Ordinal Shape Coding and Correlation for Orientation-invariant 2D Shape Matching

Michael Sole

Doctor of Engineering

University of York

Computer Science

March 2019

Abstract

The human brain and visual system is highly robust and efficient at recognising objects. Although biologically inspired approaches within the field of Computer Vision are often considered as state of the art, a complete understanding of how the brain and visual system works has not yet been unlocked. Benefits of such an understanding are twofold with respect to Computer Vision: firstly, a more robust object recognition system could be produced and secondly a computer architecture as efficient as the brain and visual system would significantly reduce power requirements. Therefore it is worthy to pursue and evaluate biologically inspired theories of object recognition.

This engineering doctorate thesis provides an implementation and evaluation of a biologically inspired theory of object recognition called Ordinal Shape Coding and Correlation (OSCC). The theory is underpinned by relative coding and correlation within the human brain and visual system. A derivation of the theory is illustrated with respect to an implementation alongside proposed extensions. As a result, a hierarchical sequence alignment method is proposed for the correlation of multi-dimensional ordinal shape descriptors for the context of orientation-invariant 2D shape descriptor matching.

Orientation-invariant 2D shape descriptor matching evaluations are presented which cover both synthetic data and the public MNIST handwritten digits dataset. Synthetic data evaluations show that the proposed OSCC method can be used as a discriminative orientation-invariant 2D shape descriptor. Furthermore, it is shown that the close competitor Shape Context (SC) method outperforms the OSCC method when applied to the MNIST handwritten digits dataset. However, it is shown that OSCC outperforms the SC method when appearance and bending energy costs are removed from the SC method to compare pure shape descriptors. Future work proposes that bending energy and appearance costs are integrated into the OSCC pipeline for further OCR evaluations.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First of all I would like to thank Professor Jim Austin for both supervising this work and providing its foundations. I wish to thank both my internal examiner Dr Nick Pears and my external examiner Professor Tim Cootes for their time and valuable feedback. I would also like to thank Cybula Ltd for sponsoring me to undertake this research. I have thoroughly enjoyed being part of Cybula and working with Jim.

I would like to thank my wife Daniela for all of her support, encouragement and time - especially towards the end of this work. I would also like to thank my parents for their support throughout the years and for the many hours spent proofreading my work.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Chapter 1

Introduction

1.1 Motivation

The human brain and visual system is highly robust and efficient at recognising objects. Although biologically inspired approaches within the field of Computer Vision are often considered as state of the art, a complete understanding of how the brain and visual system works has not yet been unlocked. Benefits of such an understanding are twofold with respect to Computer Vision: firstly, a more robust object recognition system could be produced and secondly a computer architecture as efficient as the brain and visual system would significantly reduce power requirements. Therefore it is worthy to pursue and evaluate biologically inspired theories of object recognition.

With Cybula Ltd as a sponsoring organisation, this engineering doctorate thesis provides an implementation and evaluation of a biologically inspired theory of object recognition proposed by Austin [1] via internal communication. The theory is underpinned by relative coding and correlation within the human brain and visual system. Motivated by promising results that Austin obtained from a prototype implementation of his theory, this thesis provides a thoroughly engineered and optimised implementation driven by tests and software engineering best practices to research, evaluate and extend the relative coding and correlation.

Motivated by linking the relative coding and correlation method with the literature of related work, the method is suitably categorised with respect to the Computer Vision literature and related work is drawn upon to propose an extension to the method. Additionally, an appropriate name is given to the method from categorising it within the literature, resulting in the name of Ordinal Shape Coding and Correlation (OSCC).

1.2 Relativistic Object Recognition Theory

As an extract of section 3.2 to provide context: This thesis is based upon an internal communication with Austin [1] who proposed a biologically inspired relativistic theory of object recognition. As an analogy, the biologically inspired theory is established on the relativistic input that a neuron receives. When considering a neuron in the visual field that is activated by an edge orientation of a shape, a higher level representation of the neuron could be encoded with respect to the order of its inputs from other activated neurons in the visual field. Since physical properties such as the time it takes for a neuron to receive an input from another neuron are present, the delays of activated neurons across the visual field are an ideal candidate to encode and correlate shape information. Once learnt, an ordinal coding for a particular shape could be shared amongst other neurons to invariantly recognise shapes.

1.3 Aims

The overall aim of this work is to progress Austin's relative coding and correlation theory of object recognition [1] as much as possible from a practical computer vision perspective. Consequently, aims are defined as:

1. **System verification**: verification of components within the initial prototype system provided by Austin, along with the additional components within the second iteration of the system provided by this thesis
2. **System evaluation**: evaluation of the system with respect to both point correspondence and object recognition tasks
3. **System extensions**: identify areas of improvement and propose novel solutions
4. **Well engineered system**: this is of importance for both a commercial and engineering doctorate point of view

1.4 Contributions

The ordinal shape coding and correlation implementation provided by this thesis is the second iteration with respect to an initial prototype implementation provided by Austin [1]. As a result of this thesis, the second iteration provides optimisations for the sampling algorithms used within the prototype implementation and most notably the core ordinal correlation engine is replaced with a proposed hierarchical sequence alignment algorithm. Contributions made by this thesis are as follows:

- Edge positional angle features are proposed to be encoded (see section 3.5.1) in addition to the edge segment orientation that was proposed within Austin's original prototype. The addition of this relative constraint is shown to increase the discriminability of OSCC descriptors (see chapter 4)

- Links to the sequence alignment literature have been made with respect to the correlation of OSCC coding representations that were proposed within Austin's original prototype. Consequently, this results in multiple contributions:

  - Definitions of sequence alignment methods and their effects in the context of correlating components of the original ordinal shape coding representation (see section 3.6.1)

  - Proposed hierarchy of sequence alignment methods for the complete correlation of the original ordinal shape coding representation (see section 3.6.2)

  - Proposed sequence alignment traceback method with respect to the sequence alignment hierarchy to obtain a rich set of correspondences (see section 3.6.3)

- Point correspondence and object recognition evaluations of OSCC method variations with synthetic data (see chapters 4 and 5).

- Object level recognition evaluation which compares the OSCC method with Shape Context method on the MNIST handwritten digits dataset (see chapter 6)

- Thoroughly engineered system which has been rigorously tested and has accompanying Graphical Processing Unit (GPU) and Message Passing Interface (MPI) implementations

1.5 Organisation

Chapter 2 reviews the literature around ordinal shape coding and correlation of shape with a focus on computer vision. Firstly the general local feature based pipeline is reviewed and the task of local feature description with respect to shape and ordinal encoding is focused on. Secondly the object level task of the local feature pipeline is reviewed. A review of biologically inspired computer vision approaches is undertaken

with special attention to both ordinal neural coding and local features. Finally, ordinal coding and correlation approaches are reviewed.

Chapter 3 illustrates the derivation of an ordinal shape coding and correlation implementation following the relativistic principles of Austin's biologically inspired relativistic theory of object recognition [1]. Pre-processing and sampling stages used by a prototype initially implemented by Austin are derived and illustrated up to the point of obtaining an ordinal shape coding. Original work of this thesis follows with a proposed sequence alignment hierarchy approach for the correlation of ordinal shape codings.

Chapter 4 provides a point correspondence level evaluation using synthetic data with respect to variations of the OSCC approach. An evaluation framework using the Mean Average Precision metric is used to report results. Furthermore, the framework also provides a deeper inspection into the contributions that individual shapes have towards the high level metric. Evaluation requirements are first defined followed by an evaluation of OSCC variations with respect to similarity, affine and perspective transformations as well as clutter and occlusion. Furthermore, an interest region evaluation is also undertaken to compare variations of the OSCC approach with the Generalised Shape Context, Scale Invariant Feature Transform and Gradient Location and Orientation Histogram local descriptor methods.

Chapter 5 builds upon the point correspondence evaluation chapter by providing a preliminary object recognition evaluation with respect to synthetic dataset variations combined with variations of the OSCC approach. Results are presented in the same format as the point correspondence evaluation for simplicity. Furthermore, based on local Hough voting, a simple object classification metric is defined and used for the evaluation. The classifier is trained on a reference dataset and then used to recall reference dataset variations of interpolation, translation, rotation, scale, clutter and occlusion.

Chapter 6 undertakes an object level recognition evaluation using the public MNIST handwritten digits dataset. OSCC is compared against the Shape Context method [2] with respect to published results of using 20,000 train images and the full test set. A further evaluation is undertaken where OSCC is compared against a Shape Context method variation where the Shape Context distance component is used only - rather than using the MNIST optimised KNN distance function weights along with appearance cost and bending energy components.

Finally, chapter 7 concludes the work undertaken within this thesis along with highlighting the contributions made. Future work is then proposed and discussed.

Chapter 2

Background

2.1 Introduction

Computer vision approaches can be broken down into the two high level categories of global and local feature based methods. Global features encode the overall information of an image, for example, a histogram of image gradients or colours describe global information. Local features on the other hand encode relative information with respect to a point of interest. For example, given a point of interest indicated by a blob detector, a local feature encodes relative information for a region around the interest point, such as a spatial histogram of gradients with the sampling pattern aligned with the dominant orientation of the region.

While having links to biological plausibility, local feature based methods have the advantage of naturally being robust to clutter and occlusion. Furthermore, the relativistic nature of local feature based methods is a suitable basis for relative ordinal coding and correlation of shape which is the focus of this thesis. Moreover, local feature based methods exist within the literature that are designed for describing and matching relative shape features. Consequently, the first part of this chapter focuses on reviewing local feature descriptors.

In this chapter, we first review the general local feature based pipeline which can be used to achieve object level recognition tasks. We first focus on the local descriptor task with special attention to shape based local descriptors and ordinal coding. Following this, we review methods that are used to achieve the higher level task of object recognition using local features. We then review biologically inspired approaches that have been applied to object recognition with a concentration on ordinal neural coding and local features. Finally, ordinal correlation approaches are reviewed. The topics reviewed cover the relevant properties of the relativistic object

recognition theory for which this thesis provides an implementation and evaluation for.

2.2 Local Feature Descriptors

The general local feature based object recognition pipeline can be broken down into individual tasks, where each task is a research area in its own right. Figure 2.1 provides an illustration of the pipeline tasks and information flow. It is important to note that individual approaches within the literature normally define more than one task of the local feature based object recognition pipeline. The definition and fine-tuning of preceding tasks along with the features that they output is essential to take advantage of subsequent algorithmic properties.

Figure 2.1: General local feature based object recognition pipeline

Pre-processing is the first task which is used to prepare images as input to the pipeline. This can involve sub-tasks such as noise removal and image resizing depending on the application specific domain. As a simple example, a shape classification system could have the pre-processing steps of blurring, thresholding and then centering the image around the shape with respect to its convex hull. Therefore normalising the input for the rest of the pipeline.

Feature detection is the second task which is concerned with detecting image features such as edges, corners or blobs. Desirable properties of feature detectors are repeatability and robustness to invariances such as rotation, scale and affine transformations. Feature region detectors go beyond detecting the location of a feature by adding region information. For example, the affine invariant Harris affine region detector [3] can be used to detect an elliptical region around a keypoint which can then be normalised to a circular region, this representation is often referred to as a patch within the literature. Continuing with the simple shape recognition example introduced within the pre-processing stage, a set of normalised patches obtained from the Harris affine region detector could be extracted from the shape to then be passed to the next step within the pipeline.

Feature description is the third task which is concerned with efficiently and discriminatively encoding the region around a detected feature. Local feature description approaches are commonly based on a spatial histogram of gradients. Following the shape recognition example, given the patches output from the previous feature detection task, each patch can be encoded as a spatial histogram of gradients, where a patch is divided into sub-regions from which a histogram of gradients are extracted. The resulting patch is then represented as the concatenation of sub-region gradient histograms. Sections 2.2.2 and 2.2.3 review numerical and ordinal feature descriptor approaches respectively while section 2.2.1 reviews shape specific descriptors. Shape specific descriptors can often be described as semi-local or global descriptors as they generally sample from a relatively larger region to capture shape information.

Feature matching is the fourth task which comprises of a cost function used to compare feature descriptors. The feature matching tasks can also incorporate feature matching methods that make use of the defined cost function, for example, methods such as exact or approximate Nearest Neighbor can be used along with the popular approach of coarse-to-fine matching used for efficiency. Following on with the basic shape recognition example, the Euclidean distance could be used to construct a cost matrix between the feature descriptors of two images, resulting in a set of correspondences between the images.

The final object level task takes a collection of matched feature descriptors and associated costs from the feature matching task and uses the information within a specific object level task. The high level task of shape alignment can be used for the running shape recognition example. Given a set of correspondences extracted from a pairwise cost matrix between the descriptors of two shapes, a basic approach for alignment could be the use of the least squares method for finding an affine transformation. More robust methods for image alignment along with a brief review of object level tasks using local features is presented within section 2.3.

In addition to handcrafted tasks of the general local feature based object recognition pipeline, task subset and end-to-end learning has also been applied [4] [5] [6]. It is interesting to note that a large scale evaluation of both hand crafted and learned descriptors shows that with additional normalisation, the renowned hand crafted Scale Invariant Feature Transform (SIFT) feature descriptor is either comparable to or outperforms the learned descriptors with respect to most tasks, while also being more efficient to compute [7]. However, it is shown that the learned descriptors outperform SIFT with respect to a significant amount of geometric noise.

2.2.1 Shape Specific Descriptors

The Shape Context shape descriptor coined in 2002 by Belongie et al. [8] provides a translation, rotation and scale invariant shape descriptor. A single Shape Context is encoded for a single edge point of a shape by sampling from a log-polar coordinate system with the origin set as the edge point. The local sampling pattern is used to obtain a histogram of edge point counts resulting in the descriptor. More specifically, once an edge detector has been used to extract the edges of an image, an individual Shape Context for each individual edge pixel can be obtained to describe the shape relativistically. Keeping a one to one mapping between shape context and edge pixel, the number of edge pixels can be reduced to save computation. The Shape Context encodes relative edge angle and distance by splitting ranges of angles and distances into bins using a log-polar coordinate system, therefore resulting in a spatial histogram.

For Shape Context scale invariance, the distance of an edge point is normalised by the mean distance of all edge points from the Shape Context origin. To achieve rotation invariance, the sampling pattern is aligned with the tangent with respect to the sample edge point. Early work populated each bin with the number of edge points present within its corresponding geometric area. Later work stores the strongest relative direction of the edge point angles within a bin instead of the density, this modification leads to the name of a generalised shape context [9] which have been shown to be more discriminative when reliable orientation information is available.

Similarity between two Shape Contexts is calculated by the X^2 statistic, though later work uses the Euclidean distance for efficiency [9]. Moreover, similarity between two shapes is defined as an optimisation problem for cost minimisation with a one-to-one mapping of Shape Contexts. Once the Shape Context correspondence assignment optimisation is complete, transformation estimation follows. Further work by Mori et al [9] concentrates on optimising the Shape Context query process with respect to computational time, with the aim of enabling the method to be applied to larger image databases. As a result of this, two approaches are proposed which are used during a fast pruning stage before further detailed matching on the reduced set of descriptors. The representative shape contexts approach compares a reduced number of Shape Contexts between the query image and the images in the memory, whereas the shapemes approach uses vector quantization of Shape Contexts, thus enabling an efficient search over clustered Shape Contexts.

Most notably, in 2003 Mori and Malik [10] applied generalised Shape Contexts to cluttered CAPTCHA images. They take two separate approaches, bottom-up and top-down. The bottom-up method identifies potential individual characters with a

location and score. Potential characters are concatenated in different combinations to return the score for a potential dictionary word. In contrast, the top-down approach recognises whole words at once; this is shown to be required for the more difficult CAPTCHAs where words are overlaid. The bottom-up approach is applied to single word CAPTCHAs and is shown to achieve a correct classification of 92%, while the top-down approach is applied to CAPTCHAs with overlapping words which is shown to achieve 33% correct classification.

With the motivation of matching similar shapes, Berg [11] proposed the Geometric Blur shape descriptor. As a preprocessing step, edge orientation maps are first calculated from which multiple geometric blur maps are obtained by applying a Gaussian kernel multiple times with an increasing size. For an individual edge point, a Geometric Blur descriptor is obtained by sampling the geometric blur maps with equally spaced sample points around equally spaced concentric rings. As rings get further away from the origin, their associated sample points sample from corresponding geometric blur maps obtained from an increasing Gaussian blur kernel size. Therefore as sample points get further away from the origin, they sample an increasingly blurred signal - an individual sample point samples all edge orientation maps resulting in n values from n orientation maps. All sample point values are finally concatenated to construct the descriptor.

Motivated by the use of shape features when other visual features are not available, in 2003 Mikolajczyk et al. [12] proposed a shape recognition method based on local edge features. A scale invariant edge based region detector is proposed from which a local shape descriptor is constructed as an edged based variation of the SIFT descriptor. Modification to the SIFT descriptor for the application to an edge based region is the sample pattern alignment with the underlying edge from which the dominant orientation is also obtained from. For a single region, both a low resolution 16 dimensional and high resolution 128 dimensional SIFT descriptor is constructed which are used for the higher level task of coarse to fine matching.

The Spin-Image local descriptor for 3D surface matching was proposed in 1997 by Johnson as a PhD thesis [13]. Given the 3D mesh of an object, the idea is that a spin-image can be used to represent a local surface relative to the coordinate system of a vertex within the mesh, where a vertex is the origin of its local coordinate system. Thus, each vertex paired with its spin-image representation provides an overall surface descriptor that is invariant to pose as well as being tolerant to occlusion and clutter.

The construction of a spin-image is as follows. A 3D vertex of a mesh and its normal is known as an oriented point. An oriented point is used as the origin for a

local Spin Image descriptor, from which a cylindrical coordinate system is established. Cylindrical dimensions of the plane perpendicular to the normal represent the distance and angle from the normal. The third dimension along the normal represents distance from the plane. A spin-image is represented as a 2D matrix which maps to discrete bins of the distance and angle dimensions which lie on the plane. The matrix is populated with counts of vertices that lie within the two dimensions with the use of bilinear interpolation to reduce binning boundary effects.

A comparison measure for the similarity of Spin-Images is achieved by taking correlation coefficients of their overlapping populated bins. As a result of being able to compare Spin-Images, multiple correspondences between surfaces can be identified and used to calculate a transformation from one surface to another. Johnson and Herbert [14] state that the computational complexity does not scale when comparing large amounts of Spin-Images which is especially a problem for object recognition. They overcome this problem by using Spin-Image compression with the use of Principal Component Analysis.

2.2.2 Numerical Descriptors

Between 1999 and 2004, Lowe [15] [16] proposed the Scale Invariant Feature Transform (SIFT) method which comprises of both keypoint detection and description. The SIFT detector finds keypoints along with their scale by detecting consistent interest points within the scale space pyramid. Given a detected SIFT keypoint, the SIFT descriptor encodes the keypoint's local region by using a spatial histogram of gradients. The SIFT descriptor method has set the foundations for many local descriptor methods which can be described as SIFT descriptor variations.

The SIFT descriptor spatial histogram is constructed by first calculating the signed orientation and magnitude of all pixels within the keypoint's local region. The magnitudes are then weighted by a Gaussian such that pixels closer to the keypoint's origin contribute more information to the descriptor. The keypoint's region is then split into a 4 x 4 grid of sub-regions relative to the dominant orientation of the region for rotation invariance. For an individual sub-region, a histogram of eight orientation bins is obtained with respect to associated magnitudes, trilinear interpolation is used during histogram construction to avoid binning effects. The resulting 128 dimensional SIFT descriptor represents the concatenation of sub-region orientation histograms. Illumination invariance is achieved by normalising the SIFT feature vector.

PCA-SIFT [17] is concerned with applying the PCA dimensionality reduction

method to a normalised keypoint region obtained from the SIFT detector. Although originally applied to regions obtained from the SIFT detector, other region detectors could also be used. Given a normalised keypoint region, gradients in both the x and y directions for each pixel are concatenated into a vector which is then normalised. The authors undertake an initial stage to obtain a PCA projection matrix by gathering 21,000 SIFT keypoint regions to find the top n principle components - each region was normalised to obtain a 41 x 41 patch. Therefore the dimensionality of the PCA-SIFT descriptor is equal to the selected n principle components. Experiments show that PCA-SIFT is both more discriminative and faster than SIFT when using n principle components of no more than 20.

Gradient location-orientation histogram (GLOH) [18] modifies SIFT in three ways. The first is the use of a log-polar sampling pattern with both radial and angular bins using bin counts of 3 and 8 respectively - as opposed to the standard SIFT 4 x 4 grid sampling pattern. The second modification is an increased number of orientation bins which is double the number used by standard SIFT resulting in 16 bins. The third modification is the use of PCA to reduce the higher dimensional feature vector to match the same 128 dimensions as the standard SIFT descriptor. The authors obtain a PCA projection matrix during an initial stage which comprises of 47,000 keypoint regions.

DAISY [19] is motivated by the need for a computationally efficient local descriptor for the application of dense wide baseline matching. Like GLOH, DAISY also modifies the spatial sampling pattern of SIFT. DAISY uses a circular sampling pattern with three equally spaced concentric circles which in turn have eight equally spaced sub-regions. A key modification that DAISY makes is the use of computationally efficient convolutions to calculate orientation bin values for the descriptor, as opposed to SIFT which uses weighted sums with the purpose of avoiding binning effects. More specifically, sub-regions of a concentric sampling ring are all of the same size, their orientation histograms are constructed from a Gaussian kernel with a corresponding sub-region size applied to eight orientation maps extracted from the underlying patch. This approach is computationally efficient as an initial preprocessing stage of an image can be undertaken with a pipeline of convolution kernels. The construction of each descriptor can then be achieved by gathering values from the convolution outputs and then writing them to a vector to represent a single DAISY descriptor. The authors show that DAISY is at least 43 times faster than SIFT amongst three different image sizes of 800x600, 1024x768 and 1290x960.

2.2.3 Ordinal Descriptors

Binary descriptors can be described as ordinal descriptors as they encode pixel intensity differences as a binary string. With the advantage of efficient sampling and representation, binary descriptors are significantly faster to compute and match when compared to alternative approaches. The main differences between binary descriptor methods is the sampling pattern that they use. A random sampling pattern is used by Binary Robust Independent Elementary Features (BRIEF) [20] which is not rotation invariant. Later proposed binary descriptors add rotation invariance and improve on the sampling pattern approach. ORB [21] uses a learning algorithm to find the most discriminative sample pairs, while FREAK [22] define a sampling pattern to model properties of the human visual system from which the most discriminative sample pairs are learnt. Although binary descriptors are significantly faster to compute and match compared to floating point descriptors, there is a tradeoff of discriminability as shown by the results of Balntas et al [7].

In 2006, Heikkilä et al [23] proposed the Center-Symmetric Local Binary Pattern (CS-LBP) descriptor. Based on the SIFT descriptor, the key difference is that CS-LBP features are encoded as opposed to gradients. CS-LBP descriptors are an adaptation of the Local Binary Pattern (LBP) texture descriptor. A LBP descriptor is obtained for an individual point by comparing the pixel intensity of the point with the pixel intensities of neighboring points. A binary string is constructed from the intensity comparisons and then converted into a decimal number. With the motivation of reducing the number of possible states, the CS-LBP descriptor is obtained by comparing the intensity of neighboring points that are opposite to each other instead of comparing the neighboring points with the center point. Where n is the number of neighboring sample points, the number of possible states output by the LBP descriptor is 2^n as opposed to $2^{n/2}$ for the CS-LBP descriptor.

In 2010, Gupta et al. [24] proposed the HRI-CSLTP descriptor which is the combination of two complementary descriptors. The first is the proposed Histogram of Relative Intensities (HRI) descriptor which samples a patch resemblant to SIFT. The major difference is that binned intensity values are encoded as opposed to orientations. A binned intensity value is obtained by determining which intensity bin an intensity value falls within. The second proposed descriptor is an extension to the CS-LBP descriptor with the addition of ternary codes resulting in the CS-LTP. The additional value made available by the ternary code is used to encode whether the comparison of two values is under a threshold which implies that the values are similar - the authors state that codes are less robust when the values are similar. As a result, the additional

value is used to weight the contribution of the code for the histogram. To reduce the dimensionality added by the ternary codes, two comparisons are made, one for each diagonal neighbor pair, resulting in nine unique states for a point.

In 2009, Toews and Wells proposed the SIFT-Rank ordinal descriptor [25]. The motivation behind the method is to explore the use of ordinal approaches for descriptor normalisation, as opposed to numerically normalising the vector of a local descriptor. The authors highlight that common numerical descriptor matching methods such as the Euclidean distance assume linearity while the underlying data is not necessarily linear. Ordinal coding and correlation does not make the assumption of linearity with respect to the underlying data.

Ordinal coding of the SIFT descriptor is achieved as follows. First, all of the elements within the descriptor are sorted while keeping track of the original element positions in the descriptor. A sorted index value for each element of the original descriptor is then obtained resulting in an ordinal encoding of the original descriptor. This simple mapping between numerical descriptor to ordinal descriptor is illustrated by figure 2.2. The resulting ordinal coding can then be used by ordinal correlation approaches for descriptor matching. The authors highlight the use of the Spearman and Kendall rank correlation coefficients. Toews and Wells evaluate their ordinal coding

$$\begin{pmatrix} 0 & 10 & 20 & 15 & 90 & 30 \end{pmatrix} \qquad \begin{pmatrix} 0 & 1 & 3 & 2 & 5 & 4 \end{pmatrix}$$

(a) (b)

Figure 2.2: Six dimensional numerical descriptor (a) mapped to an ordinal descriptor (b). The numerical descriptor is sorted while keeping track of the original index for each element. The sorted element index is then assigned to the corresponding original index within the ordinal descriptor. This mapping has $\mathcal{O}(n \log n)$ time complexity due to the sorting step.

descriptor method with respect to point correspondence using the local descriptor performance evaluation framework and dataset defined by Mikolajczyk and Schmid [18]. Three numerical descriptor methods of SIFT, PCA-SIFT and GLOH along with their ordinal counterparts of SIFT-Rank, PCA-SIFT-Rank and GLOH-Rank are evaluated. Results show that SIFT-Rank achieves both the highest precision and recall values for 6 pairs of images out of 8. For the remaining two pairs of images, SIFT-Rank achieves the highest recall for the JPEG compression noise treatment pair and the highest precision for rotation and zoom treatment pair.

Although SIFT-Rank shows better performance than SIFT, it is highlighted that GLOH-Rank and GLOH are similar while PCA-SIFT-Rank is much worse than PCA-SIFT. Toews and Wells speculate that despite the use of principal components in both PCA-SIFT-Rank and GLOH-Rank methods, PCA-SIFT-Rank low performance could be due to principal component correlation or relatively lower descriptor dimensionality (32 dimensions), while GLOH-Rank could have comparable performance with GLOH due to its relatively higher descriptor dimensionality (128 dimensions). It is clear that further experimentation is needed to evaluate how descriptor dimensionality affects the performance of point correspondence when using ordinal coding and correlation.

In 2009, Tang et al. [26] proposed the Ordinal Spatial Intensity Distribution (OSID) descriptor with the goal of a non-linear illumination invariant descriptor. OSID encodes a patch as a spatial histogram with n angular partitions. Each pixel within the patch is assigned an intensity order bin value as well as the angular bin partition index that it falls within. The spatial histogram encodes a histogram of intensity order bins within each angular partition. A resulting feature vector is obtained by concatenating the angular bin histograms - using both a consistent angular bin start index and a consistent direction - followed by normalisation.

With the motivation of deformation robustness, in 2010 Tant et al. [27] propose an extension to OSID resulting in soft OSID which incorporates the soft binning method, where a single value is assigned to multiple angular bins with a weighting function. Experiments are undertaken with the evaluation framework dataset of Mikolajczyk and Schmid [18]. Results show that soft OSID outperforms standard OSID as well as all descriptors provided by the evaluation framework which include SIFT, GLOH, Shape Context and Spin Image descriptors. Furthermore, soft OSID is also shown to outperform the same descriptors with respect to illumination changes.

In 2011, Fan et al. [28] show that the dominant orientation calculation of a patch that local descriptors rely on such as SIFT is error prone. An experiment is undertaken with 40 pairs of images, where a pair has an associated ground-truth rotation homography. For each individual pair of images, SIFT descriptors are calculated and matched between images using the nearest neighbor method. Results show that the SIFT descriptor is robust to a dominant orientation calculation error between the range of -20 to +20 degrees. Furthermore, it is shown that 36.23% of the ground truth correspondence across all image pairs have a dominant orientation calculation error outside the range of -20 to +20 degrees.

Motivated by the error prone dominant orientation calculation that many descriptors rely on such as SIFT, Fan et al. [28] make use of rotationally invariant gradients.

A rotationally invariant gradient between a keypoint and another point of interest - (keypoint, sample point) pair - within the keypoint's local descriptor sample area is calculated by first establishing a rotationally invariant coordinate system between the two points. The keypoint is established as the origin and the point of interest lies along the positive y-axis. The gradient is then calculated within the coordinate system between the two points resulting in a rotationally invariant gradient.

With the goal of encoding spatial information of a patch that does not rely on the error prone dominant orientation calculation, Fan et al. [28] use the ordinal intensity information of a local patch for spatial pooling. This is achieved by first sorting all points from a patch with respect to pixel intensity and then splitting the result into n equal segments. For an individual segment, a histogram of rotation invariant gradient orientations is constructed from the points within the segment. The histogram from each segment is then concatenated and the resulting vector is normalised to create a local descriptor. The final step that Fan et al. take to create their proposed local descriptor is the concatenation of the previously defined local descriptor with respect to multiple patches of different sizes for an interest point. The authors name this descriptor the Multisupport Region Order-Based Gradient Histogram (MROGH).

In 2012, Fan et al [29] proposed the Intensity Monotonic Invariant Descriptor (MRRID) with the motivation of intensity invariance. MRRID is an adaptation of MROGH which replaces the histogram of rotation invariant gradients from an individual segment with a histogram of intensity based features. Center-Symmetric Local Binary Patterns (CS-LBP) are used as the intensity based features, where an individual CS-LBP is extracted from the rotation invariant coordinate system of a (keypoint, sample point) pair.

In 2011 Wang et al [30] proposed the Local Intensity Order Pattern (LIOP) local descriptor followed by 2016 when Wang et al [31] proposed the complementary Overall Intensity Order Pattern (OIOP). Both descriptors make use of the ordinal intensity information of a local patch for spatial pooling and rotation invariant sampling approaches used by the MROGH method [32]. Principal Component Analysis is applied to concatenated LIOP and OIOP descriptors to obtain the resulting 128 dimension MIOP (Mixed Intensity Order Pattern) descriptor [31].

LIOP creates a normalised histogram of ordinal codes for a local patch associated with a keypoint. This is achieved by firstly defining the length n of an ordinal code, where the total number of unique codes is $n!$. Like the MROGH method, a rotation invariant coordinate system is established for each (keypoint, sample point) pair within the patch. A rotation invariant code is obtained from the coordinate system of a

(keypoint, sample point) pair by sampling n equally spaced pixel intensities around the sample point with respect to a given radius starting from the furthermost point along the y-axis in an anticlockwise direction. The sorted indices of the n sampled pixel intensities are then obtained and used as an ordinal code. A normalised histogram of ordinal codes is obtained from a patch to obtain the LIOP descriptor.

Motivated by distortion and noise robustness, OIOP uses information across spatial pooling regions for a sample point as opposed to LIOP which uses local information within a spatial pooling region. At the point where LIOP obtains an ordinal code for a (keypoint, sample point) pair with n sample points, OIOP obtains a quantised vector of length m for each n sample point from the m spatial pooling regions of the patch. The authors highlight that distributions of pixel intensities across the spatial pooling regions are not uniformly distributed. Consequently, an experiment was set up to learn the quantisation ranges for each spatial pooling region to remove bias. The quantised vectors are combined to produce a (keypoint, sample point) code as an n^m number. The resulting OIOP descriptor for a patch is encoded as a normalised histogram of n^m numbers obtained from all (keypoint, sample point) pairs from the patch.

In 2016 Wang et al [31] evaluate ordinal descriptor approaches that make use of both the intensity order spatial pooling and the rotation invariant sample point coordinate system methods which include: MROGH, MRRID, LIOP, OIOP and MIOP. MROGH and MRRID are shown to be less discriminative than LIOP, OIOP and MIOP with an experiment that shows the performance of each descriptor with an increasing number of support regions from one to four.

For image alignment evaluations, Wang et al [31] compare the ordinal descriptor methods using a single support region with other descriptor methods which include: SIFT, DAISY, OSID and HRI-CSLTP. For the Oxford dataset and evaluation framework [18], in general the three most discriminative descriptors are in the order of MIOP, OIOP and LIOP. Two further datasets are used for the evaluation of image alignment which include corresponding patches between images of buildings and corresponding patches between 3D objects with out-of-plane rotations. Like the Oxford dataset evaluation results, the two most discriminative descriptors are in the order of MIOP and OIOP. The third most discriminative descriptor for the Patches dataset was LIOP while the third most discriminative descriptors for the 3D objects dataset was LIOP and DAISY.

Overall, the image alignment results show that the coarse ordinal information encoded by the OIOP descriptor is more discriminative than the local ordinal informa-

tion encoded by the LIOP descriptor, while a combination of both approaches result in an even more discriminative descriptor. Furthermore, the rotation invariant sampling used by the MIOP, OIOP and LIOP has a clear advantage over the other descriptor methods that use the error prone dominant orientation calculation for spatial pooling.

2.2.4 Cross Comparison

Table 2.1 outlines the input, sampling and representation properties of the shape specific descriptors reviewed within section 2.2.1. The Geometric Blur and Shape Context variations are described as semi-local / global descriptors as they sample from larger regions compared to standard region detectors that are used to extract local patches. Although the Shape Context uses a log polar sampling pattern while the Geometric Blur doesn't, the increasing sample area for points that are further away from the concentric ring origin should have a similar effect. Geometric Blur encodes more edge orientation information by sampling from multiple edge orientation channels, while the Shape Context counts edge points and the Generalised Shape Context uses the dominant edge orientation. Therefore the Geometric Blur method should be more discriminative with the tradeoff of requiring more space. It is surprising that the Shape Context and Geometric Blur methods have not been evaluated side by side as they are very similar, it is even more surprising that both methods share a common author.

While the Shape Context and spin image are similar with respect to counting binned population occurrences, the Spin Image is designed to project 3D surface information onto a 2D spatial histogram. The edge based SIFT approach is most similar to the Geometric Blur method as they both use information from multiple edge orientations. The edge based SIFT approach constructs a histogram of edge gradients while the Geometric Blur method samples from four edge orientation signals. Another key difference between the approaches is that the edge based SIFT approach is applied to a local normalised region while the Geometric Blur was initially proposed to be used in a semi-local setting. From these observations, an evaluation should be undertaken with all of the reviewed shape specific local descriptors in both a semi-local and normalised local setting. It should be noted that the Generalised Shape Context has been evaluated within a local patch setting along with SIFT and other local point descriptors [33]. However it would be interesting to undertake this evaluation within a semi-local setting with the inherent property of sampling more shape information.

Table 2.2 outlines the input, sampling and representation properties of the local

numeric descriptors reviewed within section 2.2.2. Although the DAISY descriptor is significantly faster to compute than SIFT and GLOH, the main differences between the SIFT, GLOH and DAISY are the sample patterns. Linking the methods to the reviewed shape specific local descriptors, DAISY has a very similar sample pattern to Geometric Blur and GLOH has a very similar sampling pattern to the Shape Context. PCA-SIFT is the most different out of the local numerical descriptors, its name is slightly deceiving in this context as it applies PCA to the output of the SIFT detector and not the SIFT descriptor.

Table 2.3 outlines the input, sampling and representation properties of the local ordinal descriptors reviewed within section 2.2.3. A major divide exists within the reviewed ordinal descriptors which is the sampling approach being either spatial sub regions or ordinal intensity spatial pooling. The former is identified to suffer from dominant orientation estimation error while the latter is designed to overcome this issue. SIFT-Rank, CS-LBP and HRI-CSLTP can all be considered as sampling variations of the SIFT sampling pattern, the major difference is that SIFT-Rank applies an ordinal coding to the actual output of the SIFT descriptor while the other approaches encode the underlying relative intensity information from the sampling pattern. OSID on the other hand makes use of angular sub-regions while encoding histograms of binned pixel intensities.

Table 2.3 shows that MROGH and MRRID have the difference of sampling strategy with respect to a rotation invariant reference frame of a sample point, where MROGH uses a histogram of rotation invariant gradients while MRRID uses a histogram of CS-LBP features. Although the multiple support regions add discriminability, they also add extra computational cost for both sampling and matching. Ordinal codes used by the complementary LIOP and OIOP are shown to be more discriminative which result in only requiring a single patch. Furthermore, the complementary descriptors that use ordinal codes within and across intensity spatial pools are combined with dimensionality reduction to form the highly discriminative MIOP descriptor with a higher computational cost. Although these highly discriminative descriptors have been evaluated and shown to outperform SIFT, an evaluation has not yet been undertaken on the recent large scale HPatches local descriptor benchmark [7], which has shown that contradictory results in the literature are due to datasets not being representative enough and that SIFT is still a main contender amongst proposed methods.

Method	Input	Sampling	Representation
Shape Context	Edges	Semi-local log-polar spatial sub-regions	Spatial histogram of point counts
Generalised Shape Context	Edges	Semi-local log-polar spatial sub-regions	Spatial histogram of strongest orientations
Geometric Blur	Edges	Semi-local uniformly spaced concentric ring with increasing sample point areas as rings get further away from the origin	Spatial histogram of edge orientation strengths
Edge based SIFT	Edges	Uniformly spaced grid of sub-regions	Spatial histogram of edge orientations
Spin Image	3D surface mesh	Uniformly spaced grid of sub-regions	Spatial histogram of vertex counts

Table 2.1: Summary of shape specific local descriptors

Method	Input	Sampling	Representation
SIFT	Patch	Uniformly spaced grid of sub-regions	Spatial histogram of orientations
PCA-SIFT	Patch	Pixel gradients	Values for n principle components
GLOH	Patch	Log-polar spaced grid of sub-regions	Spatial histogram of vertex counts
DAISY	Patch	Uniformly spaced concentric ring with increasing sample point areas as rings get further away from the origin	Spatial histogram of orientations

Table 2.2: Summary of local numerical descriptors

Method	Input	Sampling	Representation
CS-LBP	Patch	Uniformly spaced grid of sub-regions	Spatial histogram of CS-LBP values
HRI-CSLTP	Patch	Uniformly spaced grid of sub-regions	Spatial histogram of concatenated CS-LTP and HRI values
SIFT-Rank	Patch	Uniformly spaced grid of sub-regions	Spatial histogram of ordinal values
OSID	Patch	Uniformly spaced angular sub-regions	Spatial histogram of intensity order bins
MROGH	Multiple patches	Ordinal intensity spatial pooling applied to multiple support regions	Intensity order histogram of rotation invariant gradients over multiple support regions
MRRID	Multiple patches	Ordinal intensity spatial pooling applied to multiple support regions	Intensity order histogram of CS-LBP over multiple support regions
LIOP	Patch	Ordinal intensity spatial pooling. Ordinal codes are obtained within spatial pools	Intensity order histogram of ordinal codes
OIOP	Patch	Ordinal intensity spatial pooling. Ordinal intensity codes are obtained across spatial pools	Intensity order histogram of ordinal codes
MIOP	Patch	PCA applied to the concatenation of LIOP and OIOP descriptors	Values for n principle components

Table 2.3: Summary of local ordinal descriptors

2.3 Object Recognition Using Local Features

Object recognition is broken down into the two categories of classification and detection within object recognition evaluation frameworks, namely the The PASCAL Visual Object Classes Challenge [34] and the Large Scale Visual Recognition Challenge (ILSVRC) [35]. The higher level object classification category is concerned with determining whether an image contains an object class or not. Whereas the lower level detection category adds the additional constraint of object localisation within an image using a bounding box. With respect to local features, the simplest approach to object classification is the use of local feature histograms to encode images of objects which can then be compared using a distance measure. The task of object detection is more complex which would require either a computationally expensive sliding window approach using feature histograms or an alternative approach using local feature correspondences which results in tackling the correspondence problem [36]. This section briefly reviews the literature with respect to how local feature descriptors are used for object level tasks.

2.3.1 Part-Based Models

Local feature Part-Based models are representations of local feature spatial configurations and constraints that make up an object. The probabilistic constellation model is a well known part-based model. Fergus et al [37] use a weakly supervised learning method based on local features to construct a constellation model. This is achieved by first detecting local scale invariant regions across a database of images for an individual category. The regions are then normalised and reduced by the use of Principal Components Analysis. A probabilistic model encoding both features and their relative positions is iteratively estimated to find the most consistent model across the training images. An object can then be detected within an image by finding a model with the highest probability above a specific threshold.

Parts-based models do not have the requirement of being probabilistic. As another weakly supervised learning method, Lazebnik et al [38] proposed the semi-local affine parts based model which is based on identifying a model of geometrically consistent semi-local affine invariant parts. Firstly, they use an affine invariant detector to extract normalised patches which are then represented by the concatenation of two complementary descriptors. Groups of local patches that are geometrically consistent across multiple training images are considered as a semi-local affine part. A model of an object is constructed by identifying geometrically consistent correspondences

between semi-local affine parts across multiple images of an object.

2.3.2 Bag-of-features

The bag-of-features model [39] is inspired by the bag-of-words model found within the information retrieval literature. The bag-of-words model creates a histogram of word counts within a document while ignoring the ordering of the words. The bag-of-features model within the computer vision literature first starts with the construction of a visual vocabulary which is defined as a set of feature clusters. Given a visual vocabulary, features can be extracted from an image and assigned to a visual vocabulary histogram while ignoring the spatial configuration of the features. By using a visual vocabulary histogram as a feature vector, classification algorithms can be used to classify images. As an example, the bag-of-features model has been used for the task of sketch-based image retrieval using semi-local descriptors as shown by Eitz et al [40]. Their results show that a semi-local descriptor region size of around 25% with respect to the image size is appropriate for the application area.

A clear disadvantage of the bag-of-features model is that it disregards the spatial information configuration of features. Motivated by this, Lazebnik [41] proposed the spatial pyramid model which divides an image into a hierarchy of sub regions, where a sub region is represented as a bag-of-features. For an individual layer within the spatial pyramid, a spatial histogram of feature categories is constructed where a feature is discretised such that it falls within a specific category. Spatial histograms for all layers are normalised and then concatenated. A pyramid matching kernel is used to compare two spatial pyramids by assigning higher weights to lower level layers that contain more fine grained spatial information. The highest level layer without any sub regions is represented as the standard bag-of-words approach. Results show that the spatial pyramid model is more discriminative than the standalone bag-of-features model. A clear disadvantage that arises from the spatial partitioning approach is the requirement that input images should be relatively similar with respect to the viewpoint of scenes or objects.

To overcome the effects of clutter and noise being encoded by the bag-of-features model, discriminative kernel based methods that are robust to these effects are proposed within the literature [42]. A comprehensive evaluation of such methods is undertaken by Zhang et al [42] where variations of detectors and descriptors are evaluated as input into a local feature distribution encoding for an individual image. The distribution encoding is achieved by obtaining a number of clusters from the

extracted feature descriptors. Furthermore, with respect to the distribution coding, variations of Support Vector Machine classifier kernels are undertaken. The authors propose the Earth Mover's Distance as a Support Vector Machine kernel which is referred to as the EMD-Kernel. Results show that the EMD-Kernel and chi-squared kernels are comparable and result in more discriminative classifiers compared to the use of other kernels including linear, quadratic and RBF.

2.3.3 Geometric Correspondence

Random sample consensus (RANSAC) is commonly used within the computer vision literature for estimating a homography from a set of correspondences that contains both inliers and outliers [43]. The iterative method has two main steps. The first is the random selection of correspondences from which a candidate homography model is obtained. The second step determines the consensus of the candidate homography model from the whole set of correspondences which results in identified inliers and outliers. After n iterations, n candidate homography models are obtained with associated consensus scores. The general and simplistic framework of RANSAC lends itself to adaptations. For example, the Progressive Sample Consensus (PROSAC) [44] adaptation makes use of correspondence scores from the underlying descriptor distances to rank the correspondences from best to worst. By starting from the highest ranked correspondences, PROSAC is shown to be significantly faster than RANSAC with the assumption that the underlying correspondence costs are trustworthy [44].

Mikolajczyk et al. [12] use the generalised Hough transform to find a consistent affine transformation from local points. The initial task of coarse to fine matching is first undertaken where two descriptors are calculated for an individual region, a lower resolution descriptor of 16 dimensions and a higher resolution descriptor of 128 dimensions. The lower resolution descriptors are used to filter out the worst matches so that the better matches can be used within the second task. The second task combines the higher resolution descriptor match costs with a local neighbourhood transformation consistency score as a transformation vote. More specifically, the generalised Hough transform is used to parameterise the affine transformation space which receives votes for transformations obtained from local regions. Consequently, maximum regions within the Hough voting accumulator matrix are used to identify the most consistent transformation. A similar approach is also undertaken by Lowe [16] who states that the Hough transform is more robust when compared to methods such as RANSAC when the number of outliers is less than 50%. A major disadvantage of

the generalised Hough transform is the amount of space that the accumulator matrix requires as dimensionality increases.

Belongie et al. [8] use the Regularised Thin Plate Spline method to align two shapes given a set of correspondences. The Thin Plate Spline method is related to the analogy of bending a piece of metal which uses the amount of bending energy required as a cost. With respect to a point set, the analogy follows that the surface of a sheet of metal is defined by the point set. Therefore a transformation of the point set can be expressed as the amount of bending energy applied to the sheet of metal. The first step that Belongie et al. [8] take to apply the Regularised Thin Plate Spline method is the construction of a pairwise cost matrix between the feature descriptors of two shapes. High cost matches are then filtered out with respect to a cost threshold. One-to-one correspondences are then obtained from the remaining cost matrix and used as input for the Regularised Thin Plate Spline method. This process is undertaken iteratively by using the estimated transformation from the Thin Plate Spline method to warp the input image which is then used as input for the next iteration. After a number of iterations, the final bending energy can be used to find the template with the least cost. Disadvantages of this approach are computational complexity along with a variety of parameters to tune.

2.3.4 Cross Comparison

Both probabilistic and non-probabilistic Part-Based methods outlined in section 2.3.1 construct part-based models as weakly supervised methods. With both approaches being based on local features, they follow the same high level process of determining geometrically consistent parts across a set of training images. Lazebnik et al [38] state that modelling affine invariance is challenging from a probabilistic modelling point of view. Furthermore, it is identified that the size of probabilistic models is limited due to complexity. However, the correspondence problem is still present for these feature based approaches. As indicated by Zhang et al [42], although local feature part-based models are intuitive, they are often too complex within an automatic learning context due to spatial configuration variability.

The review of bag-of-features approaches within section 2.3.2 reviews methods that build upon the underlying approach to make it more discriminative. Clearly, the disregard of spatial information removes spatial constraints. The use of semi-local descriptors covering around 25% of the image are shown to be of practical use with the result of incorporating a large amount of spatial information when compared to local

descriptors. However, computing larger descriptors results in a higher computational cost. The spatial pyramid model tackles the disregard of spatial information in an alternative way creating a spatial pyramid of bag-of-features sub-regions. With the advantage of computational efficiency, a potential disadvantage is the need for image normalisation such that spatial pyramid sub-regions are somewhat aligned for comparison. Kernel based methods are proposed as an alternative with the goal of providing a highly discriminative classifier that is robust enough to handle clutter. However, it should be noted that such methods are computationally expensive.

The review of geometric correspondence approaches within section 2.3.3 outlines that the Generalised Hough Transform is more robust than RANSAC when inliers becomes less than approximately 50%. A major disadvantage of the Generalized Hough Transform is the large amount of memory that is required for the accumulation matrix which becomes impractical as the number of parameters increases. Therefore special considerations are required with respect to model parameters during initialisation. Although the Thin Plate Spline method provides a non-rigid transformation estimation and associated cost, it is clear that the iterative method is not efficient, while careful attention to model parameters and correspondence input need to be undertaken for practical use.

2.4 Biologically Inspired Approaches

2.4.1 Ordinal Neural Coding

In 1988, Austin [45] proposed the grayscale N tuple pattern recognition approach based on ordinal coding. Resulting in a binary neural network with ordinal grayscale N tuples as input. As an extension to the binary N tuple approach proposed by Bledsoe and Browning [46], the goal was to remove the limitation of binary image input by enabling the use of grey scale images to provide the availability of more discriminative information. The binary N tuple approach proposed by Bledsoe and Browning [46] is based on creating a codebook of states which can be used to collectively discriminate between patterns. An example of character recognition was originally used when presenting the approach. First of all, N tuples with n elements are selected, where each tuple element is randomly assigned to a fixed image pixel location. During training, tuple elements sample their corresponding fixed grid locations of a binary image, resulting in a binary string for each tuple. Therefore the number of unique states that can be represented with a tuple size of n is 2^n. The states of each tuple

are recorded within a codebook along with the class label associated with the image. During recall, the number of matched states within the codebook is accumulated for each trained class, resulting in match scores.

The grey scale N tuple approach proposed by Austin [45] uses ordinal coding to create a binary string from corresponding sample pixels of tuple elements. This is achieved by first assigning a sorted grey scale value index to the sampled grey scale values of tuple elements. For example, given a tuple with three elements which have sampled grey scale values of $(254, 0, 50)$, the corresponding sorted indices are $(2, 0, 1)$, resulting in an ordinal code. The number of unique ordinal codes that a tuple of size n can represent is defined as the ordered bell number of n where rank ties are allowed. A further method is proposed to reduce the number of unique states due to the combinatorial complexity. This method constructs a lookup table that maps ranges of ranks to individual states. For example, given a tuple of size 3 with corresponding sampled values of $(255, 0, 50)$ resulting with the ordinal code of $(2, 0, 1)$, ranges are calculated by subtracting the minimum sampled value from the maximum and then dividing by a constant p to obtain p ranges. With $p = 2$, ranges are therefore 0 to 127.5 and 127.5 to 255. Consequently, the ordinal code of $(2, 0, 1)$ is represented as two states associated by the calculated ranges resulting in states $(0, 1)$ and (2). Binary strings are associated with each unique state to construct a codebook in the same format as the binary N tuple approach. This approach is shown to result in the maximum number of unique states shown by equation 2.1 [45], where n is the tuple size and p is the number of rank ranges.

$$max\_states = p^n - (p-1)^n \tag{2.1}$$

Austin [45] demonstrated the grey scale N tuple approach with the task of edge detection. Since a region around an edge has a larger range of values compared to a background region, confidence scores are used to assign a higher confidence to tuples that sample a larger range of values. The edge detection system was set up with a tuple size of 4 associated with an 8 x 8 sliding window. Once trained on edge patterns, the detector is applied to an image with ground truth edges. Variation combinations with respect to the number of ranks and the use of confidence scores are evaluated. Results show that all variations successfully recognise edges, while increasing the number of ranks increases confidence scores due to the larger separation of scores between ground truth edges and background. Furthermore, the use of tuple confidence scores further increases confidence scores resulting in the top two overall scores out

of the four variation combinations. The best performing variation used the tuple confidence score with the highest number of ranks experimented with.

In 1998, Thorpe and Gautrais [47] proposed an ordinal neural coding method with the goal of encoding temporal spike information. With respect to a population of neurons, the approach disregards the exact spike timing information while retaining the relative spike order information. Using the constraint of one spike per neuron, a population of n neurons can be used to code $n!$ unique states. Although the elimination of timing reduces the amount of information encoded, it is shown that the proposed ordinal method is almost comparable with respect to the amount of information encoded towards the beginning of a temporal window, which is considered to be the most relevant part. A decoding strategy is used that becomes less sensitive after each spike is received.

When training with the ordinal coding proposed by Thorpe and Gautrais [47], the descending order of a neuron is used as a weight. For example, given a population of three neurons A, B and C, and an ordinal training pattern of spikes for the neurons in the order of A, C, B, then the descending order of weights are assigned as follows: $A = 3$, $B = 1$ and $C = 2$. The decoding strategy becomes less sensitive after each spike is received from a recall pattern, for example, given an individual spike within a recall pattern, a constant value of less than 1 is raised to the power of the recall spike index and then multiplied by the associated neuron order weight of the trained pattern to obtain a match score. The sum of all match scores is taken to produce the final match score for the recall pattern.

The ordinal coding method proposed by Thorpe and Gautrais [47] has been demonstrated across a range of vision publications. Invariance to illumination is presented in [47] when linking the firing of a neuron to pixel intensity, where a neuron fires earlier when presented with a higher pixel intensity. Face detection is demonstrated with a feedforward architecture that comprises of four layers [48]: the first layer outputs activity with respect to a local binary contrast map where neurons fire earlier with higher contrast. The second layer outputs activity related to eight orientation maps, for an individual map, a neuron fires earlier with respect to a higher orientation filter response. The third layer contains an individual map for the mouth, left eye and right eye features and finally the fourth layer contains the spatial configuration of the lower level facial features. During training, the mean order of inputs is taken over a set of training examples for a population of neurons which is then used as the ordinal code for the population. Manual steps were undertaken for training the third facial feature layer and the final face feature spatial configuration

layer. These steps involved selecting regions in previous layers to isolate selectivity.

The ordinal coding face detection system was trained on 270 images of faces. Four different datasets are evaluated for recall performance using detection rate and false detection counts. The first dataset consists of the data used to train the model, the system obtains a 96.3% detection rate and with two false detections. It should be noted that the model is not exact with respect to the training data as it is based on the mean rank orderings over the dataset. The second dataset consists of unseen images of faces that were trained into the model which also contained distractors such as beards and glasses. The system achieves a lower 73.1% detection rate with four false detections. The third dataset containing images of novel faces with a lower proportion of beards and glasses when compared to the second dataset achieved a 94% detection rate with four false detections. The difference between the detection rate of dataset 1 and 2 can be explained by the lower proportion of distractor objects in dataset 2. Finally a dataset without any faces obtained a single false face detection amongst 216 images. These results show that the system has a very low false detection rate while achieving a high detection rate across the datasets. Furthermore, it is shown that the system has a sensitivity to glasses and beards as distractor objects.

2.4.2 Local Features and Retinal Sampling

Lowe [15] loosely links local feature detector and description methods with biological vision. Neuroscience research is highlighted that identifies the activation of neurons that are invariant to transformations. More specifically, it is highlighted that activations of neurons for particular features are robust with respect to feature position and scale. Furthermore, Lowe [15] states that the SIFT descriptor representation is approximately at the same representative level of lower level invariant shape features found within the visual system. A face is given as an example of a higher level feature which is beyond the representative level that is referred to.

With the motivation of a fast and compact local feature descriptor, Alahi et al [22] proposed the biologically inspired Fast Retina Keypoint (FREAK) binary descriptor. Inspiration for keypoint sampling is taken from the topology of the retina, which is described as having an exponentially decreasing density of ganglion cells from the centre of the retina coupled with an increasing receptive field size of the cells from the centre. The specific sampling pattern used is constructed from a log polar concentric ring arrangement around a key point, where a ring has a number of equally spaced sampling points. As rings get further away from the centre, the sample points have

an exponentially increasing and overlapping sample area with an associated gaussian smoothing kernel. Alahi et al [22] state that it is believed differences of gaussians are used to encode visual information within the visual system. Consequently, the proposed descriptor uses the difference of gaussians with respect to sample point pairs to encode a binary string descriptor.

Due to the large number of sample point pairs available for encoding the FREAK descriptor, Alahi et al [22] undertake an evaluation to select the most discriminative pairs using almost 50,000 keypoints to reduce the binary string length of the descriptor. Furthermore, a coarse-to-fine structure is discovered during the evaluation with respect to discriminability, consequently the structure is used for descriptor matching which is also linked to saccadic visual search. As a result, the coarse-to-fine matching approach is undertaken by first thresholding matches with the respect to the outermost sample rings of a descriptor, proceeding to the innermost sample rings when match thresholds are exceeded. Experiments show that the FREAK descriptor is faster for both keypoint description and matching with respect to the SIFT and SURF floating point descriptors and also the BRISK binary descriptor. Furthermore, experiments show that FREAK is at least as discriminative than the other three approaches.

2.4.3 Convolutional Neural Networks

In 1980, Fukushima proposed the Neocognitron [49] which provides the foundations of Convolutional Neural Networks. With the goal of progressing the understanding of how the human brain works, the Neocognitron was used to model mechanisms of the visual system to achieve pattern recognition that is invariant to translation. With a hierarchical structure, the Neocognitron model is made up of layers containing simple (S-cells) and complex (C-cells) cells based on the biological visual system model of Hubel and Wiesel [50]. Layers of the Neocognitron alternate between S-cells and C-cells where a layer takes a previous layer as input. A single layer is made up of multiple cell planes, where a cell plane is made up of spatially arranged duplicate cells. The duplicate cells provide the same output to a stimulus while taking input from their corresponding spatial arrangement of the previous layer. An S-cell plane is used to identify a particular feature across all spatial locations of a previous layer. In image processing terms, an S-cell plane has an associated convolution kernel which can be used to obtain a feature map with respect to its input. A C-cell plane is used to locally pool features for shift invariance, such that if a feature exists for any local input then a response is output. An unsupervised learning self organisation approach

was initially proposed for the Neocognitron.

In 1998, LeCun et al [2] proposed the LeNet-5 Convolutional Neural Network model for character recognition as part of a successful cheque recognition system. With the main difference between the Neocognitron being the use of back-propagation for supervised training, the LeNet-5 model has 7 layers. The first five layers have the sequence of convolutions → subsampling → convolutions → subsampling → convolutions. With respect to the Neocognitron terminology, convolution layers are S-cell layers which contain a number of feature maps and subsampling layers are C-cell layers which locally pool features for shift variance. The last two layers are a fully connected layer followed by a layer that outputs a score for each trained class. Experiments on the MNIST character dataset show that Convolutional Neural Networks require a large amount of data. It is shown that LeNet-5 which takes 32 x 32 grey scale images as input with 10 class labels overall continued to improve with respect to a lower error rate when increasing the training set from 60,000 instances to 600,000. This increase was made possible with the use of synthetic data distortions. The lowest error rate of 0.7 was achieved for the MNIST character dataset with a boosted variation of LeNet-5.

Two major disadvantages of Convolutional Neural Networks are the amount of computational resources and data that is required to train a model. Due to Moore's Law and the advancements of Graphical Processing Units (GPUs) along with large scale labelled datasets such as ImageNet [35], Convolutional Neural Networks have been shown to achieve state of the art performance for a range of computer vision tasks. This was demonstrated in 2012 by the AlexNet Convolutional Neural Network [51] which won the Large Scale Visual Recognition Challenge [35] and took seven days to train. With a variation of LeNet-5 components along with the use of random dropout [52] to avoid overfitting, AlexNet was efficiently implemented across two GPUs. With eight layers, AlexNet takes a 224 x 224 RGB image as input and processes it through the first five layers which are convolutional layers. The first, second and fifth convolutional layers have pooling applied to their outputs within the layers. The final three layers are fully connected with the final layer outputting predictions with respect to 1000 classes. Since AlexNet, a range of architectures have been proposed such as the 125 layer Microsoft ResNet [53] which won the Large Scale Visual Recognition Challenge classification task in 2015. Another example is the Spatial Transformer Network [54] module which parameterises spatial transformations which can then be learnt with the outcome of invariance to spatial transformations, such as affine transformations.

Two further disadvantages of Convolutional Neural Networks are highlighted by Szegedy et al [55]. Firstly, model inspection for the understanding of learned semantics by isolating individual units given an input image is not possible due to the conclusion that semantic information is encoded across many units. As a result, these complex models are black boxes. Secondly, it is shown that small transformations applied to images that are not perceivable to the human eye can cause misclassification [55] [56]. Images that result from these transformations are called adversarial images. Adversarial attacks are a genuine concern to industry, especially as large technology companies such as Google, Facebook and Microsoft make extensive use of Convolutional Neural Networks [57]. For example, adversarial attacks could be used to trick image spam filters while it is even more concerning that these kind of attacks could danger life with respect to sign recognition in the context of self driving cars [58].

2.5 Ordinal Correlation

2.5.1 Correlation Coefficients

As part of non-parametric statistical methods, ordinal correlation coefficients provide a measure of association between two ordinal variables. Spearman's rank correlation coefficient is a common ordinal correlation coefficient which measures the monotonic relationship between two ordinal variables. For example, table 2.4 can be used to illustrate how Spearman's rank correlation coefficient is calculated. Given variables A and B, an ordered list is constructed for each variable resulting in the A Ranks and B Ranks columns respectively. A third column is constructed to represent the difference between ranks. Given equation 2.2 for the calculation of Spearman's rank correlation coefficient without any tied ranks [25], variables A and B obtain a correlation value of 0.9, where n is the number of entries for a variable.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{2.2}$$

As highlighted in section 2.2.3, the SIFT-Rank descriptor method defines both Spearman's and Kendall's correlation coefficients as possible measures of descriptor similarity. Kendall's Rank Correlation coefficient is another common correlation coefficient which is based on concordant and discordant pairs. For example, table 2.5 can be used to illustrate how Kendall's rank correlation coefficient is calculated, where the first four columns are the same as table 2.4 used for illustrating Spearman's rank correlation

A	B	A Ranks	B Ranks	Rank Difference (d)
10	10	1	1	0
20	30	2	3	-1
30	20	3	2	1
40	40	4	4	0
50	80	5	6	-1
60	85	6	7	-1
70	75	7	5	2
80	110	8	8	0

Table 2.4: Example of calculating Spearman's rank correlation coefficient between variables A and B

coefficient. A row value is calculated within the concordant column by counting the number of higher ranks that are below the associated B Ranks row. For example, for the fifth row, the concordant column has a value of 2 and the B Ranks column has a rank value of 6. B Rank values that are below the fifth row are 7, 5 and 8. Two ranks are higher than the fifth row value of 6, therefore the concordant value for the fifth row is assigned the value of 2. The same process is used to calculate the discordant column with the modification of counting lower ranked entries as opposed to higher ranked entries. Given equation 2.3 for the calculation of Kendall's rank correlation coefficient without any tied ranks [25], variables A and B obtain a correlation value of 0.78, where c and d are the total number of concordant and discordant pairs respectively.

$$\tau = \frac{c - d}{c + d} \tag{2.3}$$

A	B	A Ranks	B Ranks	Concordant	Discordant
10	10	1	1	7	0
20	30	2	3	5	1
30	20	3	2	5	0
40	40	4	4	4	0
50	80	5	6	2	1
60	85	6	7	1	1
70	75	7	5	1	0
80	110	8	8		

Table 2.5: Example of calculating Kendall's rank correlation coefficient between variables A and B

For the example within this section Kendall's rank correlation coefficient obtained a score of 0.78 for the correlation of variables A and B while Spearman's rank correlation coefficient obtained a score of 0.9. It can be seen that Kendall's rank correlation coefficient is more sensitive as it penalises equally for each rank mismatch, this is opposed to Spearman's rank correlation coefficient which penalises less for smaller rank differences.

2.5.2 Sequence Alignment

Ordinal correlation can be achieved with the use of sequence alignment approaches. For example, given a reference sequence $A = (a, b, c, d, e, f)$ and a query sequence $B = (a, e, b, c, f, d)$, B' shows the order of elements within B with respect to A, where $B' = (0, 4, 1, 2, 5, 3)$. The longest monotonically increasing sequences that can be obtained from the ordered elements of B' are $(0, 1, 2, 5)$ and $(0, 1, 2, 3)$. As a result, the minimum alignment cost can be seen as 2 which can be used as a measure of ordinal correlation. Sequence alignment methods commonly use Dynamic Programming to find the shortest and lowest cost monotonically increasing path through a pairwise cost matrix to reduce brute force computational complexity from $\mathcal{O}(X^Y)$ to $\mathcal{O}(XY)$ where X and Y are the lengths of the query and template sequences respectively.

Main differences between sequence alignment methods are constraints and penalties applied to find the shortest monotonically increasing path through the pairwise cost matrix of two sequences. The Edit Distance [59] (also known as the Levenshtein distance) is a well known example which is a metric that identifies the minimum number of insertions, deletions and substitutions required for the alignment between two sequences. Given sequence $A = (1, 2, 3, 4, 5, 6, 7)$ and sequence $B = (1, 2, 3, 7, 4, 4, 5, 6)$, figure 2.3 shows the pairwise binary cost matrix between A and B. Furthermore, figure 2.4 shows the corresponding edit distance accumulated cost matrix which contains the edit distance value of 3 as shown by the bottom right cell.

The Longest Common Subsequence is a sequence alignment approach for finding the largest monotonically increasing subsequence between two sequences [60]. For example, given sequences $A = (1, 2, 3, 4, 5, 6, 7)$ and sequence $B = (1, 2, 3, 7, 4, 4, 5, 6)$, the longest common subsequence is $(1, 2, 3, 4, 5, 6)$. A practical application example of the Longest Common Subsequence can be given for time series classification, where a pre-processing stage is undertaken to represent time series indices as symbols [61]. The Longest Common Subsequence is clearly robust to noise with respect to the insertion of random tokens within a sequence.

Dynamic Time Warping is a sequence alignment method that allows one-to-many matches between sequence indexes [62]. It is well known within the time series classification literature as an elastic based matching method that allows local shifts and warps between indexes of time series [63]. The property of allowing one-to-many matches between sequences results in allowing non-linear mappings between sequences. As opposed to the Edit Distance and Longest Common Subsequence approaches that take symbols as input, Dynamic Time Warping can take real values as direct input. As a result, a range of cost functions can be used to create the input cost matrix for the common Dynamic Programming accumulated cost matrix stage.

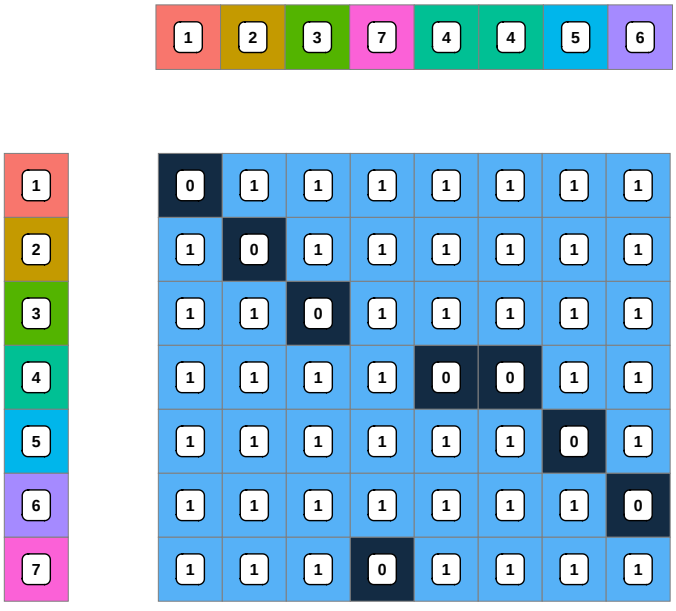


Figure 2.3: Pairwise binary cost matrix for sequences $A = (1, 2, 3, 4, 5, 6, 7)$ and $B = (1, 2, 3, 7, 4, 4, 5, 6)$.

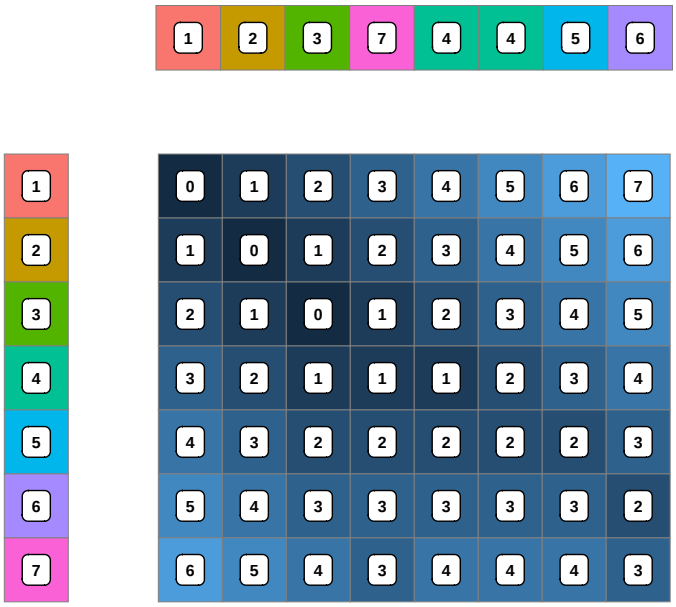


Figure 2.4: Accumulated cost matrix for the Edit Distance between $A = (1, 2, 3, 4, 5, 6, 7)$ and $B = (1, 2, 3, 7, 4, 4, 5, 6)$.

As previously stated, sequence alignment approaches that use Dynamic Programming have time and space complexity of $\mathcal{O}(XY)$ where X and Y are the lengths of the query and template sequences respectively. Although $\mathcal{O}(XY)$ is a lot better than the brute force recursive approach with a worst case complexity of $\mathcal{O}(X^Y)$, the complexity of $\mathcal{O}(XY)$ is inefficient as sequence size increases. To overcome this problem, global constraints can be applied to the Dynamic Programming matrix such that specific areas are only calculated. For example, the Sakoe-Chiba Band places a search space band across the diagonal of the matrix to reduce both time and space complexity to $\mathcal{O}(B \cdot max(X, Y))$ where B is the band width [62]. Furthermore, lower bounding techniques can be used to efficiently prune matches when searching for a best match among a collection of templates [64].

2.6 Summary

The literature reviewed within this chapter covered the relevant properties of the relativistic object recognition theory which this thesis provides an implementation and evaluation for. More specifically, firstly the relativistic nature of local feature based approaches is identified as a suitable basis within the literature for the ordinal coding and correlation method. Local feature descriptors are then reviewed with special attention to shape based local descriptors. Shape based local descriptors are identified as global or semi-local descriptors due to the larger region sample size compared to local descriptors with the purpose of encoding richer shape information. As a result, this thesis gives the appropriate name of Ordinal Shape Coding and Correlation (OSCC) to the application of the relativistic object recognition theory when applied to the domain of computer vision. As an observation, it is not clear how robust global or semi-local shape descriptors that are based on spatial histograms are with respect to heavy clutter and occlusion.

As a result of establishing local feature based approaches as a suitable basis within the literature, special attention to ordinal coding within the context of local descriptors was also given. Although the ordinal descriptors were not specifically designed for shape, they are still relevant as semi-local extensions can be easily achieved with the potential extra cost of computation. Consequently, this observation indicates that an evaluation of both local numerical and ordinal descriptors would be valuable within the context of semi-local shape descriptors. Ordinal descriptors are shown to be both discriminative and robust to non-linear monotonic illumination changes.

Biologically inspired approaches of object recognition were reviewed due to the biologically inspired nature of the relativistic object recognition theory. More specifically, ordinal neural coding along with biologically inspired local features and retinal sampling were reviewed. It is shown that local features have connections to biological plausibility and that ordinal neural coding can be used to efficiently encode and decode a large amount of information with examples of applications to computer vision problems.

Convolutional neural networks were briefly reviewed which are shown to have a biologically inspired basis. Although convolutional neural networks are considered as state of the art across many topics of computer vision literature, four disadvantages are identified where three do not align with properties of the human brain and visual system. These disadvantages are: computationally expensive and time consuming training times; requirement of large scale datasets; small transformations applied to

an image that are not perceivable by humans can cause misclassification; and the final disadvantage which is also true for the human brain and visual system is that the resulting systems are too complex to completely understand.

General methods for achieving ordinal correlation were reviewed which is a core component of the relativistic object recognition theory. Ordinal based statistical correlation coefficients and their underlying mechanisms were briefly reviewed followed by linking ordinal correlation to the sequence alignment and consequently time series classification literature. This literature is used as a basis for the main contribution of this thesis, which is a hierarchical sequence alignment method which can be used as an ordinal correlation engine within the OSCC framework. Further details of the hierarchical sequence alignment method along with the OSCC framework are provided within the next chapter.

Chapter 3

Ordinal Shape Coding and Correlation

3.1 Introduction

In the previous chapter, we reviewed the literature to cover the relevant properties of the relativistic object recognition theory which this thesis provides an implementation, proposed extensions and evaluations for. In this chapter, we provide illustrations for the derivation of the implementation with respect to the biologically inspired relativistic theory of object recognition proposed by Austin [1]. Firstly, we link the biologically inspired theory of object recognition to computer vision terminology. We then proceed to derive and illustrate each step of the implementation which is referred to as the Ordinal Shape Coding and Correlation (OSCC) method.

Derivation of the OSCC method is illustrated with the following core components: A brief overview of relative spatial configurations is given from which a translation, rotation and scale invariant point perspective can be obtained. The sampling decomposition and approximation of the point perspective is then presented which is in turn mapped to a hierarchical ordinal coding. Correlation approaches for the hierarchical ordinal coding are linked to the sequence alignment literature which leads to a novel shape descriptor correlation method - the novel shape descriptor correlation method fits in with the school of thought behind the relativistic theory of object recognition (see section 3.2).

In the next chapter we evaluate the OSCC method with respect to point correspondence evaluation requirements using ground truth point correspondences. The subsequent chapters are concerned with the evaluation of object level recognition.

3.2 Relativistic Object Recognition Theory

This thesis is based upon an internal communication with Austin [1] who proposed a biologically inspired relativistic theory of object recognition. As an analogy, the biologically inspired theory is established on the relativistic input that a neuron receives. When considering a neuron in the visual field that is activated by an edge orientation of a shape, a higher level representation of the neuron could be encoded with respect to the order of its inputs from other activated neurons in the visual field. Since physical properties such as the time it takes for a neuron to receive an input from another neuron are present, the delays of activated neurons across the visual field are an ideal candidate to encode and correlate shape information. Once learnt, an ordinal coding for a particular shape could be shared amongst other neurons to invariantly recognise shapes.

Rewriting the above in computer vision terminology leads to the following: when considering an edge pixel of a shape in an image, a point descriptor can be encoded by using the relative features of other edge pixels that form the shape. Once extracted, the point descriptor can be used by a wide range of techniques that fit into the pipeline of point descriptor shape recognition. For example, given a similarity measure for the point descriptor, a simple Nearest Neighbor approach could be used to find the most similar point descriptor in a database to determine a point correspondence.

The rest of this chapter illustrates the derivation and implementation of the biologically inspired relativistic theory of object recognition with proposed extensions - taking the encoding of relativistic information at the core of the theory into special consideration. As highlighted within the introduction chapter (see chapter 1), this thesis is motivated by promising results that Austin [1] obtained and shared through an internal communication from an initial prototype. It is also highlighted that this thesis provides a second iteration which optimises the pre-processing and sampling stages of the initial prototype, while most notably incorporating a hierarchical sequence alignment approach for ordinal correlation which is proposed within section 3.6 of this chapter.

It is important to note that all sections leading up to section 3.6 essentially illustrate the process of obtaining a shape ordinal coding representation as undertaken within the initial prototype. Section 3.6 and onwards present original work of this thesis with respect to proposing a hierarchical sequence alignment approach for the correlation of ordinal shape coding. It should also be noted that as a result of this thesis, the use of the positional angle features is proposed within section 3.5.1 in addition to the

originally used relative edge segment orientation feature.

3.3 Shape From Relative Points

As an overview, relative spatial configurations that give rise to shape can in turn be used to encode shape. A point perspective of shape can be described by the relative configuration of shape edge points with respect to a point. With the point as a reference point of a polar coordinate system, distance and angle parameters from the relative spatial configuration of shape edge points can be obtained. Figure 3.1 illustrates a point perspective of shape with a reduced number of shape edge points for simplicity.

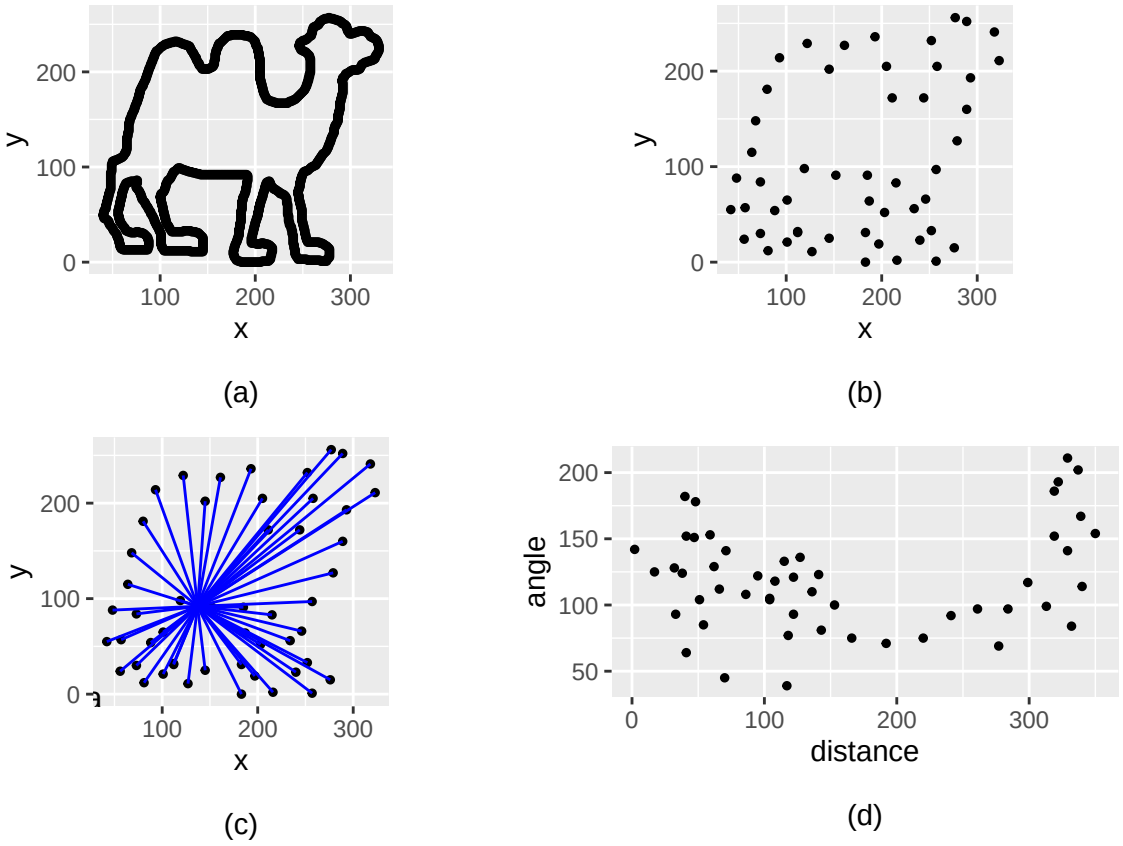


Figure 3.1: Relative shape configuration from the perspective of a point. (a) shows the original shape, (b) shows arbitrarily selected edge points, (c) shows lines connecting the selected edge points to an arbitrarily selected reference point and (d) shows the point perspective for the reference point in a polar coordinate system.

3.4 Relative Edge Point Perspective of Shape

As shown in the previous section, a point perspective of shape can be described by the relative configuration of edge points that represent the shape. By using a shape's edge point as a reference point, an invariant point perspective can be encoded by using relative edge features. Two types of edge features are described within this section which are defined as the edge segment orientation and positional angle. Figure 3.2 illustrates the difference between the features. An edge segment is defined as a group of connected edge points with a common image gradient direction.

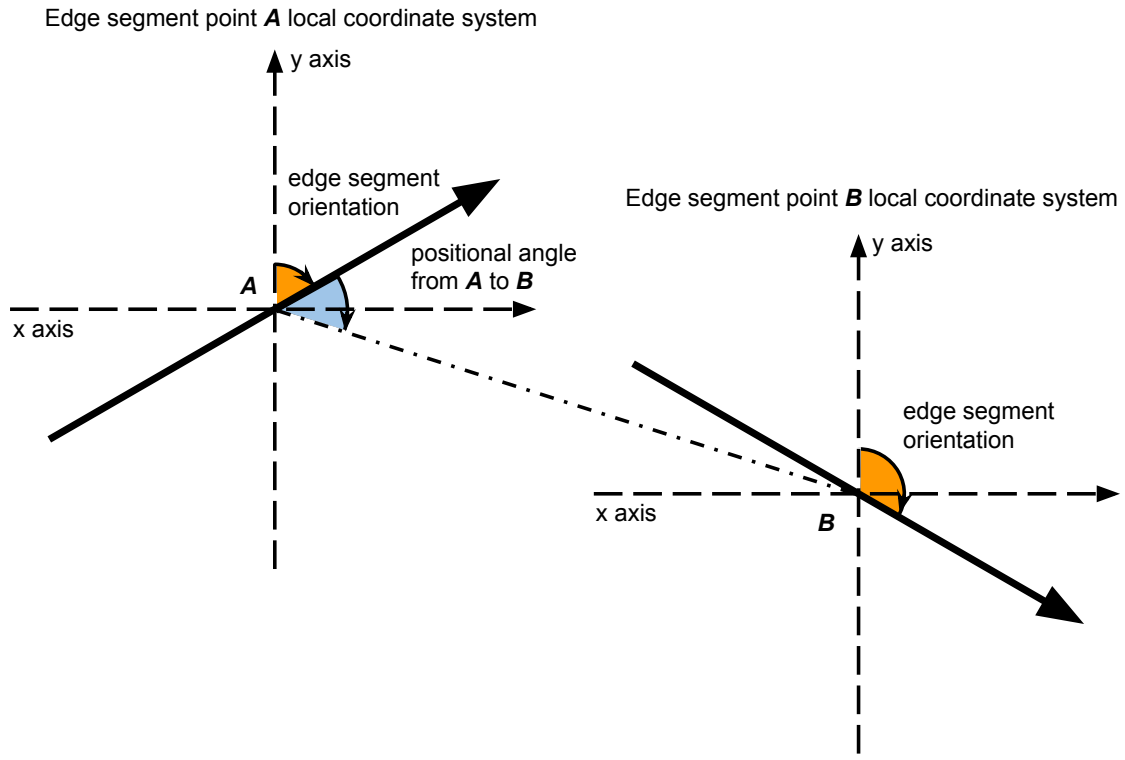


Figure 3.2: Edge segment orientation and positional angle features illustrated with two edge segments. Given two edge segments with a single arbitrarily selected point A and B respectively, each point has a local coordinate system. The edge segment orientation is defined as the image gradient direction. The positional angle from A to B is defined as the angle between the edge segment orientation of A and the point B.

3.4.1 Relative Edge Segment Orientation

Relative Edge Segment Orientation (RESO) encodes the relative edge segment orientation resulting in a translation, rotation and scale invariant encoding. For example, given edge segment orientations of 45 and 135 degrees for edge segments of points A and B respectively as illustrated by figure 3.2, the signed RESO would be $45 - 135 = -90$ degrees. Figure 3.3 illustrates rotation invariance using binned signed RESO features by showing RESO features for an arbitrarily selected edge point on the character 'A' along with the corresponding binned RESO features for the edge point of the character 'A' rotated clockwise by 45 degrees.

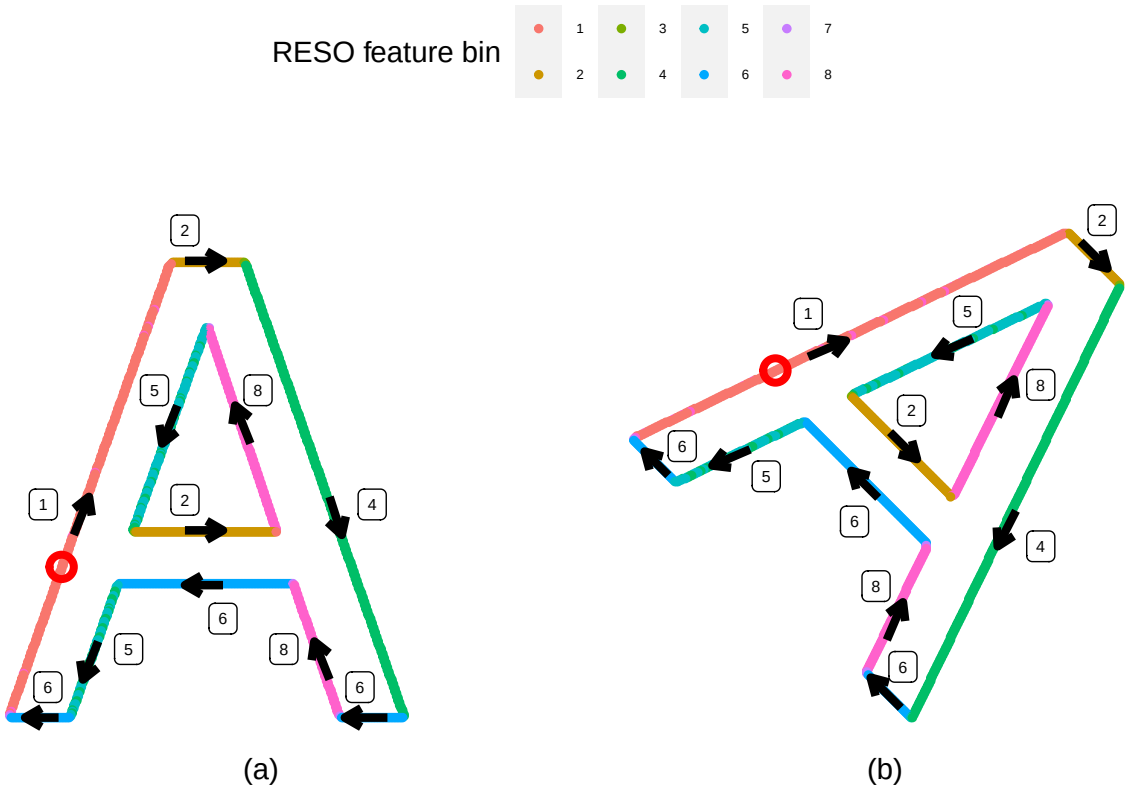


Figure 3.3: Rotation invariant point perspective using binned signed RESO features. (a) Binned RESO features with respect to a point marked with a circle. (b) Binned RESO features with respect to the point after a clockwise rotation of 45 degrees is applied to (a). Arrows and their corresponding labels indicate the binned RESO features.

Algorithm 1 defines pseudocode for obtaining binned RESO features. The algorithm extracts binned RESO features for the input image edge point *pt* from input image *img*. RESO features can be signed or unsigned as defined by the *signed* parameter. RESO features are binned specified by the *nBins* parameter.

Algorithm 1 Binned relative edge segment orientations

function RELATIVEEDGESEGMENTORIENTATIONS(img, pt, $signed$, $nBins$)
  $directions \leftarrow$ intensity direction (0 to 359) degrees for each pixel of img
  $edgeMap \leftarrow$ binary edge map from the canny edge detector applied to img
  $ptDirection \leftarrow$ intensity gradient direction for pixel (pt_x, pt_y) from $directions$
  $relativeEdgeDirections \leftarrow \{\}$
  for all $point \in edgeMap$ **do**
    $directionDiff \leftarrow ptDirection - directions(point_x, point_y)$
    if $directionDiff < 0$ **then** ▷ Assure $directionDiff$ range is $[0, 359]$
      $directionDiff \leftarrow directionDiff + 359$
    end if
    $maxValue \leftarrow 359$ ▷ Signed range is $[0, 359]$
    if $signed \neq true$ **then**
      $maxValue \leftarrow 179$ ▷ Unsigned range is $[0, 179]$
      if $directionDiff > maxValue$ **then**
        $directionDiff \leftarrow abs(directionDiff - 359)$
      end if
    end if
    $bin \leftarrow floor(nBins * (directionDiff/maxValue))$
    $relativeEdgeDirections \leftarrow relativeEdgeDirections \cup \{point_x, point_y, bin\}$
  end for
  return $relativeEdgeDirections$
end function

3.4.2 Positional Angle

Positional Angle (PA) encodes the relative angle between the edge segment orientation of an edge point and the position of another edge point as illustrated by figure 3.2. Like RESO, the PA feature is invariant to translation, rotation and scale. Algorithm 2 defines pseudocode for obtaining binned PA features. The algorithm extracts binned PA features for the input image edge point *pt* from input image *img*. PA features are binned as defined by the *nBins* parameter.

Algorithm 2 Binned positional angles

```
1: function positionalAngles(img, pt, nBins)
2:     directions ← intensity direction (0 to 359) degrees for each pixel of img
3:     edgeMap ← binary edge map from the canny edge detector applied to img
4:     ptDirection ← intensity gradient direction for pixel (pt_x, pt_y) from directions
5:     positionalAngles ← {}
6:     for all point ∈ edgeMap do
7:         positionalAngle ← angle between ptDirection and point
8:         bin ← floor(nBins * (angle/359))
9:         positionalAngles ← positionalAngles ∪ {point_x, point_y, bin}
10:    end for
11:    return positionalAngles
12: end function
```

3.5 Ordinal Shape Coding

Firstly this section highlights the available relative constraints for shape description. Secondly the process of obtaining OSCC coding is illustrated and discussed from sampling to representation. The next section illustrates and discusses OSCC correlation approaches.

3.5.1 Relative Constraints

The philosophy behind the OSCC method is that all measurements are relative. With respect to OSCC, relative constraints impose the ordinal consistency of shape edge points while remaining invariant to numeric measurements between them. From a point perspective, two types of relative constraints exist between a reference point and a sample point which are directly related to the relative edge features described within the previous section. The RESO feature can be used as a constraint to enforce the relative ordering of edge segment orientations while allowing the edge segments to be free of exact positions. The complementary PA feature can be used as a constraint to enforce the positional angle between edge segments. An intrinsic property of the PA constraint when binned is that the constraint becomes looser as the radius from the origin increases.

3.5.2 Sampling Area

As identified within Chapter 2, feature based approaches can be categorised as local, semi-local or global with respect to their sampling area. Shape based descriptors

generally fall into the category of semi-local or global to capture information from edges whereas local descriptors concentrate on a dense patch of pixel intensities. Although the OSCC method is not dependent on a specific sampling area, semi-local and global areas are more appropriate as they capture a larger amount of shape information as opposed to local areas. Ultimately, sample area selection is an application specific parameter.

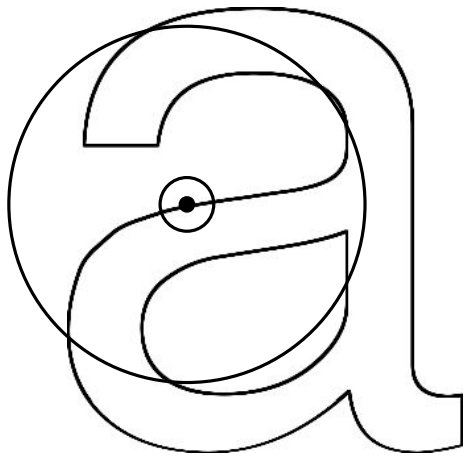


Figure 3.4: Illustration of local, semi-local and global sampling areas. An interest point marked as a solid circle. The local sampling area is between the interest point and the first ring. The semi-local sampling area is between the interest point and the second ring. The global sampling area is the space that covers the whole object

Although a strict definition does not exist between local and semi-local sampling areas, figure 3.4 roughly illustrates the sampling area categories around an interest point which is marked as a solid circle. Firstly, the space within the first ring around the interest point illustrates a local area which focuses on the immediate neighborhood around the interest point. Secondly, space between the second ring and the interest point illustrates a semi-local area which covers more shape information when compared to the local area. Lastly, the space which covers the whole object falls within the global category.

3.5.3 Sampling Decomposition and Approximation

The OSCC method is based on encoding ordinal features extracted from the point perspective of shape defined in section 3.4. Binned RESO and PA ordinal features loosen their associated constraints and also reduce complexity for both the representation and correlation of the OSCC descriptor defined in the next section.

Decomposition of the two dimensional binned relative point perspective of shape into a sequence of one dimensional signals is undertaken to further reduce complexity. A sampling pattern of concentric rings around a point applied to its relative perspective of shape is used, where an individual ring is mapped to a one dimensional signal. The point perspective of a whole shape contains information about every edge point in the image relative to a single point. Thus, if a point perspective of the whole shape was taken for each edge point, then sampling would have a worse case of $\mathcal{O}(n^2)$ for both runtime and space complexity (where n is the number of edge points). To overcome this, the point perspective of shape can be approximated by adjusting radii of the concentric ring sampling pattern. This naturally leads to further approximation approaches such as sampling around edge points that are intersected by rings. Pseudocode for the sampling pattern is defined within the next section.

3.5.4 Mapping from Sampling to Coding

Shape edge points that are intersected by the concentric ring sampling around a reference edge point are used to encode its approximate shape perspective. Binned RESO and PA ordinal features defined in sections 3.4.1 and 3.4.2 respectively are used as relative edge point features for translation, rotation and scale invariance. Figure 3.5 (a) visualises concentric sample rings for a reference edge point over its binned RESO feature perspective.

A sequence of features is obtained from a sample ring by tracing around it using a common start angle and direction. The common start angle is set as the orientation of the edge point at the origin of the point perspective to achieve rotational invariance, this is in addition to providing a relative coordinate system to calculate the relative features from. Figure 3.5 (a) visualises the concentric sample rings for a point and (b) shows the corresponding approximate ordinal encoding of RESO features.

As PA features are not included in figure 3.5 (b) for simplicity, a compact visual representation of the RESO feature descriptor is illustrated which excludes spatial information between features of individual rings. As described in section 3.4.2, the PA feature is used to enforce the positional angle of features. Consequently, this

constraint enforces spatial configurations between features of an individual ring as well as features across rings.

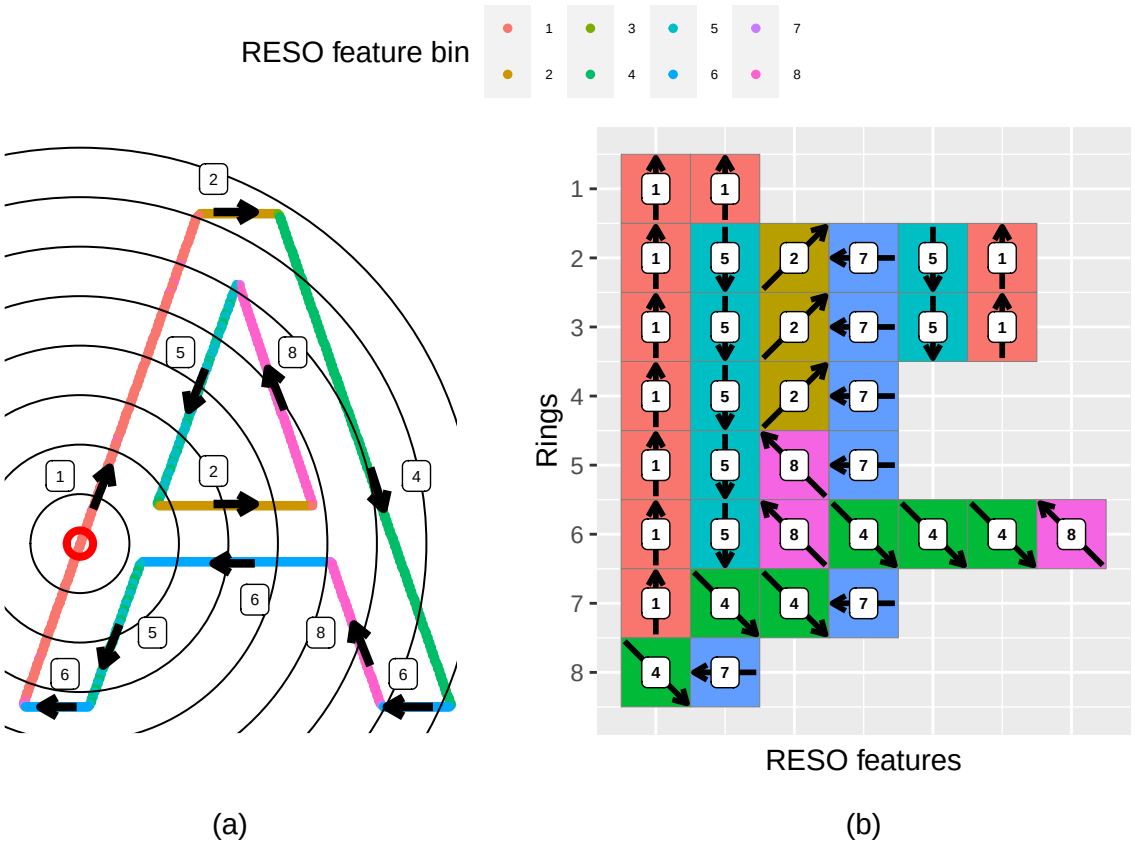


Figure 3.5: Mapping between sampling and point description. (a) visualises concentric sample rings for a point (marked with a red circle) and (b) shows the corresponding OSCC coding with respect to RESO features only for simplicity. Arrows and their corresponding labels indicate the binned RESO features.

Algorithm 3 defines pseudocode for sampling the relative perspective of shape for an edge point *pt* from the image *img* to obtain a relative coding using both RESO and PA features. The *nBins* parameter defines the number of bins to be used for both RESO and PA features. *maxRadius* defines the maximum radius for the concentric ring sampling while *rx* defines the spacing between the rings. Line 23 sorts the ring feature entries by positional angle so that they are ordered clockwise around the ring with the start angle set as the orientation of the edge point at the origin. Finally, an ordered collection of concentric sample rings are returned where a ring contains a collection of ring feature entries which are ordered by positional angle. Referenced functions *relativeEdgeSegmentOrientations* and *positionalAngles* are defined by

algorithms 1 and 2 respectively.

Algorithm 3 Point Perspective Sampling to Coding

```
1: function getPointPerspectiveCoding(img, pt, nBins, maxRadius, rx)
2:     edgeMap ← binary edge map from the canny edge detector applied to img
3:     resoFeatures ← relativeEdgeSegmentOrientations(img, pt, nBins)
4:     paFeatures ← positionalAngles(img, pt, nBins)
5:     radii ← {}
6:     lastRadius ← 0
7:     while lastRadius < maxRadius do
8:         lastRadius ← lastRadius + rx
9:         if lastRadius <= maxRadius then
10:            radii ← radii ∪ lastRadius
11:        end if
12:    end while
13:    sampleRings ← {}
14:    for all r ∈ radii do
15:        ring ← ring with pt as origin and radius of r
16:        intersectedEdgePoints ← intersected edgeMap points with ring
17:        ringPtFeatures ← {}
18:        for all edgePoint ∈ intersectedEdgePoints do
19:            resoFeature ← feature for edgePoint from resoFeatures
20:            paFeature ← feature for edgePoint from paFeatures
21:            ringPtFeatures ← ringPtFeatures ∪ {resoFeature, paFeature}
22:        end for
23:        ringFeatures ← sort ringFeatures by paFeature
24:        sampleRings ← sampleRings ∪ {ringFeatures}
25:    end for
26:    return sampleRings
27: end function
```

As a summary, OSCC retains both the relative order of features for a sample ring and the relative order of sample rings from the origin. When using the RESO features alone, the two dimensional angle and distance decomposition into one dimensional signals loses relative angle alignment of features between the one dimensional signals. The addition of PA features can be used to impose the relative angle alignment constraint within and across rings if required.

3.6 Ordinal Shape Coding Correlation

Correlation between two OSCC codings can be broken down into two steps. Firstly, correlation between individual rings of the codings can be achieved using the order of their features. Secondly, ring sequence correlation can be achieved using the order of matched rings between the two codings. This section links each step to the sequence alignment literature which leads to the discussion of different correlation approaches for each step. Further connections are made between OSCC and the sequence alignment literature resulting in a novel OSCC correlation method.

3.6.1 Ordinal Ring Correlation

An individual ring is represented as a sequence of binned relative edge features. From a string matching point of view, a ring is a string with symbols from the alphabet of all possible relative edge feature bins. Ordinal correlation of elements between two strings can be seen as a sequence alignment problem. Sequence alignment methods are based on the shortest and lowest cost monotonically increasing path through a pairwise cost matrix. The Edit Distance is an example of a sequence alignment method which is based upon minimising the number of insertions, deletions and substitutions of symbols between sequences. Insertions, deletions and substitutions of symbols within an OSCC ring can be caused by a variety of factors such as: signal noise, clutter and occlusion. Furthermore, it is important to note that rings do not have a fixed size.

The Dynamic Time Warping (DTW) method introduces two interesting properties to explore for ordinal ring correlation. The first is the mapping of one token to many tokens without penalty, figure 3.6 illustrates the row token of 4 being mapped to both consecutive column tokens of 4 without penalty as opposed to the edit distance accumulated cost matrix shown by figure 3.7. The second property introduced is the use of a real value based cost matrix between features as shown in figure 3.8 using equation 3.1 as the relative edge feature bin distance function. This is opposed to the

binary cost matrix used by methods such as the Edit Distance.

$$d(a,b,r) = \begin{cases} 2r - abs(a-b), & \text{if } abs(a-b) > r \\ abs(a-b), & \text{otherwise} \end{cases} \tag{3.1}$$

Equation 3.1 defines the pairwise distance function used for the edge feature bin cost matrix as shown by figure 3.8. Where a and b are edge feature bins and r is a constant for feature bin wrap-around. For example, with eight orientation bins from $0-7$ equally mapped to the range of $0-2\pi$ and the wrap around constant r assigned to 4, then the distance between bin 0 and bins $0-7$ is equal to $0,1,2,3,4,3,2,1$.

Algorithms 4 and 5 define pseudocode for obtaining the binary and relative feature bin distance cost matrix between two rings respectively. The first three parameters of the algorithms are the same which are the two rings to compare ($ringA$ and $ringB$) as well as a feature type defined by $featureType$ (either RESO or PA as defined in section 3.4). Algorithm 5 takes an additional parameter r which is defined as the feature bin wrap around constant. The rings to compare are obtained using algorithm 3 which has previously been defined in section 3.5.4. A $getRingFeature$ function is referred to which simply returns a feature for a particular ring given the index and type of the feature.

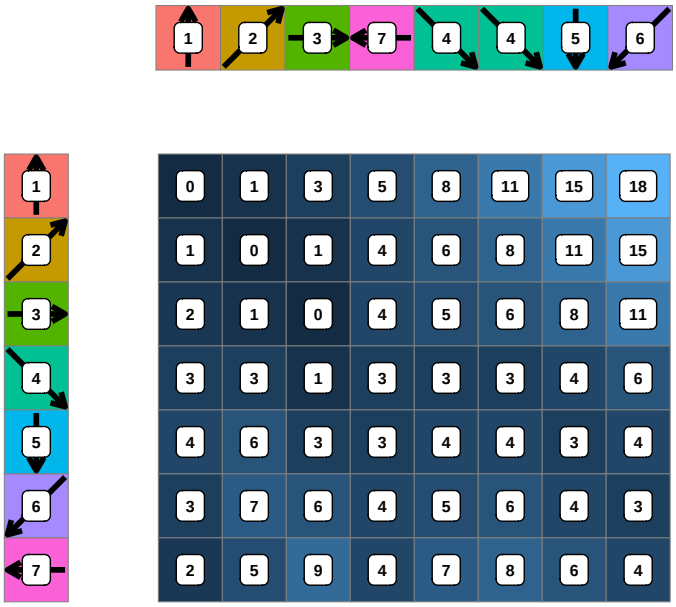


Figure 3.6: DTW accumulated cost matrix from the binary cost matrix between $A = (1, 2, 3, 4, 5, 6, 7)$ and sequence $B = (1, 2, 3, 7, 4, 4, 5, 6)$.

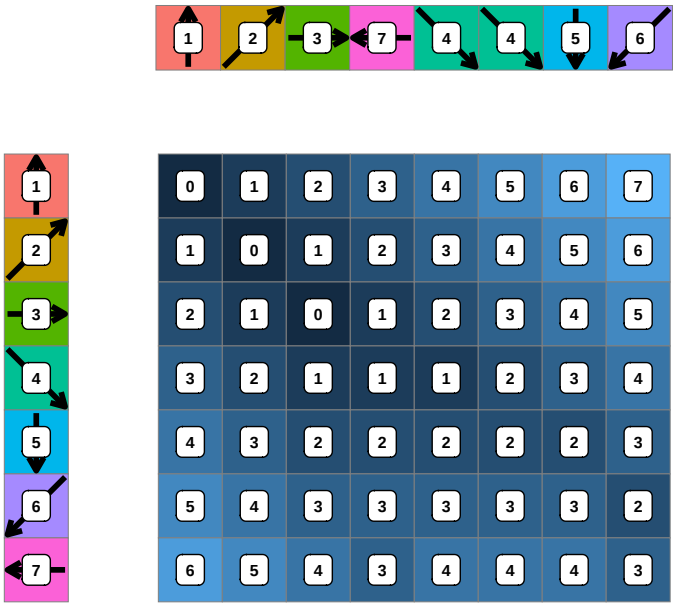


Figure 3.7: Accumulated cost matrix for the Edit Distance between $A = (1, 2, 3, 4, 5, 6, 7)$ and sequence $B = (1, 2, 3, 7, 4, 4, 5, 6)$.

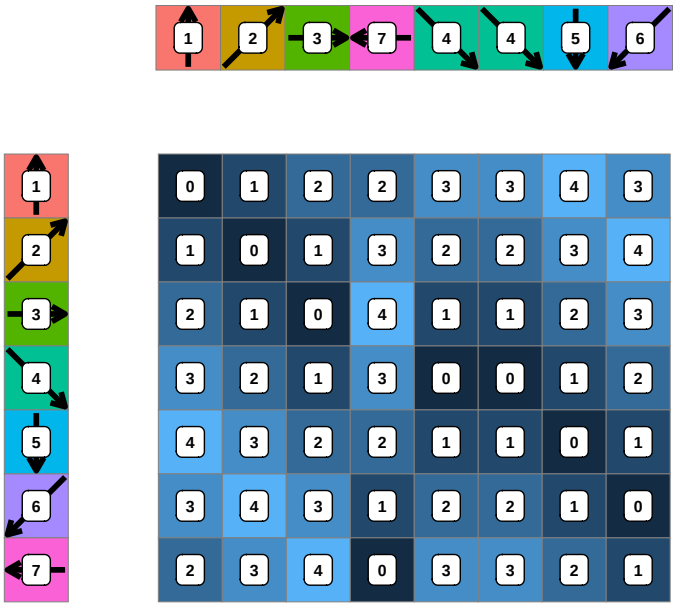


Figure 3.8: Pairwise edge feature bin distance cost matrix for sequences $A = (1, 2, 3, 4, 5, 6, 7)$ and sequence $B = (1, 2, 3, 7, 4, 4, 5, 6)$ using equation 3.1 as the distance function.

Algorithm 4 Binary cost matrix between two rings

```
1:  function getBinaryCostMat(ringA, ringB, featureType)
2:      costMat ← [ringA length][ringB length]
3:      for idxA ← 0 to ringA length do
4:          for idxB ← 0 to ringB length do
5:              ftA ← getRingFeature(ringA, idxA, featureType)
6:              ftB ← getRingFeature(ringB, idxB, featureType)
7:              costMat[idxA][idxB] ← ftA ≠ ftB
8:          end for
9:      end for
10:     return costMat
11: end function
```

Algorithm 5 Relative feature bin cost matrix between two rings

```
1: function getRelFeatureBinCostMat(ringA, ringB, featureType, r)
2:     costMat ← [ringA length][ringB length]
3:     for idxA ← 0 to ringA length do
4:         for idxB ← 0 to ringB length do
5:             ftA ← getRingFeature(ringA, idxA, featureType)
6:             ftB ← getRingFeature(ringB, idxB, featureType)
7:             diff ← abs(ftA − ftB)
8:             if diff > r then
9:                 dist ← 2r − diff
10:            end if
11:            costMat[idxA][idxB] ← diff
12:        end for
13:    end for
14:    return costMat
15: end function
```

Table 3.1 summarises sequence alignment method properties and consequences with respect to the correlation of OSCC rings. Although the Edit Distance is a metric, it could be hindered by clutter, occlusion and noise encoded by the rings. By only comparing common tokens between two rings, the Longest Common Subsequence (LCS) should be better suited to the effects of clutter, occlusion and noise encoded by the rings with the consequence of returning a non-metric distance. Subsequently repeating tokens along a ring can be handled without penalty with the use of the non-metric DTW distance. Furthermore, DTW allows real value distances between binned edge feature tokens to be compared. This property should add robustness with respect to both binning boundary effects and non-rigid transformations.

Method	Properties	Ring correlation consequences
Edit Distance.	As a metric distance, equal penalties are assigned to insertions, deletions and substitutions between tokens.	Clutter, occlusion and noise is associated with token insertions, deletions and substitutions.
Longest Common Subsequence (LCS).	As a non-metric distance, the maximum sequence length is found with respect to tokens that are common between sequences.	Correlation of only common tokens between rings potentially reduces clutter, occlusion and noise effects.
Dynamic Time Warping (DTW).	As a non-metric, a single token within one sequence can be mapped to many tokens within another sequence without penalty. In addition to binary cost matrices, real value cost matrices are also valid input.	Repeated tokens along rings can be handled without penalty in the case that a ring traces an edge. Real value binned edge feature distance can also be used as distance matrix input.

Table 3.1: Ordinal ring correlation methods and consequences

Ring correlation using the Edit Distance cost applies the standard Edit Distance algorithm using equal weights for insertion, deletion and substitution. As defined in section 3.5.1, a ring token can encode one or more relative features identified as RESO and PA. For the Edit Distance ring correlation approach, a match between two tokens is defined as the case where all features between two tokens are exactly the same. Algorithm 6 defines this feature aggregation approach with pseudocode. Furthermore, algorithm 14 (appendix A) defines pseudocode for the Edit Distance accumulated cost matrix. The resulting edit distance value can be found at the bottom right cell of the matrix. $getBinaryCostMat$ is defined by algorithm 4.

Like the Edit Distance, a match between two tokens for the Longest Common Subsequence (LCS) is defined as the case where all features between two tokens are exactly the same as defined by algorithm 6. Algorithm 15 (appendix A) defines pseudocode for the LCS accumulated cost matrix. The resulting LCS value can be found at the bottom right cell of the matrix. Ring correlation using the LCS as a cost leads to various possibilities of cost definition. The simplest approach is to

Algorithm 6 Aggregated Binary Cost Matrix

```
1: function getAggregatedFeatureBinaryCostMat(ringA, ringB)
2:     resoCostMat ← getBinaryCostMat(ringA, ringB, RESO)
3:     paCostMat ← getBinaryCostMat(ringA, ringB, PA)
4:     costMat ← [ringA length][ringB length]
5:     for idxA ← 0 to ringA length do
6:         for idxB ← 0 to ringB length do
7:             resoCost ← resoCostMat(idxA, idxB)
8:             paCost ← paCostMat(idxA, idxB)
9:             costMat[idxA][idxB] ← resoCost + paCost > 0
10:        end for
11:    end for
12:    return costMat
13: end function
```

subtract the LCS value from the smallest ring length, the properties of this approach should allow robustness to noise, clutter and occlusion. A possible disadvantage of this approach is that the same cost for rings with varying length can be returned, for example, given ring $A = (c, e)$ with two states, ring $B = (c, d, e)$ with three states and $C = (a, b, c, d, e, f, g, h)$ with eight states, $LCS(A, B) = 0$ and $LCS(A, C) = 0$. Another alternative is to normalise the LCS cost using equation 3.2 which favours longest common subsequences between sequences of similar size A and B are input sequences. The disadvantage of this alternative is the potential hindering of robustness properties with respect to noise, clutter and occlusion.

$$NLCS(A, B) = \frac{LCS(A, B)}{max(length(A), length(B)} \tag{3.2}$$

Ring correlation using the Dynamic Time Warping uses the real valued cost function defined by equation 3.1. Costs for each feature are summed for the overall cost between two tokens. For example, given token $A = (2, 7)$ and token $B = (7, 5)$ which encode RESO and PA features at index 0 and 1 respectively, using a feature bin wrap-around value of $r = 4$ for each feature, the RESO feature cost would be $d(2, 7, 4) = 3$ and the PA feature cost would be $d(7, 5, 4) = 2$, therefore the cost between tokens A and B is $3 + 2 = 5$. Algorithm 7 defines pseudocode for using this approach to compare two rings ($ringA$ and $ringB$). Furthermore, algorithm 16 (appendix A) defines pseudocode for the DTW accumulated cost matrix. The resulting DTW cost can be found at the bottom right cell of the matrix. $getRelFeatureBinCostMat$ is defined by algorithm 5.

Algorithm 7 Aggregated Relative Feature Bin Cost Matrix

```
1: function getAggregatedFeatureRelCostMat(ringA, ringB)
2:     resoCostMat ← getRelFeatureBinCostMat(ringA, ringB, RESO)
3:     paCostMat ← getRelFeatureBinCostMat(ringA, ringB, PA)
4:     costMat ← [ringA length][ringB length]
5:     for idxA ← 0 to ringA length do
6:         for idxB ← 0 to ringB length do
7:             resoCost ← resoCostMat(idxA, idxB)
8:             paCost ← paCostMat(idxA, idxB)
9:             costMat[idxA][idxB] ← resoCost + paCost
10:        end for
11:    end for
12:    return costMat
13: end function
```

3.6.2 Ordinal Ring Sequence Correlation

The previous section describes the first stage of correlation between OSCC codings which is concerned with the correlation between individual rings. This section describes the second stage which is concerned with the correlation of ring sequences. Ordinal ring sequence correlation is achieved by building upon the pairwise correlation between rings of two OSCC codings by using the order of matched rings. Like ring correlation, ring sequence correlation can also be seen as a sequence alignment problem.

Since ring sequence correlation makes use of ring correlation scores, it is important to note that each sequence alignment method considered for ring correlation in table 3.1 returns a real value correlation score and can either be a metric or non-metric. Furthermore, ring sequences can have a varying number of rings, for example, figure 3.9 shows the sample rings and corresponding OSCC codings for both a point on the image of the letter 'a' and the corresponding point of the image transformed by a scale factor of 2.

Scale invariant properties of ring sequence matching requires a single ring in one sequence to be able to be mapped to one or many consecutive rings in another sequence. With this property, along with its ability to handle real values and sequences of varying sizes, DTW is used as the ordinal correlation method for ring sequence correlation. Consequently, ordinal ring sequence correlation is non-metric.

Ordinal ring sequence correlation is illustrated as follows: figure 3.10 shows the pairwise cost matrix between rings defined in figure 3.9 (b) and (d) using the Edit Distance. Alternative ordinal ring correlation methods can also be used as previously defined by table 3.1. Figure 3.11 shows the corresponding accumulated cost matrix

output from DTW where the resulting cost can be found at the bottom right cell of the matrix. More specifically, algorithm 8 defines pseudocode for both of these steps which takes two ring sequences to compare (rsA and rsB) as input parameters - an individual ring sequence can be obtained by the output of algorithm 3 and $getRingEditDistanceAccCostMat$ is defined by 14 (appendix A).

The same underlying approach of sequence alignment is applied to both ring and ring sequence ordinal correlation steps of OSCC. This leads to a sequence alignment hierarchy: the first stage creates a pairwise cost matrix from relative edge features of two rings, an accumulated cost matrix is then created to find the best alignment score of the two rings. The second stage takes two ring sequences and creates a pairwise cost matrix using the first stage for the ring correlation scores, an accumulated cost matrix is then created to find the ring sequence correlation score for the given ring sequences.

The ring sequence correlation approach leads to a wide range of application specific customisation. For example, DTW parameters such as step size, step weights and global constraints can be modified as well as the underlying pairwise ring correlation method. Consequently, accumulated cost matrix weights can be used to decrease the sensitivity of ring correlation scores as the corresponding concentric sample rings get further away from the reference sample point. Furthermore, global constraints can be used to ensure that rings at the beginning of one sequence are not aligned with rings at the end of another sequence, therefore controlling scale invariance properties. Using the same underlying approaches for both stages in OSCC ordinal correlation hierarchy leads to desirable properties such as algorithm reuse and the retention of intrinsic algorithmic features. The next section uses the intrinsic algorithmic features from the sequence alignment hierarchy to propose a further OSCC correlation approach.

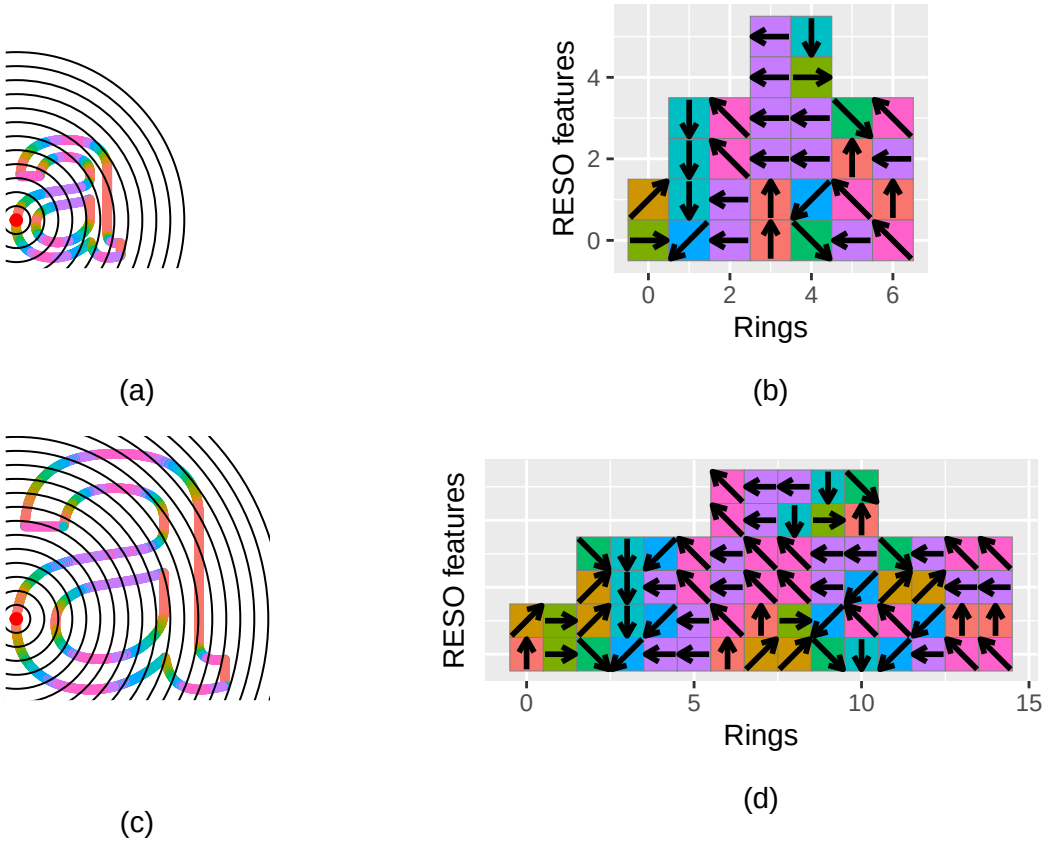


Figure 3.9: Mapping between sampling and point description for 'a' and 'a' scaled by 2. (a) shows concentric sample rings for a point and (c) visualises concentric sample rings for the same point after a scale of 2 has been applied. (b) and (d) show the corresponding OSCC coding for (a) and (c) respectively.

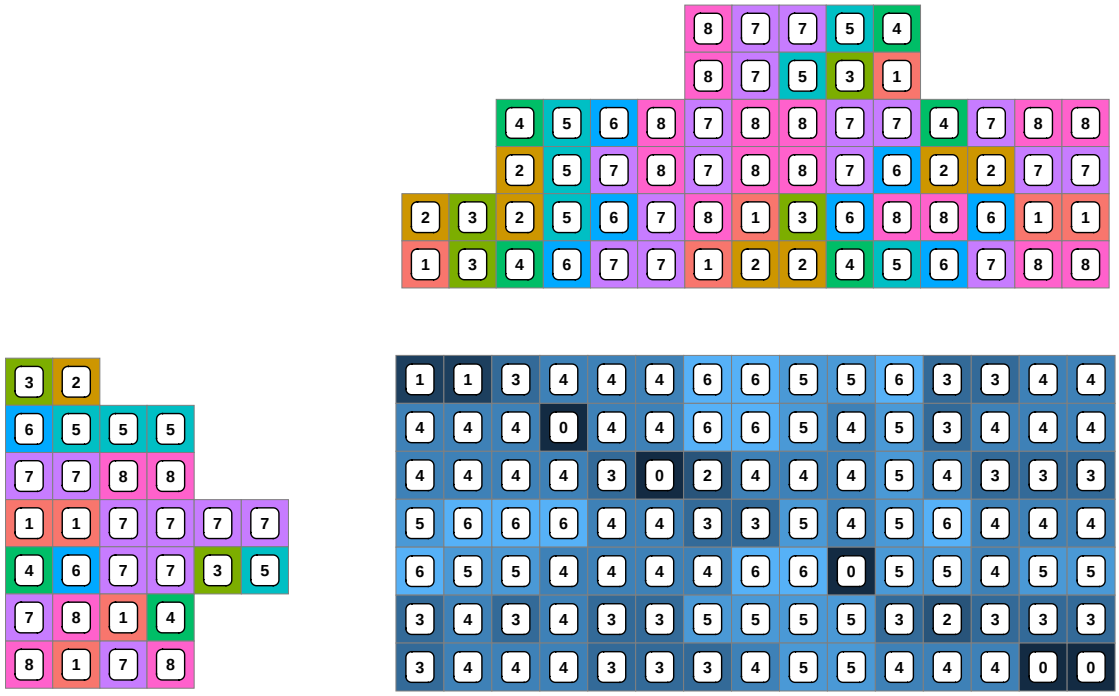


Figure 3.10: Pairwise cost matrix between the rings defined in figure 3.9 (b) and (d))

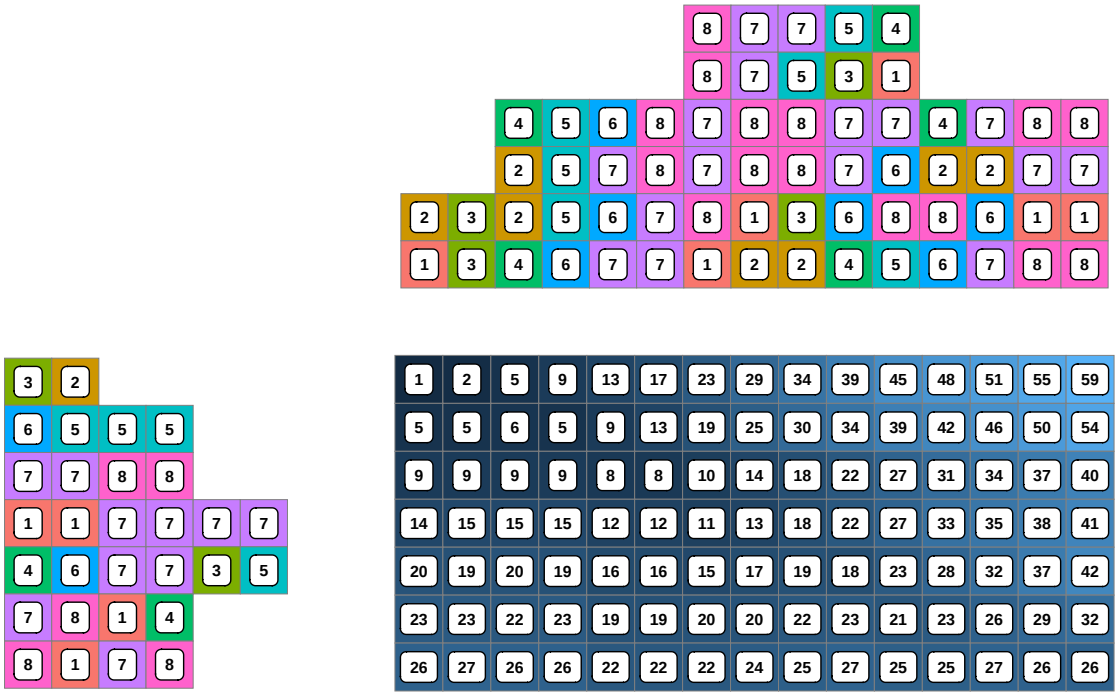


Figure 3.11: DTW accumulated cost matrix from the cost matrix between rings defined in figure 3.9 (b) and (d)). See figure 3.10 for the original cost matrix

3.6.3 Ring Sequence Alignment

In the previous section, correlation of OSCC coding is defined as a hierarchy with two stages: ring correlation and ring sequence correlation. Both stages use sequence alignment methods which can be used to return the sequence alignment path and not just the cost. This section explores the use of sequence alignment paths for OSCC coding correlation. Algorithm 9 defines pseudocode for obtaining a sequence alignment path given an accumulated cost matrix.

Algorithm 8 Ring Sequence Dynamic Time Warping Accumulated Cost Matrix

```
1: function getRingSequenceDTWAccCostMat(rsA, rsB)
2:     rsCostMat ← [rsA length][rsB length]
3:     for ringAIdx ← 1 to rsA length do
4:         for ringBIdx ← 1 to rsB length do
5:             accMat ← getRingEditDistanceAccCostMat(ringA, ringB)
6:             ringALength ← rsA[ringAIdx] length
7:             ringBLength ← rsB[ringBIdx] length
8:             cost ← accMat[ringALength][ringBLength]]
9:             rsCostMat[ringAIdx][ringBIdx] ← cost
10:        end for
11:    end for
12:    accMat ← rsCostMat
13:    for idxA ← 1 to rsA length do
14:        accMat(idxA, 0) ← accMat(idxA − 1) + accMat(idxA, 0)
15:    end for
16:    for idxB ← 1 to rsB length do
17:        accMat(0, idxB) ← accMat(0, idxB − 1) + accMat(0, idxB)
18:    end for
19:    for idxA ← 1 to rsA length do
20:        for idxB ← 1 to rsB length do
21:            sub ← accMat[idxA − 1][idxB − 1]
22:            ins ← accMat[idxA][idxB − 1]
23:            del ← accMat[idxA − 1][idxB]
24:            accMat(idxA, idxB) ← accMat[idxA][idxB] + min(sub, ins, del)
25:        end for
26:    end for
27:    return accMat
28: end function
```

Algorithm 9 Accumulated Cost Matrix Traceback

```
1: function getAccCostMatTraceback(accMat)
2:     currentRow ← num accMat rows −1
3:     currentCol ← num accMat cols −1
4:     path ← path{{currentRow, currentCol}}
5:     while currentRow > 0 AND currentCol > 0 do
6:         topLeft ← accMat[currentRow − 1][currentCol − 1]
7:         top ← accMat[currentRow][currentCol − 1]
8:         left ← accMat[currentRow − 1][currentCol]
9:         if topLeft <= top AND topLeft <= left then
10:            currentRow ← currentRow − 1
11:            currentCol ← currentCol − 1
12:        else
13:            if top < left then
14:                currentRow ← currentRow − 1
15:            else
16:                currentCol ← currentCol − 1
17:            end if
18:        end if
19:        path ← path ∪ {currentRow, currentCol}
20:    end while
21:    while currentRow > 0 do
22:        currentRow ← currentRow − 1
23:        path ← path ∪ {currentRow, currentCol}
24:    end while
25:    while currentCol > 0 do
26:        currentCol ← currentCol − 1
27:        path ← path ∪ {currentRow, currentCol}
28:    end while
29:    return path
30: end function
```

For the first stage of ring correlation, each step within a ring alignment path defines a correspondence between the originally sampled locations of the aligned tokens. Figure 3.12 illustrates a ring correlation alignment path between ring $A = (1, 2, 3, 4, 5, 6, 7)$ and ring $B = (1, 2, 7, 4, 4, 5, 6)$ where DTW has been applied to the binary cost matrix between A and B. For this example, correspondences from tokens of B to tokens of A would be $B' = ((B_0, A_0), (B_1, A_1), (B_2, A_2), (B_3, A_3), (B_4, A_4), (B_5, A_5), (B_6, A_6))$ as identified by the alignment path. A lookup table can then be used to identify the sampled locations for each token which in turn can be used to retrieve geometric correspondences. This process is defined by algorithm 10 which is described in the following paragraphs.

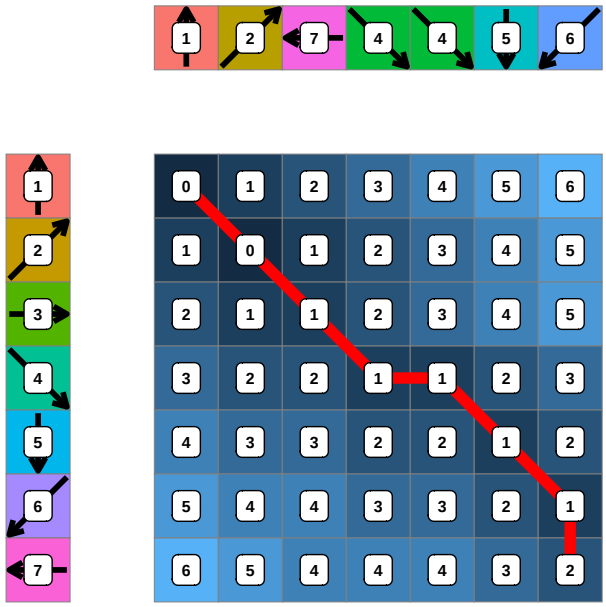


Figure 3.12: DTW accumulated cost matrix and alignment path for sequences $A = (1, 2, 3, 4, 5, 6, 7)$ and $B = (1, 2, 7, 4, 4, 5, 6)$.

For the second stage of ring sequence correlation, each step within the ring sequence alignment path defines a mapping between two rings defined by the underlying cost matrix. Figure 3.13 illustrates the ring sequence alignment path between the OSCC coding for a point on an image of the character 'a' and the corresponding point on

an image of the character 'a' scaled by 2. The ring sequence alignment path can be used to return mappings between rings which can in turn be used to return mappings between ring tokens identified by ring alignment. Figure 3.14 shows the resulting correspondences from the ring sequence alignment.

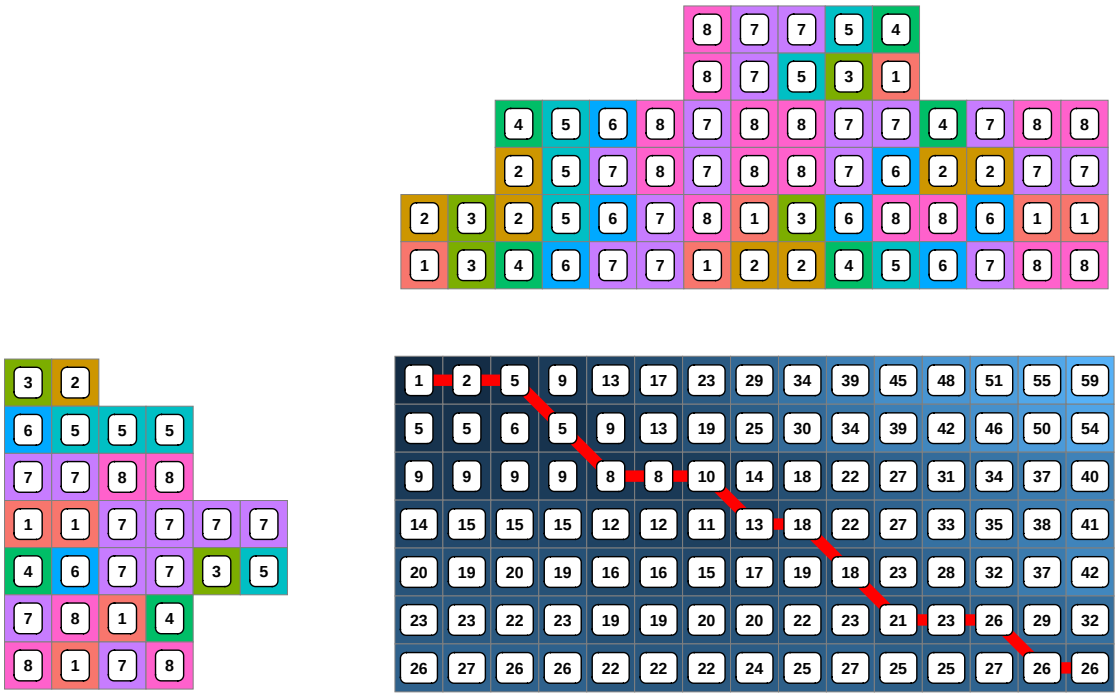


Figure 3.13: DTW accumulated cost matrix and alignment path from the cost matrix between rings defined in figure 3.9 (b) and (d)). See figure 3.10 for the original cost matrix

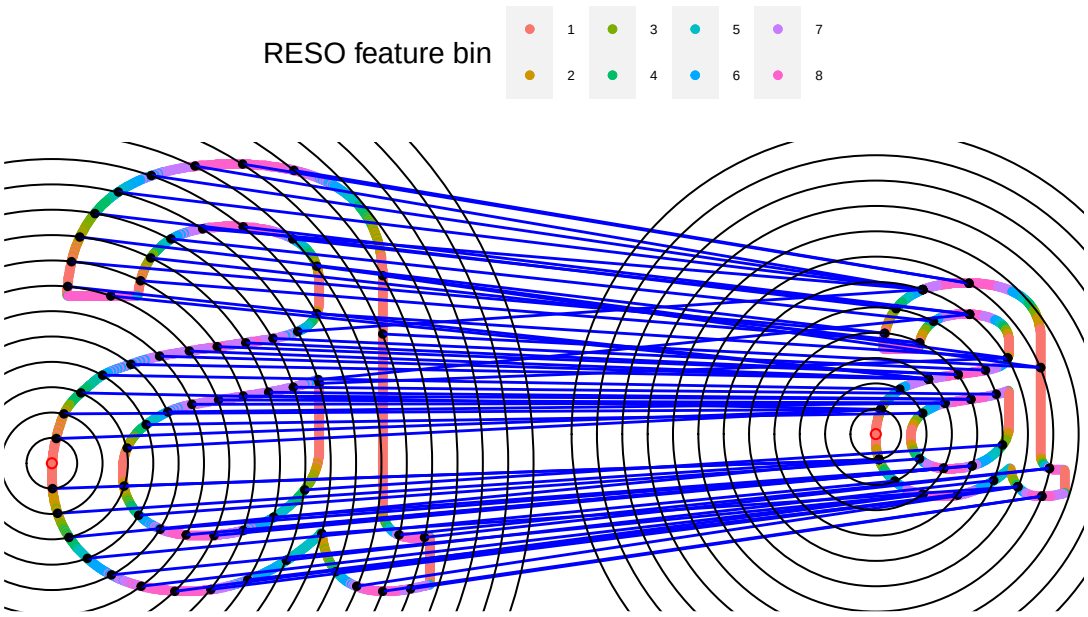


Figure 3.14: Correspondences identified by the proposed OSCC coding and correlation sequence alignment approach applied to images of figure 3.9 (a) and (c)

Algorithm 10 defines pseudocode for both stages of ring correlation alignment and the ring sequence correlation alignment (a helper function *getRingTokenCoords* is used which simply returns the original sample coordinates for a particular token of a ring). First, the ring sequence alignment path is obtained between two input ring sequences (rsA and rsB). For each alignment within the path, the two underlying rings ($ringA$ and $ringB$) are used to output an alignment path between their tokens. Each entry within the token alignment path is then used to obtain a geometric correspondence by using lookup tables which hold the sample coordinates associated with each token of each ring. *getRingSequenceDTWAccCostMat* is defined by algorithm 8, *getAccCostMatTraceback* is defined by algorithm 9 and *getRingEditDistanceAccCostMat* is defined by 14 (appendix A).

Algorithm 10 OSCC Sequence Alignment Hierarchy

```
1: function getOsccSequenceAlignmentHierarchy(rsA, rsB)
2:     rsAccMat ← getRingSequenceDTWAccCostMat(rsA, rsB)
3:     rsAlignmentPath ← getAccCostMatTraceback(rsAccMat)
4:     corres ← {}
5:     for rsRingAlignment in rsAlignmentPath do
6:         ringAIdx ← rsRingAlignment[0]
7:         ringBIdx ← rsRingAlignment[1]
8:         ringA ← rsA[ringAIdx]
9:         ringB ← rsB[ringBIdx]
10:        ringAccMat ← getRingEditDistanceAccCostMat(ringA, ringB)
11:        ringAlignmentPath ← getAccCostMatTraceback(ringAccMat)
12:        for ringTokenAlignment in ringAlignmentPath do
13:            ringATokenIdx ← ringTokenAlignment[0]
14:            ringBTokenIdx ← ringTokenAlignment[1]
15:            src ← rsA.getRingTokenCoords(ringAIdx, ringATokenIdx)
16:            dst ← rsB.getRingTokenCoords(ringBIdx, ringBTokenIdx)
17:            corres ← corres ∪ {src, dst}
18:        end for
19:    end for
20:    return corres
21: end function
```

As explained in section 3.5.4, when only using RESO features, sampling decomposition of the two dimensional angle and distance point perspective into one dimensional signals loses positional angle information across rings. Figure 3.14 shows the correspondences identified by the proposed OSCC coding correlation sequence alignment approach applied to figure 3.9 (a) and (c). Consequently, ring sequence alignment

information can be used to return location information from the aligned tokens of the aligned rings.

3.7 Parameters

This section summarises the available parameters that have been identified throughout this chapter for the OSCC coding and correlation implementation. Furthermore, additional parameters are also proposed and discussed. The parameters are categorised into the three ordered groups of sampling, ring coding / correlation and ring sequence coding / correlation.

Sampling parameters are concerned with the sampling pattern applied to a point perspective. Table 3.2 summarises the available sampling parameters - a sample point is defined as an edge pixel that is intersected by a ring (see section 3.5.3). The simple ring sampling pattern described within section 3.5.4 has uniformly spaced rings without any sample regions. This simple sampling approach can be modified to match the wide range of sampling patterns proposed within the literature. To achieve this, parameters for ring spacing and sample point region are required as defined in table 3.2.

Name	Description	Variations
Ring spacing	A concentric ring spacing function to control the radii of sample rings	Uniform or log-polar spacing
Sample point region	A sample point region is defined as an area around a sample point.	A sample point region can be assigned an area which has either a fixed or varying size with respect to the ring distance from the origin.
Sample point area	Sampling area for a particular sample point	Global or semi-local (if semi-local then a radius value will be specified)

Table 3.2: Sampling Parameters

Ring coding and correlation comprises of binning values obtained by the sampling pattern and the correlation of rings. Table 3.3 summarises the available ring coding

and correlation parameters. Either exact binning can be used as defined in section 3.5.3 or another binning approach such as soft binning can be used to reduce artefacts produced by bin boundaries - of course the selected ring correlation method would need to be able to handle the soft binning type of input. Ring correlation methods of Edit Distance, Longest Common Subsequence and Dynamic Time Warping which all have sequence alignment properties are identified within section 3.6.1. The variety of ring correlation methods results in the subsequent parameter selection for the selected method.

Name	Description	Variations
Binning	Binning of input feature values is required to reduce model size	Exact or soft binning can be used with n bins. Soft binning requires further weighting parameters
RESO features	Use of RESO features	True or False. If true then signed or unsigned.
PA features	Use of PA features.	True or False. If true then signed or unsigned.
Ring Correlation Method	Ring correlation methods find lowest cost monotonically increasing path between two rings	Edit Distance, Longest Common Subsequence and Dynamic Time Warping variations are defined and described within section 3.6.1
Rotation Invariance	Ring correlation rotation invariance	True or False

Table 3.3: Ring Correlation Parameters

Ring sequence coding and correlation takes two sequences of rings along with their pairwise ring correlation costs and applies the Dynamic Time Warping sequence alignment method for a correlation score. Table 3.4 summarises the available parameters. As described by section 3.6.2, the lowest cost monotonically increasing path with respect to ring indices of the two ring sequences is calculated. Dynamic Time Warping parameters, constraints and penalties include step size and step weights in addition to global cost matrix search space constraints. Default values

of 1 are used for both the step size and weights for simplicity. However, with the goal of constraining ring matches such that rings at the start of one sequence are not able to match rings at the end of another sequence, the Sakoe-Chiba band is used with horizontal and vertical dimensions of 0.5 with respect to the corresponding ring sequence lengths. This global constraint can also be used to control scale invariance.

Name	Description	Variations
Global Search Space Constraints	Cost matrix global search space constraints to enforce that parts of one ring sequence can't be matched with parts of another ring sequence. This can be used to control scale invariance.	Sakoe-Chiba band with horizontal and vertical dimensions proportionally assigned to corresponding horizontal and vertical ring sequences. A proportion of 0.5 is used as default.
Step Size	Cost matrix step size can affect the final match cost. A small step size has the potential of being too sensitive to noise while a large step size has the potential of missing discriminative information	Step size of 1 is used as default
Step Weights	Weights assigned to the step pattern can be used to direct a warping path. It is also possible to assign weights such that the overall cost is more sensitive to ring matches at the beginning of the ring sequences and less sensitive towards the end which corresponds to higher sensitivity around a sample point origin	Step weights of 1 are used as default

Table 3.4: Ring Sequence Correlation Parameters

3.8 Computational Complexity

A hierarchical sequence alignment structure is at the core of the OSCC method. The sequence alignment approaches used for both ring and ring sequence correlation are based on finding the lowest cost monotonically increasing path through a cost matrix. As reviewed in section 2.5.2, Dynamic Programming can be used to achieve a time and space complexity of $\mathcal{O}(XY)$ where X and Y are the lengths of the query and template sequences respectively. With the tradeoff of losing the actual alignment, the sequence alignment cost can be obtained with a reduced space complexity of $\mathcal{O}(min(X,Y))$ by memorising the previous row or column of the accumulated cost matrix. Global constraints can be used to reduce both time and space complexity such as a diagonal cost matrix band resulting in time and space complexity of $\mathcal{O}(B \cdot max(X,Y))$ where B is the band width [62].

Given two ring sequences A and B with lengths of X and Y respectively, a ring sequence cost matrix is created resulting in $\mathcal{O}(XY)$ for both time and space complexity. As previously described, this time and space complexity can both be reduced to $\mathcal{O}(B \cdot max(X,Y))$ when using a horizontal global constraint band. Each entry in the cost matrix is populated with the associated pairwise ring correlation result which has a time and space complexity of $\mathcal{O}(CD)$ where C and D are the lengths of the two rings to be correlated respectively. For the case where an alignment path is required and the horizontal global band constraint is not used, time complexity results in:

$$\mathcal{O}(r(A) \cdot r(B) \cdot t(A) \cdot t(B))$$

Where $r(A)$ returns the number of rings within the ring sequence of A and $t(A)$ returns the total number of tokens across all rings of A. Furthermore, space complexity results in

$$\mathcal{O}(t(A) \cdot t(B))$$

3.9 Summary

In this chapter, we illustrated the derivation and implementation of OSCC following the relativistic principles of Austin's biologically inspired relativistic theory of object recognition [1]. More specifically, we illustrated the pre-processing and sampling stages used by a prototype initially implemented by Austin which resulted in an ordinal shape coding. Original work of this thesis is then presented with a proposed sequence alignment hierarchy approach coined OSCC.

Each step of the derived implementation was illustrated along with the identification of available parameters. Approaches from the sequence alignment literature are linked to the relativistic correlation tasks within the implementation. This link does not only result in a variety of possible correlation approaches - the intrinsic properties of sequence alignment along with how OSCC is structured leads to novel OSCC correlation methods. These methods fit in with the school of thought behind the relativistic theory of object recognition. More specifically, a sequence alignment hierarchy is proposed for the correlation of OSCC codings. Additionally, a second method is proposed that makes use of the intrinsic alignment properties of the sequence alignment hierarchy which can be used to identify correspondence from the alignment.

The next chapter evaluates OSCC with respect to shape descriptor evaluation requirements using ground truth point correspondences. Evaluation requirements include similarity, affine and perspective transformations as well as clutter and occlusion.

Chapter 4

Synthetic Point Correspondence Evaluation

4.1 Introduction

In the previous chapter, we illustrated the derivation of the OSCC method along with a proposed hierarchical sequence alignment method. In this chapter, we evaluate OSCC with respect to point descriptor evaluation requirements using ground truth point correspondences. Evaluation requirements include similarity, affine and perspective transformations as well as clutter and occlusion. A point correspondence evaluation framework is defined that uses the Mean Average Precision (mAP) metric to report results - a standard metric used in the literature for evaluating point descriptors between images with known transformations. In the next chapter, we build upon this chapter by undertaking an object level evaluation of the OSCC method.

An alphabet character dataset is defined and used throughout the evaluations to allow comparisons between baseline results and results obtained from a variation. For each evaluation, results are presented with a high level Mean Average Precision score summarising the performance of a descriptor variation applied to the alphabet dataset with respect to a transformation. Lower level results are also presented to show how each individual alphabet character contributes to the overall score.

4.2 Experiment Design

First of all, point descriptor requirements are defined for the evaluation. The alphabet dataset is then introduced which is used throughout this evaluation chapter. A method for extracting ground truth correspondences from the alphabet dataset is described along with the Mean Average Precision evaluation metric. Exact method variation parameters are defined as well as dataset variations used for the evaluation. Finally, the overall evaluation approach is summarised.

4.2.1 Requirements

Table 4.1 defines evaluation requirements for the OSCC method. The evaluation requirements specify general point descriptor properties which are important factors to be taken into consideration when solving feature based computer vision problems. It should be noted that object level recognition tasks may require subsets of the outlined requirements. Tradeoffs between requirements are also likely when applied to application specific tasks. Therefore it is important to define and evaluate the OSCC method against common point descriptor requirements.

Requirements PT_1 to PT_3 build upon each other by adding degrees of freedom from similarity transforms to perspective transforms. Similarity transform invariance (PT_1) is important for shape descriptors as the actual shape remains the same due to the angles between lines and ratios remaining constant. Robustness against higher level transformations (PT_2 and PT_3) are important while keeping in mind that the actual underlying shape can change. Both clutter robustness PT_4 and occlusion robustness PT_5 are important for a wide range of computer vision problems.

ID	Name	Description
PT_1	Similarity transforms invariance	Translation, reflection, rotation and scale invariant
PT_2	Affine transforms robustness	Robust to shearing
PT_3	Perspective transforms robustness	Robust to perspective transforms
PT_4	Clutter robustness	Robust to clutter
PT_5	Occlusion robustness	Robust to occlusion

Table 4.1: OSCC evaluation requirements

4.2.2 Alphabet Dataset

A synthetic alphabet dataset is used within subsequent sections of this chapter to evaluate descriptors against the requirements defined in table 4.1. A synthetic dataset is used so that independent variables can be defined and controlled. In this context, an independent variable is defined as an individual transformation. All transformation functions were used from OpenCV 3.0. Further details about the dataset are presented in appendix B such as the number of edge points for each character.

The synthetic alphabet dataset (see figure 4.1) comprises of the lowercase FreeSans characters from 'a' to 'z'. All individual characters have been uniformly scaled such that the maximum dimension (width or height) is equal to 150 pixels. Amongst the various shapes provided by the FreeSans font, it is notable that the character pairs of (b, d) and (p, q) are examples of horizontal reflection while the character pairs of (b, p) and (d, q) are examples of vertical reflection. Furthermore, the character pairs of (b, q), (d, p) and (n, u) are examples of 180 degree rotation. The characters 'l', 'o', and 'x' are identical when rotated by intervals of 180 degrees. Overall, the mean number of edge points per character is 400.23 ± SD 103.43.

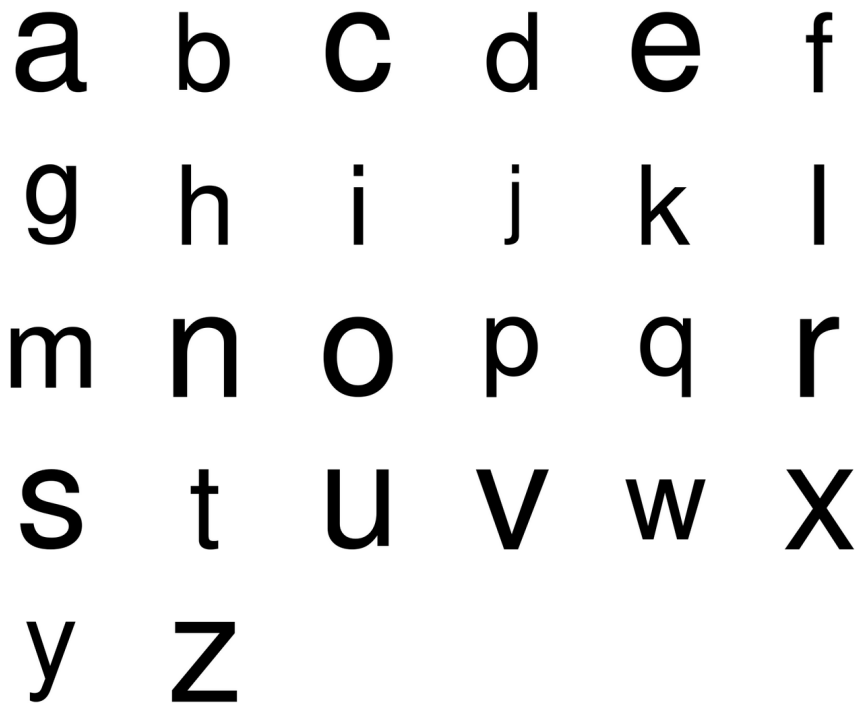
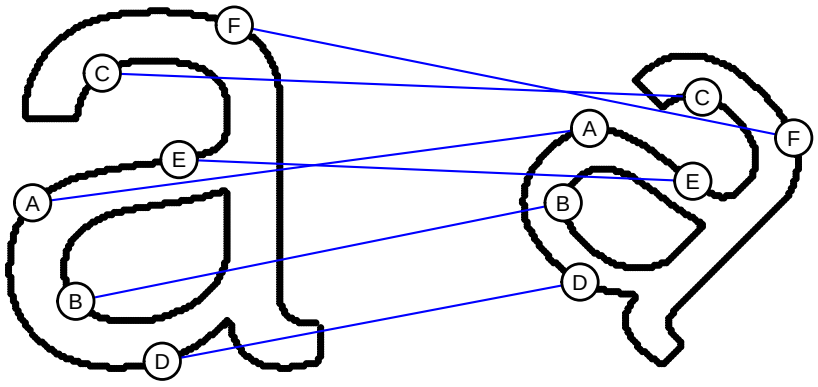


Figure 4.1: FreeSans alphabet dataset. Each character is uniformly scaled such that the maximum dimension is equal to 150 pixels.

4.2.3 Ground Truth Point Correspondences

Given a transformation function, ground truth correspondences are obtained between a reference image and a transformed image. Correspondence source and destination points are identified as keypoints in the reference and transformed images respectively. Figure 4.2 illustrates the process of obtaining ground truth correspondences. A similarity transformation of 0.75 uniform scaling and 45 degree clockwise rotation is applied to the reference image to obtain the transformed image. The Canny edge detection algorithm is applied to both the reference image and transformed image (figure 4.2 left and right respectively) to extract shape information. If an edge point within the transformed image can be mapped to an edge point within the reference image using the inverse of the given transformation, then a ground truth correspondence is extracted. A constraint is imposed such that a single transformed point can only have a single ground truth correspondence with respect to the reference image. See appendix C for further algorithm and dataset details.



(a)

Figure 4.2: Ground truth correspondence identification between a reference image and a resulting image from 0.75 uniform scaling and 45 degree clockwise rotation. The left and right images show the output of the Canny edge detector for the reference and transformed images respectively. Six arbitrarily selected correspondences are shown for illustration purposes. A correspondence is identified by an identically labelled source keypoint in the reference image and a transformed keypoint in the transformed image.

Six ground truth correspondences are arbitrarily selected for illustration purposes in figure 4.2 labelled from "A" to "F". For example, the source of correspondence "F" is indicated by the reference image keypoint labelled as "F" and the destination is indicated in the transformed image keypoint labelled as "F". The evaluation of a point descriptor with respect to the ground truth correspondences is based on the creation of a pairwise cost matrix between keypoint descriptors of the transformed and reference images (the number of keypoints used is defined by experiment parameters). Transformed keypoints are ordered along the rows and reference keypoints are ordered along the columns where the ground truth keypoint costs lie along the main diagonal, see figure 4.3 (a). The Mean Average Precision metric is the standard approach for evaluating the cost matrix which operates on the cost matrix with row-wise cost ordering, see 4.3 (b). The next section defines and illustrates the Mean Average Precision metric.

$$\begin{pmatrix} \mathbf{0} & 10 & 20 & 15 & 50 & 40 \\ 10 & \mathbf{0} & 30 & 25 & 10 & 10 \\ 20 & 30 & \mathbf{5} & 40 & 25 & 60 \\ 15 & 25 & 40 & 10 & \mathbf{4} & 20 \\ 50 & 10 & 25 & 4 & \mathbf{0} & 30 \\ 40 & 10 & 60 & 20 & 30 & \mathbf{0} \end{pmatrix} \qquad \begin{pmatrix} \mathbf{0} & 1 & 3 & 2 & 5 & 4 \\ 1 & \mathbf{0} & 3 & 2 & 1 & 1 \\ 1 & 3 & \mathbf{0} & 4 & 2 & 5 \\ 2 & 4 & 5 & 1 & \mathbf{0} & 3 \\ 5 & 2 & 3 & 1 & \mathbf{0} & 4 \\ 4 & 1 & 5 & 2 & 3 & \mathbf{0} \end{pmatrix}$$

(a) (b)

Figure 4.3: Example of a keypoint descriptor cost matrix (a) transformed into a row-wise rank order matrix (b). Transformed keypoints are ordered along the rows and reference keypoints are ordered along the columns. Ground truth correspondence costs are along the main diagonal. Row-wise best matches are highlighted as bold.

4.2.4 Mean Average Precision Metric

Information Retrieval systems provide an ordered list of relevant matches for a given query. Precision and recall based metrics are appropriate for evaluating Information Retrieval systems with unbalanced datasets. The task of evaluating ground truth point correspondences between a reference image and a transformed image presents itself as the evaluation of an Information Retrieval system with unbalanced data. Each keypoint in the transformed image is used as an individual query to return an ordered list of matches with respect to the keypoints in the reference image. A single

ordered list contains one positive ground truth correspondence amongst many negative correspondences as illustrated by figure 4.3 (b).

Mean Average Precision (mAP) is a standard metric used in the literature for evaluating local point descriptors between images with known transformations [7]. The Average Precision (AP) is used to summarise the Precision-Recall curve, see equation 4.3. The Average Precision is calculated for each keypoint query between the transformed and reference image. The mAP calculates the mean from all of the calculated AP scores related to the individual keypoint queries.

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

$$AP = \sum_{n} (R_n - R_{n-1}) P_n \tag{4.3}$$

The Average Precision function from scikit-learn 0.20.2 is used throughout this thesis which is defined by equation 4.3 [65]. Where n is the rank order list of match costs in ascending order. P_n and R_n are precision and recall values for the ordered match costs list index n respectively.

Experiments in the following sections are setup so that a single ground truth correspondence exists between a transformed image keypoint and a reference image keypoint. As a result, the previously defined Average Precision (AP) metric will only be taking a single precision score as input. Table 4.2 illustrates all possible query rank variations and output AP scores to gain an intuitive understanding of the metric. The Query ID column represents a query for a transformed keypoint "A". Query ranks are shown for the reference keypoints "A" to "F" and the AP column shows the AP for the query. The reference keypoint "A" is the ground truth keypoint for each query. As an example, query "A2" has an AP score of 0.33, since a single ground truth correspondence exists, the AP will be equal to the precision for the ground truth query rank which is 0.33.

Figure 4.4 demonstrates the format of box plots used within this chapter given the values of $(0, 5, 10, 15, 20, 25, 30, 60)$, The box plot shows that the values have a median of 17.5 and a mean of 20.63. The first, second and third quartiles are 8.75, 17.50 and 26.25 respectively. An outlier exists with a value of 60 as it is more than 1.5 of the interquartile range away from the closest interquartile range boundary. The standard error is 6.64, therefore the upper error bar is at 27.27 and the lower error bar is at 13.98.

	Query Rank						
Query ID	A	B	C	D	E	F	AP
A0	0	1	2	3	4	5	1.00
A1	1	2	3	4	5	0	0.50
A2	2	3	4	5	0	1	0.33
A3	3	4	5	0	1	2	0.25
A4	4	5	0	1	2	3	0.20
A5	5	0	1	2	3	4	0.17

Table 4.2: Query rank permutations for AP inspection. Each query shows the AP score for a permutation of the reference image keypoint query ranks. Reference image keypoint 'A' is the ground truth keypoint across the permutations.

```
Warning: `fun.y` is deprecated. Use `fun` instead.
```

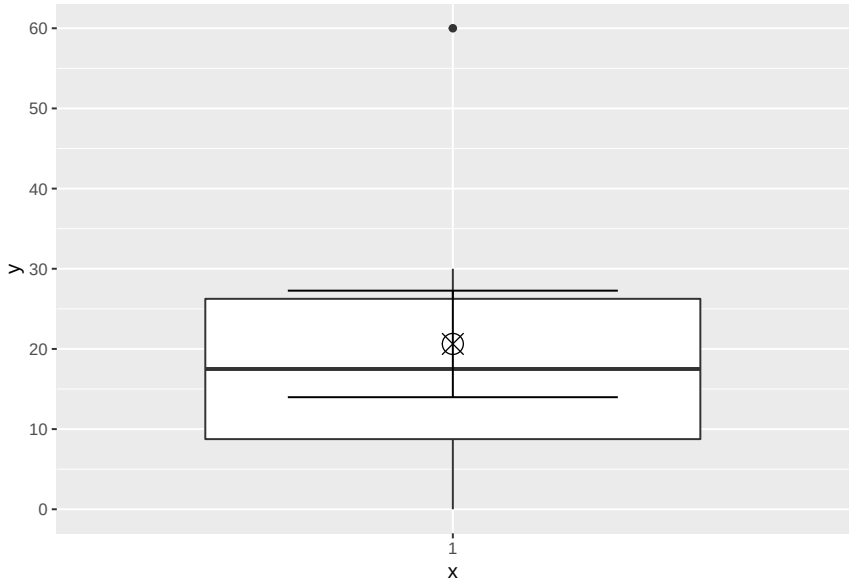


Figure 4.4: Example box plot format used throughout this chapter given the values of $(0, 5, 10, 15, 20, 25, 30, 60)$. Median 17.5. Quartiles 1, 2 and 3 $(8.75, 17.50, 26.25)$. Mean 20.63. Standard error 6.64. Error bars $(13.98, 27.27)$

4.2.5 Method Variation Parameters

This section defines the exact configuration of OSCC variations that are evaluated within this chapter. All OSCC parameters are defined and described within section 3.7 of chapter 3. Firstly, this section defines default parameters for all OSCC variations. Secondly, each individual variation to be evaluated is defined along with details of

any overridden parameter values with respect to the default parameters. Tables 4.3, 4.4 and 4.5 defines the default parameters for sampling, ring coding / correlation and ring sequence coding / correlation respectively.

Table 4.6 defines each method variation to be evaluated throughout this chapter. Variations of ring correlation approach and relative edge features are concentrated on as both parameters are core to the OSCC method. A variation exists for each combination of ring correlation method defined in section 3.6.1 / relative edge feature defined in section 3.5.1.

Point sample area	Global
Parameter	Default parameter
Ring Spacing	Uniform spacing of three pixels
Sample Point Region	Nearest-neighbor

Table 4.3: Default Sampling Parameters

Parameter	Default parameter
Binning	Exact binning using 8 bins for all relative features
RESO features	True (signed)
PA features	False
Ring Correlation Method	Edit Distance
Rotation Invariant	True

Table 4.4: Default Ring Correlation Parameters

Parameter	Default parameter
Global Search Space Constraints	Sakoe-Chiba band is used with horizontal and vertical dimensions of 0.5 with respect to the horizontal and vertical ring sequence lengths
Step Size	Step size of 1
Step Weights	All step weights set to 1

Table 4.5: Default Ring Sequence Correlation Parameters

Method ID	Overridden default parameter
ED	Equal to the defined default parameters which use the Edit Distance for ring correlation
ED-A	Same as ED with the addition of signed PA features
LCS	Ring correlation set to Longest Common Subsequence
LCS-A	Same as LCS with the addition of signed PA features
NLCS	Ring correlation set to the normalised Longest Common Subsequence as defined in section 3.6.1
NLCS-A	Same as NLCS with the addition of signed PA features
SD	Ring correlation set to Dynamic Time Warping which uses the feature distance correlation approach described in section 3.6.1. RESO features are used only.
SD-A	Same as SD with the addition of signed PA features

Table 4.6: Method Variation Definitions

4.2.6 Overall Approach

The previously outlined requirements define alphabet dataset variations (see section 4.2.1). For each dataset variation, ground truth correspondences are extracted between each original alphabet dataset character image and its corresponding dataset variation transformed image (see section 4.2.3 for the approach used and appendix B for details about the number of images and ground truth correspondences extracted for each

dataset variation).

A section exists within this chapter for each individual dataset variation. For an individual dataset variation, an individual query is undertaken between each transformed alphabet dataset character and its corresponding original alphabet dataset character. For an individual image query, a cost matrix is constructed between the edge points of the query image and the original image using an individual OSCC method to calculate the costs. As described within section 4.2.4, by making use of the cost matrix, the AP metric is obtained for each individual ground truth correspondence. Furthermore, the mAP metric is used to summarise a group of AP metrics. The high level mAP metric is used to compare method variations which are explicitly defined by tables of parameters within section 4.2.5.

4.3 Interpolation

Interpolation is ubiquitous in image transformations due to the process of transforming a discrete grid of pixels to another discrete grid of pixels. Table 4.7 shows the Mean Average Precision (mAP) ± Standard Deviation (SD) for the alphabet dataset with two variations. The None variation does not apply any interpolation to the reference dataset, therefore the reference and transformed datasets are exactly the same. The Cubic variation represents bicubic interpolation using a 4 x 4 neighborhood which is used to obtain the interpolated dataset. Furthermore, all method variations are compared against each other across interpolation variations using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test as shown by figure 4.5.

Interpolation	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
None	0.97 ±.12	0.94 ±.17	**0.99** ±.05	**0.99** ±.06	**0.99** ±.06	0.78 ±.28	0.97 ±.12	0.96 ±.16
Cubic	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
mAP ± SD	0.95 ±.14	0.91 ±.19	**0.98** ±.06	**0.98** ±.07	0.97 ±.09	0.75 ±.31	0.96 ±.13	0.93 ±.17

Table 4.7: Alphabet Dataset point correspondence interpolation variation results using mAP ± SD. The first row shows results without the use of any interpolation and the second row shows results for bicubic interpolation. The last row shows mAP ± SD across all dataset variations for each method.

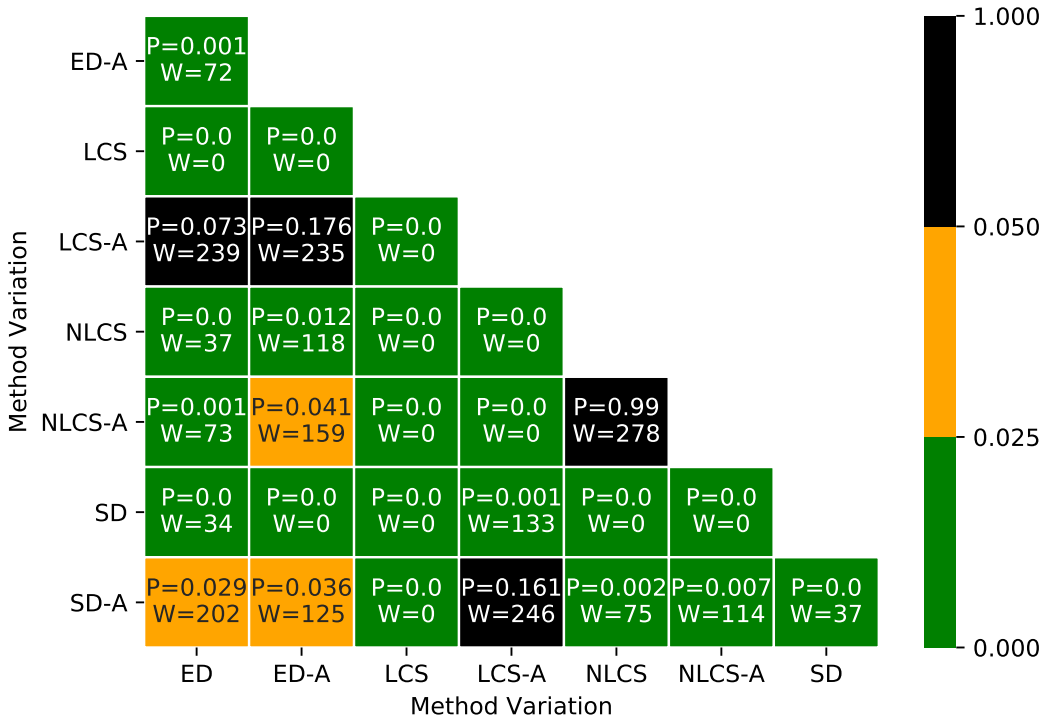


Figure 4.5: Interpolation method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Average Precision for an individual transformed alphabet character for two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix C.1 for further details.

Table 4.7 shows that the high level mAP scores for the None variation are consistently higher for all of the point descriptors compared to their corresponding Cubic variation results. This is expected as a consequence of the Cubic interpolation edge signal noise when compared to the None variation. As a result, the sampling approach used for the OSCC method in general can be affected by relative edge orientation errors. These errors can lead to bin boundary artefacts which need to be handled by the core ordinal correlation approaches. Results show that in general, methods are discriminative for both the None and Cubic interpolation as all Average Precision scores are above 0.5, which means that on average, a ground truth correspondence is at least within the top two best matching results.

The high level mAP metrics in table 4.7 show that the use of the positional angle constraint increases discriminability for the SD, LCS and ED methods. As shown by

figure 4.5, with a p-value threshold of 0.025, the median difference of AP between these methods and their positional angle constraint variations are significantly greater than zero (SD/SD-A, W=37, P=0.0), (LCS/LCS-A, W=0.0, P=0.0), (ED/ED-A, W=72, P=0.001). These results are expected as positional angles are not changed by the interpolation variations, therefore the constraint adds discriminability. However, it is interesting to observe that the median difference of AP between the ED variation and both the SD-A and LCS-A variations are not significantly greater than zero (ED/SD-A, W=202, P=0.029), (ED/LCS-A, W=239, P=0.073) with a p-value threshold of 0.025. This can be explained by the strict ED property of preferring the minimum number of differences between two rings combined with the fact that interpolation variations do not introduce shape transformations.

Along with the observation that the NLCS and NLCS-A variations achieve the highest score for both interpolation variations, the high level results show that the NLCS descriptor variations are more discriminative than the other methods with respect to interpolation - figure 4.5 shows that the median difference of AP between both NLCS and NLCS-A compared to all other method variations is significantly greater than zero (with a P value threshold of 0.05). This can be explained as the normalisation component of the method increases the cost of matching rings where the difference between ring sizes increases or decreases.

By breaking down the high level mAP summary score and inspecting its contributing components, we can see how each individual character contributes to the score as shown by figure 4.6. Additionally table 4.8 shows a detailed view of the lowest five ranked alphabet characters for each interpolation method / descriptor method combination. Outliers for each combination are shown in bold according to box plot statistics.

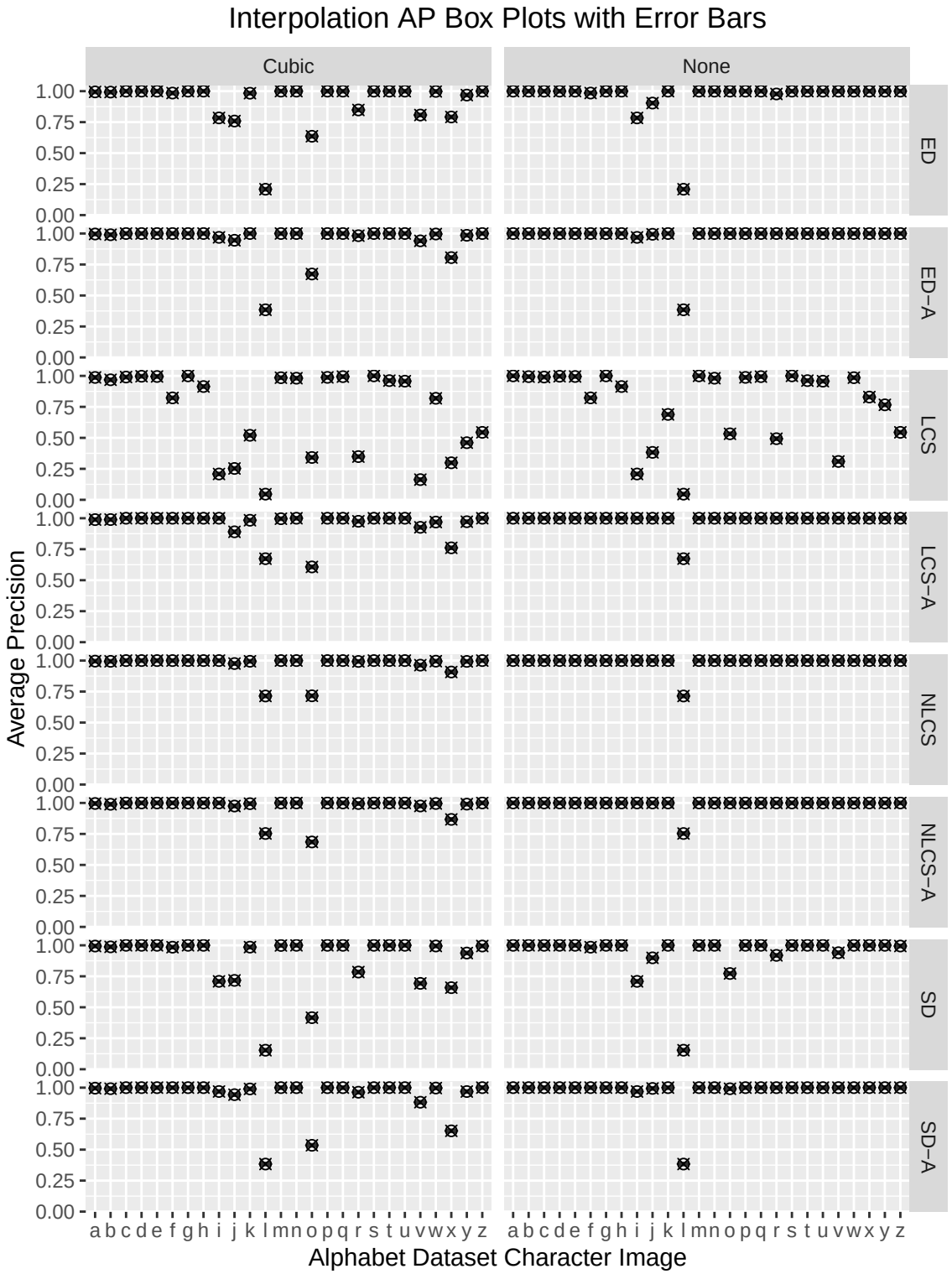


Figure 4.6: Interpolation variation Average Precision box plots with error bars for each alphabet character. An Average Precision score represents the average of precision scores obtained from each individual ground truth correspondence between a reference image and a transformed image.

Interpolation	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
None	**j, 99**	**r, 92**	**l, 75**	**l, 71**	**l, 67**	r, 49	**j, 99**	**f, 98**
	o, 99	**j, 90**				j, 38	**i, 97**	**r, 98**
	i, 97	**o, 77**				v, 31	**l, 38**	**j, 90**
	l, 38	**i, 71**				i, 21		**i, 78**
		l, 15				l, 05		**l, 21**
Cubic	j, 94	i, 71	**v, 98**	**j, 98**	**v, 93**	x, 30	**j, 95**	x, 79
	v, 88	v, 69	**j, 98**	**v, 96**	**j, 89**	j, 25	**v, 94**	i, 78
	x, 65	x, 66	**x, 87**	**x, 91**	**x, 76**	i, 21	**x, 81**	j, 76
	o, 53	**o, 42**	**l, 75**	**o, 72**	**l, 67**	v, 16	**o, 67**	**o, 64**
	l, 38	**l, 15**	**o, 69**	**l, 71**	**o, 61**	l, 05	**l, 38**	**l, 21**

Table 4.8: Lowest five ranked alphabet dataset images by Average Precision score. An individual cell shows the lowest ranked images for an interpolation / OSCC method combination. A cell contains (alphabet image, average precision score) pairs. Box plot outliers are highlighted in bold.

First of all, the None interpolation row in table 4.8 identifies how discriminative a method variation is with respect to an individual alphabet character without any interpolation applied. The NLCS-A and NLCS methods are shown to be the most discriminative with duplicate points only existing for the letter 'l'. Interestingly the standard LCS approach is shown to be the least discriminative while its positional angle constraint variation of LCS-A is shown to significantly increase discriminability. All methods are shown to be the least discriminative with respect to the character 'l', this can be explained by the fact that the image for the character 'l' is identical when rotated by 180 degrees while also having a relatively low amount of shape information. Therefore multiple points are seen to be exactly the same from a rotation invariant perspective. Characters 'i' and 'j' are the second most frequent lowest ranked characters, this can potentially be explained by the relatively low amount of shape information that they contain. Furthermore, the positional angle constraint is shown to improve discriminability for these characters.

The Cubic row in table 4.8 shows that the same Average Precision scores are obtained for the character 'l' across all methods when compared to the None interpolation variation. However, other characters are shown to be less discriminative. In addition to the most frequent characters of 'l', 'i' and 'j' being present within the lowest ranked characters for the None variation, characters 'x', 'v' and 'o' are present for the Cubic variation. Possible explanations for this observation is that in the dataset, character images of 'x' and 'o' are shown to be the same when rotated by 180 degrees while the character 'v' has a relatively low amount of shape information.

A further observation is that the positional angle constraint increases discriminability for all characters across all methods, with one exception where the NLCS method is more discriminative with respect to the letter 'o' when compared to the NLCS-A method. This can be explained by edge feature binning boundary effects.

4.4 Similarity Transforms Invariance

Similarity transforms retain shape as angles between lines and ratios remain constant. The following subsections evaluate variations of the OSCC method to satisfy requirement PT_1 (see table 4.1). Similarity transforms of translation, reflection, rotation and scale are evaluated.

4.4.1 Reflection Invariance

Both horizontal and vertical reflection transformations are evaluated in this section. Although the OSCC methods were not designed with reflection invariance in mind, it is important to understand how reflection transformations are handled as requirement PT_1 defined in table 4.1. Table 4.9 shows the high level mAP scores for each method / reflection transformation. The results show that the OSCC methods are not invariant to reflection transformations. However, if reflection is required for a higher object level recognition task, then this can be overcome by querying horizontal and vertical reflection variations of the OSCC coding to find the best match amongst the variations. This naive approach is impractical for all other transformations due to the number of possible transformation variations. It should also be noted that a shape descriptor with reflection invariance would be highly unlikely to be discriminative enough for practical applications.

Reflection	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
horizontal	0.01 ±.00	0.01 ±.01	0.01 ±.01	**0.02** ±.02	0.00 ±.00	0.00 ±.00	0.01 ±.01	0.01 ±.01
vertical	0.01 ±.00	0.01 ±.01	0.01 ±.01	**0.03** ±.03	0.00 ±.00	0.00 ±.00	0.01 ±.01	0.01 ±.01
mAP ± SD	0.01 ±.00	0.01 ±.01	0.01 ±.01	**0.03** ±.02	0.00 ±.00	0.00 ±.00	0.01 ±.01	0.01 ±.01

Table 4.9: Alphabet Dataset point correspondence reflection variation results using mAP ± SD. The last row shows mAP ± SD across all dataset variations for each method.

4.4.2 Translation Invariance

Four variations of image translations are evaluated with respect to combinations of exact and inexact discrete pixel grid mappings for shifts in both the x and y directions. Following the experimentation framework, bicubic interpolation is used with all translation transformations.

Table 4.10 shows the high level mAP score for each translation variation / OSCC method variation combination. The first row shows the baseline results for comparison. As expected, the same results as the baseline are obtained when using a translation of 1.0 in both the x and y direction as shown by the fifth row. Although the results show a slight sensitivity to translation, all results are within 0.08 with respect to the baseline. Furthermore, figure 4.7 shows the two-sided Wilcoxon Matched-Pairs Signed-Rank Test results for each pairwise method variation across all transformations.

Overall, it is shown that with the exception of (LCS-A, ED), (SD-A, ED-A) and (SD-A, LCS-A) pairs, the median difference of AP between the other pairwise method combinations are greater than zero (with a P value threshold of 0.025). Like the interpolation results, as expected, the positional angle method variations add discriminability for the translation transformations as the actual shapes do not change with the transformation.

Tx	Ty	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	0.0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
0.5	0.5	0.91 ±.18	0.85 ±.24	**0.95** ±.13	**0.95** ±.13	0.90 ±.18	0.63 ±.31	0.93 ±.17	0.88 ±.22
0.5	1.0	0.96 ±.15	0.90 ±.22	**0.98** ±.09	0.97 ±.10	0.95 ±.13	0.71 ±.28	0.96 ±.15	0.92 ±.21
1.0	0.5	0.92 ±.18	0.86 ±.22	0.95 ±.12	**0.96** ±.12	0.92 ±.16	0.66 ±.33	0.93 ±.17	0.89 ±.20
1.0	1.0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
mAP ± SD		0.93 ±.17	0.88 ±.22	**0.96** ±.10	**0.96** ±.10	0.93 ±.14	0.68 ±.32	0.94 ±.15	0.90 ±.20

Table 4.10: Alphabet Dataset point correspondence translation variation results using mAP ± SD. The last row shows mAP ± SD across all dataset variations for each method.

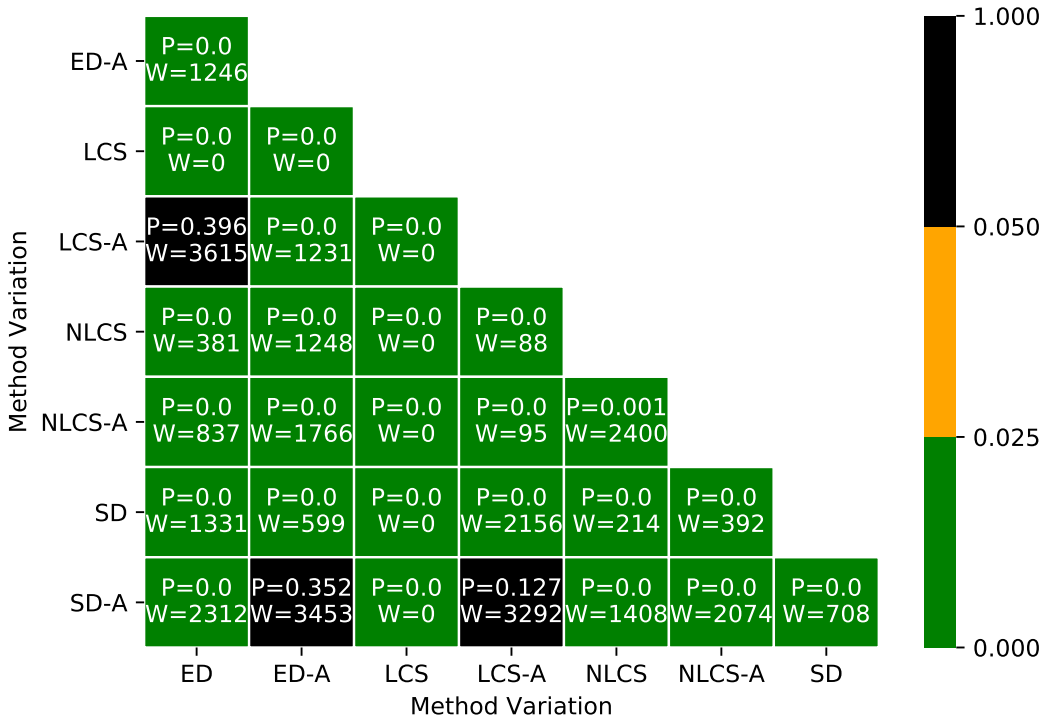


Figure 4.7: Translation method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Average Precision for an individual transformed alphabet character for two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix C.3 for further details.

The NLCS-A and NLCS methods consistently obtain the highest score across the translation variations. As shown by figure 4.7 the median difference of AP between both NLCS and NLCS-A compared to all other method variations is significantly greater than zero (with a P value threshold of 0.025). Furthermore, the median change in AP between the NLCS and NLCS-A is significantly greater than zero (W=2400, P=0.001). These results show that the positional angle constraint both increases and decreases the relative discriminability of different characters with varying translation transformations.

In addition to the previously described observation that the NLCS-A and NLCS methods consistently obtain the highest score across all translation variations, the LCS method obtains the lowest score across the variations. These observations are consistent with the observations made in relation to the interpolation results. This

is expected due to the possible edge noise introduced by mapping one discrete grid to another resulting in noise being introduced to the OSCC correlation methods. Furthermore, the high level results show that the OSCC methods are robust to translation.

Figure 4.8 shows the lower level Average Precision score distributions for individual alphabet characters. An individual box plot shows the Average Precision scores obtained from an OSCC method for an alphabet character across all translation transformations defined in table 4.10. Table 4.11 shows the last five entries of the ordered Average Precision list for each variation and method combination with the outliers highlighted as bold. The OSCC methods are clearly the least discriminative with respect to character 'l' followed by 'o', 'i', 'j', 'x' and 'v' in order. As expected, these results are consistent with the interpolation results.

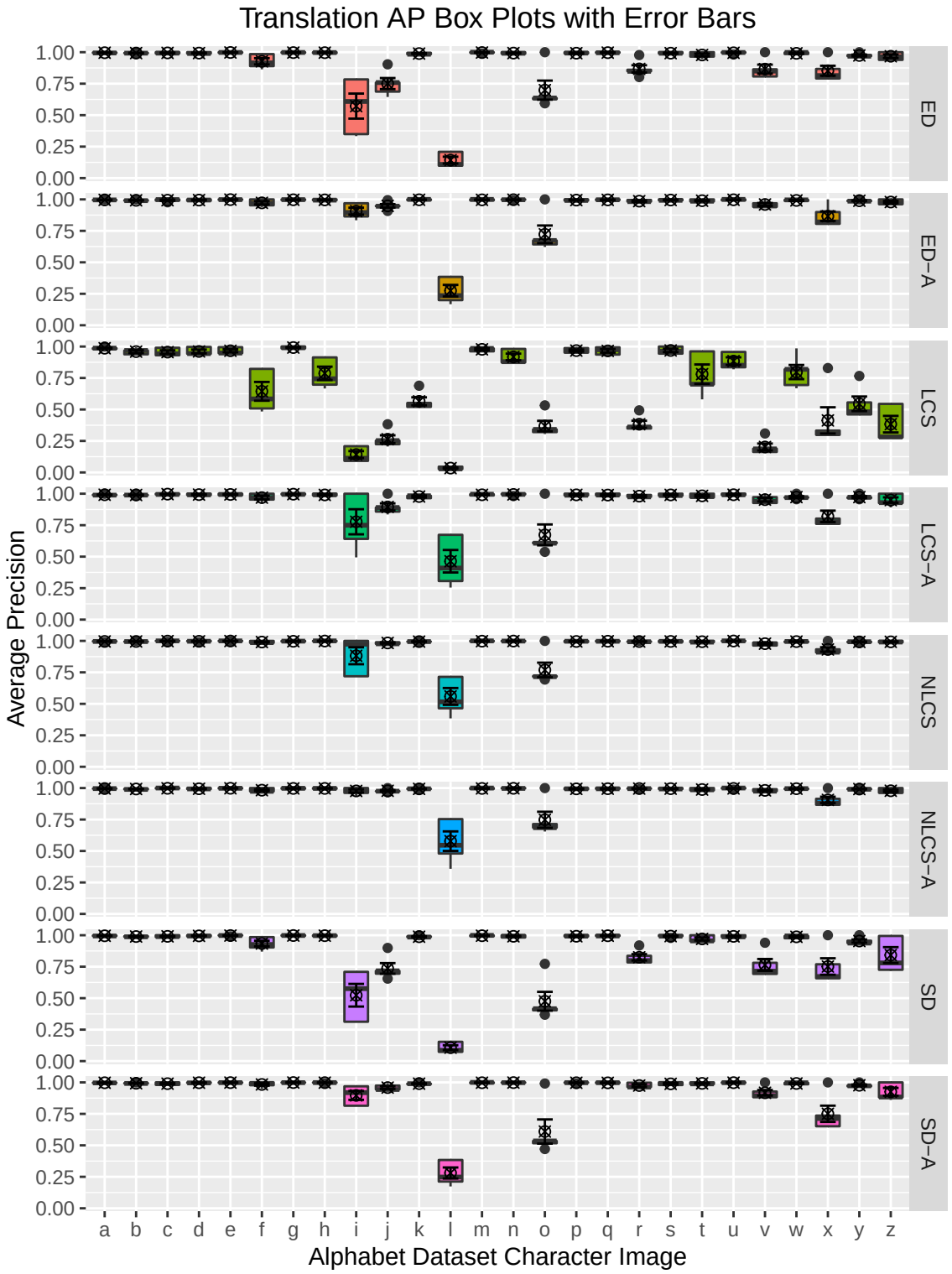


Figure 4.8: Translation variation box plots. An individual box plot shows the Average Precision scores for all translation transformation combinations applied to an individual character with respect to an OSCC method.

Tx	Ty	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	0.0	j, 94	i, 71	**v, 98**	**j, 98**	**v, 93**	x, 30	**j, 95**	x, 79
		v, 88	v, 69	**j, 98**	**v, 96**	**j, 89**	j, 25	**v, 94**	i, 78
		x, 65	x, 66	**x, 87**	**x, 91**	**x, 76**	i, 21	**x, 81**	j, 76
		o, 53	**o, 42**	**l, 75**	**o, 72**	**l, 67**	v, 16	**o, 67**	**o, 64**
		l, 38	**l, 15**	**o, 69**	**l, 71**	**o, 61**	l, 05	**l, 38**	**l, 21**
0.5	0.5	**z, 88**	v, 71	v, 97	v, 98	**j, 84**	z, 27	f, 94	r, 86
		i, 81	j, 69	**i, 94**	**x, 93**	**x, 80**	j, 21	x, 90	j, 69
		x, 74	**o, 40**	**x, 91**	**o, 72**	**o, 61**	v, 19	**i, 83**	**o, 59**
		o, 52	**i, 31**	**o, 71**	**i, 72**	**i, 49**	i, 09	**o, 64**	**i, 33**
		l, 17	**l, 07**	**l, 36**	**l, 38**	**l, 25**	l, 03	**l, 17**	**l, 09**
0.5	1.0	t, 99	j, 90	i, 99	t, 99	f, 97	j, 38	w, 98	z, 94
		w, 98	**z, 78**	t, 99	z, 99	w, 96	v, 31	**f, 97**	**f, 92**
		z, 88	**o, 77**	**f, 97**	f, 99	**z, 89**	z, 28	**z, 96**	**j, 90**
		i, 81	**i, 31**	**z, 96**	**i, 72**	**i, 64**	i, 09	**i, 86**	**i, 35**
		l, 25	**l, 07**	**l, 55**	**l, 52**	**l, 41**	l, 03	**l, 24**	**l, 10**
1.0	0.5	i, 92	x, 67	z, 96	j, 98	**j, 86**	z, 28	**j, 91**	r, 80
		z, 86	j, 66	i, 96	i, 97	**x, 78**	j, 22	**i, 90**	**j, 64**
		x, 71	i, 58	**x, 88**	**x, 91**	**i, 75**	v, 19	**x, 82**	**o, 63**
		o, 47	**o, 37**	**o, 65**	**o, 69**	**o, 54**	i, 11	**o, 62**	**i, 61**
		l, 21	**l, 09**	**l, 48**	**l, 46**	**l, 31**	l, 03	**l, 20**	**l, 11**
1.0	1.0	j, 94	i, 71	**v, 98**	**j, 98**	**v, 93**	x, 30	**j, 95**	x, 79
		v, 88	v, 69	**j, 98**	**v, 96**	**j, 89**	j, 25	**v, 94**	i, 78
		x, 65	x, 66	**x, 87**	**x, 91**	**x, 76**	i, 21	**x, 81**	j, 76
		o, 53	**o, 42**	**l, 75**	**o, 72**	**l, 67**	v, 16	**o, 67**	**o, 64**
		l, 38	**l, 15**	**o, 69**	**l, 71**	**o, 61**	l, 05	**l, 38**	**l, 21**

Table 4.11: Lowest five ranked alphabet dataset images by Average Precision score. An individual cell shows the lowest ranked images for a translation transformation / OSCC method combination. A cell contains (alphabet image, average precision score) pairs. Box plot outliers are highlighted in bold.

4.4.3 Rotation Invariance

Clockwise rotations from 45 degrees to 315 degrees with intervals of 45 degrees are evaluated in this section. Table 4.12 shows the high level mAP scores for each OSCC method / rotation transformation. In addition, figure 4.9 shows two-sided Wilcoxon Matched-Pairs Signed-Rank Test results for each pairwise method variation across all rotations. Results are further broken down by figure 4.10 which shows Average Precision score distributions, where an individual box plot shows the Average Precision scores obtained from an OSCC method for an individual character with respect to all rotation transformation variations. Furthermore, table 4.13 shows the last five entries of the Average Precision match list for each variation and method

combination with the outliers highlighted as bold.

Angle	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
45.0	0.85 ±.24	0.78 ±.28	**0.89** ±.21	**0.89** ±.22	0.84 ±.23	0.58 ±.32	0.85 ±.25	0.79 ±.27
90.0	0.92 ±.18	0.86 ±.23	**0.94** ±.14	**0.94** ±.14	0.92 ±.15	0.68 ±.32	0.91 ±.19	0.87 ±.22
135.0	0.86 ±.23	0.78 ±.27	**0.89** ±.20	**0.89** ±.22	0.84 ±.22	0.58 ±.32	0.85 ±.24	0.79 ±.27
180.0	**0.90** ±.21	0.85 ±.25	**0.90** ±.22	**0.90** ±.22	0.89 ±.21	0.65 ±.33	0.87 ±.26	0.83 ±.28
225.0	0.86 ±.23	0.78 ±.27	**0.89** ±.20	**0.89** ±.22	0.84 ±.23	0.57 ±.32	0.86 ±.24	0.79 ±.27
270.0	0.91 ±.20	0.86 ±.23	**0.93** ±.17	**0.93** ±.17	0.92 ±.18	0.67 ±.32	0.91 ±.20	0.86 ±.23
315.0	0.85 ±.23	0.78 ±.27	**0.89** ±.20	**0.89** ±.22	0.84 ±.23	0.57 ±.32	0.85 ±.24	0.79 ±.27
mAP ± SD	0.89 ±.22	0.82 ±.26	**0.91** ±.18	**0.91** ±.19	0.88 ±.20	0.63 ±.33	0.88 ±.23	0.83 ±.25

Table 4.12: Alphabet Dataset point correspondence rotation variation results using mAP ± SD. The last row shows mAP ± SD across all dataset variations for each method.

The first row table 4.13 shows the baseline results for comparison. The NLCS-A and NLCS methods are both shown to be the most discriminative methods across all transformations with the SD-A method obtaining an equal score for the rotation variation of 180 degrees. Figure 4.9 shows that that the median difference of AP between both NLCS and NLCS-A compared to all other method variations is significantly greater than zero (with a P value threshold of 0.025) while the median difference of AP between NLCS and NLCS-A is not significantly greater than zero (W=13, P=0.484). With the exception of the LCS method, all transformations are within 0.12 of the baseline. As previously observed with respect to the interpolation and translation variations, the positional angle constraint provides a more discriminative variation of the LCS method resulting in the LCS-A method.

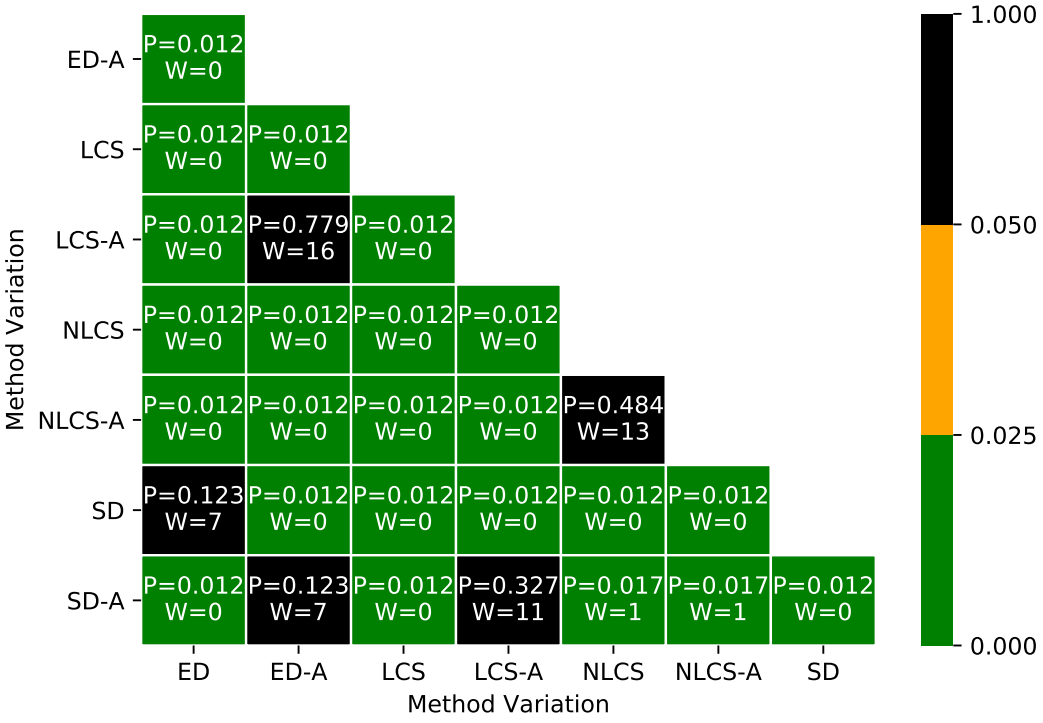


Figure 4.9: Rotation method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the mAP between two OSCC method variations for a particular rotation value as defined by 4.12. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two method variations using matched pairs from all dataset variations. See appendix C.4 for further details.

It is interesting to note that results for rotation variations of 45, 135, 225 and 315 degrees are consistently lower when compared to the results for rotation variations of 0, 90, 180 and 270 degrees. This observation can be explained by rotation transformation effects which in turn introduce noise into the OSCC sampling with respect to edge orientations. It is not thought that this observation can be explained by bin boundary effects alone, where bin boundaries are relatively located at intervals of 45 degrees starting from 22.5 degrees. It should be noted that although sensitivity to rotation transformation effects is shown, the observation that all high level transformations are within 0.12 of the baseline is not changed.

With respect to rotation transformations, table 4.13 shows that the OSCC methods are the least discriminative for character ‘l’ followed by ‘x’, ‘v’, ‘o’, ‘z’, ‘i’ and ‘j’ in order. It is interesting to note that although ‘x’ is found within the bottom five least

discriminative characters for each variation, it consistently drops ranks for each 45, 135, 225 and 315 degrees of rotation. It is thought that as 'x' is identical under a 180 degrees transformation while having a relatively low amount of shape information, this leads to less discriminability for the character.

Angle	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	j, 94	i, 71	**v, 98**	**j, 98**	**v, 93**	x, 30	**j, 95**	x, 79
	v, 88	v, 69	**j, 98**	**v, 96**	**j, 89**	j, 25	**v, 94**	i, 78
	x, 65	x, 66	**x, 87**	**x, 91**	**x, 76**	i, 21	**x, 81**	j, 76
	o, 53	**o, 42**	**l, 75**	**o, 72**	**l, 67**	v, 16	**o, 67**	**o, 64**
	l, 38	**l, 15**	**o, 69**	**l, 71**	**o, 61**	l, 05	**l, 38**	**l, 21**
45.0	**z, 71**	i, 47	**v, 87**	**v, 89**	z, 73	j, 18	v, 74	i, 53
	v, 57	v, 44	**z, 72**	**z, 71**	v, 71	i, 12	**z, 57**	z, 45
	o, 29	o, 25	**l, 40**	**l, 36**	**o, 30**	v, 10	**o, 28**	o, 26
	x, 28	x, 16	**o, 31**	**o, 33**	**l, 29**	x, 03	**x, 24**	**x, 17**
	l, 15	**l, 06**	**x, 30**	**x, 21**	**x, 13**	l, 02	**l, 14**	**l, 06**
90.0	z, 90	v, 66	j, 97	j, 97	**j, 89**	j, 25	v, 94	i, 78
	v, 90	i, 66	**z, 82**	**z, 81**	**z, 82**	i, 24	**z, 62**	**x, 53**
	x, 56	**x, 50**	**x, 66**	**x, 73**	**x, 57**	x, 21	**x, 57**	**z, 50**
	o, 44	**o, 34**	**l, 59**	**l, 57**	**l, 56**	v, 18	**o, 40**	**o, 36**
	l, 30	**l, 13**	**o, 43**	**o, 45**	**o, 44**	l, 05	**l, 36**	**l, 17**
135.0	z, 73	v, 48	v, 90	**v, 89**	z, 72	j, 18	i, 80	i, 48
	v, 64	i, 47	**z, 72**	**z, 70**	v, 70	i, 12	**z, 56**	z, 46
	o, 33	o, 26	**l, 42**	**l, 38**	**o, 37**	v, 11	**o, 34**	**o, 29**
	x, 29	x, 18	**o, 38**	**o, 35**	**l, 34**	x, 03	**x, 25**	**x, 18**
	l, 17	**l, 07**	**x, 31**	**x, 22**	**x, 13**	l, 02	**l, 16**	**l, 07**
180.0	z, 90	j, 68	j, 97	**j, 96**	**j, 88**	x, 20	i, 92	i, 59
	v, 89	i, 57	**z, 56**	**z, 57**	**x, 56**	z, 19	**x, 43**	**x, 37**
	x, 51	x, 45	**x, 53**	**x, 54**	**z, 55**	v, 17	**o, 26**	**o, 22**
	o, 37	**o, 29**	**l, 27**	**o, 26**	**o, 33**	i, 16	**z, 25**	**z, 20**
	l, 13	**l, 06**	**o, 27**	**l, 24**	**l, 24**	l, 02	**l, 11**	**l, 06**
225.0	z, 70	z, 52	**v, 87**	**v, 93**	v, 74	j, 17	v, 80	i, 53
	v, 64	v, 48	**z, 70**	**z, 71**	z, 70	v, 11	**z, 54**	z, 43
	o, 36	o, 28	**l, 41**	**l, 39**	**o, 36**	i, 11	**o, 34**	**o, 29**
	x, 29	x, 17	**o, 36**	**o, 34**	**l, 30**	x, 03	**x, 26**	**x, 18**
	l, 17	**l, 06**	**x, 33**	**x, 22**	**x, 13**	l, 02	**l, 16**	**l, 07**
270.0	j, 94	v, 67	**v, 97**	j, 97	z, 94	j, 28	v, 93	i, 68
	v, 89	x, 59	**z, 86**	**z, 85**	v, 94	x, 25	**z, 69**	x, 58
	x, 58	i, 57	**x, 69**	**x, 77**	**x, 64**	i, 19	**x, 61**	z, 57
	o, 42	**o, 32**	**o, 43**	**o, 43**	**o, 44**	v, 17	**o, 40**	**o, 34**
	l, 15	**l, 07**	**l, 32**	**l, 30**	**l, 29**	l, 03	**l, 17**	**l, 09**
315.0	**z, 67**	z, 51	**v, 89**	**v, 89**	v, 71	j, 17	v, 78	i, 52
	v, 64	v, 50	**z, 68**	**z, 68**	z, 66	i, 11	**z, 55**	z, 43
	o, 31	o, 25	**l, 40**	**l, 34**	**o, 33**	v, 09	**o, 30**	o, 27
	x, 29	x, 18	**x, 34**	**o, 32**	**l, 29**	x, 03	**x, 26**	**x, 18**
	l, 18	**l, 06**	**o, 33**	**x, 23**	**x, 13**	l, 02	**l, 14**	**l, 06**

Table 4.13: Lowest five ranked alphabet dataset images by Average Precision score. An individual cell shows the lowest ranked images for a rotation transformation / OSCC method combination. The last row shows the mAP metric ± SD across all transformations for each method. A cell contains (alphabet image, average precision score) pairs. Box plot outliers are highlighted in bold.

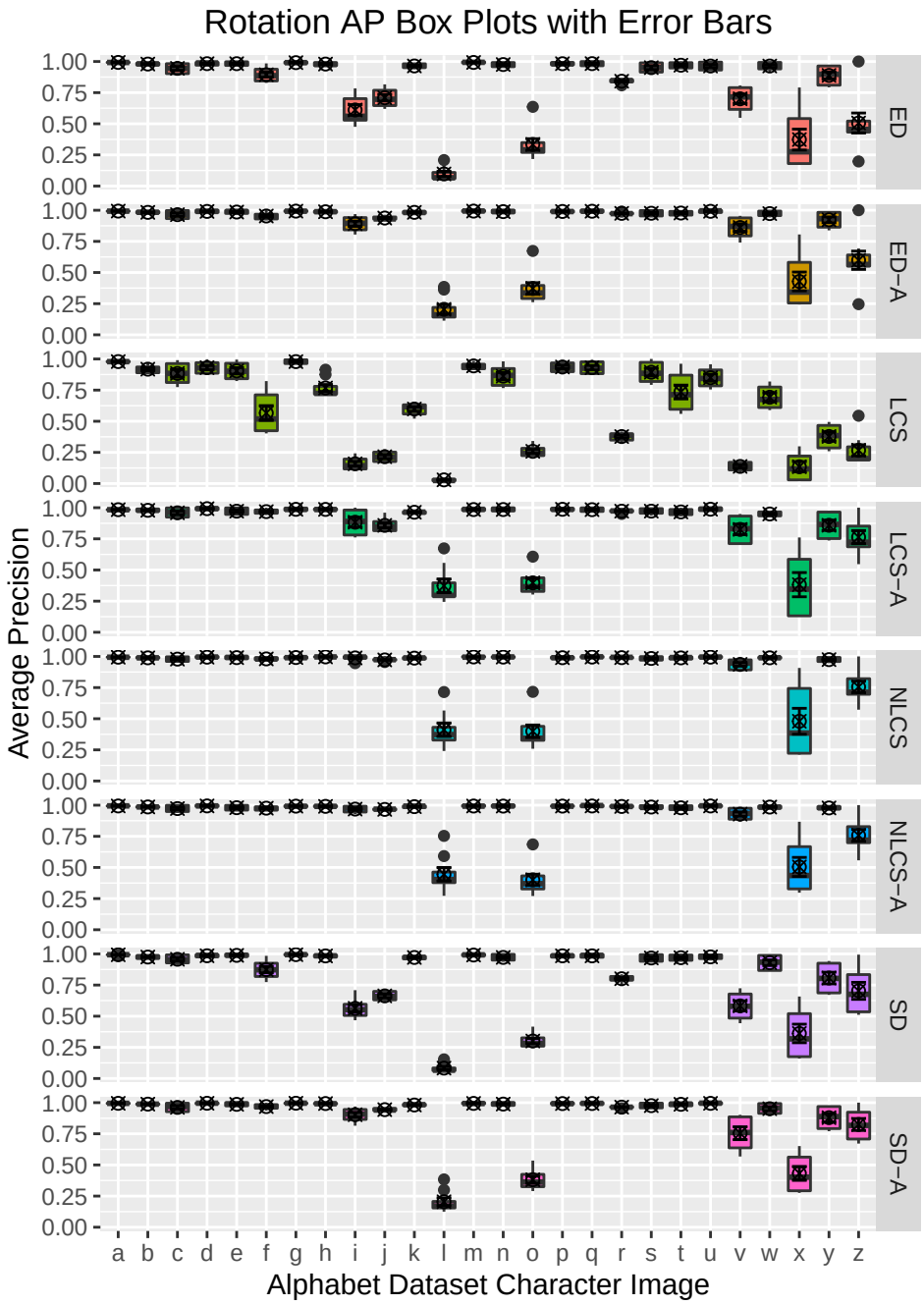


Figure 4.10: Character Rotation Transformations / OSCC method Average Precision Box Plots. An individual box plot shows the Average Precision scores for all rotation transformation combinations applied to an individual character with respect to an OSCC method.

4.4.4 Scale Invariance

Uniform scaling of 0.5, 0.75, 1.25 and 1.5 is evaluated within this section. Repeating the same approach for results inspection within the previous sections, table 4.14 shows the high level mAP scores for each OSCC method / scale transformation. Figure 4.11 shows two-sided Wilcoxon Matched-Pairs Signed-Rank Test results for each pairwise method variation across all scales. Figure 4.12 shows the Average Precision score distributions, where an individual box plot shows the Average Precision scores obtained from an OSCC method for an individual character with respect to all scale transformation variations.

Scale	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.5	**0.85** ±.20	0.77 ±.24	0.71 ±.28	0.58 ±.28	0.61 ±.30	0.48 ±.29	0.84 ±.20	0.77 ±.23
0.75	0.89 ±.19	0.81 ±.24	0.85 ±.23	0.77 ±.26	0.79 ±.26	0.57 ±.31	**0.90** ±.18	0.84 ±.22
1.0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
1.25	0.89 ±.20	0.82 ±.25	**0.93** ±.16	0.92 ±.18	0.88 ±.20	0.60 ±.32	0.91 ±.19	0.85 ±.22
1.5	0.88 ±.21	0.80 ±.26	**0.92** ±.17	0.89 ±.20	0.85 ±.22	0.59 ±.33	0.90 ±.19	0.83 ±.25
mAP ± SD	0.89 ±.19	0.82 ±.24	0.88 ±.21	0.82 ±.25	0.82 ±.26	0.59 ±.33	**0.90** ±.18	0.84 ±.22

Table 4.14: Alphabet Dataset point correspondence scale variation results using mAP ± SD. The last row shows mAP ± SD across all dataset variations for each method.

The baseline results are shown by row 3 of table 4.15 with a scale of 1. The NLCS-A method is shown to be the most discriminative for the high level results from scales of more than or equal to 1. However, the ED-A approach achieves the highest score for the scale factor of 0.75 and the SD-A approach achieves the highest score for a scale factor of 0.5. These two methods are within 0.01 of each other for scales of 0.75 and 0.5. As shown by 4.11 the median difference of AP between NLCS-A and ED-A is not significantly greater than zero (W=3850, P=0.419) with a p-value threshold of 0.025. Furthermore, the median difference of AP between both pairs of (SD-A, NLCS-A) and (SD-A, ED-A) is not significantly greater than zero as shown by (W=4132, P=0.879) and (W=3835, P=0.0421) respectively. It is also observed that all positional angle constraint variations of each method improves discriminability for

all scale factors other than the baseline.

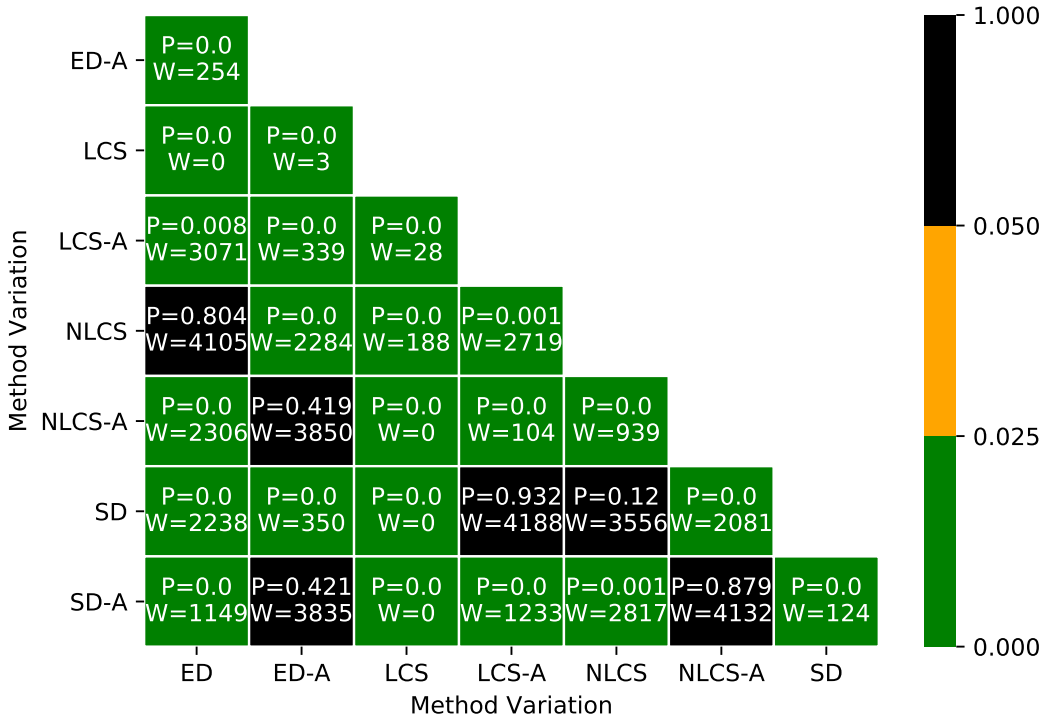


Figure 4.11: Scale method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Average Precision for an individual transformed alphabet character for two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix C.5 for further details.

A closer inspection of how each character contributes to the overall Mean Average Precision score for each method / scale variation using figure 4.12 and table 4.15 provides more insight. The Longest Common Subsequence based methods perform significantly worse for characters ‘l’ ‘i’, ‘j’ and ‘f’ with the scale factors tested of less than 1. A common feature across all of these characters is the relatively low amount of shape information that they contain. The ED / 0.5 variation has average precision scores of 0.52 and 0.54 for characters ‘i’ and ‘j’ respectively, as opposed to the LCS / 0.5 variation that has scores of 0.08 and 0.1 for characters ‘i’ and ‘j’ respectively. The only difference between the two methods that obtain significantly different results is the use of the Edit Distance or Longest Common Subsequence for ring matching. As scaling down loses information, the Longest Common Subsequence alone is not

appropriate for shapes that have a low amount of shape information as matching similar features without directly penalising for insertions, substitutions and deletions is not robust.

Scale	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.5	v, 72	i, 52	o, 39	t, 24	o, 26	x, 17	i, 77	j, 54
	z, 68	z, 51	f, 33	f, 15	f, 19	f, 15	z, 75	i, 52
	x, 51	x, 44	j, 26	j, 13	j, 18	j, 10	**x, 49**	x, 44
	o, 34	o, 25	i, 13	i, 12	i, 09	i, 08	**o, 36**	o, 30
	l, 22	**l, 07**	**l, 02**	l, 02	l, 02	l, 02	**l, 15**	**l, 07**
0.75	v, 85	z, 55	**x, 64**	f, 51	x, 48	z, 21	f, 87	j, 65
	z, 70	x, 51	**j, 59**	o, 50	o, 45	v, 17	i, 87	x, 58
	x, 55	i, 45	**o, 56**	j, 36	**j, 37**	j, 16	**x, 63**	i, 49
	o, 43	o, 31	**i, 36**	i, 23	**i, 21**	i, 08	**o, 51**	**o, 44**
	l, 21	**l, 06**	**l, 06**	**l, 05**	**l, 02**	l, 02	**l, 18**	**l, 08**
1.0	j, 94	i, 71	**v, 98**	**j, 98**	**v, 93**	x, 30	**j, 95**	x, 79
	v, 88	v, 69	**j, 98**	**v, 96**	**j, 89**	j, 25	**v, 94**	i, 78
	x, 65	x, 66	**x, 87**	**x, 91**	**x, 76**	i, 21	**x, 81**	j, 76
	o, 53	**o, 42**	**l, 75**	**o, 72**	**l, 67**	v, 16	**o, 67**	**o, 64**
	l, 38	**l, 15**	**o, 69**	**l, 71**	**o, 61**	l, 05	**l, 38**	**l, 21**
1.25	v, 83	z, 58	i, 95	i, 92	i, 82	j, 18	j, 92	j, 57
	z, 78	x, 49	z, 95	**z, 89**	**j, 74**	x, 17	i, 83	x, 56
	x, 54	i, 47	**x, 64**	**x, 63**	**x, 49**	v, 16	**x, 61**	**i, 52**
	o, 44	o, 31	**o, 54**	**o, 55**	**o, 44**	i, 10	**o, 52**	**o, 47**
	l, 17	**l, 05**	**l, 31**	**l, 20**	**l, 18**	l, 02	**l, 16**	**l, 07**
1.5	v, 79	z, 54	z, 93	j, 83	j, 75	x, 16	v, 85	j, 59
	z, 74	x, 47	**v, 90**	**x, 58**	**i, 47**	j, 16	i, 81	x, 51
	x, 53	o, 31	**x, 62**	**i, 53**	**x, 45**	v, 13	**x, 58**	**o, 42**
	o, 41	i, 30	**o, 52**	**o, 48**	**o, 42**	i, 07	**o, 49**	**i, 28**
	l, 15	**l, 05**	**l, 29**	**l, 18**	**l, 14**	l, 02	**l, 15**	**l, 07**

Table 4.15: Lowest five ranked alphabet dataset images by Average Precision score. An individual cell shows the lowest ranked images for a scale transformation / OSCC method combination. The last row shows the mAP metric ± SD across all transformations for each method. A cell contains (alphabet image, average precision score) pairs. Box plot outliers are highlighted in bold.

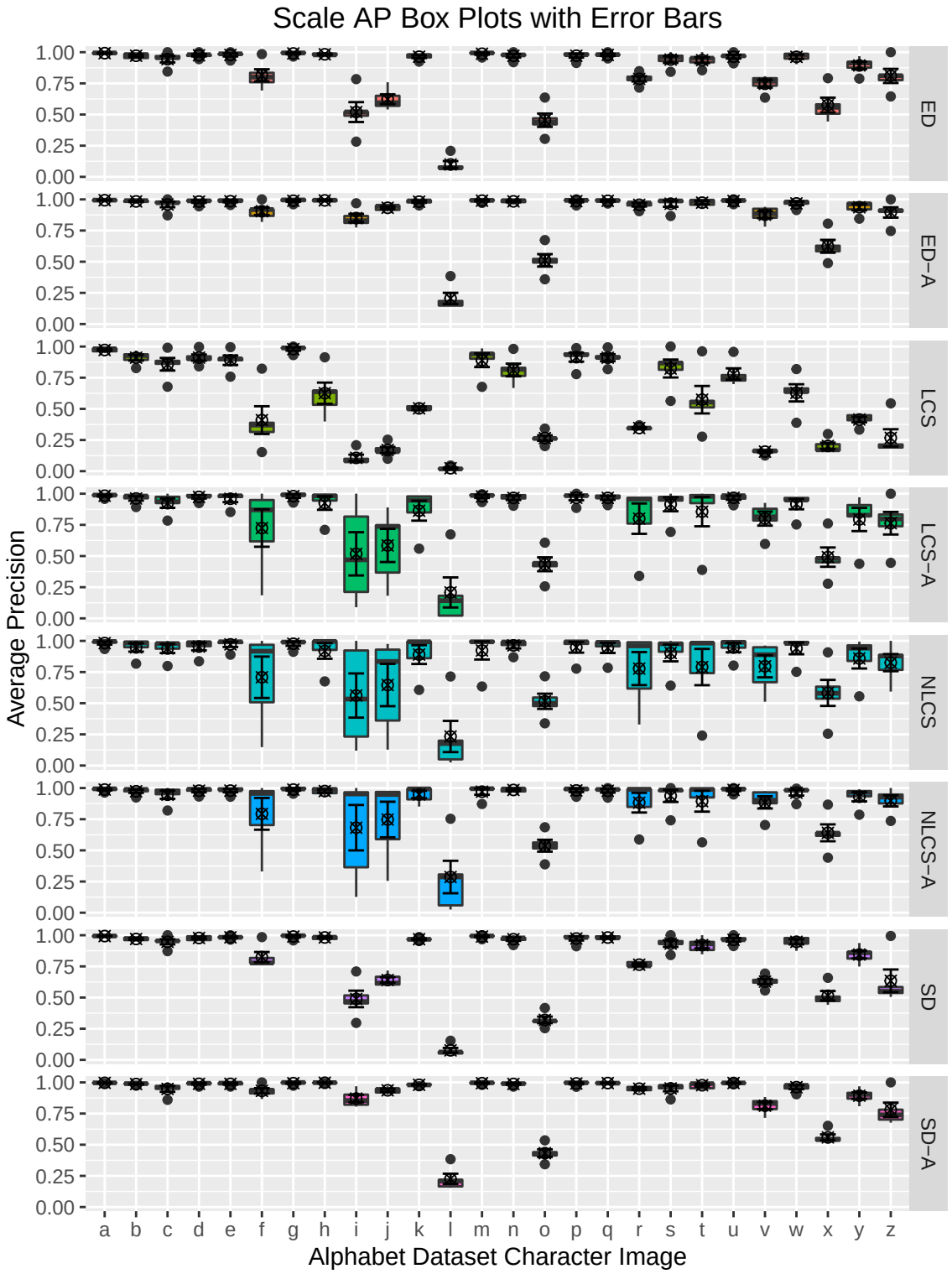


Figure 4.12: Scale variation box plots. An individual box plot shows the Average Precision scores for all scale transformation combinations applied to an individual character with respect to an OSCC method.

4.5 Affine Transforms Invariance

The additional degrees of freedom provided by affine transformations compared to similarity transformations removes the preservation of angles. Consequently, shapes can be transformed into other shapes, for example, a circle can be transformed into an ellipse. Following the requirements defined by table 4.1, this section evaluates the robustness of the OSCC method against shear transformations.

4.5.1 Shear Invariance

Shear transformations of 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6 are evaluated for both image x and y directions. Figure 4.13 illustrates all evaluated shear variations applied to the character 'o'. Tables 4.16 and 4.17 show the high level mAP scores for each OSCC method / shear transformation for image x and y directions respectively. Figure 4.15 shows the Average Precision score distributions, where an individual box plot shows the Average Precision scores obtained from an OSCC method for an individual character, with respect to all shear transformation variations in either the x or y image direction.

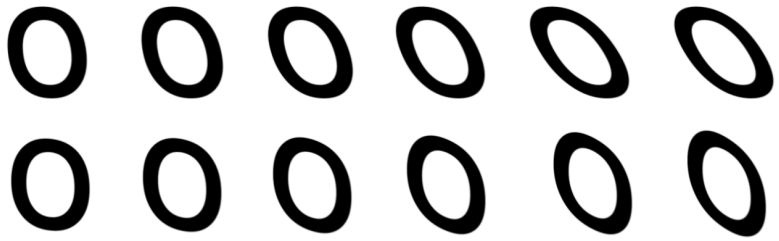


Figure 4.13: Shear x and y evaluation variations applied to character 'o'. The top row shows shear for the x direction of 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6 applied to the character 'o' from left to right. The bottom row shows shear in the y of 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6 applied to the character 'o' from left to right.

With the exception of the baseline and the shear x 0.1 variations, the SD-A method is shown to have the highest mAP score across the shear x and y variations (figure 4.14 shows that the median difference of AP between SD-A and all other OSCC method variations is significantly greater than zero). This can be explained due to its use of the relative edge feature distance instead of binary matches of relative edge feature bins. It is interesting to note that the NLCS-A method is the best for the shear x 0.1 variations while always being within 0.05 and 0.07 of the SD-A method for the shear

x and y variations respectively.

Shear x	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
0.1	0.73 ±.21	0.63 ±.22	**0.76** ±.19	**0.76** ±.19	0.61 ±.20	0.33 ±.18	0.73 ±.20	0.66 ±.21
0.2	**0.46** ±.17	0.35 ±.15	0.45 ±.21	0.44 ±.17	0.29 ±.15	0.14 ±.07	0.43 ±.18	0.39 ±.16
0.3	**0.32** ±.14	0.26 ±.11	0.29 ±.17	0.31 ±.15	0.16 ±.09	0.08 ±.04	0.27 ±.13	0.28 ±.13
0.4	**0.25** ±.12	0.20 ±.09	0.21 ±.14	0.22 ±.14	0.11 ±.06	0.06 ±.04	0.20 ±.11	0.20 ±.10
0.5	**0.17** ±.09	0.13 ±.06	0.14 ±.09	0.14 ±.08	0.06 ±.03	0.03 ±.02	0.13 ±.07	0.13 ±.07
0.6	**0.15** ±.09	0.11 ±.05	0.10 ±.07	0.11 ±.07	0.05 ±.03	0.02 ±.01	0.11 ±.07	0.11 ±.06
mAP ± SD	**0.43** ±.31	0.37 ±.30	0.42 ±.34	0.42 ±.33	0.32 ±.33	0.20 ±.27	0.40 ±.33	0.38 ±.31

Table 4.16: Alphabet Dataset point correspondence shear x variation results using mAP ± SD. The last row shows mAP ± SD across all dataset variations for each method.

Shear y	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
0.1	**0.74** ±.23	0.64 ±.23	0.73 ±.22	0.72 ±.21	0.62 ±.21	0.35 ±.18	0.70 ±.23	0.63 ±.22
0.2	**0.53** ±.23	0.44 ±.19	0.46 ±.24	0.48 ±.25	0.33 ±.21	0.16 ±.08	0.45 ±.23	0.41 ±.20
0.3	**0.40** ±.23	0.34 ±.18	0.32 ±.23	0.36 ±.25	0.22 ±.17	0.10 ±.06	0.31 ±.21	0.30 ±.19
0.4	**0.31** ±.20	0.26 ±.16	0.22 ±.18	0.26 ±.21	0.14 ±.12	0.07 ±.05	0.23 ±.18	0.24 ±.16
0.5	**0.23** ±.16	0.19 ±.12	0.16 ±.13	0.17 ±.14	0.09 ±.08	0.04 ±.02	0.18 ±.14	0.17 ±.12
0.6	**0.18** ±.13	0.15 ±.10	0.12 ±.09	0.13 ±.10	0.07 ±.07	0.03 ±.01	0.14 ±.11	0.14 ±.10
mAP ± SD	**0.47** ±.32	0.41 ±.30	0.43 ±.34	0.44 ±.34	0.35 ±.34	0.21 ±.27	0.42 ±.33	0.40 ±.31

Table 4.17: Alphabet Dataset point correspondence shear y variation results using mAP ± SD. The last row shows mAP ± SD across all dataset variations for each method.

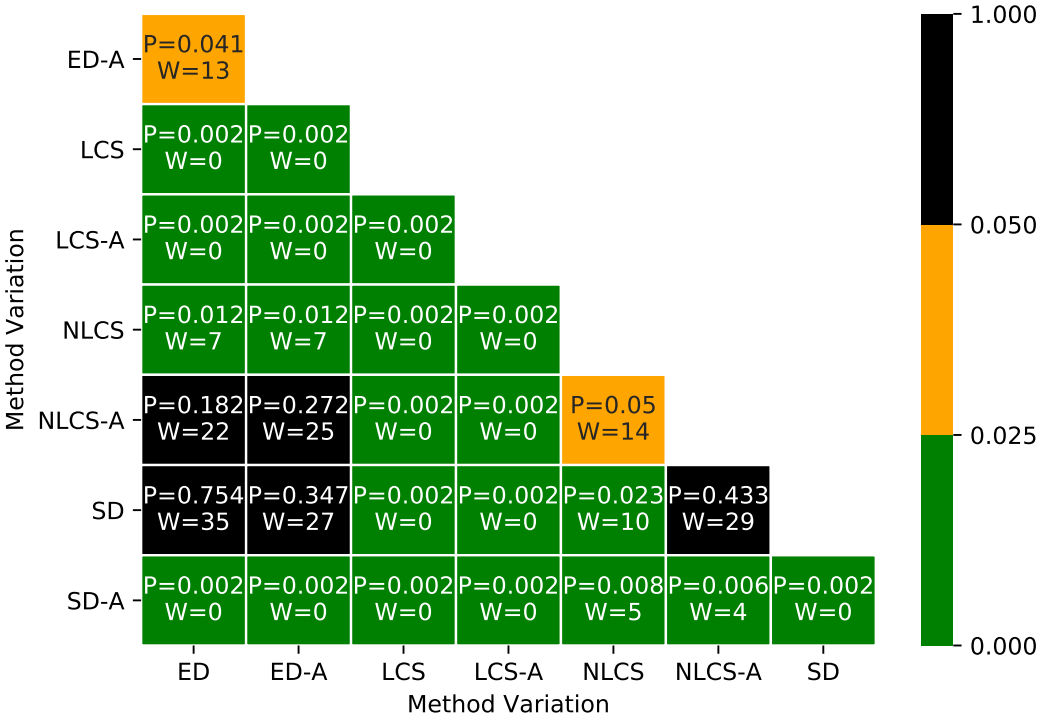


Figure 4.14: Shear method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the mAP between two method variations for a particular shear value. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two method variations using matched pairs from all dataset variations (see appendix C.6)

Figure 4.15 and tables 4.18 and 4.19 allow a further inspection into the lower level individual character Average Precision scores. As expected, character ‘o’ is shown to be the least discriminative for shear transformations as shearing changes its shape into an ellipse. Characters ‘o’, ‘l’, ‘z’, ‘c’, ‘i’ and ‘x’ are the most frequent lowest ranked for shear x across the variations and ‘o’, ‘c’, ‘z’ ‘l’, ‘g’ and ‘x’ are the most frequently lowest ranked for shear y. The character image for ‘c’ is shown to have the same structure of the character ‘o’, therefore it is not surprising for ‘c’ to be within the bottom ranks. Characters ‘z’ and ‘x’ are both defined by straight lines without any curves which is a possible reason why they are both within the lowest ranks. Overall, the results show that the SD-A method achieves an Average Precision score of at least 0.22 for all characters with respect to both shear x and y 0.3 transformation, with the exception of the following characters: ‘o’, ‘c’, ‘l’ and ‘z’. It should also be highlighted that both ‘l’ and ‘z’ are identical under a rotation of 180 degrees.

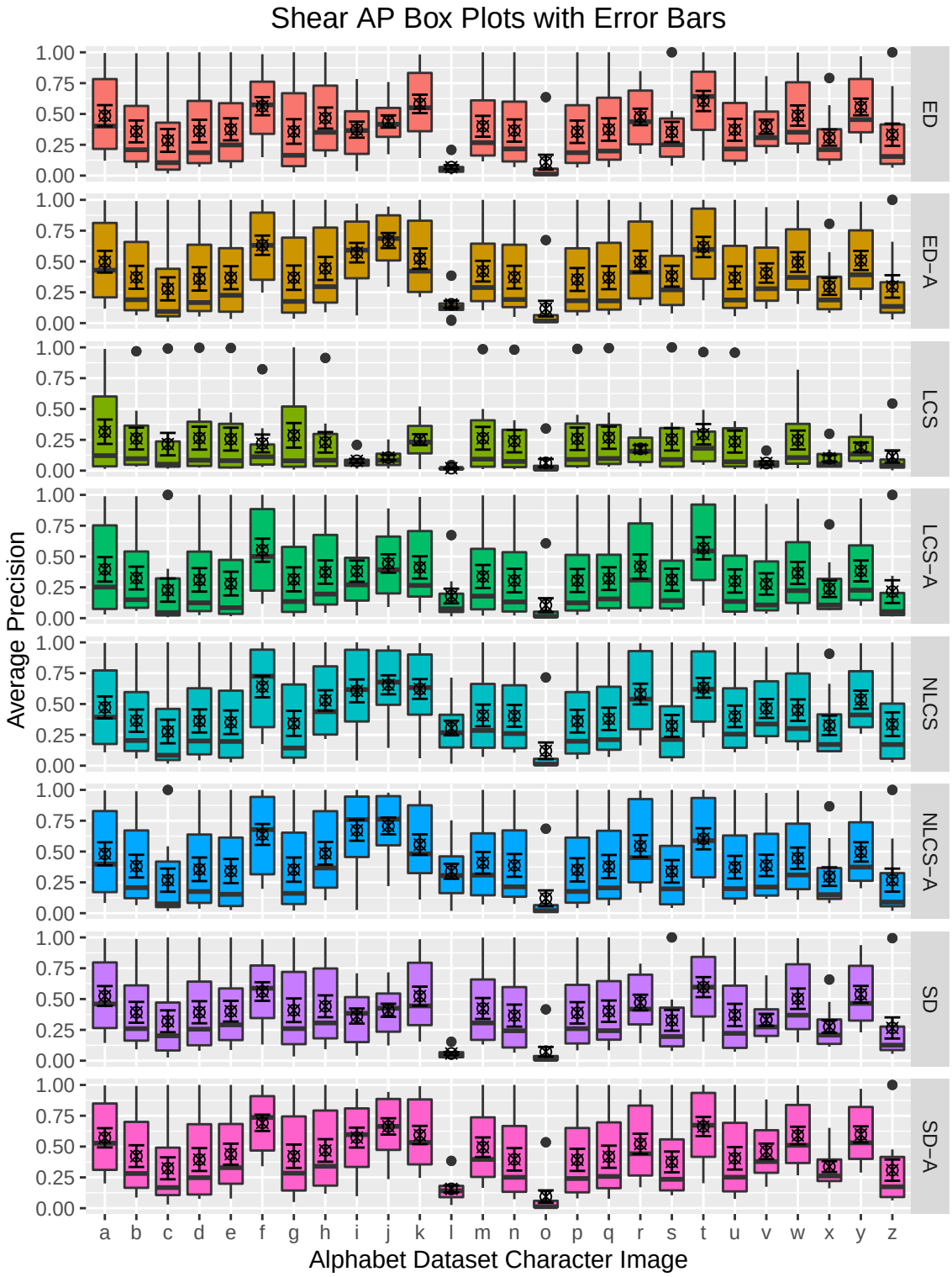


Figure 4.15: Shear x and y direction variation character box plots. An individual box plot shows the Average Precision scores for each ground truth correspondence between a reference image and all shear variation transformed images of 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6.

Shear x	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	j, 94	i, 71	**v, 98**	**j, 98**	**v, 93**	x, 30	**j, 95**	x, 79
	v, 88	v, 69	**j, 98**	**v, 96**	**j, 89**	j, 25	**v, 94**	i, 78
	x, 65	x, 66	**x, 87**	**x, 91**	**x, 76**	i, 21	**x, 81**	j, 76
	o, 53	**o, 42**	**l, 75**	**o, 72**	**l, 67**	v, 16	**o, 67**	**o, 64**
	l, 38	**l, 15**	**o, 69**	**l, 71**	**o, 61**	l, 05	**l, 38**	**l, 21**
0.1	c, 55	v, 46	x, 61	x, 66	x, 45	z, 12	s, 65	j, 54
	x, 55	i, 39	z, 61	c, 59	c, 40	v, 09	x, 58	s, 53
	z, 48	z, 31	c, 54	s, 54	z, 34	i, 07	c, 54	i, 37
	l, 19	o, 06	l, 50	**l, 40**	**l, 21**	o, 05	**l, 17**	**o, 08**
	o, 08	l, 05	**o, 09**	**o, 09**	**o, 07**	l, 02	**o, 08**	**l, 07**
0.2	s, 34	j, 28	p, 35	g, 32	x, 17	z, 05	x, 30	s, 32
	c, 30	z, 17	x, 24	s, 32	l, 11	v, 05	z, 28	c, 26
	z, 25	i, 15	z, 22	l, 21	c, 10	i, 04	c, 24	i, 17
	l, 11	**l, 02**	c, 17	c, 21	z, 10	o, 02	l, 13	**l, 03**
	o, 02	**o, 01**	**o, 02**	**o, 02**	o, 02	l, 01	o, 02	**o, 02**
0.3	d, 22	u, 20	d, 18	l, 19	e, 09	v, 04	d, 17	p, 19
	c, 20	i, 16	x, 14	g, 17	z, 05	u, 04	z, 14	d, 17
	z, 18	z, 13	z, 10	x, 16	c, 05	i, 04	c, 11	c, 13
	l, 08	l, 03	c, 08	c, 10	l, 05	o, 01	l, 11	l, 03
	o, 01	**o, 01**	o, 02	o, 01	o, 02	l, 01	o, 02	o, 01
0.4	n, 14	c, 13	s, 10	g, 10	l, 05	v, 03	z, 12	i, 12
	c, 14	i, 13	e, 10	s, 09	e, 05	n, 03	d, 12	g, 12
	z, 11	z, 10	c, 06	c, 07	z, 04	z, 02	l, 10	c, 08
	l, 08	l, 03	z, 05	z, 06	c, 04	l, 01	c, 08	l, 03
	o, 01	o, 01	o, 02	o, 01	o, 02	o, 01	o, 02	o, 01
0.5	u, 08	u, 08	n, 08	c, 06	u, 03	i, 02	c, 07	n, 07
	z, 08	n, 07	c, 07	e, 05	l, 02	u, 01	u, 06	c, 06
	n, 08	i, 04	s, 06	l, 03	n, 02	o, 01	n, 06	i, 05
	l, 03	o, 01	e, 06	z, 03	z, 02	l, 01	l, 03	o, 01
	o, 01	l, 01	o, 02	o, 01	o, 01	z, 00	o, 02	l, 01
0.6	u, 08	n, 08	s, 04	e, 04	m, 02	u, 01	u, 06	e, 07
	n, 07	z, 07	e, 03	s, 03	c, 02	n, 01	c, 05	c, 04
	z, 06	i, 05	i, 03	z, 03	z, 02	l, 01	n, 05	i, 04
	l, 03	o, 01	o, 02	l, 02	l, 01	o, 01	l, 02	o, 01
	o, 01	l, 01	l, 02	o, 01	o, 01	z, 00	o, 02	l, 01

Table 4.18: Lowest five ranked alphabet dataset images by Average Precision score. An individual cell shows the lowest ranked images for a shear transformation / OSCC method combination. The last row shows the mAP metric ± SD across all transformations for each method. A cell contains (alphabet image, average precision score) pairs. Box plot outliers are highlighted in bold.

Shear y	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.0	j, 94	i, 71	**v, 98**	**j, 98**	**v, 93**	x, 30	**j, 95**	x, 79
	v, 88	v, 69	**j, 98**	**v, 96**	**j, 89**	j, 25	**v, 94**	i, 78
	x, 65	x, 66	**x, 87**	**x, 91**	**x, 76**	i, 21	**x, 81**	j, 76
	o, 53	**o, 42**	**l, 75**	**o, 72**	**l, 67**	v, 16	**o, 67**	**o, 64**
	l, 38	**l, 15**	**o, 69**	**l, 71**	**o, 61**	l, 05	**l, 38**	**l, 21**
0.1	c, 58	v, 44	c, 50	s, 54	c, 40	z, 10	c, 51	c, 49
	z, 47	x, 35	l, 47	x, 44	x, 37	i, 10	**x, 40**	x, 38
	x, 41	z, 34	x, 41	l, 44	l, 30	v, 08	**z, 35**	z, 30
	l, 18	l, 07	**z, 36**	z, 43	**z, 24**	o, 05	**l, 16**	**l, 07**
	o, 07	**o, 05**	**o, 08**	**o, 07**	**o, 06**	l, 02	**o, 08**	**o, 07**
0.2	x, 31	c, 28	p, 31	x, 29	v, 18	c, 07	x, 25	x, 26
	z, 26	x, 23	x, 23	g, 28	l, 15	v, 05	c, 18	c, 22
	c, 26	z, 17	c, 14	z, 22	c, 08	z, 05	z, 17	z, 17
	l, 18	l, 07	z, 14	c, 15	z, 08	l, 02	l, 16	l, 07
	o, 01	**o, 01**	o, 01	o, 01	o, 01	o, 01	o, 01	o, 01
0.3	s, 24	c, 18	g, 13	b, 18	e, 08	c, 04	g, 14	g, 13
	z, 17	s, 18	e, 13	z, 12	l, 08	v, 04	l, 13	z, 13
	l, 15	z, 12	z, 06	g, 12	z, 05	z, 04	z, 08	c, 08
	c, 13	l, 06	c, 05	c, 06	c, 03	l, 02	c, 06	l, 05
	o, 01	o, 01	o, 01	o, 01	o, 01	o, 01	o, 01	o, 01
0.4	l, 15	s, 12	e, 07	s, 09	g, 05	z, 03	e, 10	z, 12
	d, 14	z, 10	g, 06	z, 09	e, 04	e, 03	g, 08	g, 07
	z, 12	c, 06	z, 03	g, 05	z, 04	c, 03	z, 06	l, 06
	c, 08	l, 06	c, 03	c, 03	c, 02	l, 02	c, 03	c, 04
	o, 00	o, 00	o, 01	o, 00	o, 01	o, 01	o, 01	o, 00
0.5	d, 10	z, 07	g, 04	z, 05	g, 03	c, 02	g, 06	b, 08
	g, 09	g, 07	e, 04	e, 05	e, 02	e, 02	e, 05	l, 05
	z, 08	l, 05	c, 02	c, 02	z, 02	l, 02	z, 04	g, 04
	c, 04	c, 04	z, 02	g, 02	c, 01	g, 02	c, 02	c, 03
	o, 00	o, 00	o, 01	o, 00	o, 00	o, 00	o, 00	o, 00
0.6	d, 08	z, 06	e, 03	z, 04	e, 01	e, 02	g, 04	e, 06
	z, 06	l, 05	g, 02	e, 03	z, 01	c, 02	e, 03	l, 05
	g, 05	g, 03	z, 02	c, 02	g, 01	a, 01	z, 03	g, 02
	c, 03	c, 03	c, 02	g, 01	c, 01	g, 01	c, 01	c, 02
	o, 00	o, 00	o, 00	o, 00	o, 00	o, 00	o, 00	o, 00

Table 4.19: Lowest five ranked alphabet dataset images by Average Precision score. An individual cell shows the lowest ranked images for a shear transformation / OSCC method combination. A cell contains (alphabet image, average precision score) pairs. Box plot outliers are highlighted in bold.

4.6 Perspective Transforms Robustness

Synthetic data is generated to inspect the robustness of OSCC methods with respect to the perspective transformation. The Unity game engine is used to generate out-of-plane vertical axis rotation ground truth correspondences with respect to the 3D model of a teddy bear. A fixed camera and an orbiting camera are used to capture approximate ground truth correspondence. More specifically, the fixed camera is used as a reference view of the model while the orbiting camera orbits the model around the vertical axis at angular intervals providing transformed views. For each angular interval of the orbiting camera, each screen point from the fixed camera is turned into a ray, if the ray hits the object then the point on the object is projected to the orbiting camera, if the hit point is not occluded by other parts of the object then a correspondence is recorded along with the reference and transformed images from the camera perspectives respectively.

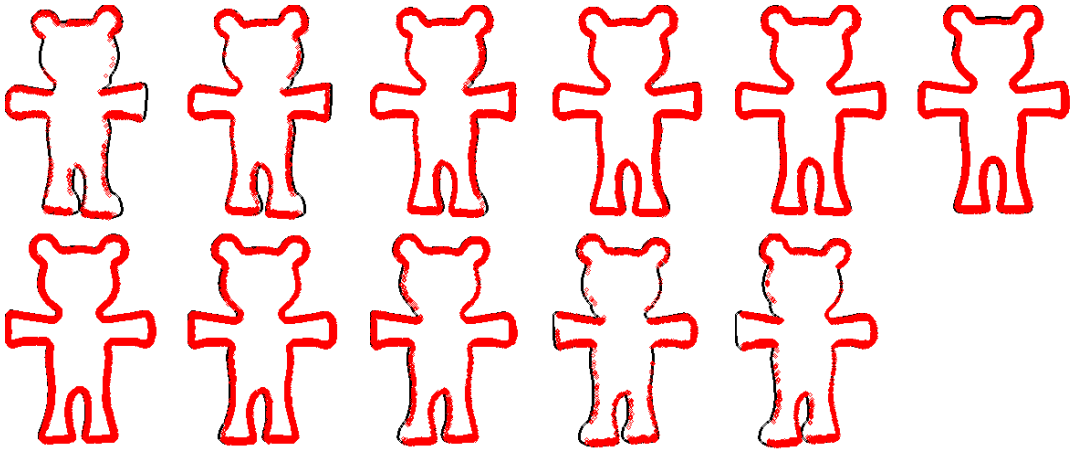


Figure 4.16: Extracted ground truth edge point correspondences for the 3D teddy bear model. Transformed image edge maps are ordered from -25 degrees to 25 degrees in 5 degree intervals with the reference image of 0 degrees being the top right image. Ground truth correspondence points between the reference and transformed image are marked by red circles.

Ground truth correspondences are filtered for a reference image and transformed image pair, such that the correspondences only exist between extracted edges. This step is required as the OSCC methods use image edge features. Figure 4.16 shows the extracted edge point correspondences for each transformed image marked by red circles overlaid on the extracted edges. The images within the figure are ordered from -25 degrees to 25 degrees with the reference image of 0 degrees being the top

right image. See appendix C.7 for information about the number of correspondences extracted for each image.

Table 4.20 shows the mAP score for each out-of-plane rotation / OSCC method while figure 4.17 shows two-sided Wilcoxon Matched-Pairs Signed-Rank Test results for each pairwise method variation across all rotations. Additionally, figure 4.18 visualises the average precision profiles for each out-of-plane rotation / OSCC method. Results for the angle of 0.0 show the discriminability of the methods for the overall shape, this is due to the reference and transformed images being exactly the same. For the angle of 0.0, the SD, NLCS-A and ED methods achieve a score of 1 with a standard deviation of 0 when rounded to two decimal places. The other methods achieve a score of 1 with a maximum standard deviation of 0.04. This shows that all methods are highly discriminative for the exact teddy bear shape with respect to ground truth correspondences.

Angle	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
-25.0	**0.82** ±.31	0.80 ±.33	**0.82** ±.33	0.79 ±.35	0.81 ±.33	0.71 ±.37	0.81 ±.32	0.79 ±.34
-20.0	**0.93** ±.21	0.92 ±.22	**0.93** ±.21	0.92 ±.22	**0.93** ±.22	0.84 ±.30	**0.93** ±.21	0.92 ±.22
-15.0	**0.97** ±.14	**0.97** ±.14	0.96 ±.16	**0.97** ±.14	0.96 ±.17	0.91 ±.23	**0.97** ±.15	**0.97** ±.14
-10.0	0.99 ±.07	0.99 ±.06	0.99 ±.07	**1.00** ±.05	0.99 ±.10	0.95 ±.18	0.99 ±.06	0.99 ±.06
-5.0	0.99 ±.06	**1.00** ±.05	**1.00** ±.03	**1.00** ±.03	0.99 ±.06	0.98 ±.11	**1.00** ±.03	**1.00** ±.04
0.0	**1.00** ±.02	**1.00** ±.00	**1.00** ±.00	**1.00** ±.02	**1.00** ±.04	**1.00** ±.03	**1.00** ±.02	**1.00** ±.00
5.0	**1.00** ±.02	**1.00** ±.04	**1.00** ±.02	**1.00** ±.00	**1.00** ±.05	0.98 ±.11	**1.00** ±.02	**1.00** ±.02
10.0	**0.99** ±.08	**0.99** ±.08	**0.99** ±.09	**0.99** ±.08	0.98 ±.12	0.96 ±.17	**0.99** ±.08	**0.99** ±.08
15.0	**0.97** ±.14	**0.97** ±.14	0.96 ±.17	**0.97** ±.15	0.95 ±.19	0.91 ±.23	0.96 ±.16	**0.97** ±.14
20.0	**0.91** ±.23	0.89 ±.25	**0.91** ±.24	0.90 ±.26	0.89 ±.25	0.82 ±.31	0.90 ±.24	0.89 ±.25
25.0	**0.85** ±.30	0.83 ±.33	0.84 ±.31	0.82 ±.33	0.84 ±.31	0.73 ±.37	0.84 ±.30	0.83 ±.32
mAP ± SD	**0.96** ±.17	0.95 ±.18	**0.96** ±.17	0.95 ±.18	0.95 ±.19	0.91 ±.24	**0.96** ±.17	0.95 ±.18

Table 4.20: Out-of-plane teddy rotation / OSCC method variation results using the mAP metric.

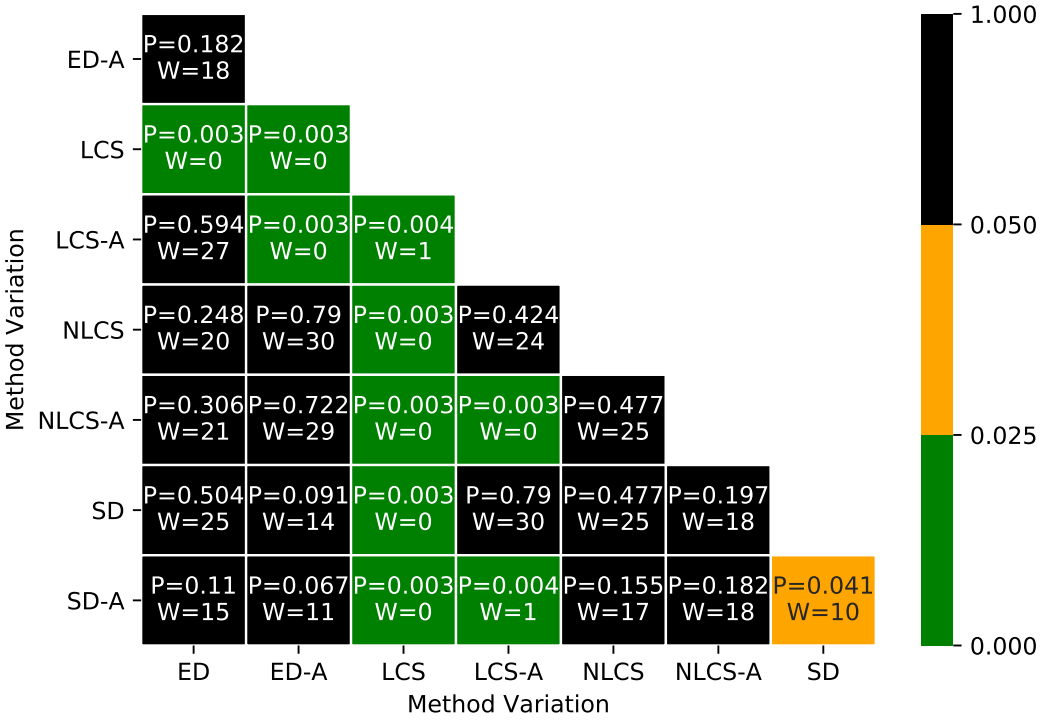


Figure 4.17: 3D teddy bear model out-of-plane rotation method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the mAP for an individual out-of-plane rotation for two OSCC methods. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two method variations using matched pairs from all dataset variations. See appendix C.7 for further details

With the exception of the LCS method, all methods achieve similar mAP and SD values across all transformations. This shows an overall robustness to the perspective transformation with respect to the out-of-plane rotated teddy bear. This observation can be explained by the relatively coarse binning of relative edge features by using a bin size of 45 degrees in addition to the teddy bear having a good amount of shape information. Figure 4.17 shows that the pairwise median difference of AP between the ED-A, NLCS, NLCS-A, SD and SD-A methods is not significantly greater than zero. Whereas the pairwise median difference between these methods and the lower scoring LCS method obtain a median difference of AP that is significantly greater than zero. The lower performance of the LCS method can be explained by the fact that it does not penalise for matching rings of different sizes with uncommon tokens.

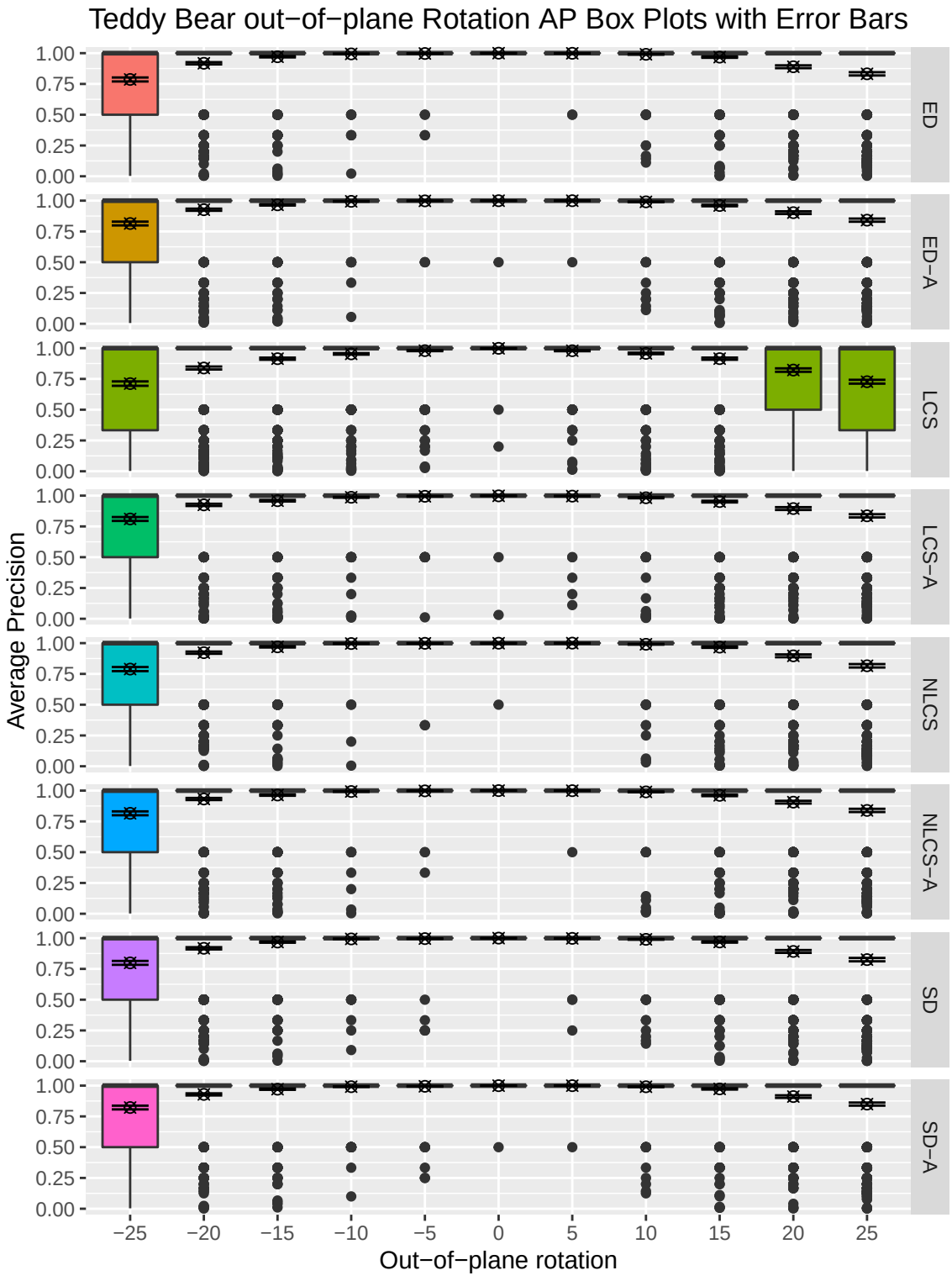


Figure 4.18: Out-of-plane teddy bear rotation / OSCC method variation results visualisation using the Average Precision metric.

4.7 Clutter Robustness

Two variations of clutter are used to inspect the clutter robustness of OSCC methods. Figure 4.19 illustrates both variations of clutter, the image to the left shows the first variation applied to the image of character 'a', where clutter is presented to the left of the character. The second variation is shown by the image to the right which has clutter applied to either side of the character 'a'. Both variations of clutter are applied to all characters of the alphabet dataset to obtain the alphabet clutter dataset.

Table 4.21 shows the mAP results for each clutter variation / OSCC method. The baseline, single clutter and multiple clutter variations are shown by columns 1, 2 and 3 respectively. Furthermore, figure 4.20 shows two-sided Wilcoxon Matched-Pairs Signed-Rank Test results for each pairwise method variation across all clutter variations

Figure 4.19: Examples of the two variations of clutter applied to the character 'a'. The first variation of clutter is illustrated by the character to the left which has clutter to its left. The second variation is illustrated by the character to the right which has clutter on either side of the character.

The NLCS-A and ED-A methods are the most discriminative when clutter is present, where the median difference of AP between NLCS-A and ED-A is not significantly greater than zero (W=1139, P=0.311) with a p-value threshold of 0.025 - while the median difference of AP between both NLCS-A and ED-A compared to all other variations is significantly greater than zero. This can be explained by the properties of the NLCS method variation which has preference for the Longest Common Subsequence for similar sized rings, therefore allowing for clutter while penalising for larger differences between ring sizes. Furthermore, the Edit Distance penalises for insertions, therefore also allowing for clutter while penalising for too much clutter. On the other hand, the LCS method performs the worst as it allows for clutter but does not penalise for differences between ring sizes with common subsequences. The SD variation approaches do not perform as well as the NLCS and ED variations as the underlying feature distance approach does not directly handle insertions introduced by clutter as a binary cost, instead the minimum feature distance

is calculated. Overall, the results show a robustness to clutter with respect to the variations applied to the alphabet dataset.

Clutter	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
1	0.72 ±.21	0.71 ±.27	**0.95** ±.12	0.88 ±.22	0.89 ±.18	0.52 ±.32	0.94 ±.15	0.85 ±.24
2	0.75 ±.21	0.67 ±.24	**0.94** ±.14	0.88 ±.20	0.86 ±.21	0.43 ±.28	**0.94** ±.15	0.84 ±.23
mAP ± SD	0.80 ±.22	0.75 ±.26	**0.95** ±.12	0.91 ±.18	0.90 ±.17	0.56 ±.33	0.94 ±.15	0.87 ±.22

Table 4.21: Alphabet Dataset point correspondence clutter variation results using the mAP metric.

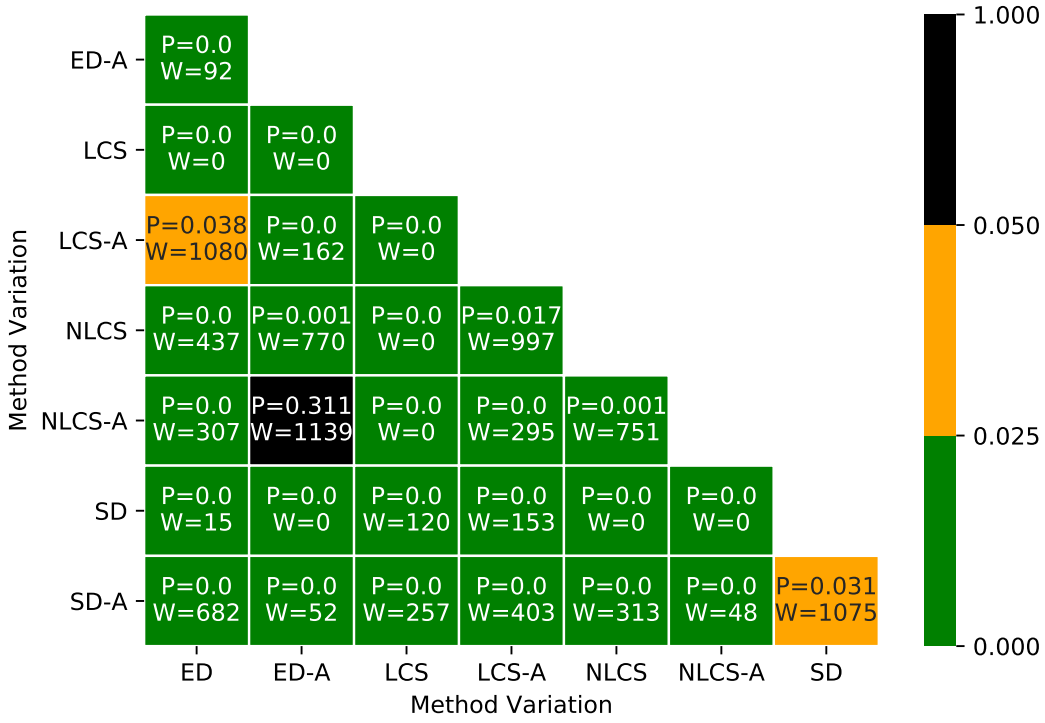


Figure 4.20: Clutter method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Average Precision for an individual transformed alphabet character for two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix C.8 for further details.

Figure 4.21 shows the lower level Average Precision scores for each clutter character / OSCC method. It is clearly shown that the positional angle constraint adds extra discriminability for the ED-A, LCS-A and NLCS-A methods. Furthermore, by visually comparing the best performing ED-A and NLCS-A clutter variation results for each character with the baseline results in the first column, the least discriminative outlier characters are identified as 'l', 'o' and 'x' which is consistent with the baseline results.

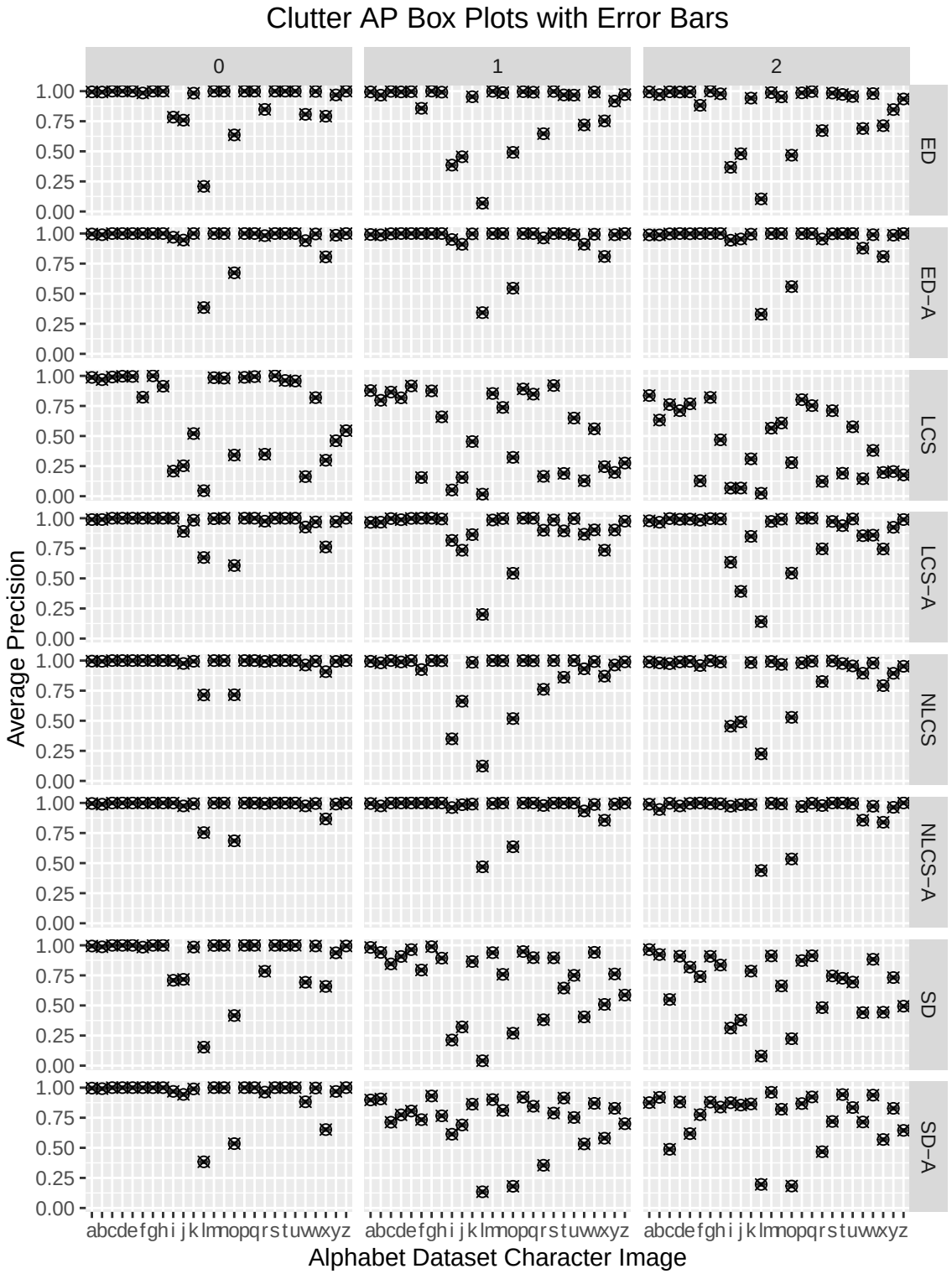


Figure 4.21: Clutter variation Average Precision scores for each alphabet character. An Average Precision score represents the average of precision scores obtained from each individual ground truth correspondence between a reference image and a transformed image.

4.8 Occlusions Robustness

A single variation of occlusion is used to inspect the occlusion robustness of OSCC methods. Figure 4.22 illustrates the occlusion variation which is defined as a horizontal line which passes through the middle of an alphabet character image. Table 4.22 shows the high level mAP scores for each occlusion variation / OSCC method and figure 4.23 shows two-sided Wilcoxon Matched-Pairs Signed-Rank Test results for each pairwise method variation across all occlusion variations. Furthermore, table 4.22 provides a lower level inspection of the Average Precision score for each character with respect to each variation.

Figure 4.22: Example of the occlusion variation applied to the three character images of 'a', 'b' and 'c'

Occlusion	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0	0.93 ±.16	0.88 ±.21	**0.97** ±.08	**0.97** ±.08	0.95 ±.10	0.71 ±.33	0.95 ±.13	0.91 ±.17
1	0.76 ±.14	0.60 ±.21	0.78 ±.17	0.73 ±.20	0.77 ±.18	0.38 ±.26	**0.82** ±.13	0.74 ±.18
mAP ± SD	0.85 ±.17	0.74 ±.25	0.87 ±.16	0.85 ±.19	0.86 ±.17	0.54 ±.34	**0.88** ±.15	0.82 ±.20

Table 4.22: Alphabet Dataset point correspondence occlusion variation results using the mAP metric.

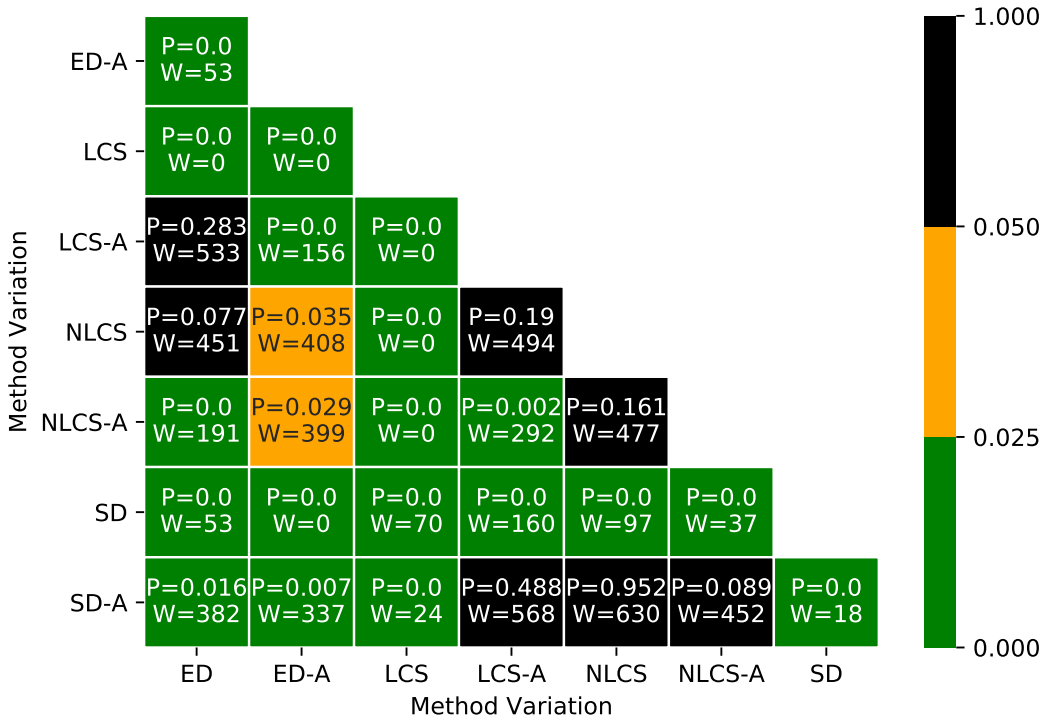


Figure 4.23: Occlusion method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Average Precision for an individual transformed alphabet character for two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix C.9 for further details.

As shown by table 4.22, for the occluded dataset, the ED-A method achieves the highest mAP score. A further inspection using figure 4.23 shows that the median difference of AP between the pairs of (ED-A, NLCS-A) and (ED-A, NLCS) is not significantly greater than zero as shown by (W=451, P=0.077) and (W=533, P=0.283) respectively with a p-value threshold of 0.025. The good performance of the ED-A method can be explained by its property of penalising for both insertions and deletions which are a result of occlusion. Furthermore, the good performance of the NLCS based approaches can be explained by their preference for matching rings of the same size while allowing for repeated tokens with the goal of being robust to noise.

A lower level inspection is provided by figure 4.24 which shows the lower level Average Precision scores for each occlusion character / OSCC method. As also observed for the clutter results, the positional angle constraint is shown to add a significant

amount of discriminability. The ED-A variation shows that the Average Precision for all characters drop but remain relatively close with respect to the baseline results which are shown by the first column. Furthermore, characters 'l' and 'o' are commonly shown to be within the bottom ranks for each method which reflect the baseline results. Overall, robustness to occlusion with respect to the alphabet dataset is shown.

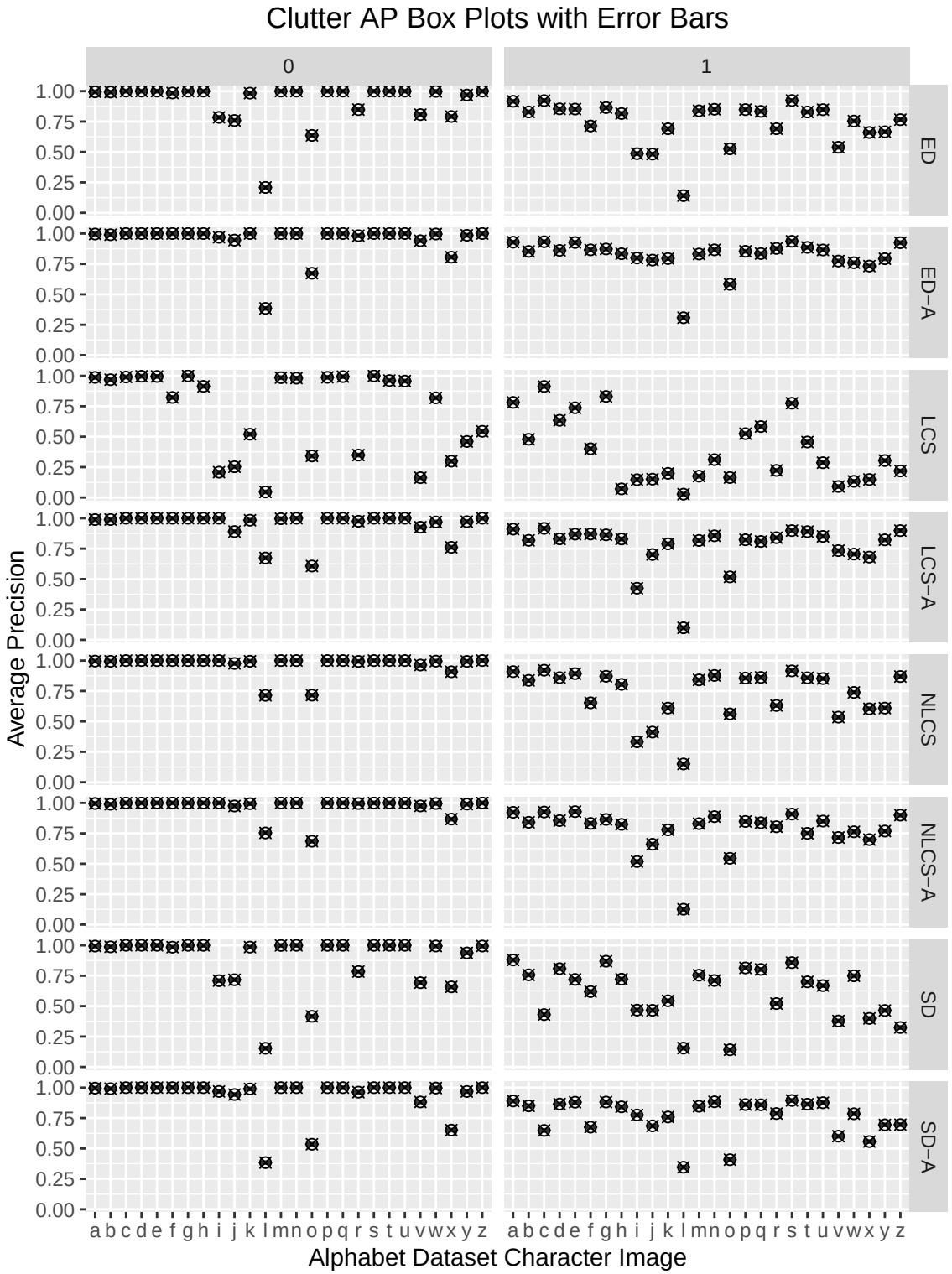


Figure 4.24: Occlusion variation Average Precision scores for each alphabet character. An Average Precision score represents the average of precision scores obtained from each individual ground truth correspondence between a reference image and a transformed image.

4.9 Interest Region Description Evaluation

This section provides an initial evaluation of OSCC method variations with respect to local interest regions. The evaluation framework defined by Mikolajczyk and Schmid [18] is firstly used to extract regions of interest from images using the affine invariant Harris-Affine detector. With a known homography between images, ground truth correspondences between the extracted regions are identified. Secondly, with respect to nearest neighbor matching of regions, the evaluation framework is used to compare OSCC method variations with the Generalised Shape Context (GSC), Scale Invariant Feature Transform (SIFT) and Gradient Location and Orientation Histogram (GLOH) local descriptor methods.

The alphabet dataset defined in section 4.2.2 is used for the interest region evaluation as it contains well defined shapes. It has been found that local region datasets do not generally contain reliable edge information that is suitable for the OSCC method. The purpose of this evaluation is to compare the OSCC method in the context of local patch descriptors instead of the semi-local / global setting that it was designed for. Furthermore, the Generalised Shape Context is used within the evaluation which was also designed for a semi-local / global setting with respect to sampling shape information.

Figure 4.25: Harris-Affine detector regions for both character image 'a' and character image 'a' rotated by 45 degrees

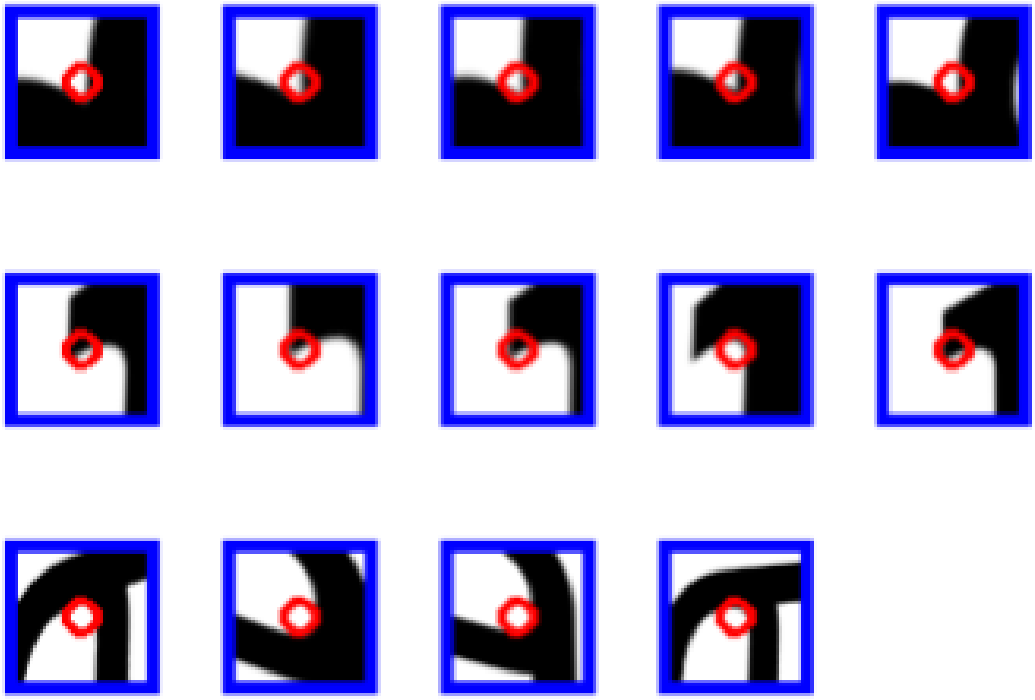


Figure 4.26: Normalised Harris-Affine Patches. The first patch of each row represents a reference patch and the other patches within a row represent ground truth correspondence patches. A blue border is added to each patch for clarity. A red circle is used to show the center of each patch.

Following the evaluation framework defined by Mikolajczyk and Schmid [18] (using the accompanying code for both patch extraction and evaluation metrics) regions are detected for each reference and transformed alphabet character image using the affine invariant Harris-Affine detector (see figure 4.25). Each detected region is then normalised to a 41 x 41 pixel patch (see figure 4.26). Ground truth patch correspondences between reference image and transformed image are identified by using the associated homography matrix. Figure 4.26 shows three examples of identified ground truth correspondence patches with respect to reference patches. The first patch of each row in the figure represents a reference patch and the rest of the patches in a row represent ground truth correspondence patches. See appendix D for further details about the obtained dataset which includes the number of pairwise patch comparisons and ground truth correspondences between reference and transformed images.

Table 4.23 shows the mAP scores for each alphabet transformation / descriptor method combination. Interp and Trans represent interpolation and translation respectively. The OSCC SD-A variation is shown to achieve the highest score for the

interpolation, translation, shear x and shear y alphabet dataset variations. GLOH is shown to achieve the highest score for the rotation and scale variations. Interestingly, it is observed that GLOH always achieves a better score than SIFT, while SIFT always achieves a better score than GSC.

Method	Interp	Trans	Rotations	Scales	Shear x	Shear y	Mean	SD
SD-A	**0.59**	**0.45**	0.24	0.34	**0.33**	**0.30**	**0.37**	0.12
SD	0.55	0.41	0.22	0.31	0.29	0.27	0.34	0.11
ED-A	0.57	0.43	0.20	0.34	0.30	0.28	0.35	0.12
ED	0.55	0.41	0.21	0.32	0.29	0.26	0.34	0.11
NLCS-A	0.57	0.43	0.21	0.32	0.32	**0.30**	0.36	0.11
NLCS	0.57	0.42	0.23	0.32	0.30	0.28	0.35	0.11
LCS-A	0.49	0.31	0.16	0.24	0.22	0.21	0.27	0.11
LCS	0.55	0.41	0.21	0.32	0.29	0.27	0.34	0.11
GLOH	0.56	0.41	**0.32**	**0.37**	0.30	0.27	**0.37**	0.10
SIFT	0.55	0.40	0.31	0.35	0.29	0.26	0.36	0.10
GSC	0.54	0.38	0.29	0.34	0.28	0.25	0.35	0.10

Table 4.23: Mean Average Precision scores for each alphabet transformation / local descriptor method combination.

With respect to the non OSCC methods and with the exception of the rotation and scale variations, SD-A and NLCS-A always obtain a higher score while ED-A always obtains a higher or equal score. However, it should be noted that all scores are fairly close. A possible explanation for the cases where the OSCC methods achieve a higher score is that they capture information at a fine level instead of a coarse level, therefore allowing higher discriminability for variations such as interpolation, translation, shear x and shear y - it should be noted that the output from the affine invariant region detector is normalised, therefore the interpolation, translation and shear patches should be similar. The relatively lower performance of the OSCC methods that is observed for rotations within table 4.23 could possibly be explained due to sampling offsets of the fine grained sampling. This could be caused due to the origin of the sampling pattern not being anchored to an edge. Figure 4.27 shows that the median difference of mAP between the highest scoring OSCC method (SD-A) and all other competitor methods is not significantly greater than zero with a p-value threshold of 0.05.

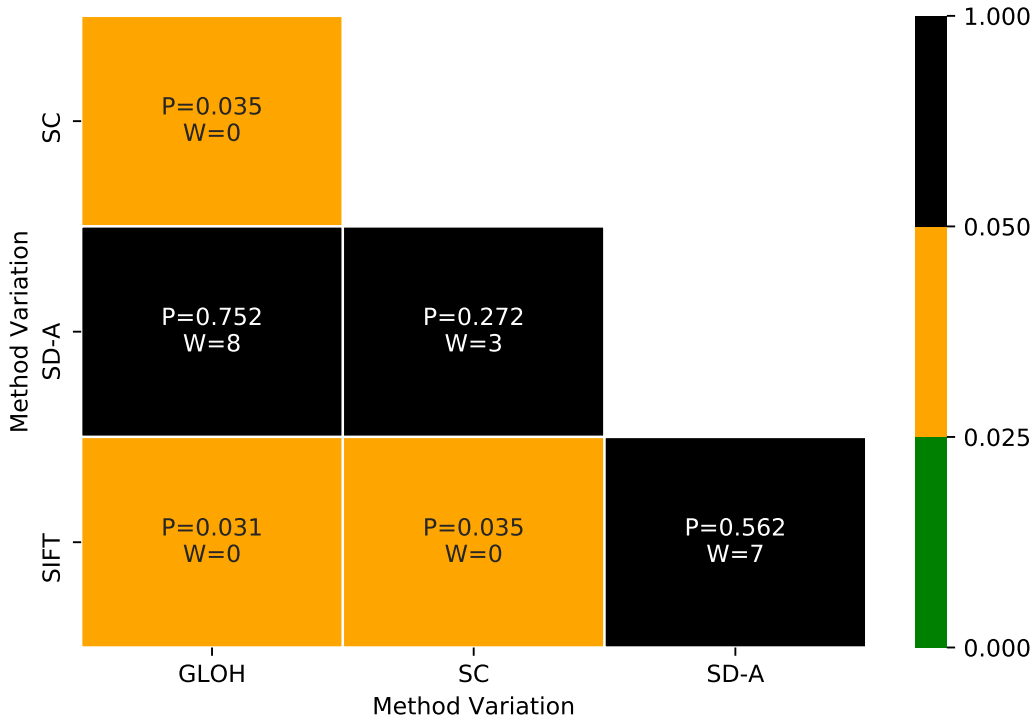


Figure 4.27: Interest region descriptor method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the mAP for an individual transformation for two interest region descriptor methods as defined by table 4.23 (using the best performing OSCC method only). An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two interest region descriptors using matched pairs from all dataset variations. See appendix D for further details.

Overall, these preliminary results give an insight into the relative performance of the OSCC methods within the context of local region description when reliable edge information is available. As previously noted, the OSCC methods have been designed to work with semi-local / global shape information. Therefore future work should concentrate on evaluating descriptors with respect to varying region sizes for the context of semi-local / global shape region description.

4.10 Summary

In this chapter, all transformations have been covered with respect to the evaluation requirements defined in table 4.1. Table 4.24 shows the mAP score for overall trans-

formation / OSCC method combinations. Each transformation has been previously inspected with its own dedicated section within this chapter. This is where the overall mAP for an individual transformation has been obtained from. Furthermore, figure 4.28 accompanies table 4.1 by showing the two-sided Wilcoxon Matched-Pairs Signed-Rank Test for each OSCC method combination across all transformations.

The high level results within table 4.1 show that the NLCS-A method obtains the highest frequency of best scores, followed by NLCS, ED-A and SD-A. ED-A achieves the best score for scale and occlusion while tieing with the best score for perspective which is also achieved by the SD-A and NLCS-A methods. Furthermore, SD-A achieves the best score for shear. Overall, the NLCS-A achieves the highest mean score across all transformations. With respect to statistical significance, figure 4.28 shows that the overall top four scoring methods with the highest mean score over all transformations (NLCS-A, NLCS, ED-A, SD-A) all have a pairwise median difference of mAP that is not significantly greater than zero (with a P value threshold of 0.05). These four methods all penalise for matching rings with different sizes while three of the four make use of the positional angle constraint. Furthermore, it is shown that the lowest scoring LCS method has a pairwise median difference of mAP with all other methods that is significantly greater than zero. This method does not use the positional angle constraint and it does not penalise for matching rings of different sizes when uncommon tokens exist between them.

	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
Interpolation	0.95	0.91	**0.98**	**0.98**	0.97	0.75	0.96	0.93
Translation	0.93	0.88	**0.96**	**0.96**	0.93	0.68	0.94	0.90
Reflection	0.01	0.01	0.01	**0.03**	0.00	0.00	0.01	0.01
Rotation	0.89	0.82	**0.91**	**0.91**	0.88	0.63	0.88	0.83
Scale	0.89	0.82	0.88	0.82	0.82	0.59	**0.90**	0.84
Shear	**0.37**	0.31	0.33	0.34	0.23	0.12	0.32	0.31
Perspective	**0.96**	0.95	**0.96**	0.95	0.95	0.91	**0.96**	0.95
Clutter	0.80	0.75	**0.95**	0.91	0.90	0.56	0.94	0.87
Occlusion	0.85	0.74	0.87	0.85	0.86	0.54	**0.88**	0.82
Mean ± SD	0.74 ± .31	0.69 ± .30	**0.76** ± .33	0.75 ± .31	0.73 ± .33	0.53 ± .27	0.75 ± .32	0.72 ± .31

Table 4.24: Overall transformation / OSCC method mAP scores

As stated in section 4.4.1, OSCC methods are not robust to reflection transformations. This is not considered to be a problem as there is a tradeoff with respect to reflection invariance and overall descriptor discriminability. For a descriptor to have

both horizontal and vertical reflection invariance, a large proportion of discriminative information would be lost. As a result, robustness to other transformations would be hindered. If reflection invariance is required for an application specific reason, then preprocessing steps can be undertaken or the encoding can be modified.

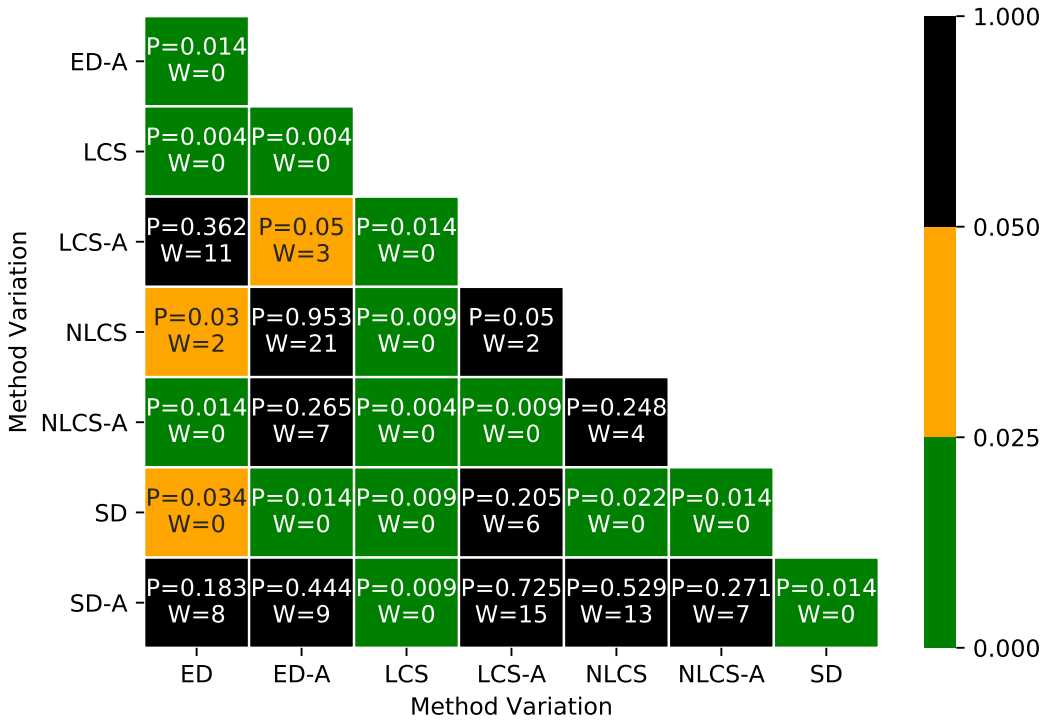


Figure 4.28: Overall method variation two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the mAP between two method variations for a particular dataset transformation defined within table 4.24. An individual entry within the figure shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and two-sided Wilcoxon test statistic (W) value for two method variations using matched pairs from all dataset transformations. As nine dataset variations are used, nine matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test

The lowest ranked characters identified for the interpolation variation are ‘l’, ‘o’, ‘x’, ‘v’ and ‘j’. Furthermore ‘l’, ‘o’, ‘i’, ‘j’, ‘x’, ‘v’ are identified as the lowest ranked characters for the translation variations. Out of these lowest ranked characters, ‘l’, ‘o’ and ‘x’ are identical when rotated by 180 degrees resulting in duplicate point perspectives for the shapes. Whereas characters ‘i’, ‘j’ and ‘v’ are identified as having a relatively low amount of shape information, resulting in a less discriminative set of descriptors for exact point correspondences. For example, character ‘v’ has a relatively

low amount of shape information as it is constructed from just two lines. With respect to the OSCC coding, edge information of the two lines is binned with respect to relative orientation and positional angle, therefore a point perspective of the two lines contains binned line segments. These binned line segments allow for more flexible matching with the tradeoff of less discriminability for shapes that have a low amount of shape information. Consequently, exact point correspondences are not always retrieved, however it is believed that the OSCC methods are discriminative enough in the context of object recognition which is to be investigated in the next chapter.

Characters 'l', 'o', 'i', 'j', 'x', 'v' also appear within the lowest ranked characters for the tested scale variations for the same reasons as previously described. As described in section 4.4.4, it is interesting to note that the Longest Common Subsequence based ring correlation methods perform worse for the scales tested of less than 1. More specifically, characters 'l', 'i', 'j' and 'f' are the least discriminative for this case. All of these characters show a relatively low amount of shape information while the Longest Common Subsequence based approaches do not directly penalise for insertions, substitutions and deletions. It should be noted that these approaches perform well for characters with relatively more shape information across all scales. As a result, these factors should be taken into consideration for object level tasks.

Characters 'o', 'l', 'z', 'c', 'i' and 'x' are the most frequent lowest ranked for shear x across the variations and 'o', 'c', 'z' 'l', 'g' and 'x' are the most frequently lowest ranked for shear y as shown in section 4.5.1. Although characters 'l' and 'i' appear to be the most similar under the shear transformation by eye, their relatively low amount of shape information combined with the relatively high amount of information being disturbed results in poor point correspondence discriminability. On the other hand, the coding and correlation methods should still be discriminative with respect to other shapes for high level object tasks. Furthermore, character shapes for 'o', 'z', 'c' and 'x' can be significantly changed with shear transformations. The high level results in table 4.24 show that the SD-A approach is the most discriminative for shear transformations. This can be explained by the use of relative edge feature bin distances as opposed to the other methods that use binary bin matches.

Perspective transformation robustness is shown by the out-of-plane teddy bear rotations within section 4.6. The shape of the teddy bear has a relatively high amount of shape information which results in high discriminability across all OSCC variations. Furthermore, the methods show robustness to perspective transformation of the teddy bear which can be explained by the coarse edge feature binning in addition to the shape information provided. As shown by the high level Mean Average Precision

scores for each method in table 4.24, all methods achieve a similar score across the out-of-plane rotations with the LCS method standing out as an outlier with the worst performance. However, the positional angle constraint is shown to improve the LCS discriminability as shown by the LCS-A method.

The NLCS-A and ED-A methods are shown by table 4.24 to be the most discriminative for the clutter transformation. As described within section 4.7, these methods perform better than the others as they both penalise the matching of different size rings as well as allowing for insertions, deletions and substitutions. The SD based variations on the other hand force all features to be matched while taking the binned distance between them. Although the LCS method allows for insertions, deletions and substitutions it does not penalise for rings of different sizes matching each other as opposed to the NLCS based methods.

For occlusion robustness, table 4.24 shows that the ED-A method performs the best. As discussed in section 4.8, this can be explained by the Edit Distance properties of directly penalising the effects of occlusion which are insertions and deletions of positional angle features. These properties are shown to give the ED-A method an edge over the other methods as previously summarised, such as the SD based approaches that force features to be matched and the LCS based approaches that penalises for shorter common subsequences.

In the next chapter, we build upon this chapter by undertaking an initial evaluation of the OSCC method with respect to higher level object recognition tasks using the alphabet dataset. Results are presented in the same style as this chapter for consistency and ease of interpretation.

Chapter 5

Synthetic Object Level Evaluation

5.1 Introduction

In the previous chapter, we undertook point correspondence level evaluations with respect to variations of the OSCC methods. The point correspondence level evaluations covered similarity, affine and perspective transformations as well as clutter and occlusion. In this chapter, we build upon the previous chapter by undertaking an initial evaluation of OSCC method variations with respect to the higher level object recognition task. More specifically, a simple object classification metric is defined which is based on local Hough voting of point correspondences. The object classifier is used to train reference images and then recall the reference images with applied transformations. Results are presented in the same format as the previous point correspondence evaluation chapter for consistency. The next chapter undertakes an object level recognition evaluation using the MNIST handwritten digits dataset.

5.2 Object Classification Metric

A local Hough style voting object level metric is used throughout this chapter as an object classification metric. This approach has been selected due to its simplicity and flexibility. Both positional angle and distance parameters of point correspondences are encoded within the accumulator matrix, resulting in an accumulator matrix of size $\mathcal{O}(XY)$, where X is the number of scale bins and Y is the number of angle bins.

As described in section 3.6.3 of chapter 3, the intrinsic property of OSCC sequence alignment allows a rich set of point correspondences to be extracted between two point descriptors. Therefore once two point descriptors have been identified as a good match, then a rich set of point correspondences can be extracted between them. The overall approach of the object classification metric first creates a pairwise cost matrix of point descriptors between a query image and a template image using the OSCC ordinal ring sequence correlation approach defined in section 3.6.2 of chapter 3. The lowest cost matches for each point descriptor of the query image with respect to the reference image are then extracted. Using these lowest cost matches between descriptors of the query and template image, a rich set of correspondences for each descriptor match are extracted using OSCC sequence alignment which are then used with the Hough style voting object level metric.

For completeness, algorithm 11 defines the process of obtaining OSCC point descriptors for an image. Algorithm 3 (*getPointPerspectiveCoding*) in chapter 3 is referred to. The *nBins* parameter defines the number of bins to be used for both RESO and PA features. *maxRadius* defines the maximum radius for the concentric ring sampling while *rx* defines the spacing between the rings.

Algorithm 11 Image Descriptor Extraction

```
1: function imgDescriptors(img, pt, nBins, maxRadius, rx)
2:     edgeMap ← binary edge map from the canny edge detector applied to img
3:     ptDescriptors ← {}
4:     for all pt ∈ edgeMap do
5:         desc ← getPointPerspectiveCoding(img, pt, nBins, maxRadius, rx)
6:         ptDescriptors ← ptDescriptors ∪ desc
7:     end for
8:     return ptDescriptors
9: end function
```

The first step of the object classification metric is defined by algorithm 12 which creates a pairwise cost matrix of point descriptors between a query image and reference image using the OSCC ordinal ring sequence correlation approach (previously defined by algorithm 8 *getRingSequenceDTWAccCostMat*) in chapter 3). The algorithm then returns the best match descriptor indices with respect to the reference image for each individual query image descriptor. The input parameters $qImgDescs$ and $rImgDescs$ correspond to the descriptors of the query and reference images respectively which can be obtained by algorithm 11.

Algorithm 12 Descriptor Best Matches

```
 1: function descBestMatches(qImgDescs, rImgDescs)
 2:     costMat ← [qImgDescs length][rImgDescs length]
 3:     for qIdx ← 0 to qImgDescs length do
 4:         for rIdx ← 0 to rImgDescs length do
 5:             qDesc ← qImgDescs[qIdx]
 6:             rDesc ← rImgDescs[rIdx]
 7:             accMat ← getRingSequenceDTWAccCostMat(qDesc, rDesc)
 8:             cost ← accMat[qDesc length −1][rDesc length −1]
 9:             costMat[qIdx][rIdx] ← cost
10:         end for
11:     end for
12:     qDescBestMatches ← []
13:     for qIdx ← 0 to qImgDescs length do
14:         for rIdx ← 0 to rImgDescs length do
15:             rIndices ← costMat[qIdx] == min(costMat[qIdx])
16:             qDescBestMatches ← qDescBestMatches ∪ [rIndices]
17:         end for
18:     end for
19:     return queryDescBestMatches
20: end function
```

The second step of the object classification metric is defined by algorithm 13 which applies the OSCC sequence alignment method to a set of best matching descriptors between the query and reference image to obtain a rich set of correspondences. The rich set of correspondences are used for a Hough style voting object level metric. Algorithm 10 (*getOsccSequenceAlignmentHierarchy*) from chapter 3 is used to obtain the rich set of correspondences from two descriptors using the OSCC sequence alignment method. The input parameters $qImgDescs$ and $rImgDescs$ correspond to the descriptors of the query and reference images respectively which can be obtained by algorithm 11. Furthermore, the input parameters $aBins$, r and $rBins$ correspond

to the number of angle bins, maximum radius and radius bins for the Hough style voting respectively.

Algorithm 13 OSCC Sequence Alignment Hough Object Level Metric

```
1: function HOUGH(qImgDescs, rImgDescs, bestMatchIndices, aBins, r, rBins)
2:     accMat ← [qImgDescs length][rImgDescs length]
3:     for qIdx ← 0 to bestMatchIndices length do
4:         qDesc ← qImgDescs[qIdx]
5:         qDescOriginPt ← qDesc.getOrigin()
6:         for rIdx ← 0 to bestMatchIndices[qIdx] length do
7:             rDesc ← rImgDescs[bestMatchIndices[rIdx]]
8:             rDescOriginPt ← qDesc.getOrigin()
9:             corres ← getOsccSequenceAlignmentHierarchy(qDesc, rDesc)
10:            for cIdx ← 0 to corres length do
11:                qDescPt ← corres[cIdx][0]
12:                qDistance ← distance(qDescOriginPt, qDescPt)
13:                if qDistance <= r then
14:                    rDescPt ← corres[cIdx][1]
15:                    rDistance ← distance(rDescOriginPt, rDescPt)
16:                    qAngle ← angleBetween(qDescOriginPt, qDescPt)
17:                    rAngle ← angleBetween(rDescOriginPt, rDescPt)
18:                    diffDist ← qDistance/rDistance
19:                    diffAngle ← qAngle − rAngle
20:                    distBin ← floor(rBins ∗ (diffDist/r))
21:                    angleBin ← floor(aBins ∗ (diffAngle/359))
22:                    accMat[distBin][angleBin] ← accMat[distBin][angleBin] + 1
23:                end if
24:            end for
25:        end for
26:        accMat ← accMat/sum(accMat)                    ▷ normalise
27:    end for
28:    return max(accMat)          ▷ normalised value of highest voted bin
29: end function
```

5.3 Experiment Design

The alphabet dataset defined within section 4.2.2 is used to inspect the OSCC approaches for the task of object recognition. Following the structure of the previous chapter, this section inspects the object recognition capabilities of OSCC method variations with transformation variations of the alphabet dataset. Appendix E provides details about each dataset transformation variation.

For each OSCC method variation / transformation variation, a pairwise cost matrix is created between all reference and transformed images. This results in a 26 x 26 cost matrix as 26 reference images and 26 transformed images are used. Furthermore, the OSCC classification metric is used to obtain the costs as defined in section 5.2. The resulting pairwise cost matrix can be used to obtain standard classifier performance metrics such as Precision and Recall as defined in section 4.2.4.

5.3.1 Method Variation Parameters

All method variations and their parameters are reused from the previous chapter as defined in section 4.2.5 which are kept constant throughout this chapter. Therefore a global sampling area is used for each point descriptor. Furthermore, the object classification metric defined in section 5.2 is used throughout this chapter which has three parameters defined as $aBins$, r and $rBins$. These parameters are fixed to 8, 15 and 5 respectively.

5.4 Interpolation

Like the point correspondence interpolation evaluation (see section 4.3), two variations of interpolation are evaluated, None and Cubic. The None variation does not apply any interpolation and the Cubic variation applies bicubic interpolation using a 4 x 4 neighborhood. Table 5.1 shows the high level mAP scores for each interpolation transformation / OSCC method. Overall, all variation combinations achieved a high score as expected. It is interesting to note that the Cubic variation for the NLCS-A and LCS methods achieve a score of 1.0 while their corresponding scores of 0.98 and 0.97 respectively are achieved for the None variation (this can be explained due to small interpolation artefacts). Following this observation, scores for the Cubic variation are consistently more than or equal to the None variation. The largest difference between the two variations is 0.04.

Figure 5.2 shows the lower level interpolation variation precision scores for each alphabet character query. First of all, it is clear that the character ‘l’ is consistently misclassified for all instances of the None variation while also being misclassified for all instances of the Cubic variation with the exception for the LCS and NLCS-A methods. This observation explains that the correct classification of the character ‘l’ is the reason why the high level mAP scores for the LCS and NLCS-A methods combined with the Cubic variation achieved a score of 1.0. A score of 0.5 means that the ground

truth trained image was either ranked second within the recall images or the ground truth trained image was ranked first while being tied with one other trained image. Overall, figure 5.1 shows that the median difference of Precision between all methods

Interpolation	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
None	0.97 ±.14	0.95 ±.19	**0.98** ±.10	0.97 ±.16	**0.98** ±.10	0.97 ±.13	**0.98** ±.10	0.97 ±.16
Cubic	0.97 ±.13	0.95 ±.19	**1.00** ±.00	0.98 ±.10	0.97 ±.14	**1.00** ±.00	0.94 ±.16	0.97 ±.15
mAP ± SD	0.97 ±.00	0.95 ±.00	**0.99** ±.01	0.97 ±.01	0.98 ±.00	**0.99** ±.01	0.96 ±.02	0.97 ±.00

Table 5.1: Overall interpolation transformation / OSCC variation mAP ± SD

is not significantly greater than zero with a p-value threshold of 0.05. All misclassified characters are either 'l', 'o' or 'q'. Characters 'l' and 'o' have many similar features as they are identical when rotated by 180 degrees. Furthermore, 'q' appears to be the same as the trained character of 'b' when rotated by 180 degrees. It should also be noted that the character 'l' has a score of less than 0.5 for method variations other than ED and ED-A. This indicates that the object classification method is not discriminative enough with respect to the underlying correspondences of 'l'.

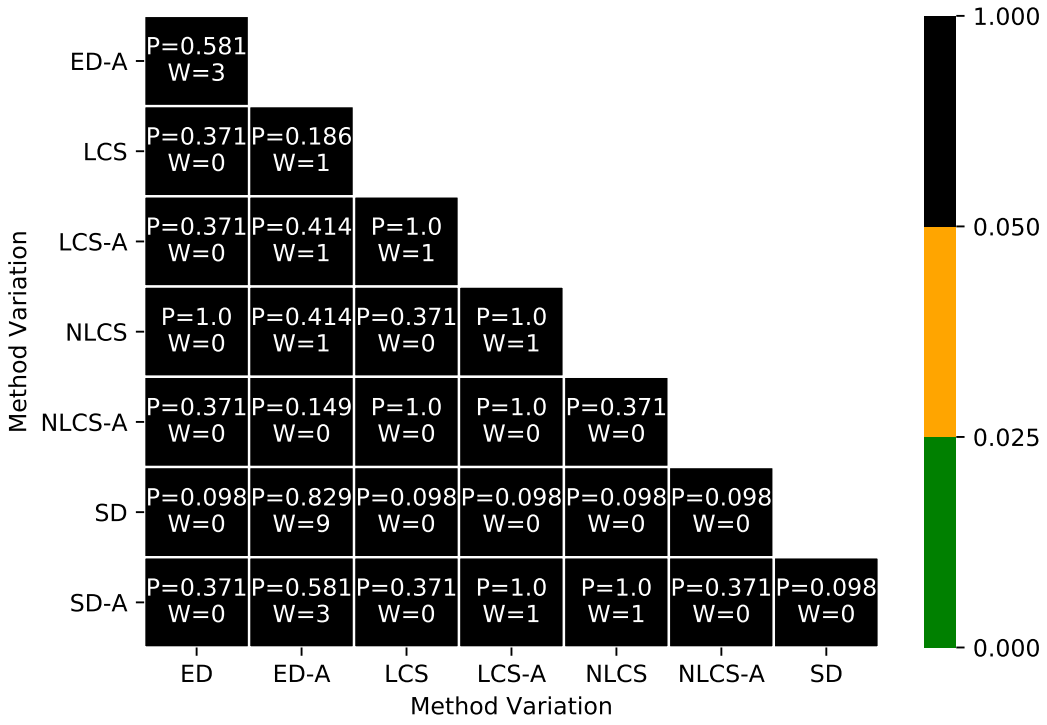


Figure 5.1: Interpolation method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix E.1 for further details.

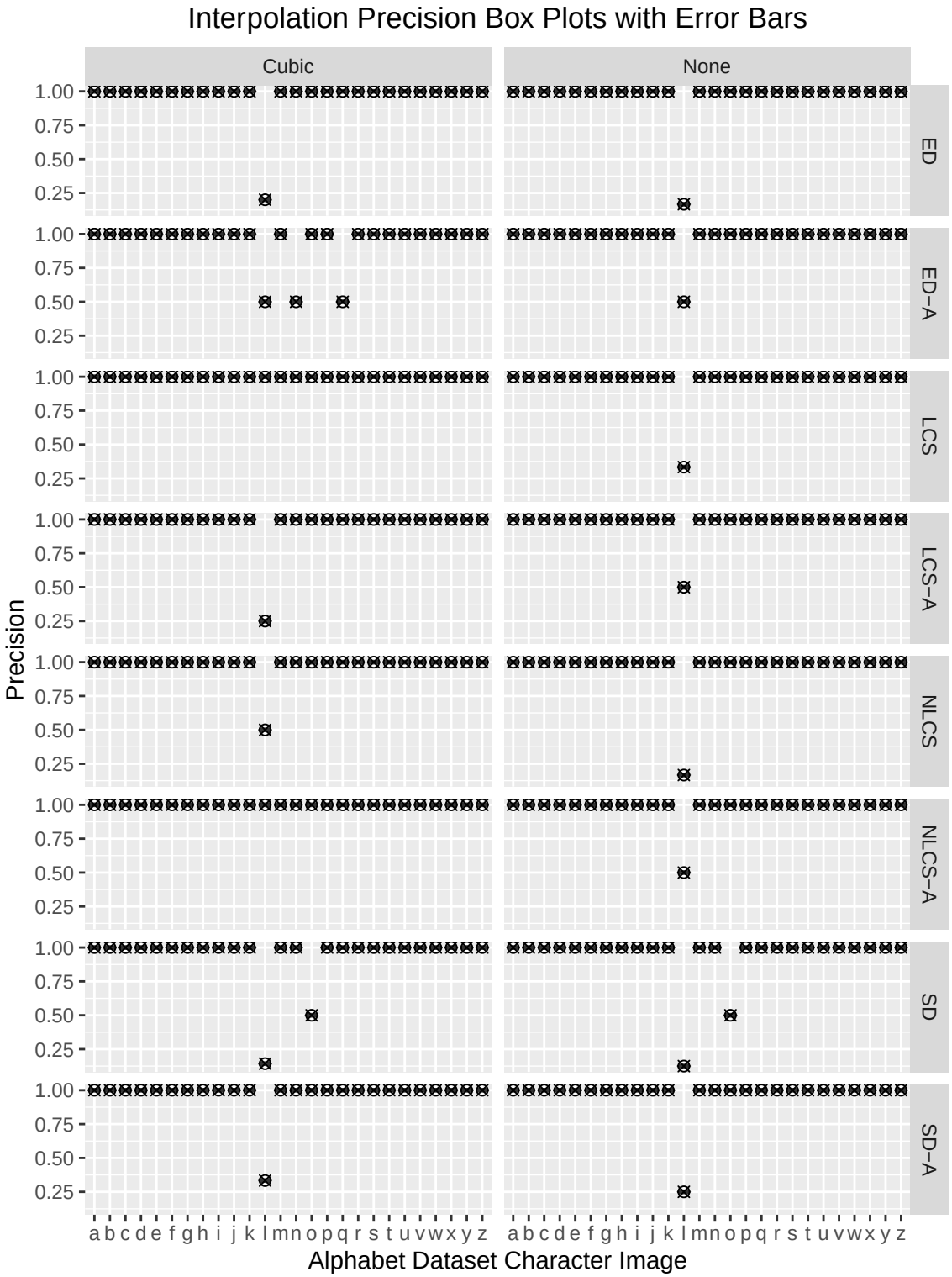


Figure 5.2: Interpolation variation precision scores for each alphabet character query.

5.5 Translation Invariance

Like the point correspondence evaluation, translations are evaluated with respect to combinations of exact and inexact discrete pixel grid mappings for shifts in both the x and y directions. Table 5.2 shows the high level mAP scores for each translation transformation / OSCC method. All combinations are shown to achieve a high score of more than 0.9 as expected. As with the interpolation results, it is observed that the median difference of Precision between all methods is not significantly greater than zero with a p-value threshold of 0.05 (see figure 5.3). This is expected as the translation variations only introduce interpolation artefacts combined with inexact discrete pixel grid mappings.

Translations	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
x 0.5 y 1.0	0.92 ±.23	0.93 ±.24	**0.98** ±.10	0.95 ±.18	**0.98** ±.10	0.94 ±.21	0.96 ±.13	0.95 ±.18
x 1.0 y 1.0	0.97 ±.13	0.95 ±.19	**1.00** ±.00	0.98 ±.10	0.97 ±.13	**1.00** ±.00	0.94 ±.16	0.97 ±.15
x 0.0 y 0.0	0.97 ±.13	0.95 ±.19	**1.00** ±.00	0.98 ±.10	0.97 ±.14	**1.00** ±.00	0.94 ±.16	0.97 ±.15
x 1.0 y 0.5	0.94 ±.20	0.91 ±.25	0.94 ±.16	0.94 ±.20	0.95 ±.18	0.95 ±.20	**0.98** ±.10	0.94 ±.22
x 0.5 y 0.5	0.92 ±.23	0.93 ±.23	0.95 ±.17	0.94 ±.20	0.94 ±.23	0.94 ±.20	**0.96** ±.18	0.94 ±.22
mAP ± SD	0.95 ±.02	0.93 ±.01	**0.97** ±.02	0.96 ±.02	0.96 ±.02	**0.97** ±.03	0.96 ±.01	0.95 ±.01

Table 5.2: Overall translation transformation / OSCC variation mAP ± SD

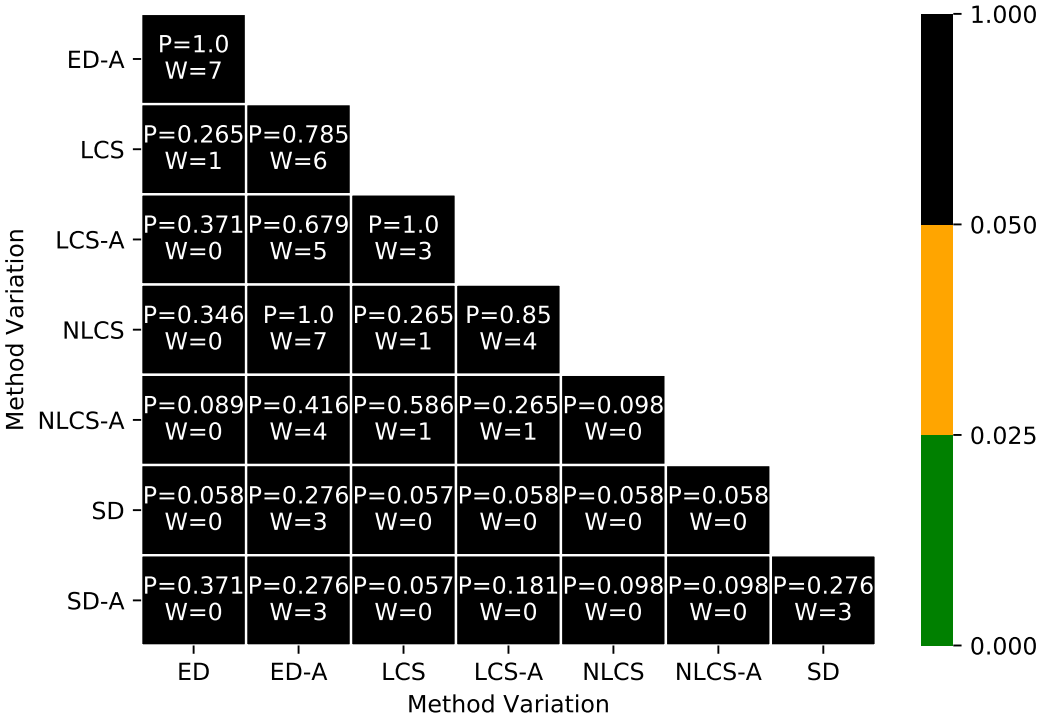


Figure 5.3: Translation method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the mAP for an individual transformation obtained by two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix E.2 for further details.

Figure 5.4 shows the lower level translation variation precision scores for each alphabet character query. It is observed that misclassifications occur for characters ‘l’, ‘n’, ‘o’, ‘q’ and ‘v’. However, characters ‘l’, ‘n’, ‘o’ are the most common misclassified characters across the methods. While characters ‘l’ and ‘o’ contain many similar features, ‘q’ is also identical to ‘b’ when rotated by 180 degrees and ‘n’ is identical to ‘u’ when rotated by 180 degrees. Overall, the translation variations show similar results to the interpolation results as expected. Misclassified characters of ‘l’, ‘o’ and ‘q’ were identified for the interpolation variations, while the translation variations introduce the additional misclassifications for ‘n’ and ‘v’. Characters ‘l’, ‘o’ and ‘v’ can all be described as having a relatively low amount of shape information as multiple occurrences of descriptors can be found for these shapes, this is especially the case for characters ‘l’ and ‘o’ due to both their rotational symmetry. Therefore the object

level metric is required to take this into account - in this case it does not.

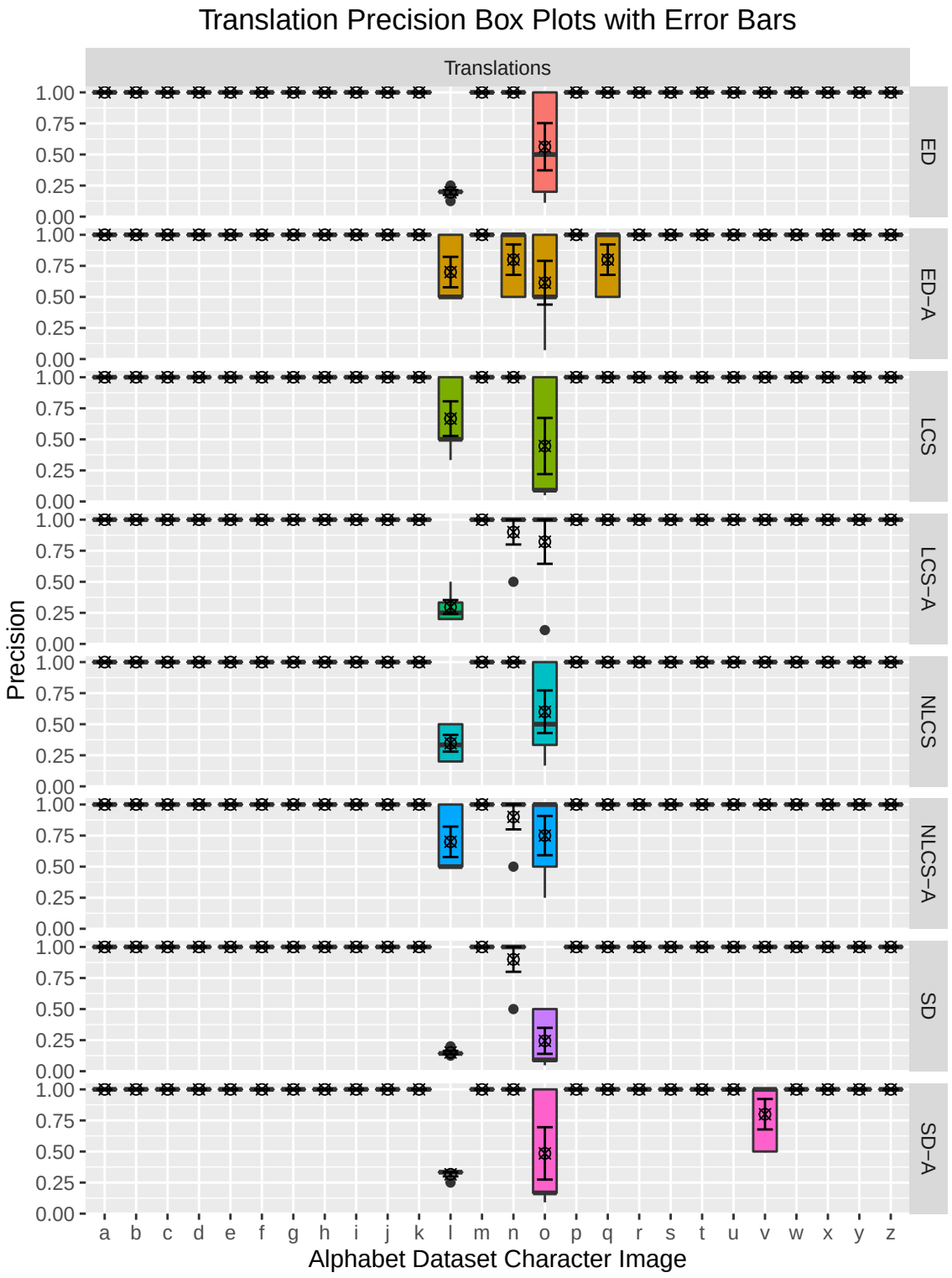


Figure 5.4: Translation variation precision scores for each alphabet character query.

5.6 Rotation Invariance

As with the point correspondence evaluation for rotation transformations (see section 4.4.3), clockwise rotations from 45 degrees to 315 degrees with intervals of 45 degrees are evaluated within this section. Table 5.3 shows the high level mAP scores for each rotation transformation / OSCC method. The NLCS and ED-A methods are shown to achieve the highest scores across the rotation variations with the exception of the 180 degree rotation where LCS achieved the highest score. The highest scoring overall method can be observed as the NLCS method whereas the lowest scoring method NLCS-A which adds the positional angle constraint - as shown by figure 5.5 the median difference of AP between both methods is not significantly greater than zero (P=0.051, W=2) with a p-value threshold of 0.05.

Rotations	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
45	0.90 ±.28	0.88 ±.28	0.86 ±.29	**0.91** ±.26	0.89 ±.29	0.79 ±.33	**0.91** ±.26	0.88 ±.29
90	0.88 ±.22	0.84 ±.26	0.85 ±.25	0.88 ±.22	0.88 ±.21	0.86 ±.30	**0.90** ±.21	0.84 ±.28
135	0.71 ±.35	0.70 ±.37	0.68 ±.35	**0.79** ±.33	0.70 ±.35	0.68 ±.39	0.74 ±.35	0.72 ±.35
180	0.72 ±.32	0.72 ±.34	0.76 ±.31	0.74 ±.28	0.78 ±.29	**0.79** ±.30	0.73 ±.32	0.75 ±.33
225	0.82 ±.29	0.83 ±.31	0.77 ±.30	**0.88** ±.26	0.81 ±.30	0.73 ±.35	0.84 ±.28	0.82 ±.31
270	0.88 ±.26	0.89 ±.23	0.92 ±.20	0.94 ±.18	0.92 ±.20	0.89 ±.26	**0.96** ±.16	0.89 ±.23
315	0.85 ±.31	0.83 ±.32	0.85 ±.33	**0.89** ±.29	0.86 ±.32	0.78 ±.35	0.84 ±.31	0.86 ±.31
mAP ± SD	0.82 ±.07	0.82 ±.07	0.81 ±.07	**0.86** ±.06	0.84 ±.07	0.79 ±.07	0.84 ±.08	0.82 ±.06

Table 5.3: Overall rotation transformation / OSCC variation mAP ± SD

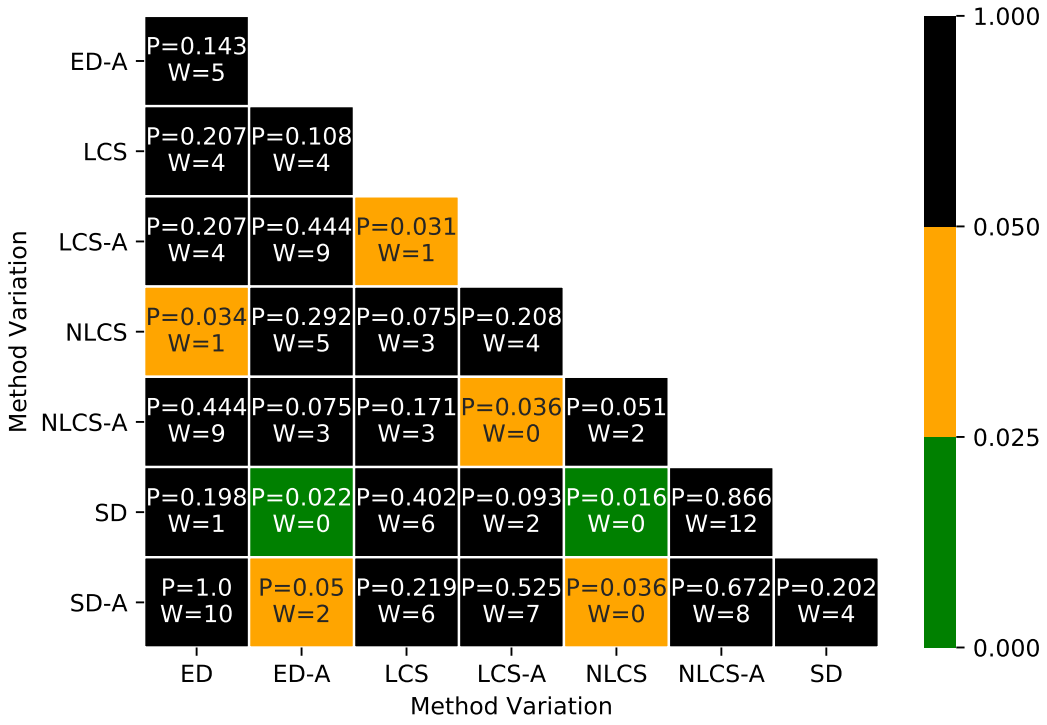


Figure 5.5: Rotation method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the mAP for an individual transformation obtained by two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix E.3 for further details.

Figure 5.6 shows the lower level rotation variation precision scores for each alphabet character query. Characters with the highest frequency of misclassification with an Average Precision score of less than 0.5 are identified as ‘l’, ‘o’, ‘r’, ‘u’ and ‘x’. While ‘l’, ‘o’ and ‘x’ have rotational symmetry of 180 degrees, ‘u’ is identical to ‘n’ when rotated by 180 degrees which the trained memory also contains. Furthermore, ‘r’ has a relatively low amount of shape information. Observations between the results of ED and ED-A show that the ED-A method achieves higher scores for the characters ‘i’ and ‘l’ which can be directly explained by the added positional angle constraint of ED-A.

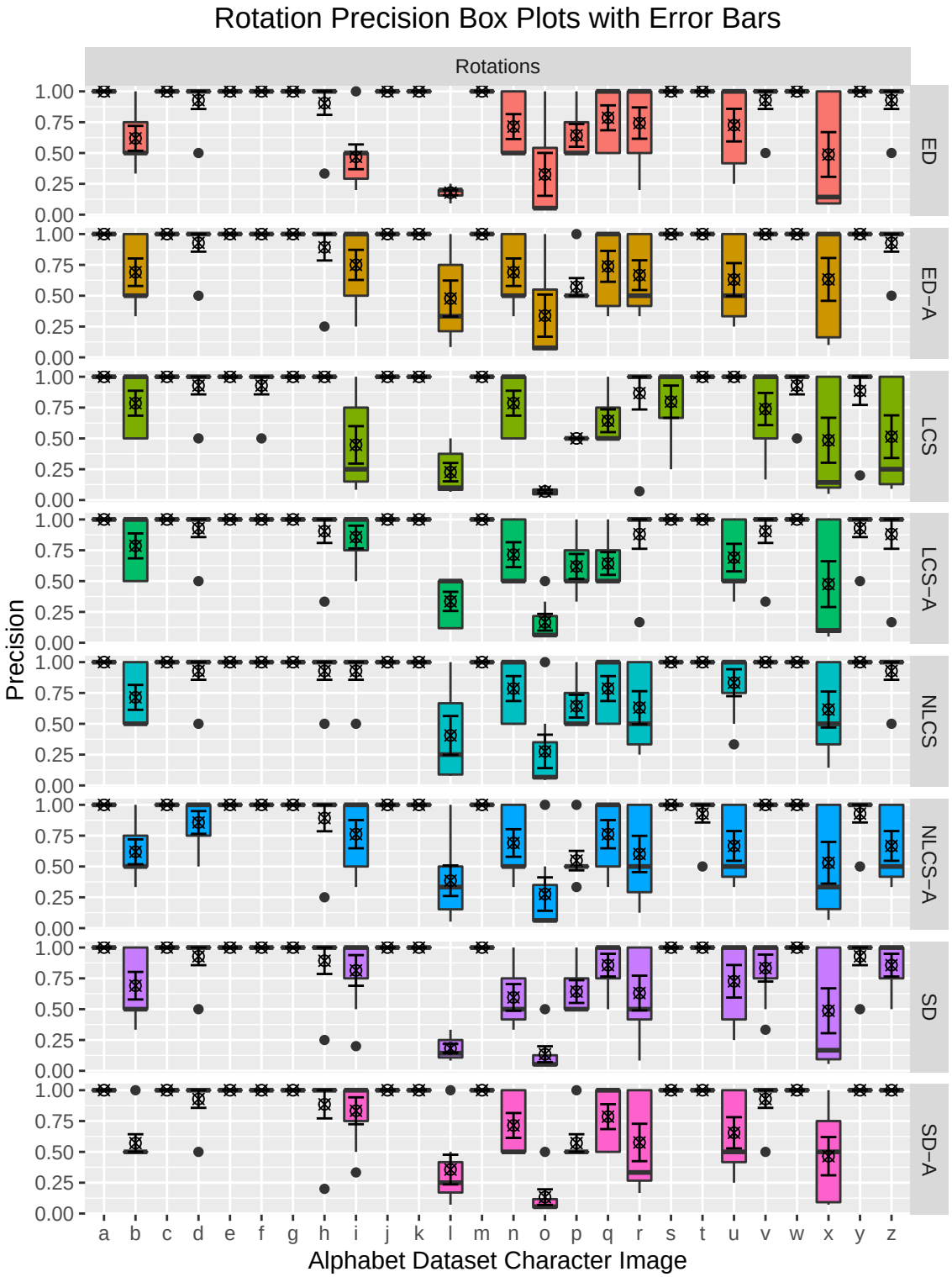


Figure 5.6: Rotation variation precision scores for each alphabet character query.

5.7 Scale Invariance

Uniform scaling of 0.75, 1.25 and 1.5 is evaluated within this section with respect to object recognition. Table 5.4 shows the high level mAP scores for each scale transformation / OSCC method. It is observed that overall, the NLCS method achieves the best score across the transformations. While the LCS-A method achieves the second best tied score with the ED-A method. Figure 5.7 shows that with the exception of the LCS method, the median difference of AP between all methods is not significantly greater than zero with a p-value threshold of 0.025. For the LCS method, it is observed that the median difference of AP between the LCS-A method (P=0.021, W=56) and the NLCS method (P=0.016, W=35) is significantly greater than zero with a p-value threshold of 0.025.

Scales	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
0.75	0.87 ±.28	0.82 ±.31	0.82 ±.28	**0.92** ±.24	**0.92** ±.23	0.81 ±.29	**0.92** ±.23	0.89 ±.26
1.25	0.85 ±.28	0.89 ±.27	**0.92** ±.23	0.89 ±.26	0.86 ±.27	0.86 ±.31	0.91 ±.23	0.85 ±.29
1.5	0.84 ±.31	0.80 ±.33	0.83 ±.32	**0.86** ±.30	**0.86** ±.30	0.79 ±.35	0.82 ±.31	0.83 ±.33
mAP ± SD	0.85 ±.01	0.84 ±.04	0.86 ±.04	**0.89** ±.02	0.88 ±.03	0.82 ±.03	0.88 ±.05	0.86 ±.03

Table 5.4: Overall scale transformation / OSCC variation mAP ± SD

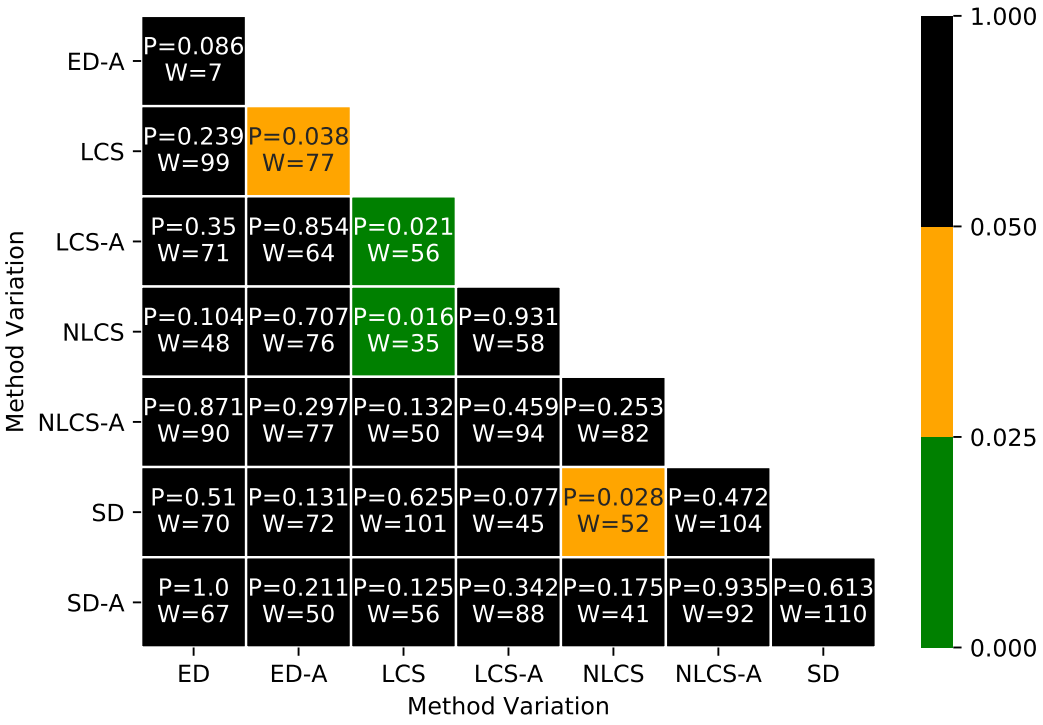


Figure 5.7: Scale method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix E.4 for further details.

Figure 5.8 shows the lower level scale variation precision scores for each alphabet character query. It is observed that the positional angle constraint increases the discriminability for the character 'l' across all variations. Common misclassified characters with scores of less than 0.5 are identified as 'i', 'l', 'o' and 'x'. These characters have previously been identified as characters that are commonly misclassified due to the relatively small amount of shape information that they contain.

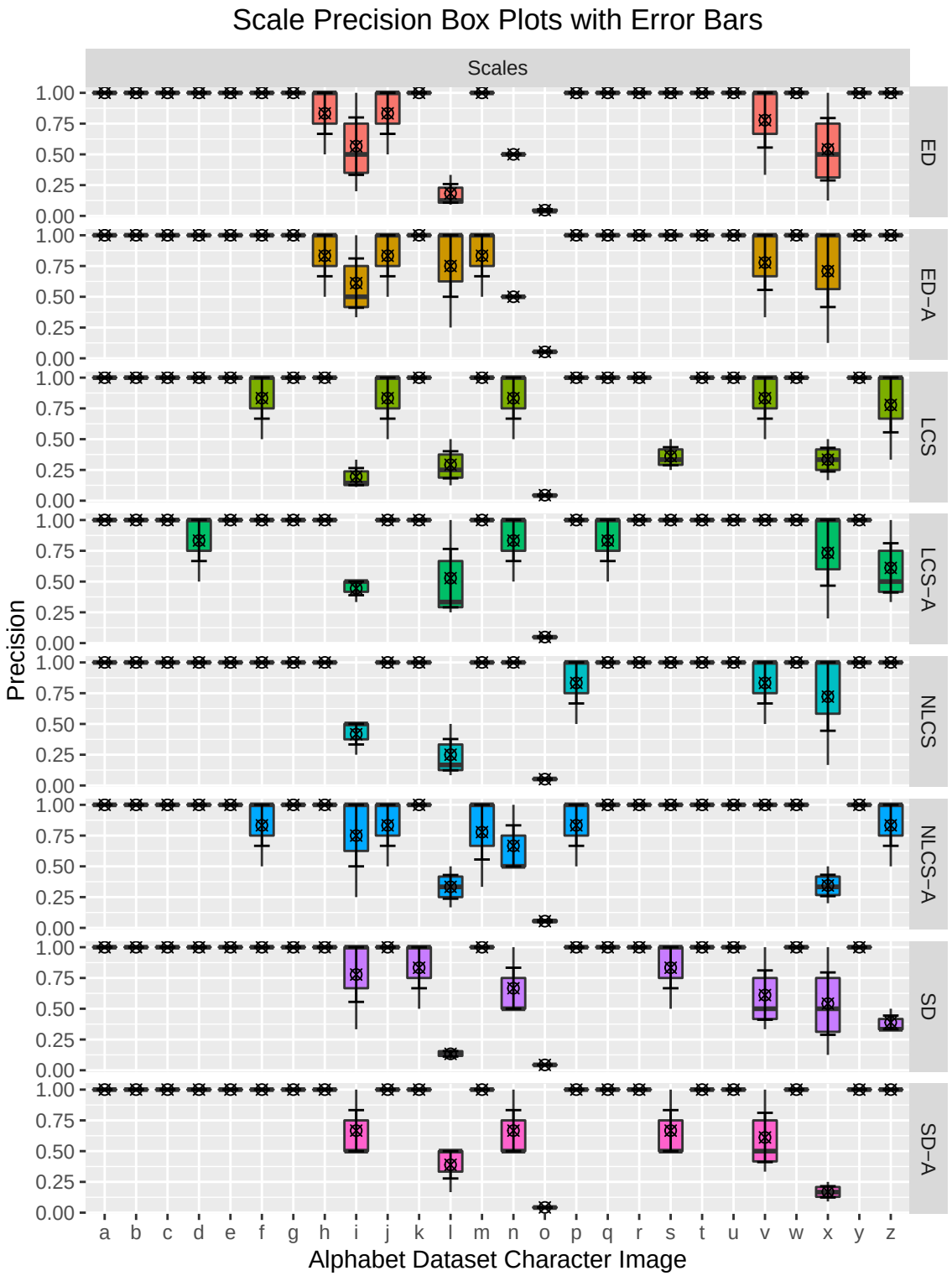


Figure 5.8: Scale variation precision scores for each alphabet character query.

5.8 Clutter Robustness

The same two variations of clutter for the alphabet dataset are used as within the point correspondence clutter evaluation section (see section 4.7). Table 5.5 shows the high level mAP scores for each clutter transformation / OSCC method. It is shown that the ED-A method obtains the highest score for the single clutter variation and the SD-A method obtains the best score for the multiple clutter variation. Furthermore, the SD and ED-A methods obtain the highest score across the transformations and the LCS method obtains the lowest. Figure 5.9 shows that the median difference of AP between the best performing methods (SD and ED-A) is not significantly greater than zero with a p-value threshold of 0.05. Furthermore, the median difference of AP between the lowest performing method (LCS) is significantly greater than zero when compared to all other methods with a p-value of 0.025.

Clutter	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
1	0.73 ±.38	0.80 ±.35	0.58 ±.40	0.79 ±.33	0.54 ±.41	0.43 ±.37	**0.81** ±.31	0.71 ±.38
2	**0.84** ±.31	0.80 ±.34	0.55 ±.40	0.68 ±.36	0.70 ±.39	0.39 ±.37	0.79 ±.31	0.79 ±.33
mAP ± SD	0.78 ±.05	**0.80** ±.00	0.57 ±.02	0.73 ±.05	0.62 ±.08	0.41 ±.02	**0.80** ±.01	0.75 ±.04

Table 5.5: Overall clutter transformation / OSCC variation mAP ± SD

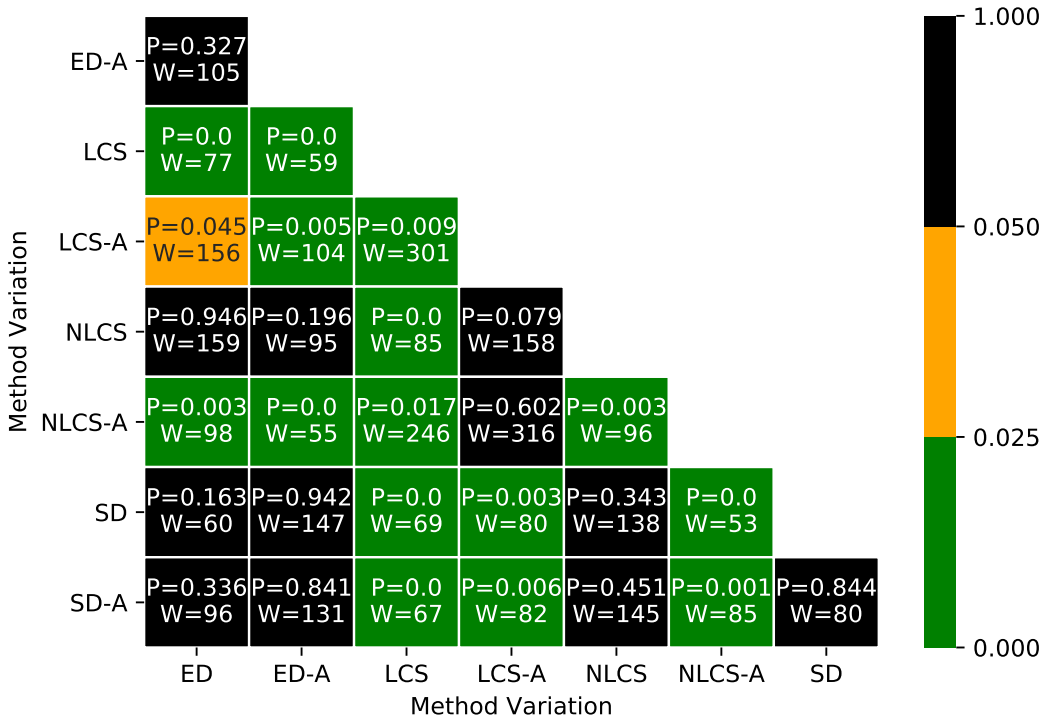


Figure 5.9: Clutter method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix E.5 for further details.

Figure 5.10 shows the lower level clutter variation precision scores for each alphabet character query. The SD method contains misclassifications for characters ‘b’, ‘h’, ‘i’, ‘j’, ‘l’, ‘o’ and ‘r’ with an average precision score of less than 0.5. Other than characters ‘b’ and ‘h’, the other characters have a relatively small amount of shape information. The SD-A improves discriminability for characters ‘j’, ‘t’ and ‘y’ while losing discriminability for characters ‘c’, ‘q’, and ‘s’ where ‘s’ is rotationally symmetric resulting in less information and ‘q’ is identical to ‘b’ when rotated.

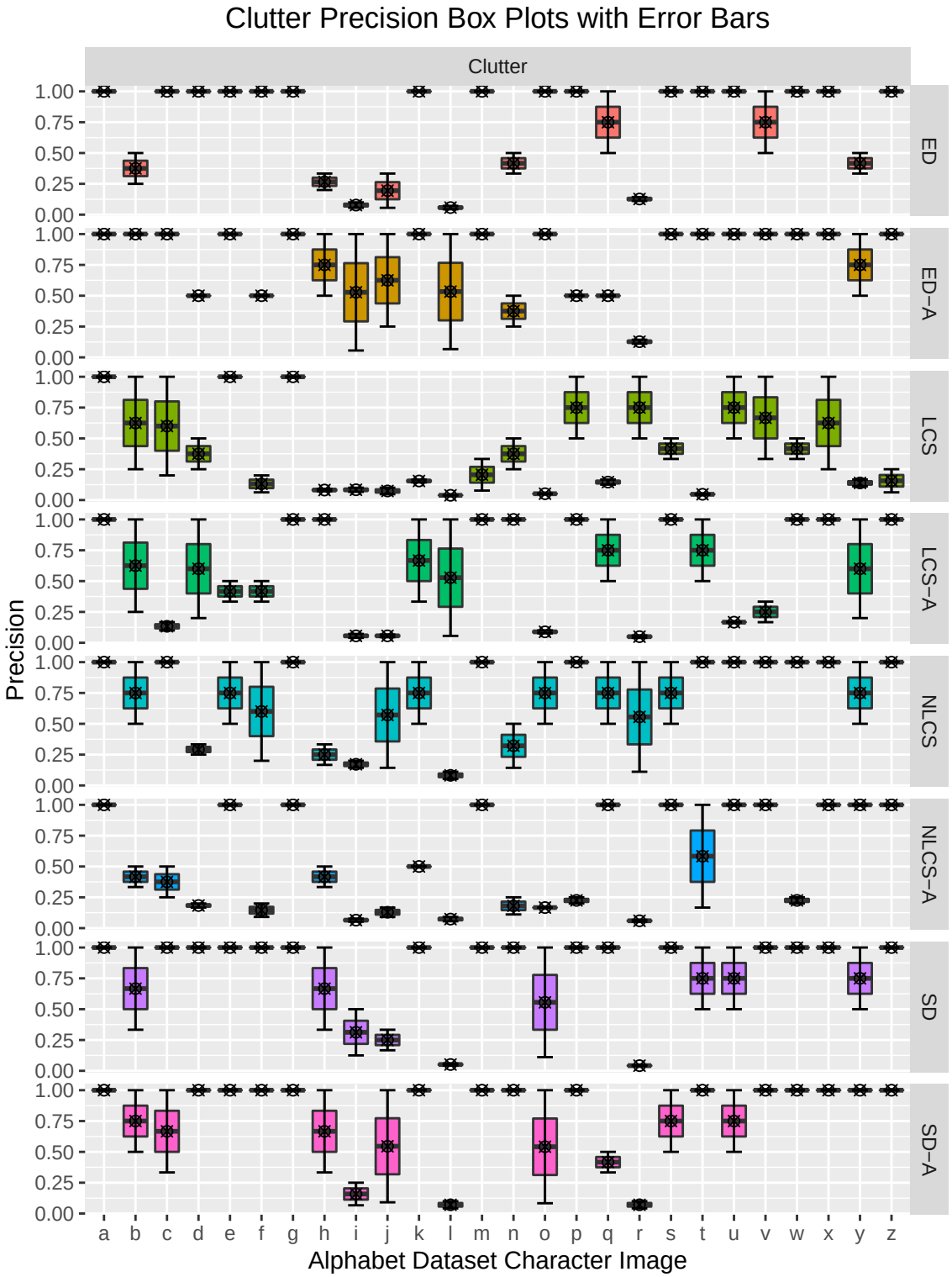


Figure 5.10: Scale variation precision scores for each alphabet character query.

5.9 Occlusion Robustness

This section uses the same occlusion dataset as used for the point correspondence occlusion evaluation within section 4.8. Table 5.6 shows the high level mAP scores for each occlusion transformation / OSCC method. Highest scoring methods to lowest scoring methods are ED, LCS-A, SD-A / NLCS-A, NLCS, SD and LCS. Figure 5.11 shows that with the exception of the LCS method, the median difference of AP between methods is not significantly greater than zero with a p-value threshold of 0.05.

Occlusion	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
mAP ± SD	0.94 ±.18	0.88 ±.25	0.94 ±.20	0.93 ±.21	0.95 ±.20	0.79 ±.35	0.91 ±.22	**0.96** ±.13

Table 5.6: Overall occlusion transformation / OSCC variation mAP ± SD

Figure 5.12 shows the lower level occlusion variation precision scores for each alphabet character query. The ED method is shown to misclassify both characters 'b' and 'q' with an Average Precision score of 0.5. Since 'b' is identical to 'q' when rotated by 180 degrees, these results are expected. It is interesting to note that the addition of the positional angle constraint reduces the Average Precision score with respect to 'q' while characters 'l' and 'r' are misclassified. The LCS method is shown to be the least discriminative as it fails to recall characters 'h', 'l' and 'o' with a high Average Precision score. The LCS-A method resolves this issue by recalling all characters with an Average Precision score of 1.0 with the exception of 'q' with a score of 0.5 and 'l' with a score under 12.5.

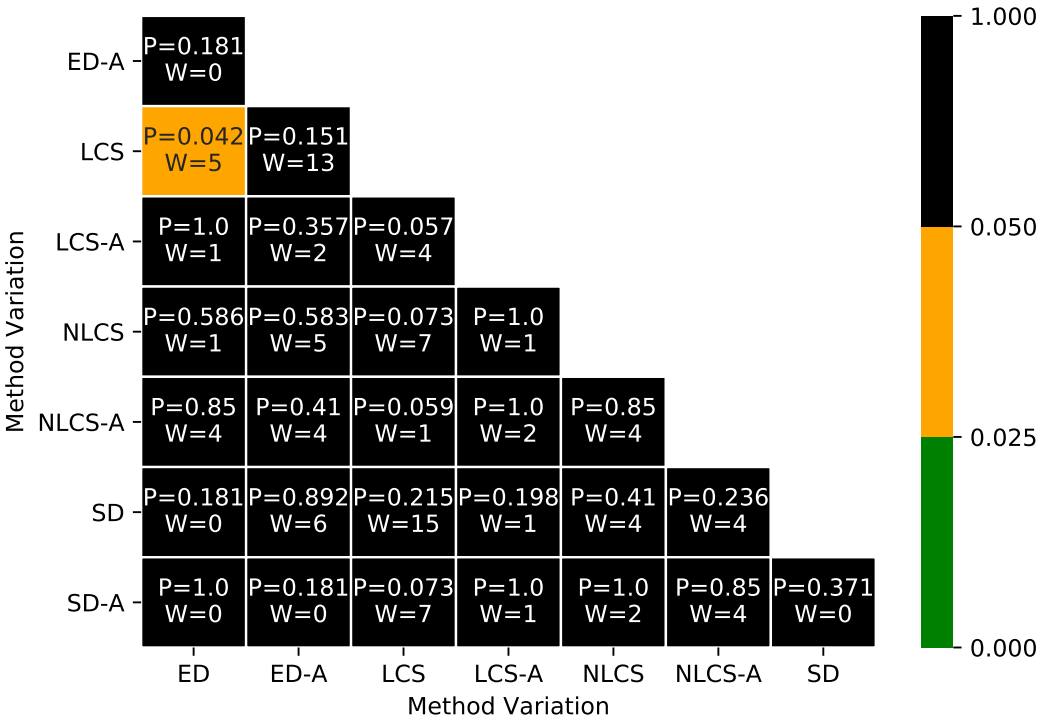


Figure 5.11: Occlusion method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. An entry within the table shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. See appendix E.6 for further details.

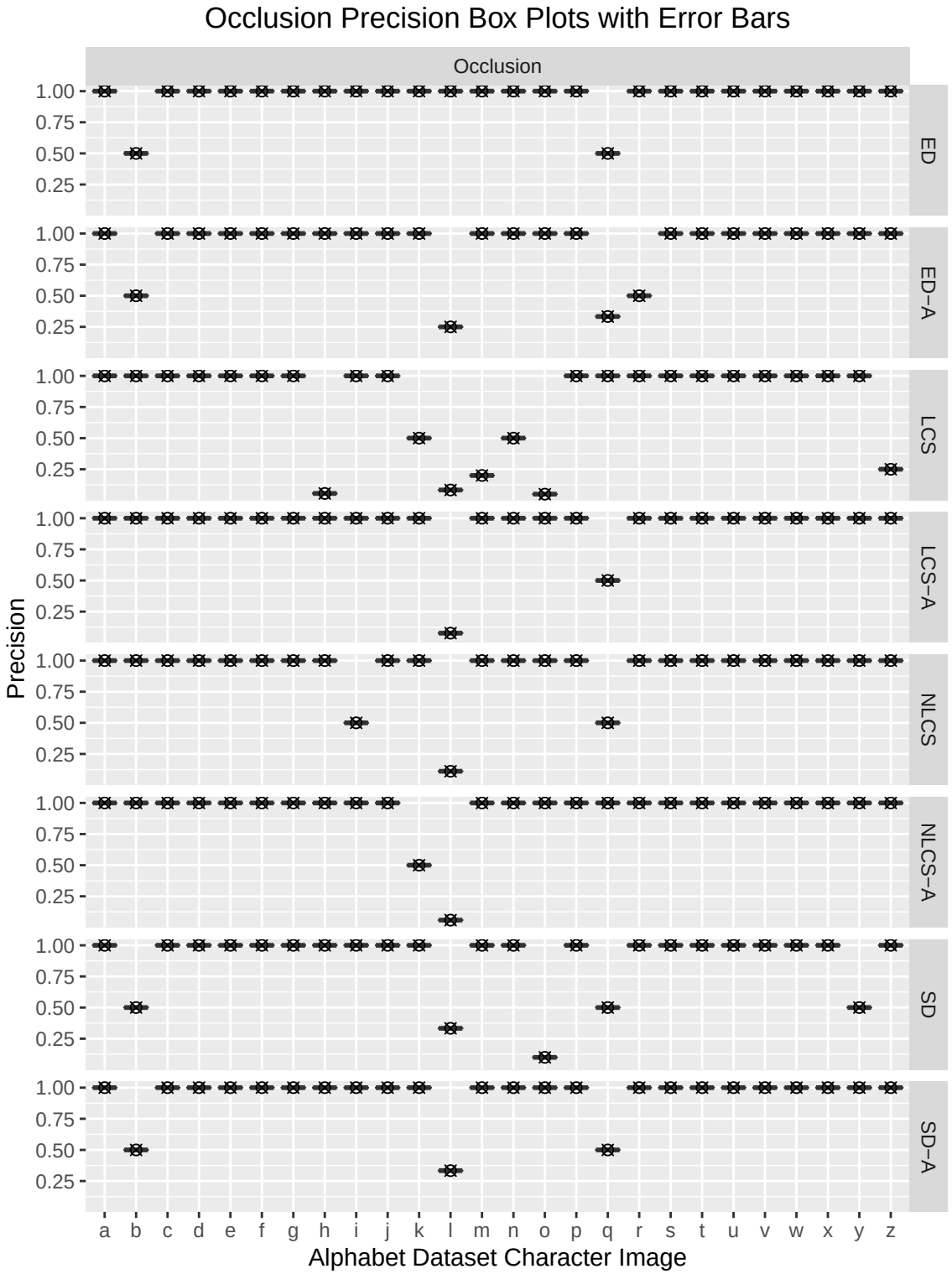


Figure 5.12: Occlusion variation precision scores for each alphabet character query.

5.10 Summary

This chapter built upon the previous point correspondence evaluation chapter by undertaking an initial evaluation of OSCC method variations with respect to the higher level task of object recognition. Firstly, a simple object classification metric was defined which is based on Hough voting of local point correspondences. Once the classifier was trained on a set of reference alphabet character images, transformations were applied to the images to obtain the mAP scores with respect to transformation / OSCC method variations.

Interpolation and translation dataset variation results showed that a majority of alphabet characters were classified without any false positives. Several misclassifications occurred for the character 'q' are expected as the character 'b' exists within the trained data which is rotationally symmetric. False positives occur for both characters 'l' and 'o'. These false positives can be explained by the fact that when rotated by 180 degrees, the characters appear to be identical. Therefore one point on the reference image will correspond to multiple points on the trained image. Consequently, the object level classification metric needs to take rotational symmetry into consideration which is proposed as future work in the last chapter.

Misclassifications with respect to the identified rotational symmetry property that the object level classifier does not yet handle can be seen throughout the evaluation results. Furthermore, misclassifications are also observed due to rotation invariance and the memory containing characters that are the same from the perspective of rotation invariance, for example, 'b' and 'q'. With the exception of these cases, these preliminary results show robustness to the interpolation, translation, rotation, scale, clutter and occlusion variations presented. Overall performance is shown by table 5.7. Figure 5.13 shows that the median difference of mAP between all methods is not significantly greater than zero with a p-value threshold of 0.025.

	SD-A	SD	NLCS-A	NLCS	LCS-A	LCS	ED-A	ED
interpolation	0.97	0.95	**0.99**	0.97	0.98	**0.99**	0.96	0.97
translation	0.95	0.93	**0.97**	0.96	0.96	**0.97**	0.96	0.95
rotations	0.82	0.82	0.81	**0.86**	0.84	0.79	0.84	0.82
scales	0.85	0.84	0.86	**0.89**	0.88	0.82	0.88	0.86
clutter	0.78	**0.80**	0.57	0.73	0.62	0.41	**0.80**	0.75
occlusion	0.94	0.88	0.94	0.93	0.95	0.79	0.91	**0.96**
mAP ± SD	**0.89** ±.07	0.87 ±.06	0.86 ±.14	**0.89** ±.08	0.87 ±.12	0.79 ±.19	**0.89** ±.06	0.88 ±.08

Table 5.7: Overall transformation / OSCC variation mAP ± SD

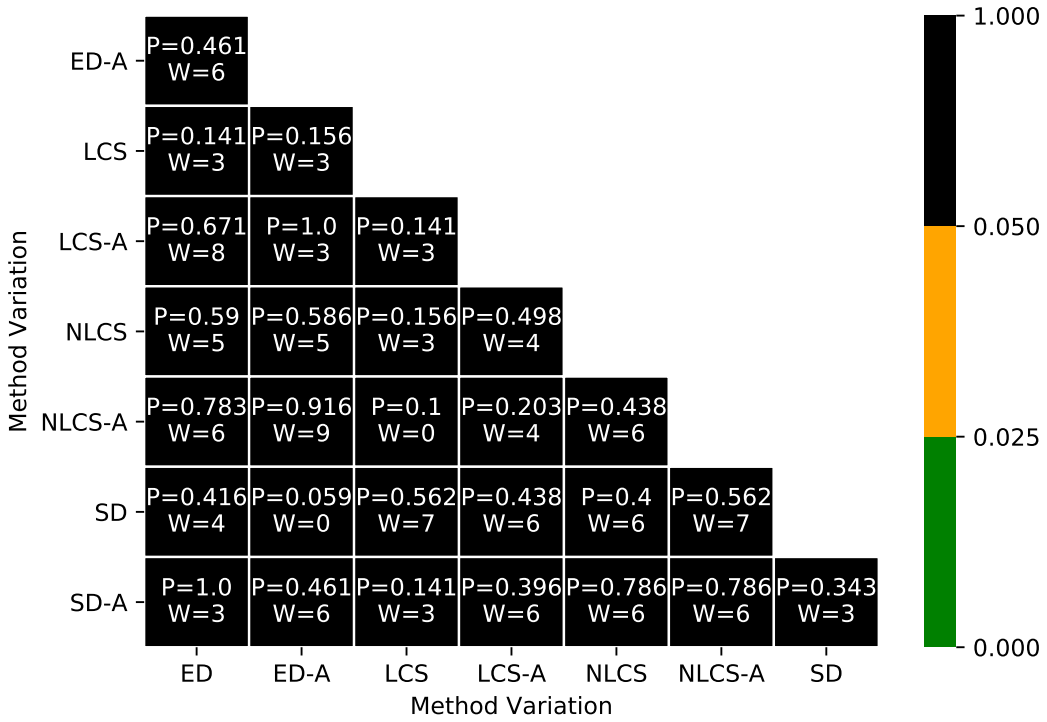


Figure 5.13: Overall transformation / method variation comparison using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test. An individual matched pair is treated as the high level mAP for a particular transformation by two different OSCC method variations from table 5.7. An entry within the table of the figure shows the pairwise two-sided Wilcoxon Matched-Pairs Signed-Rank Test p-value (P) and Wilcoxon test statistic (W) value for two OSCC method variations using matched pairs from all dataset variations. As six dataset variations are used, six matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test

Chapter 6

MNIST Handwritten Digits Evaluation

6.1 Introduction

In the previous chapter, object level recognition with OSCC was evaluated with respect to synthetic data. In this chapter, we compare OSCC against the Shape Context (SC) method using the MNIST handwritten digits dataset [2]. Firstly, OSCC is directly compared against the SC results obtained by Belongie et al. [66] using 20,000 train images and 10,000 test images. Secondly, for investigatory purposes, OSCC is directly compared against the SC implementation provided by Belongie et al. [67] using the SC distance component only and not the MNIST specific optimised weights for the SC object level recognition distance function. The next chapter highlights the conclusions of this thesis and identifies future work.

6.2 Experiment Design

OSCC is directly compared against the Shape Context (SC) method throughout this chapter which is a close competitor with respect to using point descriptors for shape recognition. Furthermore, the public MNIST dataset is used within this chapter as published SC results exist for the dataset which can be used for comparison. The overall experiment design mimics the approach used by Belongie et al. [66] for the SC method: a KNN classifier ($k = 3$) is used with an object level distance function. This chapter is broken down into two parts:

- Using 20,000 training instances with the complete test set, section 6.4 compares the OSCC error rate with the SC error rate reported by Belongie et al. [66].
- As an investigatory task using the SC MATLAB implementation provided by the authors [67], an evaluation is undertaken to compare the core SC distance component with the OSCC method. This is opposed to the SC MNIST specific KNN classifier with optimised distance function weights which also takes the appearance cost and bending energy into consideration.

6.3 Dataset

The MNIST handwritten digits dataset [2] comprises of 70,000 grayscale images which is often used as a benchmark for machine learning algorithms. The dataset was collated from the NIST handwritten digit datasets which contain examples from a range of different writers. Each image is 28 x 28 pixels with a single class label. The dataset has a predefined train and test split with 60,000 and 10,000 instances respectively. Furthermore, the dataset is imbalanced as shown by table 6.1.

6.3.1 OSCC Parameters

The SD-A OSCC variation is used throughout this chapter along with the OSCC specific object level recognition metric defined in section 5.2. The SD-A OSCC variation is defined as the exact configuration shown by tables 6.2, 6.3 and 6.4 (see section 3.7 for a definition of the parameters). Furthermore, parameters for the OSCC specific object level recognition metric are defined by table 6.5 (see section 5.2 for a definition of the parameters). Like the Shape Context method, 100 edge points were uniformly sampled for each image from which 100 point descriptors were constructed.

Class	Train Instances	Test Instances
0	5,923	980
1	6,742	1,135
2	5,958	1,032
3	6,131	1,010
4	5,842	982
5	5,421	892
6	5,918	958
7	6,265	1,028
8	5,851	974
9	5,949	1,009
Total	60,000	10,000

Table 6.1: MNIST handwritten digits dataset distribution

Parameter	Value
Ring Spacing	Uniform spacing of three pixels
Sample Point Region	Nearest-neighbor
Point sample area	Global

Table 6.2: OSCC SD-A MNIST sampling parameters

Parameter	Value
Binning	Exact binning using 8 bins for all relative features
RESO features	True (signed)
PA features	True (signed)
Ring Correlation Method	Dynamic Time Warping
Rotation Invariant	False

Table 6.3: OSCC SD-A MNIST ring correlation parameters

Parameter	Value
Global Search Space Constraints	Sakoe-Chiba band is used with horizontal and vertical dimensions of 0.5 with respect to the horizontal and vertical ring sequence lengths
Step Size	Step size of 1
Step Weights	All step weights set to 1

Table 6.4: OSCC SD-A MNIST ring sequence correlation parameters

Parameter	Value
$aBins$	8
r	13
$rBins$	5

Table 6.5: OSCC MNIST object level metric parameters

6.4 Published Shape Context Results Comparison

This section first compares the overall results obtained by the OSCC method with the published SC results using 20,000 train images with the whole test set. Results are then further inspected by removing class 0 which causes higher error rates for the OSCC method.

6.4.1 Overall Results

Belongie et al. [66] report that the SC method obtained an error rate of 0.63% with 20,000 training examples using a KNN classifier ($k = 3$). As the test set contains 10,000 images, the classifier made 63 errors - the 63 errors are identified by the authors using (predicted, actual) pairs for each error. For a direct comparison with OSCC, 20,000 images are trained (the first 2000 train images for each digit are used) and the full test set is queried using a KNN classifier ($k = 3$). Overall results are shown by table 6.6

The 63 errors identified by Belongie et al. [66] are extracted and used to populate table 6.6 for the SC method. The table shows that the SC method obtains a better

score for all metrics across all characters with the exception of the OSCC method obtaining a lower error rate for class 0.

	OSCC				Shape Context			
Class	Precision	Recall	F1 Score	Error	Precision	Recall	F1 Score	Error
0	0.733	0.997	0.845	**0.003**	**0.994**	**0.998**	**0.996**	0.006
1	0.982	0.897	0.937	0.103	**0.998**	**0.996**	**0.997**	**0.002**
2	0.989	0.952	0.970	0.048	**0.994**	**0.995**	**0.995**	**0.006**
3	0.976	0.989	0.982	0.011	**0.994**	**0.998**	**0.996**	**0.006**
4	0.980	0.931	0.955	0.069	**0.994**	**0.994**	**0.994**	**0.006**
5	0.995	0.911	0.951	0.089	**0.998**	**0.990**	**0.994**	**0.002**
6	0.985	0.953	0.969	0.047	**0.992**	**0.998**	**0.995**	**0.008**
7	0.979	0.939	0.958	0.061	**0.982**	**0.996**	**0.989**	**0.018**
8	0.975	0.890	0.931	0.110	**1.000**	**0.994**	**0.997**	**0.000**
9	0.921	0.968	0.944	0.032	**0.992**	**0.984**	**0.988**	**0.008**
Median	0.979	0.945	0.953	0.055	**0.994**	**0.995**	**0.995**	**0.006**
Mean	0.951	0.943	0.944	0.057	**0.994**	**0.994**	**0.994**	**0.006**
SD	0.075	0.035	0.036	0.035	**0.005**	**0.004**	**0.003**	**0.005**

Table 6.6: Precision, recall, f1 and error rate metrics for each class across both the OSCC and SC methods using 20,000 train images. The best score for each metric across the methods is highlighted with respect to an individual class.

Table 6.7 shows the error rate obtained by the OSCC and SC methods for each class. The p-value is obtained from the two-sided Wilcoxon Matched-Pairs Signed-Rank Test where an individual pair is the error rate obtained by both classifiers for an individual class. As a result, the median difference of error rate between the methods is significantly greater than zero (P=0.004, W=1) with a p-value threshold of 0.05. Therefore it is shown that with 20,000 train images, the SC method outperforms the OSCC method.

Overall, the OSCC method makes the following number of errors for each class from 0 - 9 respectively `(3, 117, 50, 11, 68, 79, 45, 63, 107, 32)`. As a result, OSCC obtains an error rate of 5.75% with respect to the full 10,000 test image set, whereas the reported error rate for the SC method is 0.63% [66] - the SC method makes the following number of errors for each class from 0 - 9 respectively `(6, 2, 6, 6, 6, 2, 8, 19, 0, 8)`.

Class	OSCC Error Rate	Shape Context Error Rate
0	**0.003**	0.006
1	0.103	**0.002**
2	0.048	**0.006**
3	0.011	**0.006**
4	0.069	**0.006**
5	0.089	**0.002**
6	0.047	**0.008**
7	0.061	**0.018**
8	0.110	**0.000**
9	0.032	**0.008**
Median	0.055	**0.006**
Mean	0.057	**0.006**
SD	0.035	**0.005**
p-value	0.004	

Table 6.7: MNIST error rate comparison between the OSCC and SC methods with 20,000 train images. (P=0.004, W=1) is obtained by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test where an individual pair is the error rate obtained by both classifiers for an individual class

Figures 6.1 and 6.2 show a confusion matrix with errors only for the OSCC and SC methods respectively. Figure 6.3 shows the differences between the confusion matrices by subtracting the OSCC confusion matrix from the SC confusion matrix. A positive value shows a lower error rate obtained by the OSCC method whereas a negative value shows a lower error rate obtained by the SC method.

For the OSCC method, it is shown that the highest error rate for each class from 1 - 9 is caused by classification errors with class 0 - with the exception of 7 being classified as 9 as shown by figure 6.1. Furthermore, the highest error rate for the SC method is caused by misclassifying 7 as 9 and the second highest error rate is caused by misclassifying 9 with 4 as shown by figure 6.3. It is interesting to note that while ignoring class 0 for the OSCC method, misclassifying 7 as 9 and 4 as 9 are amongst the top relative error rates for each method. This can be explained by the similar shape features that they share.

The largest misclassification differences between the two methods is dominated by class 0 as shown by figure 6.3. It is possible that the misclassification dominance of class 0 with respect to the OSCC method is due to the amount of common edge feature combinations that the digit contains due to its shape. As a result, this could cause the underlying sequence alignment correlation mechanisms of the OSCC method

to be less discriminative - especially with the inherent variability of handwritten digits. The next section inspects the results further by excluding class 0 as it dominates misclassification errors for the OSCC method.

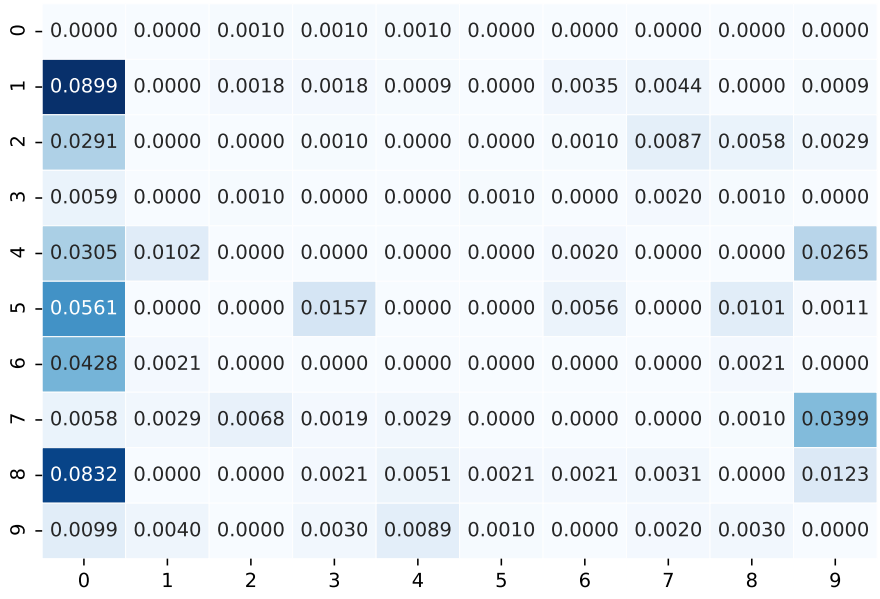


Figure 6.1: OSCC MNIST confusion matrix showing errors only (row-wise normalisation is applied before setting the diagonal values to 0) for 20,000 train images and the full test set.

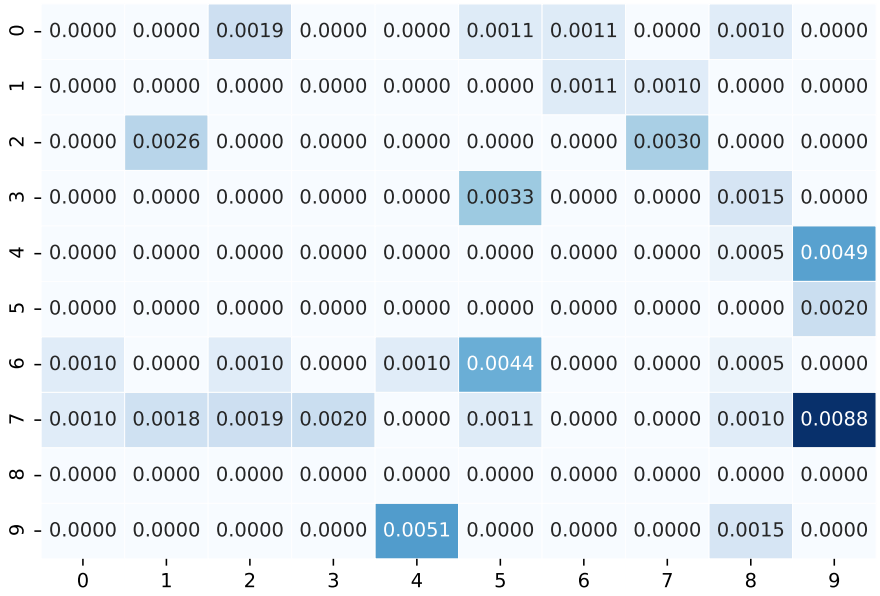


Figure 6.2: SC MNIST confusion matrix showing errors only (row-wise normalisation is applied before setting the diagonal values to 0) for 20,000 train images and the full test set.

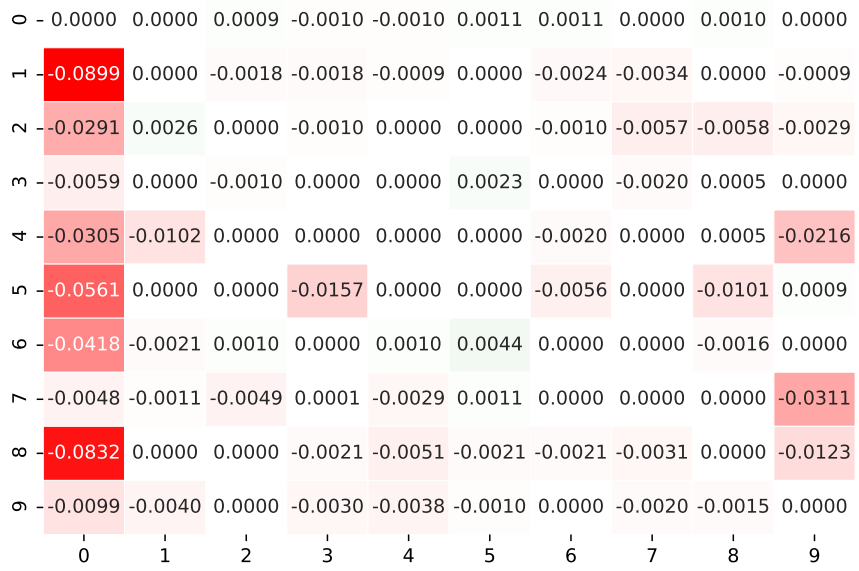


Figure 6.3: Differences between the SC and OSCC confusion matrix showing errors only (row-wise normalisation is applied before setting the diagonal values to 0). The OSCC confusion matrix (see figure 6.1) subtracted from the SC confusion matrix 6.2.

6.4.2 Results Inspection

As identified in the previous section, class 0 causes a relatively high error rate for the OSCC method. This section inspects the results further by removing class 0 from both the train and query dataset to see how the results are affected. Consequently, table 6.8 shows the high level results for both the OSCC and SC methods. With class 0 removed, the overall mean error rate is shown to be 0.025 for OSCC which is under half of the mean error rate obtained when class 0 is present (see table 6.6). Although the SC method still obtains the best scores for a majority of cases, it is shown that the OSCC method achieves an equal error rate for class 3 while obtaining a better error rate for class 6.

	OSCC				Shape Context			
Class	Precision	Recall	F1 Score	Error	Precision	Recall	F1 Score	Error
1	0.980	0.980	0.980	0.020	**0.998**	**0.996**	**0.997**	**0.002**
2	0.989	0.977	0.983	0.023	**0.994**	**0.997**	**0.996**	**0.006**
3	0.976	0.994	0.985	**0.006**	**0.994**	**0.998**	**0.996**	**0.006**
4	0.981	0.956	0.969	0.044	**0.994**	**0.994**	**0.994**	**0.006**
5	0.993	0.964	0.978	0.036	**0.998**	**0.991**	**0.994**	**0.002**
6	0.973	0.995	0.984	**0.005**	**0.993**	**0.999**	**0.996**	0.007
7	0.975	0.941	0.957	0.059	**0.982**	**0.996**	**0.989**	**0.018**
8	0.974	0.972	0.973	0.028	**1.000**	**0.995**	**0.997**	**0.000**
9	0.919	0.975	0.946	0.025	**0.992**	**0.984**	**0.988**	**0.008**
Median	0.976	0.975	0.978	0.025	**0.994**	**0.996**	**0.996**	**0.006**
Mean	0.973	0.973	0.973	0.027	**0.994**	**0.994**	**0.994**	**0.006**
SD	0.020	0.016	0.013	0.016	**0.005**	**0.004**	**0.003**	**0.005**

Table 6.8: Precision, recall, f1 and error rate metrics for each class across both the OSCC and SC methods excluding class 0. The best score for each metric across the methods is highlighted with respect to an individual class.

The error rates obtained by OSCC and SC with class 0 removed from both the test and train sets is shown by table 6.9. The p-value is obtained from the two-sided Wilcoxon Matched-Pairs Signed-Rank Test where an individual pair is the error rate obtained by both classifiers for an individual class. The median difference of error rate between the methods is significantly greater than zero (P=0.021, W=1) with a p-value threshold of 0.05. Therefore it is shown that with class 0 removed, the SC method outperforms the OSCC method. Overall, the OSCC method makes the following number of errors for each class from 1 - 9 respectively `(23, 24, 6, 43, 32, 5, 61, 27, 25)`. As a result, OSCC obtains an overall error rate of 2.73%,

whereas the SC method obtains an overall error rate of 0.61% with the following number of errors for each class from 1 - 9 respectively `(2, 6, 6, 6, 2, 7, 18, 0, 8)`.

Class	OSCC Error Rate	Shape Context Error Rate
1	0.020	**0.002**
2	0.023	**0.006**
3	**0.006**	**0.006**
4	0.044	**0.006**
5	0.036	**0.002**
6	**0.005**	0.007
7	0.059	**0.018**
8	0.028	**0.000**
9	0.025	**0.008**
Median	0.025	**0.006**
Mean	0.027	**0.006**
SD	0.016	**0.005**
p-value	0.021	

Table 6.9: Error Rate comparison between the OSCC and SC methods excluding class 0. (P=0.021, W=1) is obtained by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test where an individual pair is the error rate obtained by both classifiers for an individual class

Results are broken down further with figure 6.4 which shows the differences between confusion matrices by subtracting the OSCC confusion matrix from the SC confusion matrix. A positive value shows a lower error rate obtained by the OSCC method whereas a negative value shows a lower error rate obtained by the SC method - errors are shown only. The top two largest error rate differences where SC obtains a lower error rate than OSCC is for misclassifications of 7 as 9 and 4 as 9. Furthermore, the top two largest error rate differences where OSCC obtains a lower error rate than SC is for the misclassifications of 2 as 1 and 6 as 5. These results show that after class 0, class 9 dominates misclassification errors for the OSCC method when compared to the SC method. It is possible that this is caused due to the variability of the handwritten digit 9 which could introduce a large number of edge feature variation combinations that are encoded and correlated by the underlying OSCC sequence alignment approach, therefore reducing the discriminability.

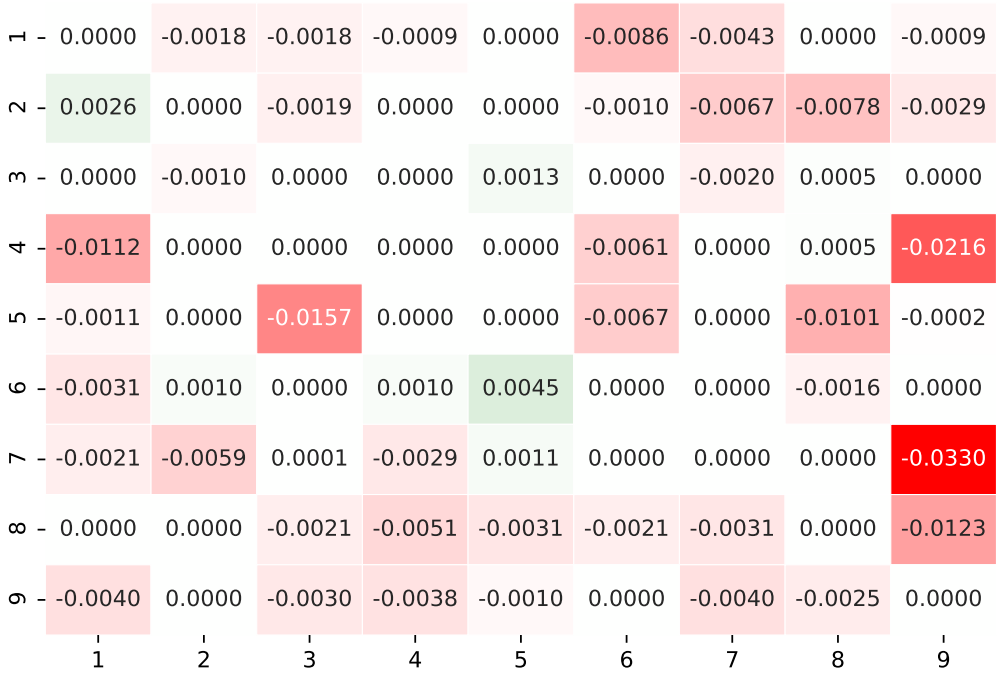


Figure 6.4: Differences between the SC and OSCC row-normalised confusion matrices by subtracting the OSCC confusion matrix from the SC confusion matrix excluding class 0 and showing errors only (by setting the diagonal values to 0).

6.5 SC Distance Component Isolation

With the goal of comparing OSCC with a core SC component, this section compares the core SC distance component D_{sc} [66] with the OSCC method using the SC code provided by the authors [67]. As a result, this section concentrates on comparing pure shape descriptor methods without the additional factors introduced by the SC method which includes appearance and bending energy costs. More specifically, for the SC method, the SC distance component is used only as the distance function for the KNN classifier. This variation is referred to as SC-D for the rest of this chapter. This is opposed to the SC MNIST specific KNN classifier with optimised distance function weights which also takes the appearance cost D_{ac} and bending energy D_{be} into consideration [66] (see appendix F for further details). Furthermore, the OSCC method and the OSCC specific object level metric configurations remain exactly the same as the previous section as defined by section 6.3.1.

The object level metrics are compared with a KNN classifier ($k = 3$) to mimic the setup defined in the previous section. The first 500 training instances for each class

from the MNIST handwritten digits dataset train split were used as the training set and all 10,000 testing images were used from the test set. Therefore a classifier was trained with 5,000 instances in total and then the 10,000 test images were used to query the classifier. As a result, with 5,000 train images and 10,000 test images, an individual classifier made 50,000,000 image comparisons. A limit of 5,000 training images was used due to the amount of time required to obtain the results for both the SC-D and OSCC object level distance functions.

Overall results are presented by table 6.10 which shows the precision, recall, f1 score and error rate metrics for each class across both the OSCC and SC-D methods. The best score for each metric between the methods for an individual class is highlighted. It is shown that the OSCC method obtains a lower error rate than the SC-D method for all classes with the exception of classes 0, 5 and 7. Furthermore, the SC-D method obtains a higher precision score for classes 1 and 9 when compared to OSCC.

	OSCC				SC-D			
Class	Precision	Recall	F1 Score	Error	Precision	Recall	F1 Score	Error
0	**0.933**	0.994	**0.962**	0.006	0.931	**0.995**	0.962	**0.005**
1	0.974	**0.967**	**0.970**	**0.033**	**0.980**	0.891	0.933	0.109
2	**0.982**	**0.951**	**0.966**	**0.049**	0.932	0.942	0.937	0.058
3	**0.967**	**0.984**	**0.975**	**0.016**	0.952	0.927	0.939	0.073
4	**0.978**	**0.943**	**0.960**	**0.057**	0.954	0.932	0.943	0.068
5	**0.992**	0.936	**0.963**	0.064	0.933	**0.946**	0.939	**0.054**
6	**0.963**	**0.986**	**0.975**	**0.014**	0.941	0.978	0.959	0.022
7	**0.958**	0.909	**0.933**	0.091	0.937	**0.916**	0.927	**0.084**
8	**0.953**	**0.930**	**0.941**	**0.070**	0.906	0.915	0.911	0.085
9	0.889	**0.975**	**0.930**	**0.025**	**0.896**	0.930	0.912	0.070
Median	**0.965**	**0.959**	**0.963**	**0.041**	0.935	0.931	0.938	0.069
Mean	**0.959**	**0.957**	**0.958**	**0.043**	0.936	0.937	0.936	0.063
SD	0.028	**0.027**	**0.016**	**0.027**	**0.022**	0.029	0.016	0.029

Table 6.10: Precision, recall, f1 and error rate metrics for each class across both the OSCC and SC-D methods using 5,000 train images. The best score for each metric across the methods is highlighted with respect to an individual class.

Table 6.11 shows the error rate score obtained for each class from each classifier (extracted from table 6.10 for the context of statistical significance). The p-value is obtained from the two-sided Wilcoxon Matched-Pairs Signed-Rank Test where an individual pair is the error rate obtained by both classifiers for an individual class. As a result, the median difference of error rate between the classifiers is significantly

greater than zero (P=0.049, W=8) with a p-value threshold of 0.05. Therefore it is shown that overall, the OSCC method outperforms the SC-D method (using the SC-D distance component D_{sc} only without the MNIST optimised KNN distance function weights) with a train set size of 5000.

Class	OSCC Error Rate	SC-D Error Rate
0	0.006	**0.005**
1	**0.033**	0.109
2	**0.049**	0.058
3	**0.016**	0.073
4	**0.057**	0.068
5	0.064	**0.054**
6	**0.014**	0.022
7	0.091	**0.084**
8	**0.070**	0.085
9	**0.025**	0.070
Median	**0.041**	0.069
Mean	**0.043**	0.063
SD	**0.027**	0.029
p-value	0.049	

Table 6.11: Error Rate comparison between the OSCC and SC-D methods with 5,000 train images. (P=0.049, W=8) is obtained by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test where an individual pair is the error rate obtained by both classifiers for an individual class

Figures 6.5 and 6.6 show confusion matrices with errors only for the OSCC and SC-D methods respectively. Furthermore, figure 6.7 visualises the differences between the confusion matrices by subtracting the OSCC confusion matrix from the SC-D confusion matrix. A positive value shows a lower error rate obtained by the OSCC method whereas a negative value shows a lower error rate obtained by the SC-D method.

Misclassifications with the highest error rate for the OSCC method are identified as 7 being misclassified as 9 and 4 being misclassified as 9 (these are consistent with section 6.4.2 when ignoring class 0 with 20,000 train instances). It is interesting to note that class 0 does not dominate error rates as observed for the 20,000 train image evaluation (see section 6.4). It is possible that this observation can be explained due to the larger train set containing a larger number of instances for class 0 which are visually similar to other handwritten digits. More specifically, as identified for the

20,000 train image evaluation, it is possible that the misclassification dominance for class 0 is due to the amount of common feature combinations that instances of the digit contain. These common feature combinations can be explained by both the shape and handwritten variations of the digit. The detrimental effect of a large number of edge feature variation combinations being encoded and correlated by the underlying OSCC sequence alignment approach can lead to decreased discriminability. In addition to parameter optimisation, this clearly shows that larger test sets are required as future work for this evaluation which is identified as a constraint at the start of this section.

The top two highest error rates obtained by the SC-D method are class 4 being misclassified as class 9 and class 7 being misclassified as class 2. Whereas the two highest error rates obtained by the SD method for 20,000 train instances are for class 7 being misclassified as class 9 and class 4 being misclassified as class 9. Although there is a difference between both dataset size and method variation, a consistency exists for class 4 being misclassified as class 9. These misclassifications can be explained by the visual similarity between the digits which share a similar shape and topology.

The largest two error rate differences where OSCC obtains a lower error rate are for class 1 being misclassified as class 8 and class 7 being misclassified as 2 (see figure 6.7). Whereas the largest two error rate differences where SC-D obtains a lower error rate are for class 7 being misclassified as class 9 and class 2 being misclassified as class 8. Focusing on the higher error rate of the two, misclassifying 7 as 9 has previously been discussed with respect to the OSCC method such that the variability of the handwritten digit could introduce a large number of edge feature combinations with a negative impact on the underlying sequence alignment correlation approaches.

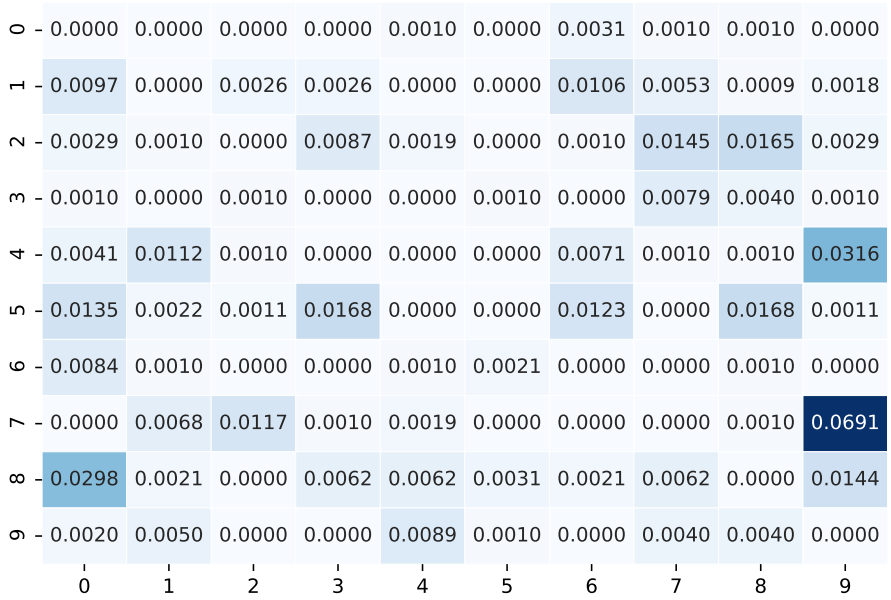


Figure 6.5: OSCC MNIST confusion matrix showing errors only (row-wise normalisation is applied before setting the diagonal values to 0) for 5,000 train images and the full test set.

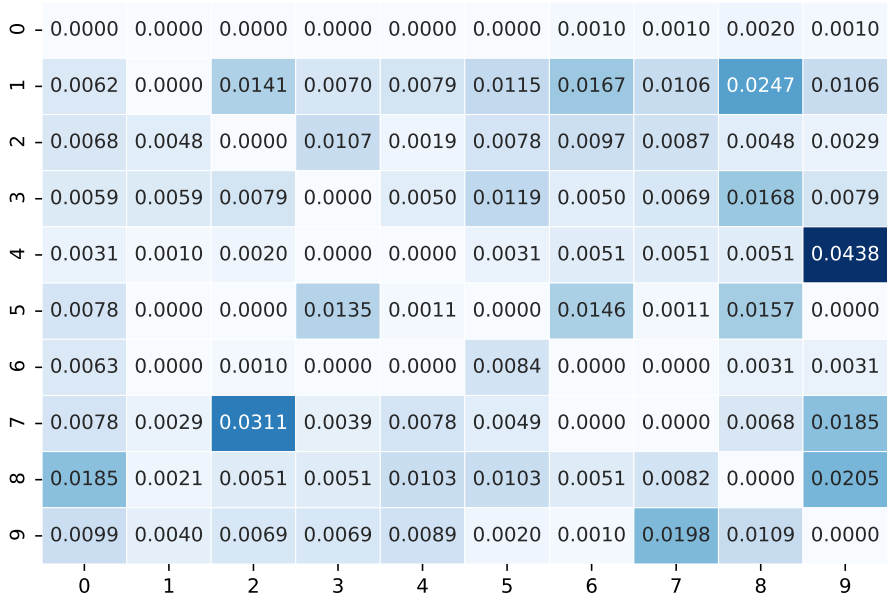


Figure 6.6: SC-D MNIST confusion matrix showing errors only (row-wise normalisation is applied before setting the diagonal values to 0) for 5,000 train images and the full test set.

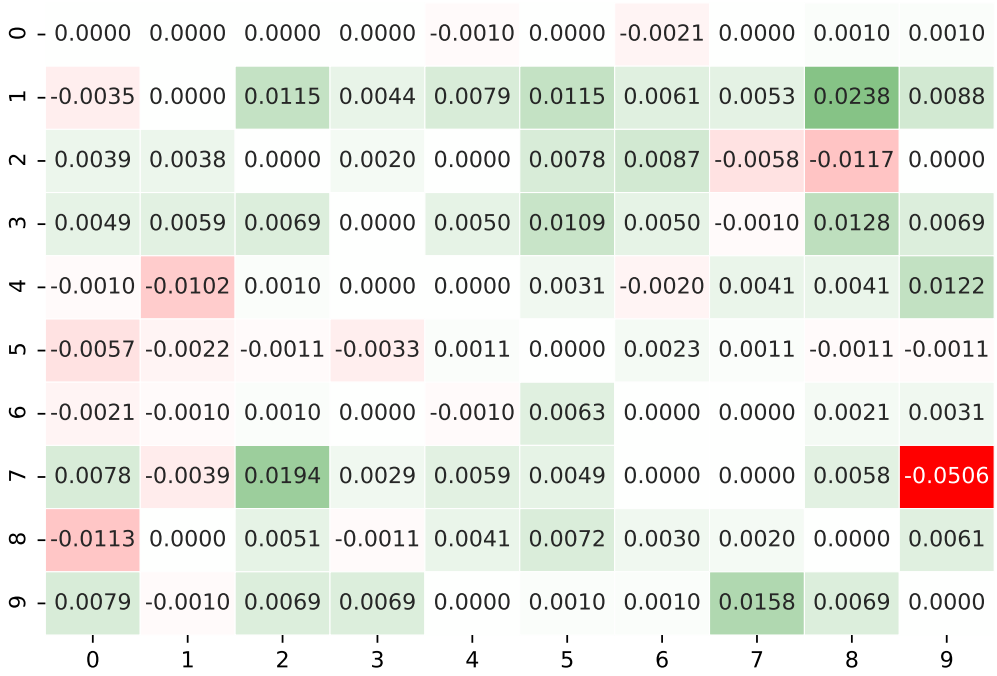


Figure 6.7: OSCC and SC-D confusion matrix differences. Plot of the OSCC confusion matrix 6.5 subtracted from the SC-D confusion matrix (see figure 6.6) showing errors only (by setting the diagonal values to 0)

6.6 Summary

OSCC was directly compared against the SC method with two different approaches:

- As the main approach, OSCC was compared with the results obtained by Belongie et al. [66] using 20,000 train instances with the complete test set. Overall it was shown that the SC method obtained a lower error rate of 0.63% whereas OSCC obtained an error rate of 5.75%. Results were further inspected by removing class 0 which caused a relatively large number of misclassifications for OSCC, as a result the SC method obtained an error rate of 0.61% and OSCC obtained an error rate of 2.73%. For both of these cases it was shown that the SC method outperforms OSCC using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

- A comparison between OSCC and the SC-D approach was undertaken using SC code provided by the authors [67]. The SC-D approach is referred to as only using the SC distance - rather than using the MNIST optimised KNN distance function weights along with appearance cost and bending energy components.

Using a train set size of 5000 (500 instances for each class) and the full test set, it was shown that the OSCC method outperforms the SC-D method using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

Overall, the SC method outperforms the OSCC method with respect to the MNIST dataset as shown by the evaluation with the results obtained by Belongie et al. [66]. Furthermore, class 0 and 9 are shown to dominate classification errors for the OSCC approach. It is believed that the variability of handwritten digits introduce a large number of common feature combinations which could cause the underlying sequence alignment correlation mechanisms of the OSCC method to be less discriminative. As a result, OSCC parameter optimisation should be considered for the specific task of handwritten digit recognition - this chapter has shown that it is worthy to pursue this task with the relatively low error rate that OSCC obtains. Furthermore, due to the relatively low error rate obtained by OSCC without parameter optimisation, it is concluded that this chapter shows that it is worth undertaking evaluations using larger MNIST train dataset sizes.

Chapter 7

Conclusions and Future Work

7.1 Introduction

This chapter concludes the work undertaken within this thesis and proposes future work. A summary of the thesis follows within this section which links all of the thesis aims (see section 1.3) with their corresponding outcomes.

With a focus on computer vision, chapter 2 reviewed the literature around ordinal coding and correlation of shape. More specifically, topics of local feature description, local feature object level tasks, biologically inspired computer vision approaches and ordinal neural coding along with ordinal coding and correlation approaches are reviewed. As a result, an appropriate name of Ordinal Shape Coding and Correlation (OSCC) is given to the resulting implementation of this thesis. Furthermore, the reviewed sequence alignment literature was used as a basis of the proposed sequence alignment hierarchy for the correlation of OSCC codings.

Chapter 3 provided illustrations for the derivation of the OSCC implementation. More specifically, pre-processing and sampling stages used by a prototype initially implemented by Austin [1] are derived and illustrated up to the point of obtaining an OSCC coding. Original work of this thesis then follows with a proposed sequence alignment hierarchy approach for the correlation of OSCC codings. Furthermore, a second correlation approach is proposed for higher level object recognition tasks that makes use of the intrinsic sequence alignment properties provided by the initially proposed sequence alignment hierarchy. OSCC parameters are identified and summarised followed by an inspection of computational complexity.

Chapter 3 meets aims 1 and 3 of this thesis (see section 1.3) which are system verification and system extensions. System extensions include the proposed use of edge angle features (see section 3.5.1), the proposed use of a novel hierarchical

sequence alignment method for the correlation of OSCC coding (3.6.2) in addition to a further approach that makes use of the intrinsic properties of the hierarchical sequence alignment method to obtain point correspondences (see section 3.6.3).

Chapter 4 provided a point correspondence level evaluation with respect to synthetic dataset variations combined with variations of the OSCC correlation approach. Dataset variations of similarity, affine and perspective transformations were used for the evaluation as well as clutter and occlusion. A detailed inspection into the high level performance metrics was provided allowing observations of how each individual shape contributed to the final high level score. Furthermore, an evaluation was undertaken within the context of local region description for a comparison with GSC, SIFT and GLOH methods. This chapter meets aims 1 and 2 of this thesis (see section 1.3) which are system verification and system evaluation with respect to the task of point correspondence.

Chapter 5 provided an object recognition evaluation with respect to dataset variations combined with variations of the OSCC approach. Dataset variations of interpolation, translation, rotation, scale, clutter and occlusion were used for the evaluation. Firstly, a simple object classification metric was defined based on local Hough voting of local point correspondences. The classifier was trained on a reference dataset which was then used to recall instances of the reference dataset with specific transformations applied. Results were presented in the same style as the point correspondence evaluation for simplicity. This chapter meets aims 1 and 2 of this thesis (see section 1.3) which are system verification and system evaluation with respect to the task of object recognition.

Chapter 6 provided an object level recognition evaluation with respect to the public MNIST handwritten digits dataset. OSCC was compared against the Shape Context (SC) method [2] with respect to published results of using 20,000 train images and the full test set. It was shown that OSCC obtains an error rate of 5.75% whereas the reported error rate for the SC method is 0.63%. Furthermore, the two-sided Wilcoxon Matched-Pairs Signed-Rank Test was used to identify that the SC method outperforms OSCC. A further evaluation was undertaken using the SC distance component only (referred to as the SC-D variation) - rather than using the MNIST optimised KNN distance function weights along with appearance cost and bending energy components. Using the two-sided Wilcoxon Matched-Pairs Signed-Rank Test it was shown that the OSCC method outperforms the SC-D method with 5000 train images. This chapter meets aims 1 and 2 of this thesis (see section 1.3) which are system verification and system evaluation with respect to the task of object recognition.

As a result of this thesis, a thoroughly engineered system which has been rigorously tested and has accompanying Graphical Processing Unit (GPU) and Message Passing Interface (MPI) implementations have been created. This meets the final aim (see section 1.3) of a well engineered system.

7.2 Conclusions

In this section, we draw conclusions from each evaluation undertaken within this thesis. In general, we conclude that further larger scale evaluations are required (using optimised parameters) with respect to data and methods to compare against.

7.2.1 Point Correspondences

From observations made with respect to the point correspondence evaluation chapter (see chapter 4), we conclude that the OSCC methods can be used as a discriminative descriptor within the context of translated, rotated and scaled shapes. Furthermore, the method can also be used as a discriminative descriptor within the context of moderately cluttered and occluded shapes. However, it is also concluded that the method is less discriminative for shapes with relatively less shape information such as 'v' while also being less discriminative for shapes with rotational symmetry such as 'x'. Consequently, the method is less discriminative for shapes that share both properties such as 'l'. As a result, higher level object recognition methods will need to be able to handle these properties. It is further concluded that larger scale evaluations with more complex shapes are required.

7.2.2 Local Interest Regions

From observations made with respect to the local interest region description evaluation (see section 4.9). It is concluded that small patches extracted from interest region detectors do not contain enough reliable edge information as input for the OSCC method. However, when providing binary images for reliable edge information, preliminary results suggest that the OSCC method is comparable to GSC, SIFT and GLOH methods for interpolation, translation, scale and shear transformations with the exception of rotation transformations. Furthermore, we conclude that larger scale evaluations are required.

7.2.3 Object Recognition

From observations made with respect to the synthetic object recognition evaluation chapter (see chapter 5), it is concluded that combined with the simple object classification metric defined within section 5.2, OSCC can be used for object recognition with respect to shapes that are not rotationally symmetric. Therefore, it is also concluded that further attention is required to improve the simple object classification metric to handle the case of rotationally symmetric shapes. Furthermore, with respect to the MNIST Handwritten Digits chapter (see chapter 6), it is identified that future work should include further optimisation stages to select the best OSCC parameters using a sub-set of training data for further evaluations with the competitor SC method.

7.3 Future Work

7.3.1 Appearance and Bending Energy Cost Evaluation

Section 6.5 shows that OSCC outperforms the SC method in the context of comparing pure shape descriptor methods. This is undertaken by removing additional steps introduced by the SC method which includes appearance and bending energy costs. Therefore future work is proposed to add the appearance and bending energy costs to the OSCC method which will be referred to as OSCC-AB. Consequently, an evaluation is proposed to compare results obtained by OSCC-AB with the results reported by Belongie et al. [66] for the SC method when applied to the MNIST handwritten digits dataset.

7.3.2 Further Evaluations

A large scale point correspondence / interest region evaluation within the context of shape recognition has been identified as future work within sections 7.2.1 and 7.2.2. Since the OSCC method has been designed to use semi-local / global areas to capture shape information, a large scale evaluation with respect to evaluating multiple local region sizes and semi-local / global shape descriptor methods would be beneficial. It would also be interesting to evaluate local descriptors within this context to gain insight into methods that are suitable for semi-local / global shape recognition.

The evaluation of further sequence alignment methods, constraints and penalties within the proposed hierarchical sequence alignment approach is identified as future work. It is of special interest to add a modification to the SD ring alignment method

(defined in section 4.2.5 of chapter 4) such that tokens that exceed a state distance threshold are ignored instead of being forced to match, therefore combining the SD approach with the Longest Common Subsequence approach.

OSCC parameter optimisation is proposed as future work with respect to the MNIST dataset evaluation. Further evaluations with respect to other public datasets other than the MNIST dataset are also of particular interest.

7.3.3 Improved Object Level Classifier

As identified within section 7.2.3 a simple but basic object level classifier metric has been used for the object recognition evaluations (see chapters 5 and 6). An outstanding issue with the metric is that it does not handle rotationally symmetric shapes which is due to time constraints of this work. A simple extension to handle rotationally symmetric shapes would be to update the relative angle binning function to handle such conditions with the possible cost of losing some discriminative information. In addition to improving the existing object level classifier, an evaluation of object level classifiers with respect to correspondences obtained from the OSCC method is proposed as future work.

It is also interesting to think of using a dense set of OSCC descriptors and their associated traceback paths to construct a graph from which belief propagation can be applied. Initial work on this was started but due to time constraints it is considered as future work.

7.3.4 Improved Sampling Method

This work has provided optimisations for the sampling method initially proposed by Austin [1] within the initial prototype implementation. This work has mainly focused on the correlation of the initially proposed OSCC coding with respect to the hierarchical sequence alignment approach proposed within this thesis. As a result, the sampling method itself has received little attention. Future work is proposed to evaluate different sampling patterns with special attention to sampling patterns that are biologically inspired.

7.3.5 Sensitivity Analysis

Section 3.7 identifies a large number of available parameters for the OSCC approach. For example, the proposed hierarchical sequence alignment method for OSCC allows

variations of sequence alignment methods to be used which in turn have a set of parameters. Therefore a sensitivity analysis is proposed as future work to determine the best parameter combinations for specific problems.

7.4 Summary

In this chapter, firstly we began by summarising the work undertaken within this thesis and identified links to each thesis aim. Therefore all aims are shown to be met. Secondly, conclusions of the thesis are identified and discussed. In general, OSCC has been evaluated with respect to point correspondence discriminability, local interest region discriminability and object recognition.

It is concluded that OSCC has been demonstrated to be discriminative for shapes that are not rotationally symmetrical. Furthermore, with respect to published results that the competitor SC method obtained for the MNIST handwritten digits dataset, it is shown that OSCC obtains an error rate of 5.75% whereas the SC method obtains an error rate of 0.63%. However, using a balanced dataset of 5000 training images and the MNIST full test set, it is shown that OSCC outperforms SC when comparing pure shape descriptor methods by removing the additional factors introduced by the SC method which includes appearance and bending energy costs. More specifically, OSCC obtained an error rate of 0.043% whereas the isolated SC distance component obtained an error rate of 0.063%.

Future work is proposed to optimise OSCC parameters specifically for the MNIST dataset in addition to integrating appearance and bending energy costs for further OCR evaluations. Further future work is defined and discussed which includes a proposed modification to improve discriminability for rotationally symmetric object recognition in addition to further evaluations and variations.

Appendix A

OSCC Algorithms

As an extension to chapter 3, OSCC dynamic programming algorithms defined in this appendix are as follows - where the resulting cost can be found at the bottom right cell of the output matrix:

- Algorithm 14 defines pseudocode for the Edit Distance accumulated cost matrix
- Algorithm 15 defines pseudocode for the LCS accumulated cost matrix
- Algorithm 16 defines pseudocode for the DTW accumulated cost matrix

Algorithm 14 Ring Edit Distance Accumulated Cost Matrix

```
1: function getRingEditDistanceAccCostMat(ringA, ringB)
2:     costMat ← getAggregatedFeatureBinaryCostMat(ringA, ringB)
3:     accMat ← [ringA length+1][ringB length+1]
4:     for idxA ← 0 to ringA length +1 do
5:         accMat(idxA, 0) ← idxA
6:     end for
7:     for idxB ← 0 to ringB length +1 do
8:         accMat(0, idxB) ← idxB
9:     end for
10:    for idxA ← 1 to ringA length +1 do
11:        for idxB ← 1 to ringB length +1 do
12:            sub ← accMat[idxA − 1][idxB − 1] + costMat[idxA − 1][idxB − 1]
13:            ins ← accMat[idxA][idxB − 1]
14:            del ← accMat[idxA − 1][idxB]
15:            accMat(idxA, idxB) ← min(sub, ins, del)
16:        end for
17:    end for
18:    return accMat
19: end function
```

Algorithm 15 Ring Longest Common Subsequence Accumulated Cost Matrix

```
1: function getRingLCSAccCostMat(ringA, ringB)
2:     costMat ← getAggregatedFeatureBinaryCostMat(ringA, ringB)
3:     accMat ← [ringA length+1][ringB length+1]
4:     for idxA ← 0 to ringA length +1 do
5:         accMat(idxA, 0) ← 0
6:     end for
7:     for idxB ← 0 to ringB length +1 do
8:         accMat(0, idxB) ← 0
9:     end for
10:    for idxA ← 1 to ringA length +1 do
11:        for idxB ← 1 to ringB length +1 do
12:            if costMat[idxA − 1][idxB − 1] == 0 then
13:                accMat(idxA, idxB) ← accMat[idxA − 1][idxB − 1] + 1
14:            else
15:                ins ← accMat[idxA][idxB − 1]
16:                del ← accMat[idxA − 1][idxB]
17:                accMat(idxA, idxB) ← max(ins, del)
18:            end if
19:        end for
20:    end for
21:    return accMat
22: end function
```

Algorithm 16 Ring Dynamic Time Warping Accumulated Cost Matrix

```
1: function getRingDTWAccCostMat(ringA, ringB, RESO, PA, r)
2:     costMat ← getAggregatedFeatureRelCostMat(ringA, ringB)
3:     accMat ← [ringA length+1][ringB length+1]
4:     for idxA ← 1 to ringA length do
5:         accMat(idxA, 0) ← accMat(idxA − 1) + costMat(idxA, 0)
6:     end for
7:     for idxB ← 1 to ringB length do
8:         accMat(0, idxB) ← accMat(0, idxB − 1) + costMat(0, idxB)
9:     end for
10:    for idxA ← 1 to ringA length do
11:        for idxB ← 1 to ringB length do
12:            sub ← accMat[idxA − 1][idxB − 1]
13:            ins ← accMat[idxA][idxB − 1]
14:            del ← accMat[idxA − 1][idxB]
15:            accMat(idxA, idxB) ← costMat[idxA][idxB] + min(sub, ins, del)
16:        end for
17:    end for
18:    return accMat
19: end function
```

Appendix B

Synthetic Alphabet Dataset

The synthetic alphabet dataset (see figure B.1) comprises of the lowercase FreeSans characters from ‘a’ to ‘z’. All individual characters have been uniformly scaled such that the maximum dimension (width or height) is equal to 150 pixels. Figure B.2 shows the number of edge points for each image.

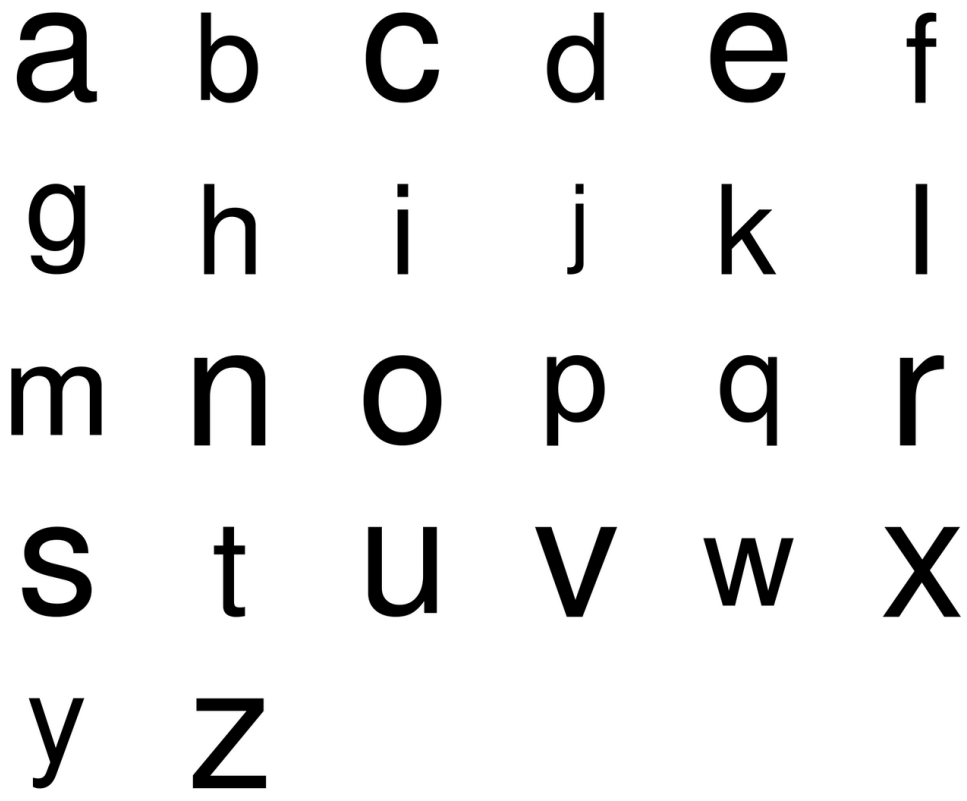


Figure B.1: FreeSans Alphabet Dataset. Each character is uniformly scaled such that the maximum dimension is equal to 150 pixels.

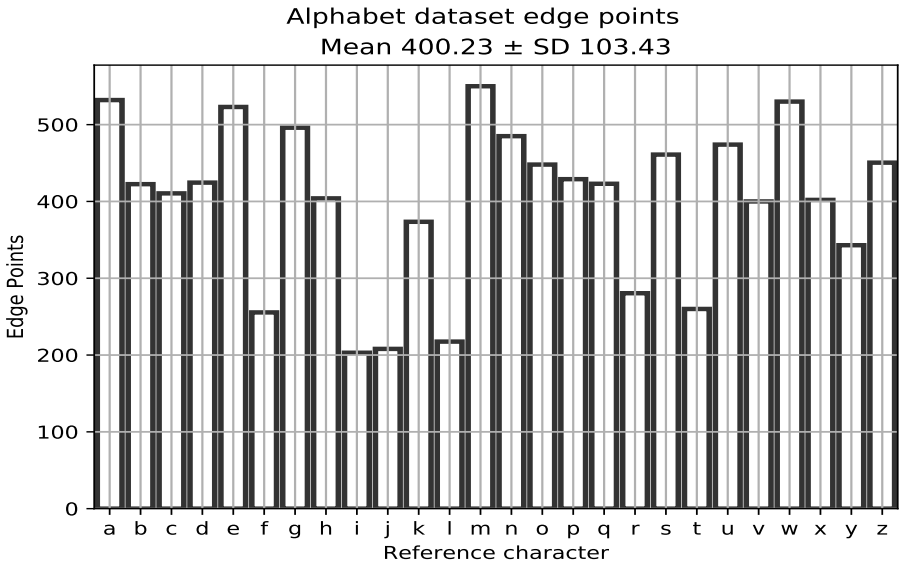


Figure B.2: Number of edge points for each character in the alphabet dataset.

Appendix C

Synthetic Point Correspondence Evaluation

This appendix describes the dataset variations used throughout chapter 4. All parameters for the OSCC method variations are defined in the method variation parameters section 4.2.5 of chapter 4 These parameters are fixed throughout the synthetic point correspondence evaluation. Furthermore, reference images are defined as all 26 images within the alphabet dataset (see appendix B).

Algorithm 17 defines pseudocode for the ground truth point correspondence extraction method. Three input parameters are defined as $refPtSet$, M and $distThresh$ where $refPtSet$ is the reference image edge point set, M is a transformation matrix and $distThresh$ is a maximum ground truth correspondence mapping distance threshold. $transform$ and $invTransform$ functions are used which simply applies or inverts a transformation with respect to a point set. It can be seen that no more than one ground truth correspondence can exist between a reference image point and a transformed image point. Furthermore, $distThresh$ is set to 5 throughout the thesis.

Algorithm 17 Ground Truth Point Correspondence Extraction

```
1: function gtPtCorresExtraction(refPtSet, M, distThresh)
2:     gtCorres ← {}
3:     transPtSet ← transform(refPtSet, M)
4:     invTransPtSet ← transform(transPtSet, M)
5:     for ptIdx ← 0 to transPtSet length do
6:         candidate ← NULL
7:         candidateDistance ← MAX_VALUE
8:         for rIdx ← 0 to refPtSet length do
9:             dist ← distance(invTransPtSet[ptIdx], refPtSet[rIdx])
10:            if dist < candidateDistance AND dist < distThresh then
11:                candidate ← refPtSet[rIdx]
12:                candidateDistance ← dist
13:            end if
14:        end for
15:        if candidate ≠ NULL then
16:            gtCorres ← gtCorres ∪ {refPtSet[rIdx], transPtSet[ptIdx]}
17:        end if
18:    end for
19:    return gtCorres
20: end function
```

C.1 Interpolation

Two interpolation variations are used for the experiment. The first is the None variation which does not apply any interpolation to the reference dataset. The second is the Cubic variation which represents bicubic interpolation using a 4 x 4 neighborhood. For each individual reference character image, figure C.1 shows the number of ground truth correspondences obtained between a single reference image and its corresponding interpolated images. Since two interpolation variations are used, a single reference image has two sets of correspondences.

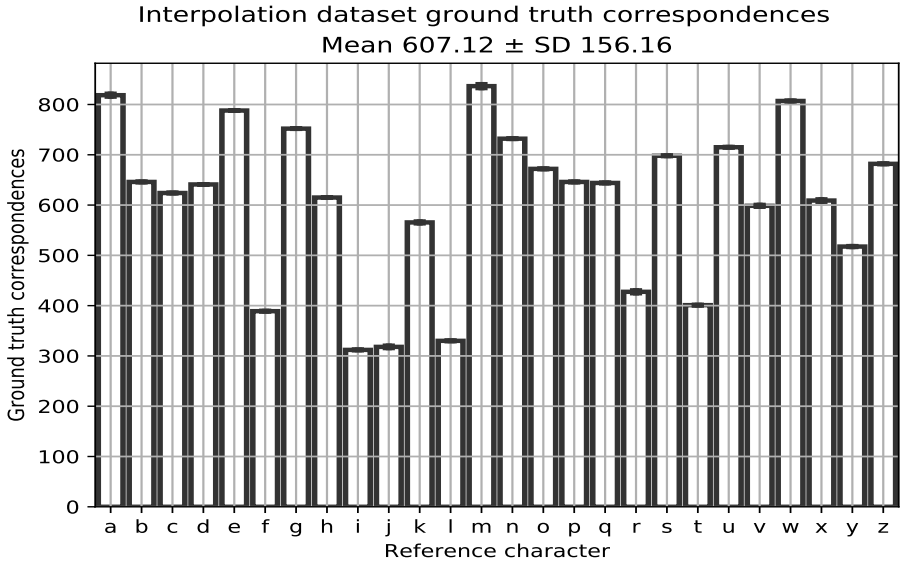


Figure C.1: Number of extracted ground truth correspondences for the interpolation variations dataset is shown for each reference character image. There are two ground truth correspondence counts for each reference character as two interpolation variations are used.

Due to a small number of interpolation variations, a matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.3) is defined as the average precision obtained for an individual alphabet character with respect to two OSCC methods. As two dataset variations are used, 52 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 52 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

C.2 Reflection

Both horizontal and vertical reflection variations are used for the experiment. For each individual reference character image, figure C.2 shows the number of ground truth correspondences obtained between a single reference image and its corresponding transformed images. Since two reflection variations are used, a single reference image has two sets of correspondences.

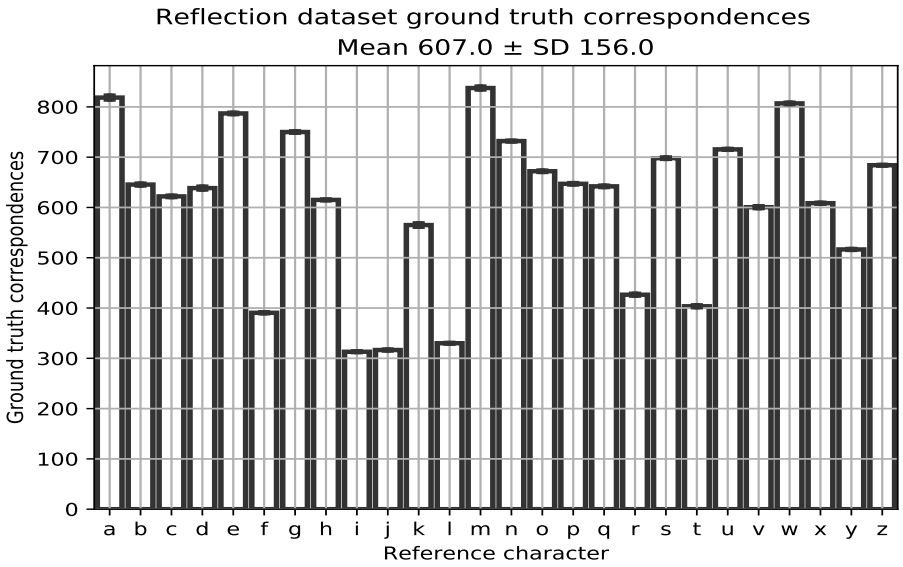


Figure C.2: Number of extracted ground truth correspondences for the reflection variations dataset is shown for each reference character image. There are two ground truth correspondence counts for each reference character as two reflection variations are used.

C.3 Translation

Four translation variations are used for the experiment which are: $(x+0.5, y+0.5), (x+0.5, y+1), (x+1, y+0.5), (x+1, y+1)$. For each individual reference character image, figure C.3 shows the number of ground truth correspondences obtained between a single reference image and its corresponding transformed images. Since four variations are used, a single reference image has four sets of correspondences.

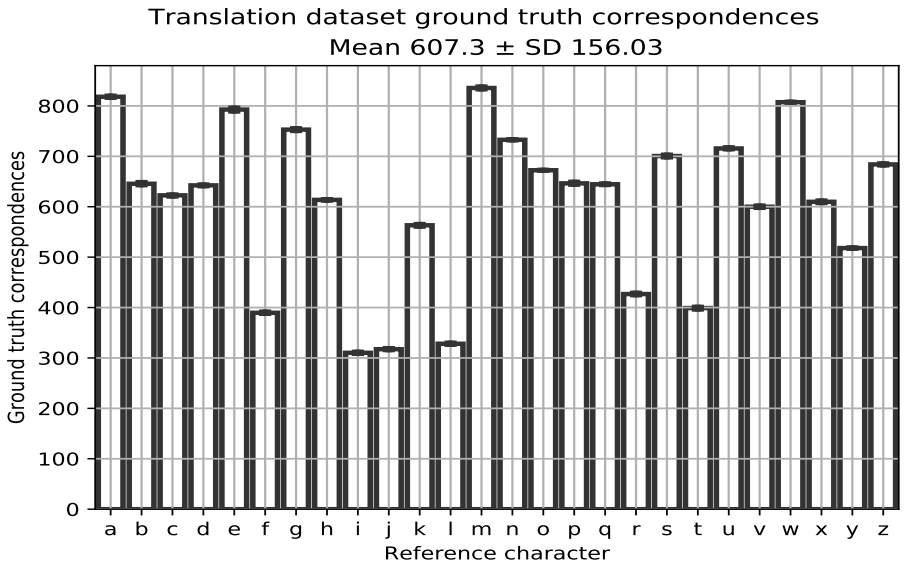


Figure C.3: Number of extracted ground truth correspondences for the translation variations dataset is shown for each reference character image. There are four ground truth correspondence counts for each reference character as two translation variations are used.

Like the interpolation evaluation, due to a small number of interpolation variations a matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.4.2) is defined as the average precision obtained for an individual alphabet character with respect to two OSCC methods. As four dataset variations are used, 104 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 104 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

C.4 Rotation

Clockwise rotations from 45 degrees to 315 degrees with intervals of 45 degrees are evaluated. Therefore seven rotation variations are used. For each individual reference character image, figure C.4 shows the number of ground truth correspondences obtained between a single reference image and its corresponding transformed images. Since seven variations are used, a single reference image has seven sets of correspondences.

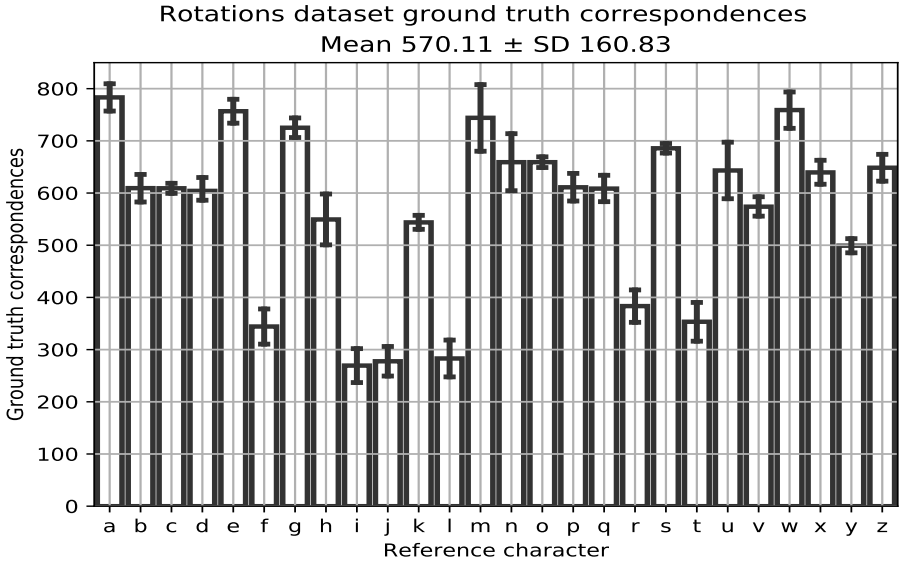


Figure C.4: Number of extracted ground truth correspondences for the rotation variations dataset is shown for each reference character image. There are seven ground truth correspondence counts for each reference character as seven rotation variations are used.

A matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.4.3) is defined as the mAP for a single rotation variation with respect to two OSCC methods. As seven dataset variations are used, seven matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

C.5 Scale

Uniform scaling of 0.5, 0.75, 1.25 and 1.5 is evaluated. Therefore four scale variations are used. For each individual reference character image, figure C.5 shows the number of ground truth correspondences obtained between a single reference image and its corresponding transformed images. Since four variations are used, a single reference image has four sets of correspondences.

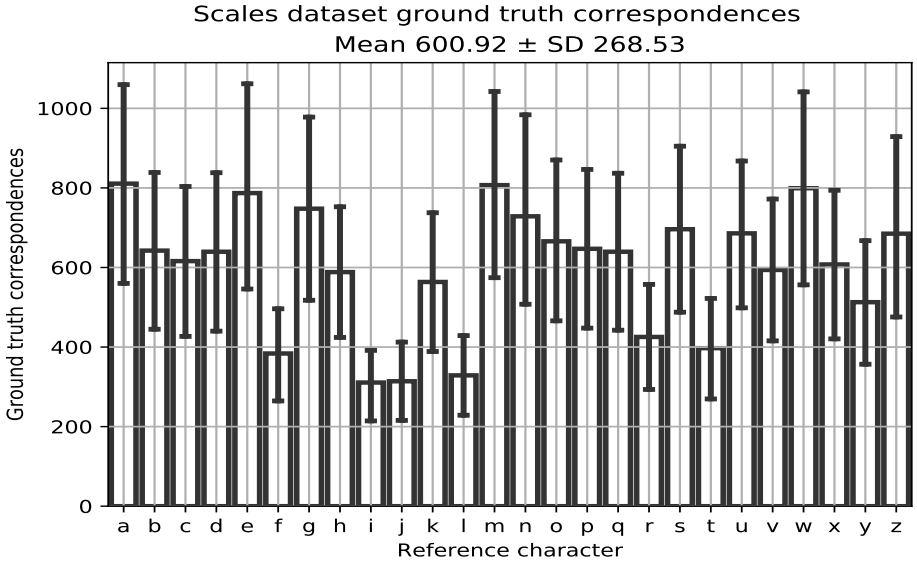


Figure C.5: Number of extracted ground truth correspondences for the scale variations dataset is shown for each reference character image. There are four ground truth correspondence counts for each reference character as four scale variations are used.

Like the interpolation and translation evaluation, due to a small number of interpolation variations a matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.4.4) is defined as the average precision obtained for an individual alphabet character with respect to two OSCC methods. As four dataset variations are used, 104 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 104 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

C.6 Shear

Shear transformations of 0.1, 0.2, 0.3, 0.4, 0.5 and 0.6 are evaluated for both image x and y direction. Therefore twelve shear variations are used. For each individual reference character image, figure C.6 shows the number of ground truth correspondences obtained between a single reference image and its corresponding transformed images. Since twelve variations are used, a single reference image has four twelve sets of correspondences.

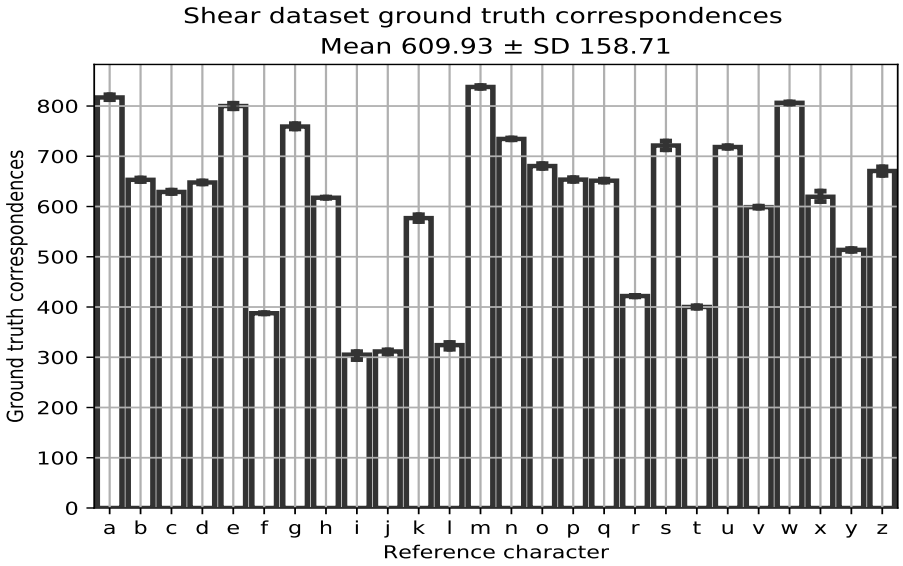


Figure C.6: Number of extracted ground truth correspondences for the shear variations dataset is shown for each reference character image. There are twelve ground truth correspondence counts for each reference character as twelve shear variations are used.

A matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.5.1) is defined as the mAP for a single shear variation with respect to two OSCC methods. As twelve dataset variations are used, twelve matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

C.7 Perspective

Section 4.6 of chapter4 defines the process of obtaining ground truth correspondences for the 3D teddy bear. Figure C.7 shows the number of ground truth correspondences extracted between the reference image and transformed images. Furthermore, a matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.6) is defined as the mAP for a single rotation variation with respect to two OSCC methods. As eleven dataset variations are used, eleven matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

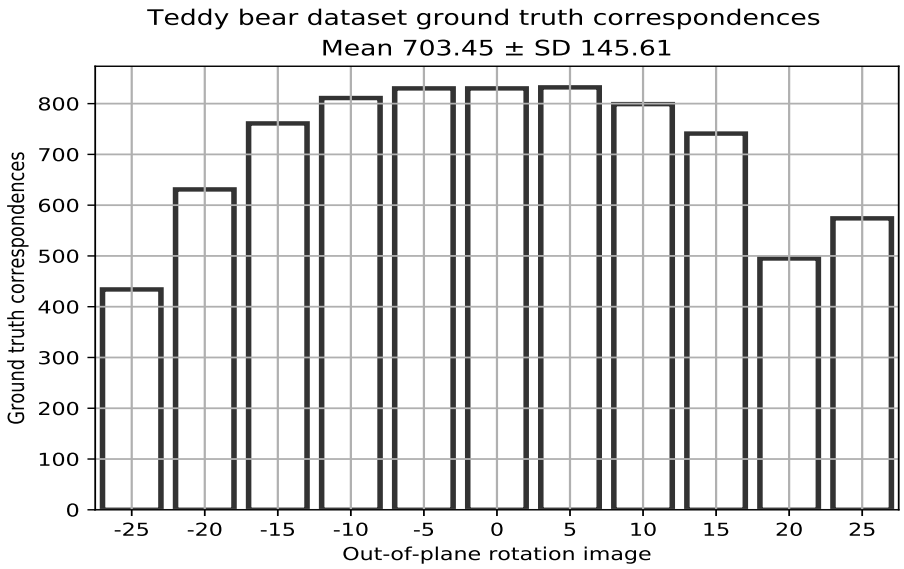


Figure C.7: Number of extracted ground truth correspondences for the 3D teddy bear out-of-plane variation dataset

C.8 Clutter

The clutter variations are used for the experiment. For each individual reference character image, figure C.8 shows the number of ground truth correspondences obtained between a single reference image and its corresponding cluttered images. Since two variations are used, a single reference image has two sets of correspondences.

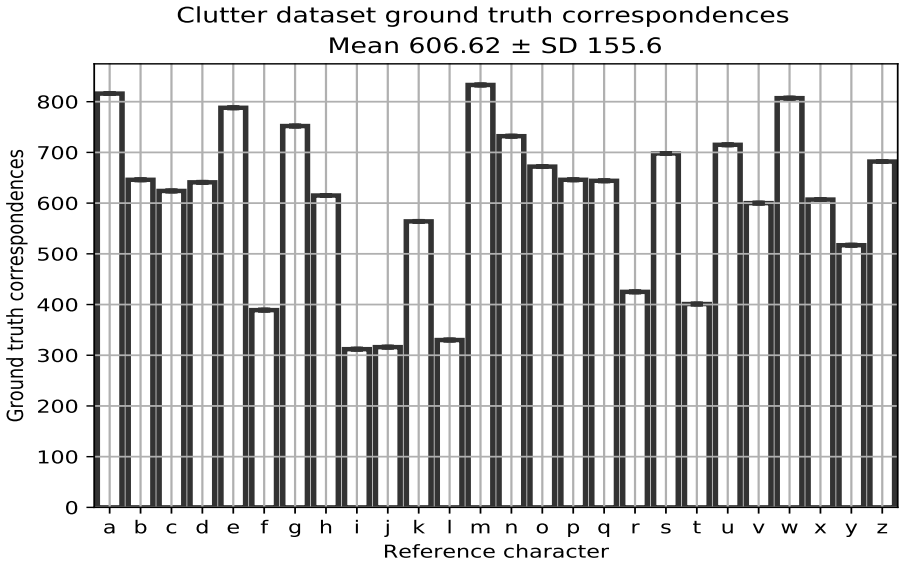


Figure C.8: Number of extracted ground truth correspondences for the clutter variations dataset is shown for each reference character image. There are two ground truth correspondence counts for each reference character as two translation variations are used.

A matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.7) is defined as the average precision obtained for an individual alphabet character with respect to two OSCC methods. As two dataset variations are used, 52 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 52 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

C.9 Occlusion

One clutter variation is used for the experiment. For each individual reference character image, figure C.9 shows the number of ground truth correspondences obtained between a single reference image and its corresponding cluttered images. Since one variation is used, a single reference image has one set of correspondences.

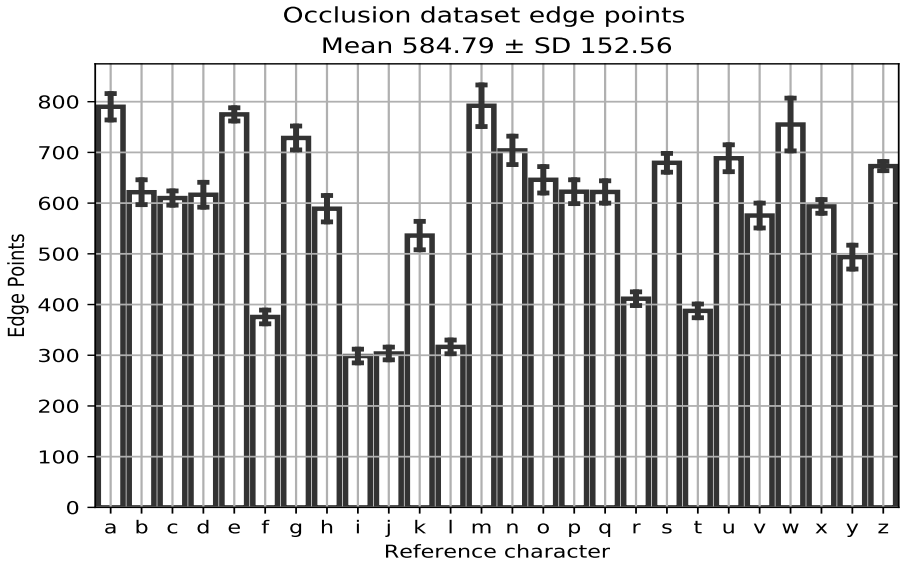


Figure C.9: Number of extracted ground truth correspondences for the occlusion variation dataset is shown for each reference character image. There is one ground truth correspondence count for each reference character as one clutter variation is used.

A matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 4 section 4.8) is defined as the average precision obtained for an individual alphabet character with respect to two OSCC methods. As one dataset variation is used, 26 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 26 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

Appendix D

Interest Region Descriptor Evaluation

As stated in section 4.9 of chapter 4 the evaluation framework defined by Mikolajczyk and Schmid [18] is used with the accompanying code. Local regions were detected and extracted for the alphabet dataset (see appendix B) using the provided code.

Figure D.1 shows the number of pairwise region comparisons between a reference and transformed image across all dataset transformations. The minimum number of pairwise region comparisons for character 'o' is 15. Furthermore, figure D.2 shows the number of ground truth region correspondences between a reference and transformed image.

In section 4.9 of chapter 4 a matched pair used by the two-sided Wilcoxon Matched-Pairs Signed-Rank Test is defined as the mAP for a single dataset variation with respect to two methods. As six dataset variations are used, six matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

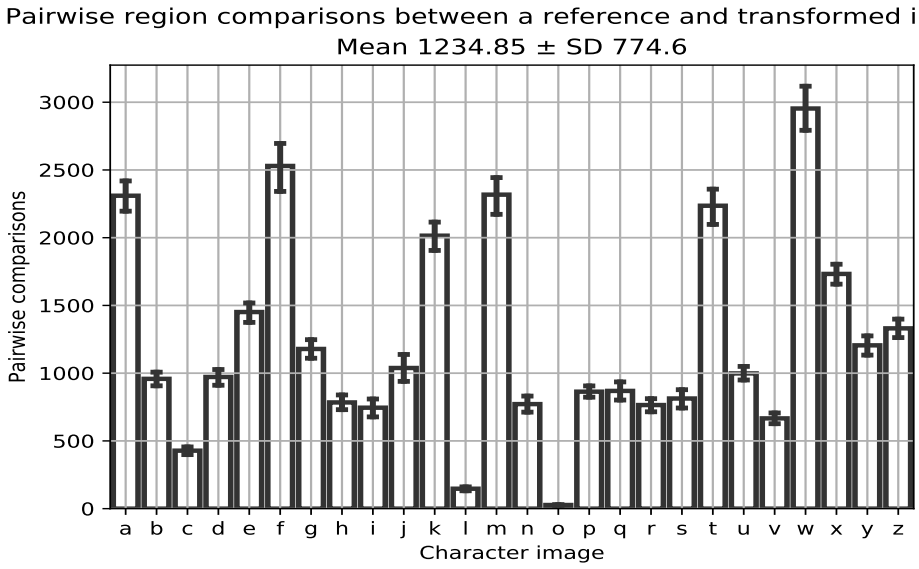


Figure D.1: Pairwise local region comparisons between a reference and transformed image for all transformations. The minimum number of pairwise region comparisons for character 'o' is 15.

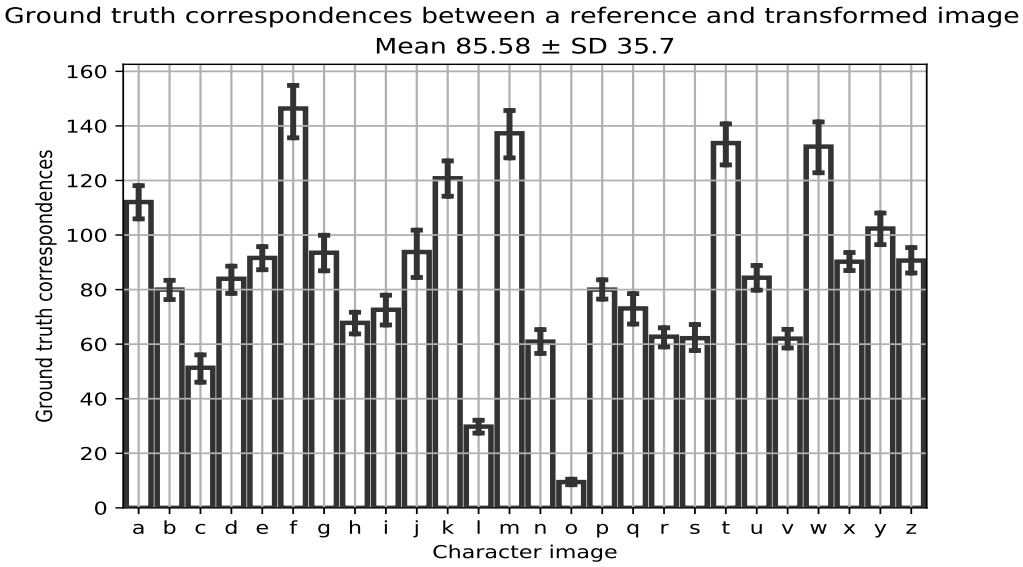


Figure D.2: Ground truth region correspondences between a reference and transformed image for all transformations.

Appendix E

Synthetic Object Level Evaluation

This appendix describes the algorithms and dataset variations used throughout chapter 5. All evaluations within this chapter follow the same format - for an individual dataset variation, all reference images (all 26 images from the alphabet dataset defined by appendix B) are compared with all transformed images (26 images in total where the transformation has been applied to each reference image) using the OSCC object level classification metric (chapter 5 section 5.2).

E.1 Interpolation

The interpolation variations dataset used for the evaluation is composed of two variations. The first is the None variation which does not apply any interpolation to the reference dataset. The second is the Cubic variation which represents bicubic interpolation using a 4 x 4 neighborhood. Figure E.1 shows the number of edge points for each character in the interpolation variations dataset.

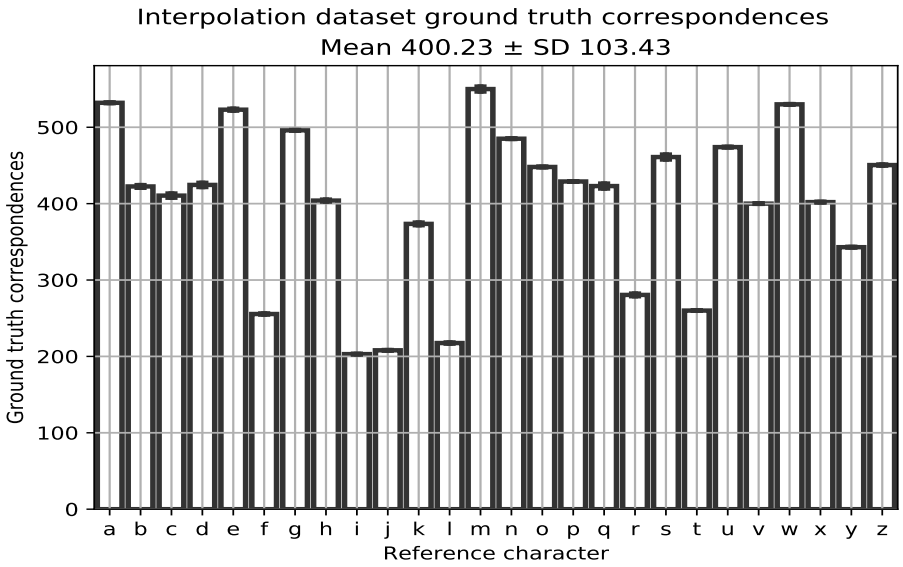


Figure E.1: Number of edge points for each character in the interpolation variations dataset.

For the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 5 section 5.4), an individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. As two dataset variations are used, 52 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 52 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

E.2 Translation

The translation variations dataset used for the evaluation is composed of five variations: $(x+0.5, y+0.5), (x+0.5, y+1), (x, y), (x+1, y+0.5), (x+1, y+1)$. Figure E.2 shows the number of edge points for each character in the translation variations dataset. For the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 5 section 5.5), an individual matched pair is treated as the mAP for an individual transformation obtained by two different OSCC method variations. As five dataset variations are used, five matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

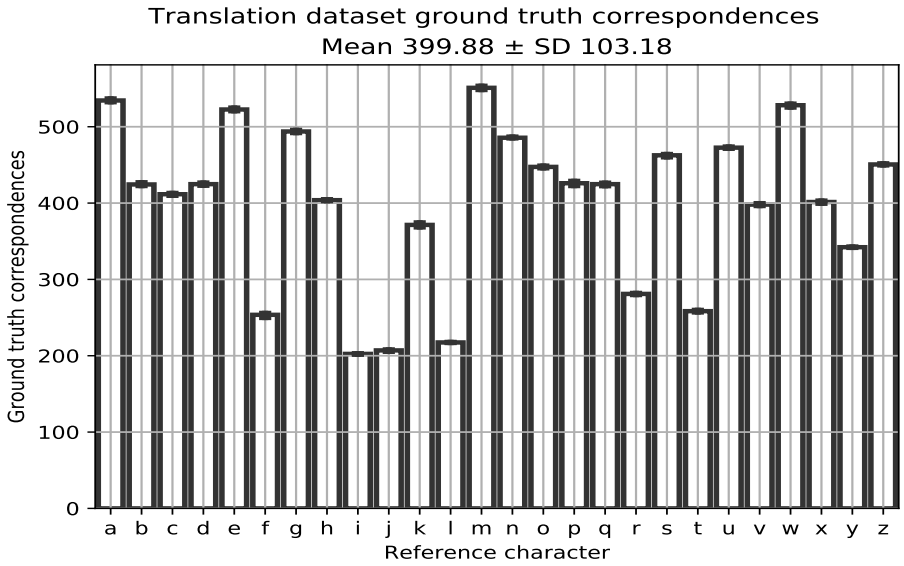


Figure E.2: Number of edge points for each character in the translation variations dataset.

E.3 Rotation

The rotation variations dataset used for the evaluation is composed of seven variations: 45 - 315 degrees in 45 degree increments. Figure E.3 shows the number of edge points for each character in the rotation variations dataset. For the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 5 section 5.6), an individual matched pair is treated as the mAP for an individual transformation obtained by two different OSCC method variations. As seven dataset variations are used, seven matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test.

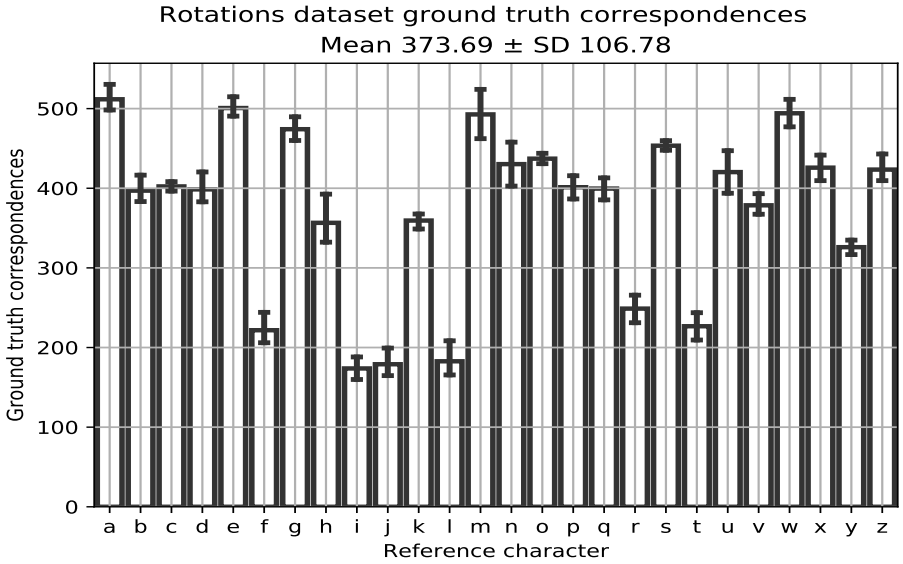


Figure E.3: Number of edge points for each character in the rotation variations dataset.

E.4 Scale

The scale variations dataset used for the evaluation is composed of three variations: 0.75, 1.25 and 1.5. Figure E.4 shows the number of edge points for each character in the scale variations dataset. For the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 5 section 5.7), an individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. As three dataset variations are used, 78 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 78 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

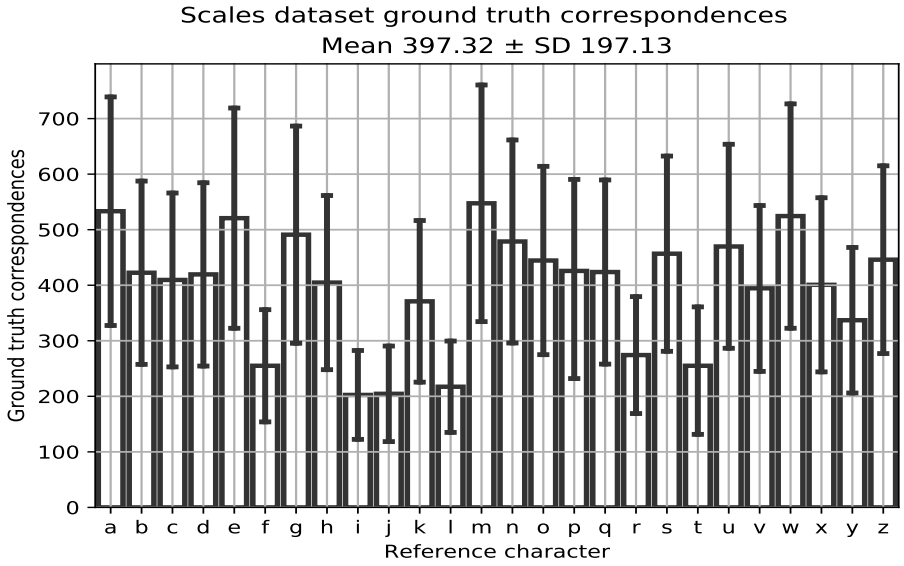


Figure E.4: Number of edge points for each character in the scale variations dataset.

E.5 Clutter

The clutter variations dataset used for the evaluation is composed of two variations. Figure E.5 shows the number of edge points for each character in the clutter variations dataset. For the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 5 section 5.8), an individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. As two dataset variations are used, 52 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 52 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

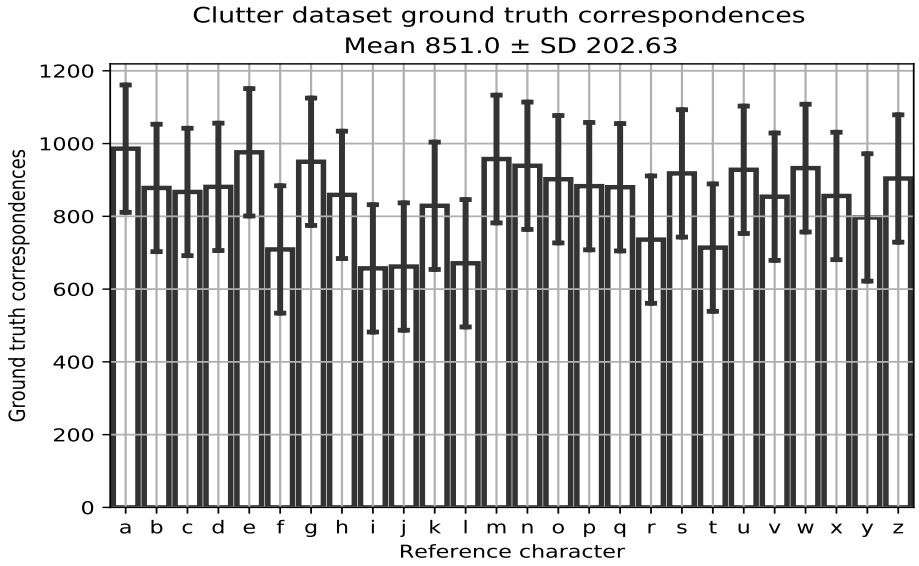


Figure E.5: Number of edge points for each character in the clutter variations dataset.

E.6 Occlusion

The occlusion variations dataset used for the evaluation is composed of one variation. Figure E.6 shows the number of edge points for each character in the clutter variations dataset. For the two-sided Wilcoxon Matched-Pairs Signed-Rank Test (chapter 5 section 5.9), an individual matched pair is treated as the Precision for an individual transformed alphabet character obtained by two different OSCC method variations. As one dataset variation is used, 26 matched pairs are used for each two-sided Wilcoxon Matched-Pairs Signed-Rank Test. 26 is calculated by multiplying the number of dataset variations by the number of images within the dataset which is 26 alphabet character images.

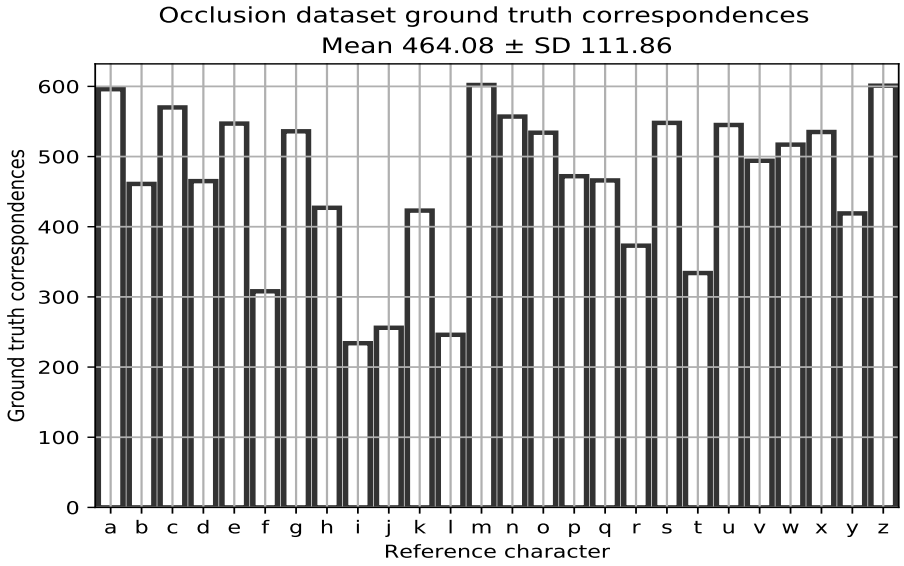


Figure E.6: Number of edge points for each character in the occlusion variations dataset.

Appendix F

MNIST Handwritten Digits Evaluation

All OSCC method settings are fixed throughout the MNIST Handwritten Digits Evaluation chapter 6. Tables of parameters are defined by section 6.3.1 of chapter 6, which detail the exact SD-A OSCC variation parameters used. The object level recognition metric defined in chapter 5 section 5.2 is used for the SD-A OSCC approach with parameters $aBins$, r and $rBins$ set to 8, 13 and 5 respectively. Furthermore, like the Shape Context method, 100 edge points were uniformly sampled for each image from which 100 point descriptors were constructed.

Section 6.5 of chapter 6 compares the SC distance component D_{sc} [66] with the OSCC method using the SC code provided by the authors [67]. This approach is referred to as the SC-D method in this thesis - this setup excludes the SC MNIST specific KNN classifier with optimised distance function weights along with the appearance cost D_{ac} and bending energy D_{be} [66]. The `demo_2.m` file provided by the SC authors is used which contains a demo using MNIST data [67]. All parameter values remained exactly how they were originally defined by the authors within the source code. The value assigned to the `sc_cost` variable on line 218 was used to report the SC distance component value for the last warping iteration. Two modifications were made to the source code:

- The display flag was set to false so that user interface components were not displayed during the evaluation
- A command line interface wrapper was added so that two MNIST digits could be used as input and the SC distance component value was returned as output

A script was written around the command line interface wrapper which iterated

through the MNIST train and test set to obtain the SC distance component between train and test images - the gathered costs were subsequently used by a KNN classifier ($k = 3$) to provide the final object recognition metrics. As detailed by section 6.5 of chapter 6, for the evaluation, 5000 MNIST train images were used (the first 500 for each class) along with the full test set. A Docker image was created that encapsulated the script along with the command line interface wrapper so that the code could be easily executed on various machines.

References

[1] J. Austin, University of York; (Internal Communication) Biologically inspired relativistic theory of object recognition with accompanying code, 2013.

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, and others, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[3] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *European conference on computer vision*, 2002, pp. 128–142.

[4] S. A. Winder and M. Brown, "Learning local image descriptors," in *2007 ieee conference on computer vision and pattern recognition*, 2007, pp. 1–8.

[5] S. Winder, G. Hua, and M. Brown, "Picking the best daisy," in *2009 ieee conference on computer vision and pattern recognition*, 2009, pp. 178–185.

[6] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative learning of deep convolutional feature point descriptors," in *Proceedings of the ieee international conference on computer vision*, 2015, pp. 118–126.

[7] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *CVPR*, 2017.

[8] Belongie and Malik, "Matching with shape contexts," in *2000 proceedings workshop on content-based access of image and video libraries*, 2000, pp. 20–26.

[9] G. Mori, S. Belongie, and J. Malik, "Efficient shape matching using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1832–1837, 2005.

[10] G. Mori and J. Malik, "Recognizing objects in adversarial clutter: Breaking a visual captcha," in *2003 ieee computer society conference on computer vision and*

pattern recognition, 2003. Proceedings., 2003, vol. 1, pp. I–I.

[11] A. Berg, "Shape matching and object recognition [ph. D. Thesis]," *Computer Science Division, University of California, Berkeley, Calif, USA*, 2005.

[12] K. Mikolajczyk, A. Zisserman, and C. Schmid, "Shape recognition with edge-based features," in *British machine vision conference (bmvc'03)*, 2003, vol. 2, pp. 779–788.

[13] A. Johnson, "Spin-images: A representation for 3-d surface matching," PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1997.

[14] A. E. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3D scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, May 1999.

[15] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh ieee international conference on computer vision*, 1999, vol. 2, pp. 1150–1157 vol.2.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] R. Sukthankar, "PCA-sift: A more distinctive representation for local image descriptors," in *Proceedings of the 2004 ieee computer society conference on computer vision and pattern recognition, 2004. CVPR 2004.*, 2004, vol. 2, pp. II–II.

[18] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.

[19] E. Tola, V. Lepetit, and P. Fua, "A fast local descriptor for dense matching," in *2008 ieee conference on computer vision and pattern recognition*, 2008, pp. 1–8.

[20] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*, 2010, pp. 778–792.

[21] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to sift or surf," 2011.

[22] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *2012 ieee conference on computer vision and pattern recognition*, 2012, pp. 510–517.

[23] M. Heikkilä, M. Pietikäinen, and C. Schmid, "Description of interest regions with center-symmetric local binary patterns," in *Computer vision, graphics and image processing*, 2006, pp. 58–69.

[24] R. Gupta, H. Patil, and A. Mittal, "Robust order-based methods for feature description," in *2010 ieee computer society conference on computer vision and pattern recognition*, 2010, pp. 334–341.

[25] M. Toews and W. Wells, "SIFT-rank: Ordinal description for invariant feature correspondence," in *2009 ieee conference on computer vision and pattern recognition*, 2009, pp. 172–177.

[26] F. Tang, S. H. Lim, N. L. Chang, and H. Tao, "A novel feature descriptor invariant to complex brightness changes," in *2009 ieee conference on computer vision and pattern recognition*, 2009, pp. 2631–2638.

[27] F. Tang, S. H. Lim, and N. L. Chang, "An improved local feature descriptor via soft binning," in *2010 ieee international conference on image processing*, 2010, pp. 861–864.

[28] B. Fan, F. Wu, and Z. Hu, "Aggregating gradient distributions into intensity orders: A novel local image descriptor," in *CVPR 2011*, 2011, pp. 2377–2384.

[29] B. Fan, F. Wu, and Z. Hu, "Rotationally invariant descriptors using intensity order pooling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2031–2045, Oct. 2012.

[30] Z. Wang, B. Fan, and F. Wu, "Local intensity order pattern for feature description," in *2011 international conference on computer vision*, 2011, pp. 603–610.

[31] Z. Wang, B. Fan, G. Wang, and F. Wu, "Exploring local and overall ordinal information for robust feature description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 11, pp. 2198–2211, Nov. 2016.

[32] P. Moreels and P. Perona, "Evaluation of features detectors and descriptors based on 3D objects," in *Tenth ieee international conference on computer vision (iccv'05) volume 1*, 2005, vol. 1, pp. 800–807 Vol. 1.

[33] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[34] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[35] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[36] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1681–1707, 2011.

[37] R. Fergus, P. Perona, and A. Zisserman, "Weakly supervised scale-invariant learning of models for visual recognition," *International journal of computer vision*, vol. 71, no. 3, pp. 273–303, 2007.

[38] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-local Affine Parts for Object Recognition," in *British Machine Vision Conference (BMVC '04)*, 2004, pp. 779–788.

[39] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In workshop on statistical learning in computer vision, eccv*, 2004, pp. 1–22.

[40] M. Eitz, K. Hildebrand, T. Boubekeur, and M. Alexa, "Sketch-based image retrieval: Benchmark and bag-of-features descriptors," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 11, pp. 1624–1636, 2011.

[41] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 ieee computer society conference on computer vision and pattern recognition (cvpr'06)*, 2006, vol. 2, pp. 2169–2178.

[42] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.

[43] O. Chum, J. Matas, and J. Kittler, "Locally optimized ransac," in *Joint pattern recognition symposium*, 2003, pp. 236–243.

[44] O. Chum and J. Matas, "Matching with prosac - progressive sample consensus," in *2005 ieee computer society conference on computer vision and pattern recognition (cvpr'05)*, 2005, vol. 1, pp. 220–226 vol. 1.

[45] J. Austin, "Grey scale n tuple processing," in *Pattern recognition*, 1988, pp. 110–119.

[46] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," in *Papers presented at the december 1-3, 1959, eastern joint ire-aiee-acm computer conference*, 1959, pp. 225–232.

[47] S. Thorpe and J. Gautrais, "Rank order coding," in *Computational neuroscience: Trends in research, 1998*, J. M. Bower, Ed. Boston, MA: Springer US, 1998, pp. 113–118.

[48] R. Van Rullen, J. Gautrais, A. Delorme, and S. Thorpe, "Face processing using one spike per neurone," *Biosystems*, vol. 48, nos. 1-3, pp. 229–239, 1998.

[49] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980.

[50] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, pp. 106–154, 1962.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[52] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[54] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," *CoRR*, vol. abs/1506.02025, 2015.

[55] C. Szegedy *et al.*, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.

[56] S. H. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *CoRR*, vol. abs/1702.02284, 2017.

[57] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553,

p. 436, 2015.

[58] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "DARTS: deceiving autonomous cars with toxic signs," *CoRR*, vol. abs/1802.06430, 2018.

[59] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, p. 707, 1966.

[60] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *Proceedings seventh international symposium on string processing and information retrieval. SPIRE 2000*, 2000, pp. 39–48.

[61] H. Huang and K. Kawagoe, "Time series classification method based on longest common subsequence and textual approximation," in *Seventh international conference on digital information management (icdim 2012)*, 2012, pp. 130–137.

[62] M. Müller, "Dynamic time warping," *Information retrieval for music and motion*, pp. 69–84, 2007.

[63] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining*, 2012, pp. 262–270.

[64] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[65] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[66] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[67] S. Belongie, J. Malik, and J. Puzicha, 2005. [Online]. Available: `https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/code/sc_demo/`.