



1-1-2014

Shape Representations Using Nested Descriptors

Jeffrey Byrne

University of Pennsylvania, jebyrne@gmail.com

Follow this and additional works at: <http://repository.upenn.edu/edissertations>

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Byrne, Jeffrey, "Shape Representations Using Nested Descriptors" (2014). *Publicly Accessible Penn Dissertations*. 1220.
<http://repository.upenn.edu/edissertations/1220>

This paper is posted at ScholarlyCommons. <http://repository.upenn.edu/edissertations/1220>
For more information, please contact libraryrepository@pobox.upenn.edu.

Shape Representations Using Nested Descriptors

Abstract

The problem of shape representation is a core problem in computer vision. It can be argued that shape representation is the most central representational problem for computer vision, since unlike texture or color, shape alone can be used for perceptual tasks such as image matching, object detection and object categorization.

This dissertation introduces a new shape representation called the nested descriptor. A nested descriptor represents shape both globally and locally by pooling salient scaled and oriented complex gradients in a large nested support set. We show that this nesting property introduces a nested correlation structure that enables a new local distance function called the nesting distance, which provides a provably robust similarity function for image matching. Furthermore, the nesting property suggests an elegant flower like normalization strategy called a log-spiral difference. We show that this normalization enables a compact binary representation and is equivalent to a form a bottom up saliency. This suggests that the nested descriptor representational power is due to representing salient edges, which makes a fundamental connection between the saliency and local feature descriptor literature. In this dissertation, we introduce three examples of shape representation using nested descriptors: nested shape descriptors for imagery, nested motion descriptors for video and nested pooling for activities. We show evaluation results for these representations that demonstrate state-of-the-art performance for image matching, wide baseline stereo and activity recognition tasks.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Computer and Information Science

First Advisor

Jianbo Shi

Keywords

activity recognition, computer vision, descriptors, matching, shape, stereo

Subject Categories

Computer Sciences

SHAPE REPRESENTATIONS USING NESTED DESCRIPTORS

Jeffrey Byrne

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy

2014

Supervisor of Dissertation

Jianbo Shi
Professor of CIS

Graduate Group Chairperson

Val Tannen
Professor of CIS

Dissertation Committee:

Kostas Daniilidis, Professor of CIS (chair)
CJ Taylor, Professor of CIS
Jean Gallier, Professor of CIS
Anthony Hoogs, Kitware Inc. (external)

ABSTRACT

SHAPE REPRESENTATIONS USING NESTED DESCRIPTORS

Jeffrey Byrne

Jianbo Shi

The problem of shape representation is a core problem in computer vision. It can be argued that shape representation is the most central representational problem for computer vision, since unlike texture or color, shape alone can be used for perceptual tasks such as image matching, object detection and object categorization.

This dissertation introduces a new shape representation called the nested descriptor. A nested descriptor represents shape both globally and locally by pooling salient scaled and oriented complex gradients in a large nested support set. We show that this nesting property introduces a nested correlation structure that enables a new local distance function called the nesting distance, which provides a provably robust similarity function for image matching. Furthermore, the nesting property suggests an elegant flower like normalization strategy called a log-spiral difference. We show that this normalization enables a compact binary representation and is equivalent to a form a bottom up saliency. This suggests that the nested descriptor representational power is due to representing salient edges, which makes a fundamental connection between the saliency and local feature descriptor literature. In this dissertation, we introduce three examples of shape representation using nested descriptors: nested shape descriptors for imagery, nested motion descriptors for video and nested pooling for activities. We show evaluation results for these representations that demonstrate state-of-the-art performance for image matching, wide baseline stereo and activity recognition tasks.

Table of Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Related Work | 3 |
| 1.2 | Globally Local Shape Representations | 4 |
| 1.3 | Primary Contributions | 6 |
| 1.3.1 | Nested Shape Descriptors | 8 |
| 1.3.2 | Nested Motion Descriptors | 9 |
| 1.3.3 | Nested Pooling | 10 |
| 1.4 | Summary of Results | 12 |
| 2 | Related Work | 13 |
| 2.1 | Introduction | 14 |
| 2.1.1 | Scope | 14 |
| 2.1.2 | Outline | 16 |
| 2.2 | Local Shape Representations | 17 |
| 2.2.1 | Local Feature Descriptors | 18 |
| 2.2.2 | Local Motion Descriptors | 19 |
| 2.3 | Global Shape Representations | 21 |
| 2.3.1 | Graphical Representations of Shape | 21 |
| 2.3.2 | Quadratic Assignment Problem | 22 |
| 2.3.3 | Geometric Graph Representations | 24 |
| 2.3.4 | Topological Graph Representations | 27 |
| 2.4 | Analysis of Global Shape Representations | 41 |
| 2.4.1 | Task Independent Comparison | 44 |
| 2.4.2 | Is Shape Similarity Geometric? | 45 |
| 2.4.3 | Is Shape Similarity Topological? | 46 |
| 2.4.4 | Are We Revisiting the Classical Theory of Categorization? | 47 |
| 2.5 | Summary | 52 |
| 3 | Nested Shape Descriptors | 54 |
| 3.1 | Introduction | 54 |
| 3.2 | Related Work | 56 |
| 3.3 | Nested Shape Descriptors | 57 |
| 3.3.1 | Hawaiian Earrings and Nested Pooling | 59 |
| 3.3.2 | Nested Shape Descriptors | 61 |
| 3.3.3 | The Seed-of-Life Descriptor | 62 |
| 3.3.4 | Seed-of-Life Examples | 63 |
| 3.3.5 | Nesting Distance | 64 |

| | | |
|----------|---|------------|
| 3.3.6 | Proofs | 68 |
| 3.3.7 | Rotation Invariance | 71 |
| 3.4 | Experimental Results | 73 |
| 3.4.1 | Experimental System | 73 |
| 3.4.2 | Middlebury Stereo and Trade Study | 74 |
| 3.4.3 | VGG-Affine | 75 |
| 3.4.4 | Local Distance Functions | 77 |
| 3.4.5 | Photorealistic Virtual City | 78 |
| 3.4.6 | Storage Weighted Matching | 83 |
| 3.4.7 | Dense Strided Descriptors | 85 |
| 3.4.8 | Saliency | 86 |
| 3.5 | Summary | 91 |
| 4 | Nested Motion Descriptors | 93 |
| 4.1 | Introduction | 93 |
| 4.2 | Related Work | 95 |
| 4.3 | Nested Motion Descriptors | 96 |
| 4.3.1 | Overview | 97 |
| 4.3.2 | Complex Steerable Pyramid | 100 |
| 4.3.3 | Phase Gradients and Component Velocity | 102 |
| 4.3.4 | Robust Component Velocity | 105 |
| 4.3.5 | Robust Phase Pooling | 107 |
| 4.3.6 | Construction of the Nested Motion Descriptor | 108 |
| 4.3.7 | Invariance to Camera Motion | 110 |
| 4.3.8 | Motion Saliency | 112 |
| 4.4 | Experimental Results | 113 |
| 4.4.1 | Experimental System | 115 |
| 4.4.2 | Experimental Datasets | 116 |
| 4.4.3 | Motion Saliency | 118 |
| 4.4.4 | Activity Recognition | 120 |
| 4.5 | Summary | 123 |
| 5 | Nested Pooling | 125 |
| 5.1 | Introduction | 125 |
| 5.2 | Related Work | 128 |
| 5.3 | Nested Pooling | 129 |
| 5.4 | Penn Functional Scene Element (Penn-FSE) Dataset | 131 |
| 5.4.1 | Penn-FSE Dataset Examples | 133 |
| 5.5 | Experimental Results | 134 |
| 5.5.1 | Experimental System | 136 |
| 5.5.2 | Penn-FSE Results | 137 |
| 5.6 | Summary | 139 |
| 6 | Applications to Perception for Unmanned Aerial Systems | 141 |
| 6.1 | Shipboard Landing using Nested Descriptors | 141 |
| 6.1.1 | Deck Pose Estimation | 142 |
| 6.1.2 | Nested Shape Descriptors | 143 |
| 6.1.3 | Planar Homography | 144 |

| | | |
|----------|---|------------|
| 6.1.4 | Direct Linear Transform | 145 |
| 6.1.5 | Nested Shape Reprojection Error | 147 |
| 6.1.6 | Deck Pose Estimation Algorithm | 147 |
| 6.2 | Classification in Aerial Imagery using Nested Pooling | 150 |
| 7 | Conclusions | 153 |

List of Illustrations

| | | |
|-----|---|----|
| 1.1 | Master draughtsmen and the use of shape in art. (left) “Woman Covering Her Face with her Hand” by Matisse, (middle) “A Woman Sleeping” by Rembrandt, (right) “Three Studies of a Dancer” by Degas. | 2 |
| 1.2 | Summary of contributions of this thesis. | 6 |
| 1.3 | Thesis contributions for Nested Shape Descriptors | 8 |
| 1.4 | Thesis contributions for Nested Motion Descriptors | 10 |
| 1.5 | Thesis contributions for Nested Pooling | 11 |
| 2.1 | Wittgenstein’s Joke [1]. This highlights the need for interpretation to create a useful representation of a stimulus, or on this context, why not use a 3D representation of a duck/rabbit? | 14 |
| 2.2 | Taxonomy and comparison of local feature descriptors. | 19 |
| 2.3 | Examples of cycle and boundary groups [2] | 33 |
| 2.4 | Persistent Homology [3]. (top) Rips complexes \mathcal{R}_ϵ for increasing ϵ . Colors correspond to k -simplices. (bottom) “Barcode” representation of persistent homology groups for increasing ϵ . The vertical dotted lines correspond to the Rips complex at a specific ϵ , and the homology groups present with this representation. | 35 |
| 2.5 | Optimal Homologous Cycle Matching [4]. Given a reference cycle (red) for a given homology group, optimal homologous cycle matching finds the minimum weight homologous cycle (green). | 37 |
| 2.6 | Alignment does not capture all similarities | 47 |
| 3.1 | Nested shape descriptors pool scaled and oriented gradients over large geometric structures called <i>Hawaiian earrings</i> . (left) Hawaiian earrings with k -fold rotational symmetry define a member of the nested shape descriptor family called the <i>seed-of-life descriptor</i> (right) Two Hawaiian earrings substructures in the seed-of-life descriptor are highlighted in grey. | 55 |
| 3.2 | Taxonomy and comparison of local feature descriptors. | 57 |
| 3.3 | Why nesting? (left) Occlusions corrupt half of a generic grid descriptor covering the occluded region (red X’s), while the nesting distance selects the best subset of supports in the nested descriptor that cover only the object (green checkmarks). (middle) Viewpoint changes for long and thin foreground structures introduce errors in grid descriptor matching due to large changes in the background. The nesting distance selects the subset of supports during matching that cover the foreground and are the correct scale to allow for background variation. (right) Scale changes without scale invariant detectors introduce errors in grid descriptor matching due to changes in local support. The nesting distance uses a subset of both large and small scale supports, ignoring intermediate scale supports with corruption. | 58 |

| | | |
|------|---|----|
| 3.4 | (top) Logarithmic spiral property of the nested shape descriptor provides <i>normalization</i> and <i>binarization</i> . The log-spiral and it's reflection shown in grey form an elegant flower-like structure. (bottom) An NSD is formed at each interest point by (left) nested pooling of scaled and oriented gradients and (right) log-spiral difference and binarization. | 63 |
| 3.5 | Nesting property of the nested shape descriptor. (left) Seed of life, (middle) Hawaiian earring, (right) Cocentric nesting. | 64 |
| 3.6 | Logarithmic spiral property of the nested shape descriptor provides <i>normalization</i> and <i>binarization</i> . (right) The log-spiral and it's reflection shown in grey form an elegant flower-like structure. | 64 |
| 3.7 | Nested shape descriptors with increasing lobes. (top row) $\mathbb{K}_1 - \mathbb{K}_5$, (bottom row) $\mathbb{K}_6 - \mathbb{K}_{10}$ | 65 |
| 3.8 | Example stereo image matching using the nesting distance and nested shape descriptors. Colors encode corresponding interest points between the reference image (middle) and the observed image using the nesting distance (left) and Euclidean distance (right). The Euclidean distance is affected by occlusions at the image boundary (left ellipse) resulting in local misalignments, while the nested distance is more robust to these occlusion effects. | 66 |
| 3.9 | Construction of a nested shape descriptor. An NSD can be considered a "flattening" of the steerable pyramid. Supports of fixed sizes at different levels of the pyramid result in exponentially increasing descriptor supports. | 67 |
| 3.10 | Rotation invariant nested shape descriptor. (left) Relative orientations between lobe and gradient orientation are pooled over all lobes and (right) each lobe is summarized with a single pooled relative orientation value. | 72 |
| 3.11 | Rotation invariant nested shape descriptor. | 73 |
| 3.12 | Trade study for the seed-of-life descriptor. (left-right) parameter analysis of orientations, lobes, scales and pooling. | 74 |
| 3.13 | Example matching results from the VGG-Affine dataset.(top-bottom) wall, graf, boat, leuven, bikes, ubc | 76 |
| 3.14 | VGG-Affine image matching results. (top) "graf", "bikes", "ubc", "boat", (bottom) "wall", "trees", "leuven" and composite. Both SOL and BSOL outperform SIFT and BRISK, and Binary-SOL is the first binary descriptor to outperform SIFT on this benchmark. | 77 |
| 3.15 | Matching score for "bark" in the VGG-Affine dataset | 78 |
| 3.16 | Evaluation of the nesting distance on VGG-Affine dataset. See text for a discussion. | 79 |
| 3.17 | Example ground truth correspondence for rendered imagery in the PVC dataset. Shown are a random subset of five hundred pixels such that colors encode corresponding pixels between the right and left images. (top left) camera 1 with translation correspondence (bottom left) camera 2 with rotation correspondence (top right) camera 3 with translation and rotation correspondence (bottom left) camera 4 with translation and rotation correspondence. | 80 |
| 3.18 | Photorealistic Virtual City - Translation only results. (left) Aggregate (right) Time of day | 81 |
| 3.19 | Photorealistic Virtual City - Translation only results per location | 82 |
| 3.20 | Photorealistic Virtual City - Rotation only results. (left) Aggregate (right) Time of day | 83 |
| 3.21 | Photorealistic Virtual City - Rotation only results per location | 84 |

| | | |
|------|---|-----|
| 3.22 | Photorealistic Virtual City - Translation and Rotation results. (left) Aggregate, (right) Time of day | 85 |
| 3.23 | Photorealistic Virtual City - Rotation and Translation results per location | 86 |
| 3.24 | Photorealistic Virtual City - Storage Weighted Matching Results | 87 |
| 3.25 | Photorealistic Virtual City - Dense Stride. (top) Stride=32, (bottom) Stride=64. Colors encode matching densely extracted interest points for each stride. See text in section 3.4.7 for a discussion. | 88 |
| 3.26 | Dense Stride evaluation. See text in section 3.4.7 for a discussion. | 89 |
| 3.27 | Popout examples for NSD saliency map. (left to right) Convexity, orientation, contrast, orientation. (top) input image, (bottom) saliency map where red is high saliency and blue is low saliency, and maximum saliency shown with a black '+'. | 90 |
| 3.28 | Qualitative saliency results from the MIT Saliency Benchmark. | 91 |
| 3.29 | Qualitative saliency results from the MSRA salient object database. | 91 |
| 4.1 | Nested motion descriptors | 94 |
| 4.2 | From nested shape descriptors to nested motion descriptors. Nested shape descriptors pool oriented and scaled gradients magnitude which captures the contrast of an edge in an image. Nested motion descriptors pool <i>relative phase</i> which captures <i>translation</i> of an edge. Projecting the structure of the nested motion descriptor onto a single image ("collapsing" the descriptor) will form the structure of the nested shape descriptor. | 97 |
| 4.3 | Nested Motion Descriptors (NMD). (left) An input video is decomposed into a set of frames of length $2^k v$, where k is the number of scales in the pyramid decomposition and v is a fixed velocity tuning parameter. (middle) The relative magnitude and phase is computed for each orientation and scale subband in a steerable pyramid decomposition from the first frame to subsequent frames on a log-scale. Frames further away in time are represented with a large scale coarse motion, and frames close in time are represented with a small scale fine motion. Shown is the 0° orientation subband only. (right) For each dense interest point in the current frame t , we pool the robust component velocity derived from relative phase in a set of circular pooling regions all intersecting at the center interest point. Log-spiral normalization computes the difference between phases in neighboring scales and positions along a log-spiral curve. The phase pooling aggregates component velocities, so this difference computes an acceleration which is invariant to constant velocity of the camera. The result is a nested motion descriptor at this interest point that is invariant to camera motion. | 98 |
| 4.4 | Pyramid decomposition and reconstruction with the complex steerable pyramid. | 100 |
| 4.5 | An example of the magnitude and phase response of a complex filter to a translating 1D step edge signal. (left column, top to bottom) (a) step edge signal (b) impulse response of 1D quadrature filters (c) magnitude of complex filter response to step edge (d) phase and spatial phase gradient ($ \vec{\phi} $) of complex filter response showing linearity of phase. (right column, top to bottom) (e) A step edge translating left to right. (f) the magnitude response (g) the temporal phase gradient (ϕ_t). Observe that at the edge, the spatial phase gradient $ \vec{\phi} = 1$ and the temporal phase gradient is $\phi_t = \pm 2$, which measures a spatial shift of $\frac{\phi_t}{ \vec{\phi} } = \pm 2$ | 101 |

| | | |
|------|---|-----|
| 4.6 | Robust Phase Pooling. The temporal phase gradient is noisy due to the measurement of phase in regions where phase is unstable, such as the region on the grass and in the crowd. The phase stability measure provides an estimate of locations of stable phase. Only the stable phase is used for pooling, resulting in pooled phase that captures the motion of the background and foreground of the golfer in the scene. This pooled phase is used to construct the nested motion descriptor. | 105 |
| 4.7 | Perspective views of the spatiotemporal pooling regions of the nested motion descriptor. (left) $az=90^\circ$, $el=90^\circ$, the temporal axis is pointed into the page. We overlay the nested shape descriptor onto this view, which shows that the NMD has an equivalent pooling structure to the NSD (middle) $az=45^\circ$, $el=25^\circ$, with the temporal axis pointed into the page, (right) $az=90^\circ$, $el=0^\circ$, with the Y axis pointed out of the page. This view shows that the temporal pooling regions increase proportionally to spatial scale. The slope of the line connecting the centers is determined by the velocity tuning of the descriptor. A video visualization of this descriptor is available at http://youtu.be/RfJJHmXnRAw | 107 |
| 4.8 | (top) Logarithmic spiral property of the nested motion descriptor provides <i>normalization and binarization</i> . The log-spiral and it's reflection shown in grey form an elegant flower-like structure. (bottom) An NMD is formed at each interest point by (left) nested pooling of scaled and oriented gradients and (right) log-spiral difference and binarization. | 109 |
| 4.9 | The nested motion descriptor is invariant to global camera motion. (top) A video sequence of a rock climber where the camera is following the climber up the rock face. For a given fixed interest point on the background, we compute the nested motion descriptor. Observe that the robust component velocities for this interest points are the same. (bottom) When computing the log-spiral difference, the constant velocity due to the camera motion is removed, leaving only <i>acceleration</i> | 111 |
| 4.10 | The nested motion descriptor represents salient motion in video. We show a semi-transparent saliency map for motion overlayed on each frame of video. This saliency map shows salient responses in red and non-salient in blue. The salient responses show the foreground motion of the basketball dribbling and suppresses the motion of the camera in the background. | 113 |
| 4.11 | Examples frames from the six activity classes in the KTH actions dataset. | 115 |
| 4.12 | Examples frames from 14/51 activity classes in the Human Motion Database (HMDB). | 117 |
| 4.13 | Motion saliency for basketball dribbling. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization suppresses the camera motion and highlights the salient motion of the basketball dribbling in the scene. A video visualization of this motion saliency is available at http://youtu.be/t6D1c6M98aE | 118 |
| 4.14 | Motion saliency for KTH jogging. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization highlights the salient motion of the runners legs and arms, while the motion without log-spiral normalization saturates with the motion of the mean velocity of the body. A video visualization of this motion saliency is available at http://youtu.be/zzhos41j-QE | 119 |

| | | |
|------|---|-----|
| 4.15 | Motion saliency for HMDB rock climbing. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization suppresses the significant camera motion in the scene focusing on the salient motion of the rock climbers only. A video visualization of this motion saliency is available at http://youtu.be/MShHPal5KsU | 120 |
| 4.16 | Motion saliency for HMDB hug. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization focuses on the subtle hand movements that form a hug and suppresses the background motion of the camera. A video visualization of this motion saliency is available at http://youtu.be/yx1g9rvXvuQ | 121 |
| 4.17 | Activity recognition results on UCF sports actions. (top) Classification rate, (bottom) confusion matrices. The class index for each result: 'Diving-Side', 'Golf-Swing-Back', 'Golf-Swing-Front', 'Golf-Swing-Side', 'Kicking-Front', 'Kicking-Side', 'Lifting', 'Riding-Horse', 'Run-Side', 'SkateBoarding-Front', 'Swing-Bench', 'Swing-SideAngle', 'Walk-Front'. Class confusion results show that HOG-3D is inflating the classification results for 'kicking' by leveraging the background context of the football pitch, while the NMD is penalized for suppressing this background due to camera motion. See text for a discussion. | 122 |
| 4.18 | Activity classification results on KTH actions (top) Precision-recall curves for each of six activity classes (middle) average precision per class, mean classification rate, (bottom) confusion matrices. For all results, the class indexes are ordered: boxing, handclapping, handwaving, jogging, running, walking. NMD results are improved for boxing and handclapping, but worse for jogging. See text body for a discussion. | 124 |
| 5.1 | Recognition of human-scene interactions is the classification of functional scene elements such as bike racks, newspaper boxes or trashcans using only activities performed during usage. FSEs such as bike racks (1) and (2) may vary widely in appearance, but exhibit similar <i>weakly causal</i> usage patterns over time which can be used for classification. | 126 |
| 5.2 | Nested pooling is the encoding of an activity as a max pooled set of motion prototypes in a nested set of support regions centered at object usage event, where each support region inceases on a log scale. Each gray region is a pooling region which is represented by a histogram of prototype responses within the pooling region. Observe that the inner support regions are fully contained within the outer support regions, forming <i>nested pooling</i> | 129 |
| 5.3 | Why nesting for activity representation? Nested pooling preserves partial order for locally unordered action prototypes (helmet/lock) and unknown temporal scale variations (loiter/depart). | 130 |
| 5.4 | Penn Functional Scene Element (Penn-FSE) dataset. Shown are representative frames from of 12 of 24 videos in the full dataset. The dataset includes annotations for 177 functional scene elements and 463 usage annotations in over 8 hours of video. . . . | 131 |
| 5.5 | Data collection locations for the Penn Functional Scene Element dataset. The dataset contains 24 data collection sites in western Philadelphia. | 132 |
| 5.6 | Examples of functional scene element usages in Penn-FSE. (rows) bench, bikerack, crosswalk, door, newsbox | 134 |
| 5.7 | Examples of functional scene element usages in Penn-FSE. (rows) parking kiosk, road, sidewalk, subway, trashcan | 135 |

| | | |
|------|--|-----|
| 5.8 | Examples of variability of functional scene element usages in Penn-FSE. (rows) trashcan, trashcan, bench, subway, bike rack, parking kiosk, sidewalk | 136 |
| 5.9 | Summary results. (left) Confusion matrix for nested pooling with mean classification rate 0.45, (right) Precision recall for each class. Class indexes for confusion matrix are in the order of the legend in the precision-recall curve. | 138 |
| 5.10 | Precision recall and average precision on Penn-FSE dataset for nested pooling compared to baseline pyramid pooling and bagged pooling. (rowwise) Class [AP_{exp} , $AP_{baseline}$]: (1) bench [0.26,0.33], (2) bikerack [0.36,0.39], (3) crosswalk [0.61,0.28], (4) door [0.52,0.32], (5) newsbox [0.63,0.64], (6) parkingkiosk [0.98,0.66], (7) road [0.55,0.29], (8) sidewalk [0.32,0.25], (9) subway [0.53,0.66], (10) trashcan [0.27,0.23]. Mean classification rate over all classes shows that nesting provides a 22% improvement over bagging and 61% improvement over temporal pyramid. . . | 139 |
| 6.1 | Application of the nested shape descriptors to visual landing zone pose estimation. Colors encode the matching of the observed landing zone with the known markings (red=right, blue=left), and the green square encodes the detected position of the landing zone. Nested shape descriptors provide broadly selective scale matching without requiring scale invariant interest points. | 143 |
| 6.2 | LAIR precision-recall performance evaluation | 151 |
| 6.3 | LAIR precision-recall performance evaluation (continued) | 152 |
| 6.4 | LAIR performance evaluation summary. Shown are summary statistics and confusion matrix result for classification performance. The summary also includes annotations for the best and worst classes, and highlights those classes that are most often confused. | 152 |

Chapter 1

Introduction

Shape is arguably the most important property in visual perception [5]. Shape is the primary property used for visual categorization, and unlike other properties such as texture, color, motion or depth, shape alone can be used to predict other category properties. Central to the discussion of shape in a computational vision context, is the *shape representation*. The issue of representation is a fundamental problem in vision, leading many to argue that representation and generalization of shape is *the* problem in vision [6][7][1][8].

Artists have long known the power of shape for capturing visual form. Master draughtsmen such as Matisse, Rembrandt and Degas have captured such intangible qualities as strength, solitude or elegance in only a few well chosen strokes. Figure 1.1 shows some examples of the masters at work, where using only contours, they are capture the subtleties of the human form. Our perception of these qualities would not be any clearer by adding color to Matisse’s “Woman Covering Her Face With Her Hand”, or adding texture to Rembrandt’s “A Woman Sleeping”, or knowing the range to arms and legs of “Three Studies of a Dancer” by Degas. The artists knew that shape alone can capture these qualities, and that it is a powerful cue to inform our perception of the world.

What is a shape? Shape is an intuitive concept for most people since we perceive shapes every moment of the day. However, it is a difficult concept to define unambiguously, which has led to different operational definitions in different fields. For example, cognitive scientists define shape in terms of *objective shape* and *shape equivalence* [9]. Objective shape refers to the concept that objects in the world have a measurable volume, they have a surface boundary with a measurable surface orientation independent of any observer. Marr defines objective shape as “the geometry of an



Figure 1.1: Master draughtsmen and the use of shape in art. (left) “Woman Covering Her Face with her Hand” by Matisse, (middle) “A Woman Sleeping” by Rembrandt, (right) “Three Studies of a Dancer” by Degas.

object’s physical surface” [10] and Palmer observes “objective shape is no different in principle from the well established belief that each object has an objectively definable size, position, orientation” [5]. These objects have 3D extent that is the same for all observers, and therefore objectively definable independent of perception. Intuitively, one can think of the objective shape in terms of computer graphics, where the objective shape of an object is that which is captured by a 3D model, used to render an image independent of the camera viewing it. *Shape equivalence* refers to those shapes that are perceived to be the same object by observers. For example, viewing a 2D square that undergoes a translation is still perceived to be a square. The same holds for a scaling or small in-plane rotation. Observers viewing such objects that undergo transformations of translation, rotation or scaling perceive the same object following the transformation. The shape before the transformation is equivalent to the shape after. In general, shape equivalence refers to the variation due to the pose of the viewer relative to a fixed objective shape. A viewer can change position (translation) move towards or away from (scale) or tilt their head (rotation), and “the shape” perceived remains equivalent, even though the retinal image may change significantly.

Statisticians define shape in terms of a *statistical shape model* [11]. Different views of a shape are assumed to have common keypoints such that these *landmarks* can be put into correspondence. Given a set of shape images, a shape model is that which remains after optimally aligning landmarks, when the translation, rotation and scale effects are removed. Alignment proceeds using *Procrustes analysis*, which performs an optimal estimate of the aligning transform for a set of known landmarks to a reference coordinate system. This assumes correspondence of a known set of shape landmarks, which have been extracted from an image, such as points of high curvature. Then, once

landmarks are in correspondence, deformation models can be learned from the remaining (non-similarity) alignment errors. In computer vision, this type of analysis has led to active shape models [12][13] for shape based models of faces.

Geometry defines shape in terms of an equivalence class under a group of transformations. A *group* is a finite or infinite set of elements along with a binary group operation that satisfies closure, associativity, identity and inverse properties. In the context of shape, the group is the set \mathbb{R}^2 under similarity transformations. A similarity transformation is a group operator $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ such that for any set of points $p \in \mathbb{R}^2$, the points $T(p)$ are *similar* or related by a similarity. Formally, $\|p_i - p_j\| \Leftrightarrow \alpha \|T(p_i) - T(p_j)\|$ for all $p \in \mathbb{R}^2, \alpha \in \mathbb{R}$. In other words, for a similarity transform, angles are preserved and distances are preserved up to a scale factor.

Computer vision does not provide one definition of shape, rather the literature provides many task specific representations. Shape representations can be broadly organized into two main categories: template based and graph based. *Template based* approaches represent shape in terms of a fixed or deformable template, where a template is a fixed, relative spatial distribution of features [14, 15, 16, 17, 18, 19, 20, 21, 12, 22]. *Graph based* approaches represent shape in terms of graphs, where features or parts have a variable spatial distribution where conditional dependencies are organized in a graphical structure [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. Both template and graph based approaches represent the geometry of an object in terms of the relative position of features, however a template assumes a fixed relative position with small allowable deformations while a graph assumes a variable relative position of parts encoded by the graph.

1.1 Related Work

In chapter 2, we describe the related work on shape representations and shape matching. Shape representations can be broadly organized into two main categories: template based and graph based. *Template based* approaches represent shape in terms of a fixed or deformable template, where a template is a fixed, relative spatial distribution of features [14, 15, 16, 17, 18, 19, 20, 21, 12, 22]. *Graph based* approaches represent shape in terms of graphs, where features or parts have a variable spatial distribution organized in a graphical structure [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. Both template and graph based approaches represent the geometry of an object in terms of

the relative position of features, however a template assumes a fixed relative position with small allowable deformations while a graph assumes a variable relative position of parts encoded by the graph.

The dominant local shape representation is the *local feature descriptor*. In chapter 2 we survey the state of the art in local feature descriptors [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. We organize this survey by introducing a taxonomy for comparing and contrasting local feature descriptors in terms of five criteria: preprocessing, support, pooling, normalization and descriptor distance. Preprocessing refers to the filtering performed on the input image, support patterns are the geometric structure used for constructing the descriptor and pooling is the aggregation of filter responses over the support structure. We compare and contrast our contribution in the context of this related work.

The dominant global shape representation is *attributed graph matching*. In chapter 2, we survey the state-of-the-art for graph based shape representations. We organize the survey by grouping graph based representations into geometric methods and topological methods, where each method is organized by representations of increasing abstraction. First, we describe methods for constructing attributed graphs using local feature descriptors. Next, we describe geometric methods for graph matching. We use the unifying framework of weighted graph matching posed as relaxations of a quadratic assignment problem, and we describe invariant shape properties maintained during various tree, bipartite and general graph matching approximations. Topological methods for shape representation are less well established for image matching, but they provide the potential for global constraints such as interior, surrounded and connected to augment geometric representations. We use the unifying framework of simplicial homology, and describe the persistent homology, a technique for recovering the homology given noisy data.

1.2 Globally Local Shape Representations

The problem of shape representation can be further decomposed into sub-problems of representation, similarity and inference. Representation is the problem of abstraction of an image into *features* that capture the local or global properties of the image at each point in an image. Similarity is the problem of computing an affinity function, distance function or matching score for sets of candi-

date matching features, forming the cost function C . Finally, inference is the problem of selecting, searching or optimizing an optimal assignment A .

In general, this problem can be described as the shape representation and matching problem. Given two images I and I' and a set of pixel locations P and P' , an optimal shape matching A^* is the assignment function $A : P \rightarrow Q$ that minimizes a given cost function C

$$A^* = \arg \min_A C(Q, A(P)) \quad (1.1)$$

Shape representation and matching is a challenging problem due to the effects of occlusions, geometric scene variation, camera pose variation, scene illumination and articulated shape variation. Imagery collected from different viewpoints of a scene vary due to the structure of the scene. Specifically, occluded surfaces that are visible in one image and not visible in the other introduce pixels that have no matches. Changes in distance and camera orientation introduce scale and rotation variations in the imagery that must be addressed. Articulated objects introduce pose variations that change the appearance and shape between two images. These challenges must be addressed by a robust shape matching approach.

A shape representation can be global or local depending on the *image support* used in the representation. The image support is defined as the set of pixels used to construct the shape representation. A global shape representation is holistic and uses pixels sampled from the entire image to construct the representation. These global shape representations are typically graph based [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34] where shape is encoded in the adjacency structure of the graph. In contrast, a local shape representation is a representation that is localized at a specific position in an image, with finite small scale support. Local shape representations are universally template based, defined using properties of a local patch centered at an interest point. A near universal local shape representation is the *local feature descriptor* which captures the local distribution of oriented edges within a local patch [35, 36, 31, 37, 38, 39, 16, 40, 41, 42, 43, 44, 45, 46]. Global shape representations are typically defined in terms of local shape representations, where local shape defines the nodes in a graph as *parts* and the graph structure encodes the geometric relationships between these parts.

Local feature descriptors exhibit a fundamental tradeoff between selectivity and support. A

| Approach | Shape Representation | Benefits | Applications |
|----------------------------------|---|---|--|
| Nested shape descriptors | Binary 2D local feature descriptor | Binary, global, robust distance function, salient edges | Image matching, wide baseline stereo, dense matching, visual landing |
| Nested motion descriptors | Binary spatiotemporal local motion descriptor | Invariant to camera motion, salient motion | Activity recognition |
| Nested pooling | Pooling of motion prototypes in a nested set of support regions | Representation of weak causality | Functional scene element recognition |

Figure 1.2: Summary of contributions of this thesis.

local feature descriptor constructed using local support suffers from the aperture problem, where the small support of a local image patch does not contain enough unique identifying properties to provide an unambiguous match. To compensate for the aperture problem, the support of the local feature descriptor can be increased to provide additional information for an unambiguous representation, however this increased support introduces representational errors due to occlusions and pose variations that can corrupt the representation. An ideal local shape representation is ”just local enough”.

In this thesis, we focus on the problem of globally local or *glocal* shape representations. This shape representation is local such that it is centered at a single point in an image, however it is defined with support covering the entire image making it global. This thesis introduces the *nested descriptor* which is a globally local feature descriptor used to represent the globally local shape in an image. To address the selectivity and support tradeoff, we introduce a robust distance function which is able to reject outliers as the support size of the descriptor increases. Furthermore, when extending the nested descriptor to motion, we introduce a camera invariant representation that is able to maintain large support but it not corrupted by the effects of the global camera motion.

1.3 Primary Contributions

The primary contributions of this thesis are shape representations using *nested descriptors*. This thesis makes three primary contributions: nested shape descriptors, nested motion descriptors and

nested pooling. Using the decomposition for shape representation into problems of representation, similarity and inference described in section 1.2, this contribution for shape representation can be described as follows.

- **Shape representation in imagery.** *Nested shape descriptors.* We define a new globally local shape representation for images. We demonstrate that this new binary local feature descriptor captures *salient edges* in an image. We compare this to the state of the art in local feature descriptors for the task of image matching, wide baseline stereo matching, dense interest point matching and storage weighted matching. We show state of the art results. We describe this contribution in chapter 3.
- **Shape similarity in imagery.** *Nesting distance for nested shape descriptors.* This is a robust local distance function unique to the nested descriptors that outperforms the Euclidean distance for similarity computations. We show that this distance provides robust matching of a nested descriptor due to occlusions. We describe this contribution in chapter 3.
- **Shape representation in video.** *Nested motion descriptors.* We define a new globally local shape representation for video. This is an extension of the nested shape descriptors to video that is invariant to global camera motion. We evaluate this new representation for the task of activity recognition against the state of the art in local motion descriptors and show strong results. We describe this contribution in chapter 4.
- **Shape similarity in video.** *Phase correction for nested motion descriptors.* We show that a straightforward correction of local phase in a video can be used to remove the effect of the dominant camera motion from an interest point. This correction provides for a descriptor which captures the shape of the local foreground motion, so that the hamming distance of local motion descriptors captures the similarity of the foreground motion and not the effects of the camera motion. We describe this contribution in chapter 4.
- **Shape representation in video.** *Nested pooling.* We show that a bag of features representation for functional scene element recognition can be improved using a nested pooling strategy over a bagged or pyramid pooling strategy. We describe this contribution in chapter 5. However, the improvement is not uniform across all classes, so we conclude that this is a negative

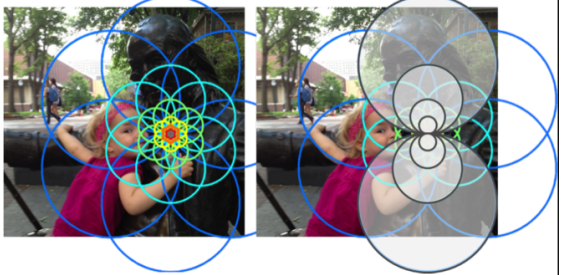
| | |
|---|--|
| <p><u>OBJECTIVE & BENEFITS</u></p> <ul style="list-style-type: none"> • A nested shape descriptor is a binary local feature descriptor for improved image matching • The benefits of the NSD <ul style="list-style-type: none"> • Binary representation → <i>Fast</i> • Local distance function → <i>Robust</i> • Global support → <i>High Performance</i> • NSD has a direct application to many early vision tasks <ul style="list-style-type: none"> • Stereo, optical flow, structure from motion, egomotion, visual mapping | <p><u>KEY INSIGHTS & CONTRIBUTIONS</u></p> <ul style="list-style-type: none"> • NSD is the first binary descriptor to outperform SIFT on standard image matching benchmark • NSD significantly outperforms state-of-the-art in binary descriptors • The nesting distance allows large support for selectivity without sacrificing performance due to occlusions. • The NSD provides a representation of <i>salient edges</i> using log-spiral normalization |
| <p><u>KEY CHALLENGES & STATE OF THE ART</u></p> <ul style="list-style-type: none"> • Shape representation using local feature descriptors dominate for image matching tasks • Researchers have generated a zoo of local feature descriptors of varying shape and speed • Local feature descriptors exhibit a fundamental tradeoff of invariance vs. selectivity <ul style="list-style-type: none"> • Large support for selectivity • Small support for invariance • An ideal descriptor is both large scale and robust to occlusions, geometric variation and scale |  <p><i>A nested shape descriptor is a new binary local feature descriptor that is fast, robust and accurate</i></p> |

Figure 1.3: Thesis contributions for Nested Shape Descriptors

result for further investigation.

The primary contributions of this thesis are summarized in figure 1.2. We explore each contribution in turn.

1.3.1 Nested Shape Descriptors

In this thesis, we propose a new family of binary local feature descriptors called nested shape descriptors. These descriptors are constructed by pooling oriented gradients over a large geometric structure called the Hawaiian earring, which is constructed with a nested correlation structure that enables a new robust local distance function called the nesting distance. This distance function is unique to the nested descriptor and provides robustness to outliers from order statistics. In this paper, we define the nested shape descriptor family and introduce a specific member called the seed-of-life descriptor. We perform a trade study to determine optimal descriptor parameters for the task of image matching. Finally, we evaluate performance compared to state-of-the-art local feature descriptors on the VGG-Affine image matching benchmark, showing significant performance gains.

Our descriptor is the first binary descriptor to outperform SIFT on this benchmark.

In chapter 3, we introduce the nested shape descriptor and nesting distance using key concepts of nested pooling and log spiral normalization. We perform a trade study to determine optimal descriptor parameters for the task of image matching. Finally, we evaluate performance compared to state-of-the-art local feature descriptors on the VGG-Affine image matching benchmark and Photo-realistic Virtual City dataset, showing significant performance gains.

The key contributions of the nested shape descriptors are summarized in figure 1.3. The NSD is the first binary descriptor to outperform SIFT on a standard image matching benchmark. NSD significantly outperforms the state of the art in binary descriptors. The nesting distance is the first robust local distance function. Finally, the nesting distance allows for large support without sacrificing performance due to occlusions.

Finally, we motivate the structure of the nested shape descriptor in terms of bottom up saliency. We show that the nested shape descriptor and the log spiral normalization is as representation of salient edges in an image. We hypothesize that this salient edge representation provides a significant performance improvement for shape representation.

1.3.2 Nested Motion Descriptors

In this thesis, we propose a new family of binary local motion descriptors called nested motion descriptors. This descriptor provides a representation of *salient motion* that is invariant to global camera motion, without requiring an explicit optical flow estimate. The key new idea underlying this descriptor is that appropriate sampling of scaled and oriented gradients in the complex steerable pyramid exhibits *phase offset* due to camera motion. This phase offset can be measured in the complex steerable pyramid, then removed using the log-spiral normalization. This correction provides invariance to camera motion without an explicit estimate of optical flow. This approach is inspired by phase constancy [47], component velocity [48] and motion without movement [49, 50], which uses phase shifts as a correction for translation without an explicit motion field estimate. Finally, the phase corrected video is used to construct a nested motion descriptor using the approach for nested shape descriptors introduced in [51]. The nested shape descriptor is a state-of-the-art binary local feature descriptor, which we extend to representation of motion in video. This descriptor uses log-spiral normalization to represent salient edges, therefore the nested motion descriptor represents

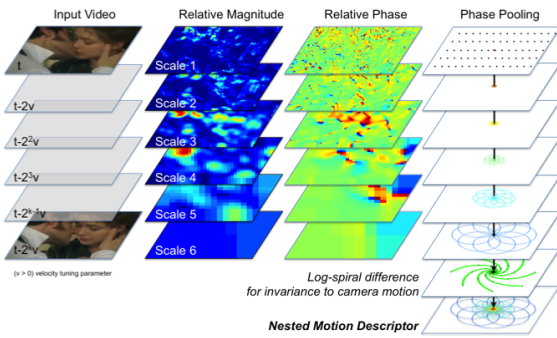
| | |
|--|--|
| <p>OBJECTIVE & BENEFITS</p> <ul style="list-style-type: none"> • Local motion descriptor for activity recognition • The benefits of the NMD <ul style="list-style-type: none"> • Binary representation \rightarrow <i>Fast</i> • Global support \rightarrow <i>Selective</i> • Salient \rightarrow <i>High performance</i> • Does not require an explicit optical flow estimation | <p>KEY INSIGHTS & CONTRIBUTIONS</p> <ul style="list-style-type: none"> • Nested motion descriptors are an extension of nested shape descriptors to video • Invariant to global camera motion by performing phase correction independently across scales <ul style="list-style-type: none"> • Inspired by “motion without movement” and phase based optical flow constraints • Representation of <i>salient motion</i> using log-spiral normalization |
| <p>KEY CHALLENGES & STATE OF THE ART</p> <ul style="list-style-type: none"> • “Activity recognition in the wild” is affected by global camera motion • Global camera motion introduces motion at every pixel that corrupts the representation of activity. • Recent work has focused on motion descriptors based on <i>dense trajectories</i> to track pixels and represent relative motion along a trajectory • These rely on optical flow which is a challenging early vision problem that introduces artifacts and over-smoothing. |  |

Figure 1.4: Thesis contributions for Nested Motion Descriptors

salient motion. Figure 1.4 summarizes the contributions of this descriptor.

In chapter 4, we define the nested motion descriptor family and we evaluate performance compared to state-of-the-art local motion descriptors on the the KTH actions, UCF sports actions and HMDB activity recognition datasets.

1.3.3 Nested Pooling

In this thesis, we describe a new pooling strategy for representation of functional scene elements called *nested pooling*. Bag-of-words based representations of activities rely on spatiotemporal pooling regions to perform max-pooling of learned prototypes to construct prototype histograms based representation of an activity. Nested pooling represents an activity as a bag-of-words model, however instead of pooling over a uniform region as in traditional bag of words models [52], or spatial pyramid based pooling as in spatial pyramid matching [15], the pooling regions are *nested*. This representation is inspired by a general class of local feature descriptors called *nested shape descriptors* [51]. We show that this nested pooling is well suited for modeling weakly causal activities

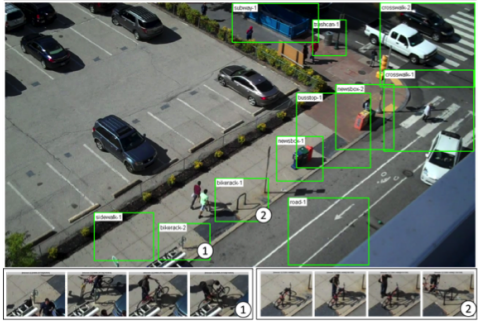
| | |
|---|--|
| <p><u>OBJECTIVE & BENEFITS</u></p> <ul style="list-style-type: none"> • Recognition of functional scene elements and functional buildings in complex human-scene interactions <ul style="list-style-type: none"> – bike racks, newspaper boxes or trashcans in urban video from usages by pedestrians. • Functional recognition will complement appearance based recognition to provide semantic labeling for recognition of functional objects in the world model. | <p><u>KEY INSIGHTS & CONTRIBUTIONS</u></p> <ul style="list-style-type: none"> • Spatiotemporal feature based representation of human activities that captures “weak causality” or partially ordered activities during object usage without expensive inference. • New contributions: <ul style="list-style-type: none"> – Nested pooling to represent weak causality over large temporal scales |
| <p><u>KEY CHALLENGES & STATE OF THE ART</u></p> <ul style="list-style-type: none"> • Current approaches to activity recognition <ul style="list-style-type: none"> – Max-pooled spatiotemporal features – Activity Templates – Graphical models • Current approaches need a middle ground between <ul style="list-style-type: none"> – Fast recognition using bags of spatiotemporal features without causality – Slow inference of graphical models that represent causality |  <p><i>Nested pooling is a new representation for weakly causal activities</i></p> |

Figure 1.5: Thesis contributions for Nested Pooling

commonly found with functional scene elements. This approach can be considered a middle ground in single level representations of human activities [53] between spatiotemporal feature based representations which ignore causality [54, 55] and sequence or graphical model based activity representations [56, 57] which represent causality by computationally expensive optimization of sequence alignments or probabilistic inference of optimal activity states. Nested pooling combines the best properties of these two approaches, which enables a representation of *weak* causality while maintaining the fast exemplar based recognition of unordered representations. Figure 1.5 summarizes the contributions of the nested pooling.

Chapter 5 introduces the nested nested pooling. In this chapter, we describe nested pooling structure and show results on the newly curated Penn Functional Scene Element (Penn-FSE) dataset. We show cross validation results on Penn-FSE over ten functional scene elements, and we justify the benefit of the nesting property by showing a 22% improvement in mean classification rate relative to non-causal bagged and pyramid representations.

1.4 Summary of Results

In this section, we summarize the results of our analysis from this work. These conclusions are derived from studies and performance evaluation performed for each primary contribution on existing benchmarks and new datasets.

- The nested shape descriptor is the first binary descriptor to outperform SIFT on the VGG-affine benchmark.
- The nested shape descriptor outperforms DAISY and other local feature descriptors on wide baseline matching benchmark. This sets a new performance standard for wide baseline matching.
- The nested shape descriptor significantly outperforms all other descriptors when considering a storage weighted matching metric, comparing matching performance as a function of storage requirements. This new metric sets a new performance standard for local feature descriptors.
- The nesting distance is the first provably robust local distance function to be proposed for local feature descriptors.
- The nesting distance outperforms the Euclidean distance for the task of image matching.
- Nested motion descriptors are the first motion descriptor that does not require an explicit optical flow solution.
- Nested motion descriptors are the first globally local motion descriptors which provide invariance to global camera motion.
- Nested motion descriptors outperform HOG-HOF and HOG-3D on standard activity recognition datasets.
- Nested pooling provides a 22% improvement over gridded or pyramid based descriptors for the task of scene element recognition.

Chapter 2

Related Work

In this chapter, we survey shape representations in computer vision. In the past fifteen years, there has been significant progress in shape representations in the literature, as described in surveys [58][59][60][39, 61]. In this review, we decompose shape representations into those representing local shape vs. global shape, and describe each in detail. For local shape, we focus on local feature descriptors in imagery and video, and provide a taxonomy of these descriptors for comparing and contrasting design choices. For global shape, we focus on graph based shape representations which are well suited for modelling part based compositions of objects. We group graph based representations into geometric methods and topological methods, where each method is organized by representations of increasing abstraction. For geometric methods, we use the unifying framework of weighted graph matching posed as relaxations of a quadratic assignment problem, and we describe invariant shape properties maintained during various tree, bipartite and general graph matching approximations. For topological methods, we use the unifying framework of simplicial homology, and describe the persistent homology, a technique for recovering the homology given noisy data, and optimal homologous cycle matching for matching topologically invariant cycles. Finally, we perform analysis of this survey by comparing and contrast these representations both in a task independent analysis based on representational power and computational tractability.

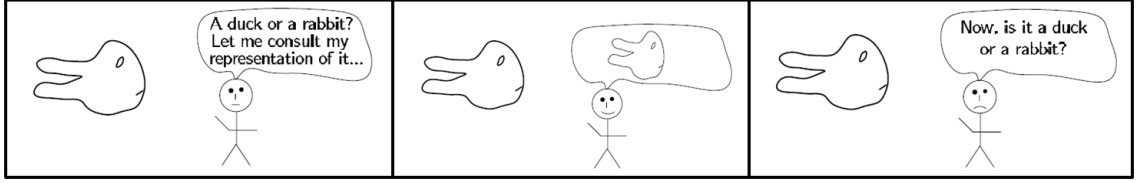


Figure 2.1: Wittgenstein’s Joke [1]. This highlights the need for interpretation to create a useful representation of a stimulus, or on this context, why not use a 3D representation of a duck/rabbit?

2.1 Introduction

2.1.1 Scope

What is the scope of this review? Our goal is to perform a survey of local and global shape representations, however this goal is still quite a broad. Are we interested in 2D or 3D shape? Shape for a specific task such as object detection? Local shape representations in terms of invariant local features? Global representations, part based or category structure? Are we interested in shape registration, matching or shape similarity algorithms, and data structures associated with efficient solutions? How about classification and learning of shape within a category? What about other non-shape or non-visually grounded features such as texture, motion and color? What about the shape of a scene?

We limit the scope of the review to the following design choices:

Graph based global methods: We focus the global shape representations on graph based representations rather than template based representations. The line between these two representations is often indistinct, as many graph based presentations use local templates for representation of parts, and some deformable template representations can be modelled as a graph such as a Markov random field. Where appropriate, we discuss the overlap between these broad categorizations.

Graph matching based global methods: We focus on graph matching, also called registration, or alignment correspondence, rather than graph similarity. Graph matching provides alignment between a reference model and an observation such that edge relations (e.g. geometry or topology) are preserved. This matching may include matching constraints such as one-to-one, many-to-one or many to many matching to reflect abstract representations of shape [8]. Recent work on *graph kernels* [62] for structured prediction compute the similarity of two general graphs by considering

the similarity of polynomial comparable substructures such as trees, cycles, walks or paths. These approaches are motivated by the intractability of subgraph isomorphism for general graphs (see section 2.3.3), and the need for an efficient computation of similarity of two graphs suitable for use in a kernel machine for discriminative classification. These approaches do not provide correspondence and therefore do not represent global geometric shape during matching. These approaches do consider local geometric shape in terms of matching substructures, but the final similarity is analogous to a “bag of substructures” representation which is not global. Therefore, we focus on geometric and topological matching rather than non-geometric comparisons.

Exemplar similarity: We focus on similarity computations to exemplars [6][7][1][63], rather than category models. This scope is appropriate for applications to shape matching. We will argue in section 2.4.4 that any fixed category model, even category models learned at training time, is classical categorization revisited. Therefore, since the classical theory of categorization has been widely discredited, a category model of shape stands on a questionable foundation. As a result, we will not consider graph based shape representations for categories such as recursive compositional models [28, 64], hierarchical generative models [65][66, 52, 67], hierarchical object parsing [21, 68], stochastic image grammars [26], composition systems [27, 24]. This effectively descopes the entire literature of generative models for visual category representation. Note that this does not mean that categorization cannot be performed, it simply states that categorization that relies on a fixed category model is questionable.

2D: We focus on 2D shape or view based representations rather than representation of 3D object shape. A large body of literature exists on “shape from X” recovery of 3D shape from imagery. Edelman [1] traces the history of 3D reconstruction as a shape representation, back to the influential work of Marr [10] and the 2.5D sketch. The motivation for 3D representations was to provide *object constancy* or viewpoint invariance for specific object identification across views, by first reconstructing the 3D geometry from multiple views. However, Edelman argues that 3D shape representations still suffer from drawbacks namely: (i) the difficulty of recovering correspondence, (ii) the need for task specific representations and (iii) the fact that 3D representations do not aid in further higher level processing such as categorization or detection. As observed by many researchers, a 3D representation still needs to be interpreted. Figure 2.1 shows an old joke due to Wittgenstein. A 3D

representation requires a “homonculus” or little man sitting inside our vision box looking at the 3D representation for interpretation [1]. In short, a 3D reconstruction still requires *similarity* to an internal labeled 3D representation, on top of the challenge of 3D reconstruction itself. Therefore, we focus on 2D representations and the challenge of similarity.

shape matching: Optimal shape representations are task dependent [8][1]. We focus on the representation for the task of shape matching. This task of matching an instance of an object in two images, can be contrasted with related tasks. Object detection is the problem of assigning a bounding box at a position and scale that surrounds all each instance of a given object category in an image. Object classification is the labelling of images that contain at least one instance of a given object category without localization. Object recognition [69], also called object identification or specific object recognition is object classification for a known unique object instead of an object category. Object segmentation is the grouping and labelling of pixels corresponding to foreground and background for each instance of a given object category in an image.

Basic level categories: We focus on shape representations of basic level categories [70, 71, 8]. Basic level categories are the highest level category for which members have similar shape, and does not include such categories as functional objects or scenes. There have been surveys for shape representations of popular object categories including faces [72] and human body [73], however a general shape representation for basic level categories should not be specialized for any one category.

2.1.2 Outline

There are many different ways of organizing and categorizing the literature on shape representations, such that each focuses on a different unifying theme. For example, we could focus on shape matching, comparing and contrasting algorithmic complexity and data structures associated with different frameworks. We could focus shape representations only, comparing features and parts extracted from imagery used to compose shapes. We could compare and contrast graph based methods with template based methods or texture based methods.

The structure of this review is representations of shape of increasing abstraction and invariance. We start with *local shape representations* in the form of local feature descriptors which capture only the local properties of shape in terms of distributions of oriented and scaled gradients in a local

patch centered at a single point in an image. We then broaden the representation to global shape representations based on graphs, and we group the problem into two major categories of graph based shape representations: geometric and topological graph based representations. *Geometric graph based representations* represent shape in terms of distances and angles between parts, such that nodes encode parts and edges encode geometric relations. This approach can be organized in terms of the underlying graph structure. We describe methods based on trees, bipartite graphs and general graphs, where the underlying graphical structure allows for more expressive representations of shape, at the cost of more expensive matching. *Topological representations* represent shape in terms of topological invariants such as connectivity or holes in a graph. These representations are less well evaluated for shape representations in computer vision, so we survey the computational topology literature and describe approaches based on persistent homology and homologous cycle matching that have the potential to provide invariant features for shape representation.

In this review, section 2.2 describes the local feature descriptors used to capture the local shape in both imagery and video. Section 2.3.1 describes global shape representations in terms of graphs and describes common components to geometric and topological shape representations. Section 2.3.3 describes three graphical structures: trees, bipartite graphs and general graphs and the graph constructions and graph matchings used for representation and detection. Section 2.3.4 describes topology based representations which represent shape in terms of homologies. Finally, we compare and contrast the different categories and draw conclusions for about the representations in sections 2.4 - 2.5.

2.2 Local Shape Representations

A *local shape representation* is a representation that is localized at a specific position in an image, with finite small scale support. For example, a local shape representation of an eye in a face would be constructed using the support of a small image patch surrounding the eye. Local shape representations are universally template based, defined using properties of a local patch centered at an interest point. A common design choice is the *local feature descriptor* which captures the local distribution of oriented edges within a local patch [35, 36, 31, 37, 38, 39, 16, 40, 41, 42, 43, 44, 45, 46]. Global shape representations are typically defined in terms of local shape representations, where

local shape defines the nodes in a graph as *parts* and the graph structure encodes the geometric relationships between these parts.

In this section, we describe a taxonomy of local feature descriptors for representing the local edge properties of a patch and the local motion properties of a video clip.

2.2.1 Local Feature Descriptors

A local feature descriptor is a representation for local 2D shape by pooling and normalizing oriented gradients over specific support regions. These local feature descriptors are commonly used to provide node attributes for use in graph matching. In this section, we describe the state-of-the-art.

Local feature descriptors have emerged in the past ten years as the dominant representation for shape matching. There exist standard benchmarks for performance evaluation [39, 74, 61], and a zoo of detectors and descriptors [38, 16, 37, 40, 42, 43, 44, 45, 46]. introduced with the trend of faster and faster matching while maintaining approximately equivalent performance to SIFT [35]. Local feature descriptors have been successfully deployed for a wide range of shape matching tasks including: stereo, optical flow, structure from motion, egomotion estimation, tracking, geolocation and mapping.

There have been many local feature descriptors proposed in the literature in the past ten years. From oldest to newest, the primary developments have been: SIFT [35], PCA-SIFT [36], Shape context [31], Local Binary Patterns [37], SURF [38], GLOH [39], Sparse localized features (SLF) [16], compressed HoG (cHoG) [40], DAISY [42], BRISK [43], BRIEF [44], ORB [45] and FREAK [46].

The trend in local feature descriptor research has been to show comparable performance to SIFT on the VGG-affine benchmark [39, 74, 61], with ever faster computation. Work has progressed from PCA-SIFT [36] and SURF [38] which show close performance to SIFT with lower dimensionality and faster preprocessing. Recent work has focused on introducing binary features from local comparison tests [44, 43, 45, 46] which enables fast distance metric based on Hamming distance and faster derivatives [75]. These developments have been driven by the need for faster processing to support mobile deployment.

A taxonomy for comparing and contrasting local feature descriptors can be described in terms of five criteria: preprocessing, support, pooling, normalization and descriptor distance. Preprocessing

| Descriptor | Preprocessing | Support | Pooling | Normalization | Distance |
|---------------|----------------------------|-----------------------|---|--------------------------|---------------------------|
| SIFT | Oriented gradients | Cartesian grid | Histogram | Truncated norm | L2 |
| SURF | Integral image | Cartesian grid | Histogram | Truncated norm | L2 |
| PCA-SIFT | Oriented gradients | Cartesian grid | histogram, PCA | Truncated norm | L2 |
| Shape Context | Edge detection | Log-polar grid | Sum | Global norm | Bipartite matching |
| GLOH | Oriented gradients | Log-polar grid | Histogram, PCA | PCA | L2 |
| SLF | Laplacian pyramid | Cartesian grid | Max | Global norm | L2 |
| cHoG | Oriented gradients | Trees | Histogram | Global norm | L2 |
| DAISY | Convolved orientation maps | Overlapping log-polar | Patch sampling | Support norm | L2 |
| BRIEF | Gaussian filter | Log-polar patch | Gaussian sampled binary comparisons | None | Hamming |
| BRISK | Gaussian filter | Log-polar patch | Deterministically sampled binary comparison | None | Hamming |
| ORB | Gaussian filter | Log-polar patch | Gaussian sampled binary comparisons | None | Hamming |
| FREAK | Gaussian filter | Log-polar patch | Retinally sampled binary comparisons | None | Hamming |
| NSD | Steerable pyramid | Nested log-polar | Nested pooling | Log-spiral normalization | Hamming, Nesting distance |

Figure 2.2: Taxonomy and comparison of local feature descriptors.

refers to the filtering performed on the input image, support patterns are the geometric structure used for constructing the descriptor and pooling is the aggregation of filter responses over the support structure. Figure 2.2 shows this taxonomy and a comparison of dominant local feature descriptors.

2.2.2 Local Motion Descriptors

Activity recognition has a long history in the computer vision literature. Recent surveys of action recognition capturing the state of the art are available [53, 76] and a critical review of action recognition benchmarks [77]. Classic activity recognition datasets [78] focused on tens of actions collected with a static camera of actors performing scripted activities, however the state-of-the-art has moved to recognition of hundreds of activities captured with moving cameras and poor quality video of "activities in the wild" [79][80][81].

The literature on motion representation can be decomposed into approaches focused on local motion descriptors, mid-level motion descriptors or global activity descriptors. Higher level motion representations are typically focused on representing semantic activity categories, and learning

mid-level representations suitable for action recognition. Examples include discriminative mid level features [82], actemes [83], motionlets [84], motion atoms and phrases[85]. In general, these higher level representations build upon local motion representations to extract activity specific discriminative motion patterns. In this section, we will focus on local motion representations only, which are most relevant to the nested motion descriptor.

A *local motion descriptor* is a representation of the local movement in a scene centered at a single interest point in a video. Examples of local motion descriptors include HOG-HOF [86, 87], cuboid [88], extended SURF [89] and HOG-3D [90]. These descriptors construct spatiotemporal oriented gradient histograms over small spatial and temporal support, typically limited to tens of pixels spatially, and a few frames temporally. HOG-HOF includes a histogram of optical flow [86, 87], computed over a similar sized spatiotemporal support. Furthermore, recent evaluations have shown that activity recognition performance is significantly improved by considering dense regular sampling of descriptors [91][92], rather than sparse extraction at detected interest points, such as spatio-temporal interest points (STIP) [54].

An interesting recent development has been the development of local motion descriptors that are invariant to dominant camera motion. A translating, rotating or zooming camera introduces global pixel motion that is irrelevant to the motion of the foreground object. Research has observed that this camera motion introduces a global translation, divergence or curl into the optical flow field [93], and removing the effect of this global motion significantly improves the representation of foreground motion for activity recognition. The motion boundary histogram [86, 94, 95] computes a global motion field from optical flow, then computes local histograms of derivatives of the flow field. This representation is sensitive to local changes in the flow field, and insensitive to global flow. Motion interchange patterns [96, 97, 98] compute a patch based local correspondence to recover the motion of a pixel, followed by a trinary representation of the relative motion of neighboring patches. First order differential motion patterns [93] compute ... Finally, dense trajectories [94, 95, 99] concatenate HOG-HOF or co-occurrence HOG [100], and motion boundary histograms for a tracked sequence of interest points forming a long term trajectory descriptor. The improved dense trajectories [99] with fisher vector encoding is the current state-of-the-art on large datasets for action recognition [101].

2.3 Global Shape Representations

2.3.1 Graphical Representations of Shape

Graphical representations of shape refer to the abstraction of an image into an *attributed graph*, such that the image encodes a *graph embedding*. A grayscale image I may be defined as a function $I : \mathbb{Z}^2 \rightarrow \mathbb{R}$, such that I is a mapping between integer valued pixel coordinates (i, j) and pixel intensity $I(i, j)$. A graph embedding in an image is a graph $G = (V, E)$ such that each node is associated with a pixel coordinate (i, j) . An attributed graph is a graph $G = (V, E, \alpha)$ that has been augmented with a set of node and edge attributes $\alpha_V(v)$ $\alpha_E(u, v)$ for all $v \in V, (u, v) \in E$. These attributes encode local image properties from the graph embedding. Recent work has considered attributes for higher order simplexes in hypergraphs [102], however in this section we consider pairwise node and edge attributes only, and postpone the discussion of higher order simplex attributes to section 2.3.4.

Graph based shape representations for can be described in terms of attributes, structure, construction and matching. Attributes refer to those local image properties associated with each node and edge as determined from the graph embedding. Node attributes may be organized in order of increasing support in an image, centered at the embedding coordinates. *Pixel support* refers to attributes such as pixel intensity, oriented gradient filter response or corner response at only the embedding coordinate. *Patch support* refers to attributes derived from a fixed, local region of interest centered at the embedding coordinate, such as a grayscale patch, intensity histogram or oriented gradient histogram. *Region or contour support* refers to attributes derived from perceptual organization of an image, such as segmentations or boundary detections. Finally, *part support* refers to attributes derived from part responses, such that a part is itself an object with shape and is used in compositional models to compose shape in terms of simpler component shapes.

Edge attributes capture pairwise geometric or topological relationships between nodes. For example, geometric edge attributes may include length between two nodes, the angle between a node and a reference orientation, or the scale normalized length between two nodes. As discussed in section 2.1.1:similarity, we consider only attributed graphs in this review. Other methods such as probabilistic graphical models enable efficient inference techniques by encoding *conditional dependence* in edge relations, however as discussed we will not consider these cases. We will revisit this issue in section 2.4.

Graph construction is the process of graph embedding and attribute extraction, and graphs may be explicitly or implicitly constructed. Explicit construction is the computation of a graph from an image, prior to any processing or matching. Implicit construction postpones the graph constructing to the matching phase. Implicit construction follows Marr’s principle of least commitment [10], such that graph embedding decisions for nodes and edges are delayed until there is further information from the matching process, rather than committing to mistaken embedding and corrupting the match. Recent work in graph matching has transitioned from explicit to implicit graph construction, and we will use this as a comparison criterion in section 2.4.

Graph matching is the problem of finding correspondences between two graphs such that relational structure is preserved. We will discuss this problem in context of the unifying framework of the *quadratic assignment problem* in section 2.3.2 following standard definitions of graphs and graph properties in [103].

2.3.2 Quadratic Assignment Problem

Graph matching is the problem of finding correspondences between two graphs such that relational structure is preserved. This is a fundamental problem in computer vision, machine learning and pattern recognition since structured data is widespread in such forms as part based object recognition, structured prediction, and shape representations. For recent surveys of graph matching, see [104, 105]. In general, graph matching can be posed as a weighted graph matching problem, with special cases of subgraph isomorphism and maximum common subgraph.

Weighted graph matching can be posed as follows. Given two attributed graphs $G = (V, E, \alpha)$, $G' = (V', E', \alpha')$, let X be an $|V| \times |V'|$ permutation matrix, such that $X(i, i') = 1$ if nodes (i, i') are matched and zero otherwise. Let W be an $|V||V'| \times |V||V'|$ weight matrix determined from attributes (α, α') such that $w_{ii', jj'} \in \mathbb{R}$ encodes the compatibility of matching (i, i') and (j, j') . Let x be an $|V||V'| \times 1$ columnwise vector representation of X such that $x_{ij} = X(i, j)$ and x_i^T is the i th row and x_j is the j^{th} column. Then,

$$\begin{aligned}
x_{QAP}^* = \arg \max \quad & x^T W x \\
\text{s.t. } \forall (i, j) \quad & \mathbb{1}^T x_j = m \\
& x_i^T \mathbb{1} = n \\
& x_{ij} \in \{0, 1\}
\end{aligned} \tag{2.1}$$

The constraints $\mathbb{1}^T x_j = m$ and $x_i^T \mathbb{1} = n$ are *mapping constraints*. Let $\mathbb{1}$ be a vector of ones, then $\mathbb{1}^T x_j$ is a column sum for columns x_j of X , and $x_i^T \mathbb{1}$ is a row sum of X . If $m = n = 1$, then the mapping constraints are *one-to-one* such that each node in G must be mapped to exactly one node in G . If $n \geq 1$ and $m = 1$, then the mapping constraints are *many-to-one* for many nodes in G' to one node in G . Similarly, if $n \geq 1$ and $m \geq 1$ then the mapping constraints are *many-to-many*. These mapping constraints enable the graph matching to encode constraints such as isomorphism or homomorphism, and allows the graph matching to be robust to imperfect graph construction.

The optimization in (2.1) is an instance of a *quadratic assignment problem*, such that x_{QAP}^* is a maximum weight edge preserving matching where $(u, v) \in E \Leftrightarrow (X(u), X(v)) \in E'$ [106]. The quadratic assignment problem is a classic problem in combinatorial optimization that can be motivated as a *facilities localization* problem. Consider the problem of assigning a given set of facilities to locations, where there are costs for a given assignment due to the cost of the flow of goods between facilities and the costs of assigning a facility at a given location. The goal is to determine an optimal assignment given these costs. Formally, given N facilities and M locations, let A be an $N \times N$ matrix defining the weight between facilities, and B be an $M \times M$ matrix for weights between locations. Solve for an assignment x that minimizes (2.1) such that the cost of assigning facility i to location j is W_{ij} .

The optimization in (2.1) is an integer quadratic program which is NP-complete, so approximate solutions are necessary. Approximation algorithms that have been explored in the literature include combinatorial search [106], graduated assignment [107], spectral [108, 109], semidefinite programming [110], and graph edit distance [111]. These approaches require a construction of the quadratic objective weights W which is quadratic in the size $|V||V'|$. Robust performance has been demonstrated [107, 109], but practical problem sizes are limited to hundreds of nodes due to the quadratic objective, and weights are limited to pairwise interactions.

The optimization in (2.1) can be made efficient for constrained graph structures. For example,

if the graph G is bipartite, then the quadratic assignment problem reduces to the *linear assignment problem* for which polynomial time integer solutions are available [31]. Similarly, if the graph G is a tree, then there exist tree edit distance algorithms [112] based on dynamic programming [113][114].

2.3.3 Geometric Graph Representations

Graph representations model an object or object category using a *graph*, such that alignment is isomorphic to *graph matching*. In general, graph matching approaches can be described in terms of the structure being preserved. Given two attributed graphs $G = (V, E, \alpha)$, $G' = (V', E', \alpha')$ a structure preserving matching $f: V \rightarrow V'$ is an optimal solution $f^* = \operatorname{argmin} c(f)$, subject to structure preserving matching constraints and assignment costs c . Exact structure preserving methods, such as subgraph isomorphism, maximum common subgraph and weighted graph matching preserve edge relations, such that $(u, v) \in E \Leftrightarrow (f(u), f(v)) \in E'$. In contrast, inexact graph matching, such as graph edit distance problem [111, 115] require approximate solutions.

Recent work in the vision literature has focused on geometry preserving linearizations [30, 116, 117] of the quadratic assignment problem in (2.1). Similarity invariant matching [116] solves for the optimal permutation matrix X and linearized similarity transformation parameters θ to minimize an assignment cost and an $L1$ -norm linear deformation cost. Locally affine invariant matching [117] solves for the optimal assignment X given $L1$ -norm barycentric coordinate preservation costs for each node, where barycentric coordinates are locally affine invariant and defined in terms of neighboring graph nodes. Both approaches use an $L1$ -norm in the objective, and exhibit linear constraints ([116] includes a linearization of the similarity constraints) resulting in a linear programming relaxation.

These geometric approaches provide fast and efficient matching, but they can suffer from ambiguity when the input graph does not satisfy the assumptions of the geometric transform model, such as cases of non-similarity transformations or degenerate triangulations. Furthermore, assignment weights are limited to node assignment weights only, ignoring informative assignment weights for edges and other higher order structures [102]. Finally, these methods must discretize X to a final binary permutation matrix for valid and invalid matches. Poor geometric alignments with large deformation costs may still be valid (e.g. articulated objects), and good geometric alignments with small deformations may be invalid. These issues will be revisited in the topological methods in

section 2.3.4.

Approaches to graph matching can be compared using graph construction and graph topology. Graph construction refers to the approach used to create an attributed graph that represents an image, including constructing nodes and edges and assigning attributes. Graph topology refers to the structure of a graph, such as trees, bipartite or general graphs. The graph structure is closely related to the efficiency and optimality of the matching, so there are tradeoffs between the fidelity of the representation vs. the optimality of the matching. In this section, we will describe three different graph structures.

2.3.3.1 Trees

Trees provide a useful abstraction to provide *part based representations* of shape. A part based representation of shape decomposes an object into a discrete set of component subshapes or *parts* that are configured to create an object. Parts may be large and sparse such as the decomposition of a face into semantic parts such as eyes, nose and mouth. Alternatively, parts may be small and dense such as contour fragments composed into a holistic shape representation. However, in all part based representations, local parts are *composed* provide a structural decomposition of a global representation of shape into a configuration of local parts. A graph $G = (V, E)$ is a tree if it is connected and acyclic. The nodes of a tree represent parts, the edges of the tree represent a composition of discrete and independent parts into a whole.

The motivation for a part based representation is invariance, compositionality, reuse, and computational tractability [8][5]. By decomposing an object into a set of parts that can be recomposed into multiple different objects in different configurations, parts can be reused to represent many object shapes. Similarly, by decomposing parts into a tree based or hierarchical representation or hierarchy of parts, the tree structure can enable efficient matching by taking advantage of the acyclic graphical structure.

Part based representations can be described in terms of part representations, structural representations and detection framework. Part representations capture the local appearance or geometry of a small subset of an object such that each part captures some local property of an object. Part representations may be appearance based or contour based, local or global, invariant (affine, rotation) or non-invariant (patches). Structural representations describe the tree structure, which may be flat

as in the case of constellation or star trees, or hierarchical with multiple levels of part interactions. Finally, the detection framework considers how the part representations are detected in an image, localized and composed into an object. Optimization approaches include dynamic programming, belief propagation, generalized hough transform and graph matching.

2.3.3.2 Bipartite Graphs

A graph $G = (V, E)$ is a bipartite graph if and only if the vertex set V can be partitioned into two disjoint subsets V_1 and V_2 such that $V = V_1 \cup V_2$ and no edge in the edge set E has endpoints in the same subset [103]. Bipartite graphs are more general than the star graphs described in section 2.3.3.1, since a star graph is a special case of a complete bipartite graph. To see this, observe that the star graph nodes can be partitioned into two subsets containing the central node and the leaf nodes forming a bipartite graph, and that every pair (v_i, v_j) such that $v_i \in V_1$ and $v_j \in V_2$ has an associated edge $e = (v_i, v_j) \in E$, forming a complete bipartite graph.

Bipartite graphs are useful for representing one to one matching. A perfect matching M for a bipartite graph G is a subset of edges ($M \subset E$) such that each vertex is incident to at most one edge of M . A minimum weight perfect matching is a perfect matching of minimum cost where the cost of a matching is given by $c(M) = \sum_{(i,j) \in M} c_{ij}$. The minimum weight perfect matching problem can be posed as a linear assignment problem which is formulated as an integer linear program as follows

$$\begin{aligned}
x_{LAP}^* &= \arg \max \quad Wx \\
\text{s.t.} \quad &\mathbf{1}^T x_j = 1 \\
&x_i^T \mathbf{1} = 1 \\
&x_{ij} \in \{0, 1\}
\end{aligned} \tag{2.2}$$

It can be shown that the constraint matrix A in equation (2.2) capturing the matching constraints (after dropping the integer constraints) is totally unimodular, so the integer linear program has an efficient integer solution. Minimum weight perfect matching can also be solved efficiently by using special purpose algorithms such as the Hungarian algorithm or reducing to a maximum network flow problem [118].

Shape contexts [31] are a representative approach to shape representations that uses bipartite matching. Shape contexts use a descriptor to capture the local shape, and bipartite graph matching

to align a reference shape with an observation. Complete bipartite graphs are constructed by first performing edge detection in an image, then subsampling the resulting edge map. The subsampled locations provide a graph embedding for V_1 , and the attributes of each node are a shape context descriptor. This descriptor counts the number of nearby edges using a log-polar histogram centered at the embedding coordinate and concatenates this log-polar histogram into a vector representation d_i . The bipartition V_2 is a reference shape, and a complete bipartite graph is constructed with edges between image nodes V_1 and reference nodes V_2 . Minimum weight perfect matching is performed by setting the assignment weights W according to kernel weight between shape context descriptors $w_{ij} = K(d_i, d_j)$, where a common kernel is the Gaussian kernel.

Since shape contexts using bipartite matching, there are no edges between nodes within V_1 and V_2 . This means that the matching is performed on nodes only. It is assumed that the geometry is encoded in the shape context descriptors, since each log-polar histogram has large support often covering a large fraction of the object. However, since there are no explicit edges to preserve in the matching, this approach cannot guarantee preserving geometric relationships. In practice, bipartite matching has been subsumed by many to one matching on general graphs.

2.3.3.3 General Graphs

A general graph is an arbitrary graph $G = (V, E)$ without additional constraints on edges or nodes. Unlike the graphs considered in previous sections for which the special graph structure enabled efficient matching algorithms, graph matching in general is NP-hard problem as it is isomorphic to the quadratic assignment problem outlined in section 2.3.2.

2.3.4 Topological Graph Representations

2.3.4.1 Computational Topology

Computational topology is the study of the algorithmic questions in topology and topological questions in algorithms [119][120]. For example, topological problems are those about invariant properties of connectivity and continuity, without requiring spatial notions such as straightness, distance or convexity. A topological question may ask about the number of “holes” in a topological space, allowing deformations but not cutting or gluing, whereas an algorithmic question in topology may

be the computation of the number of holes in a discrete representation of this space. Similarly, a topological question in algorithms may be, given a discrete representation of a topological space, does it preserve the topology of the underlying continuous space? Computational topology grew out of the desire to extend discrete results in computational geometry for point sets, polygons and polyhedra in to continuous domains, curved surfaces and higher dimensions. The success of computational geometry and the interesting overlap of computer science and topological questions hold promise of furthering both fields.

However, a challenge for collaboration between computer science and topology is the lack of common language. Motivated computer scientists without appropriate training in topology may find the topological literature unapproachable. Topology has an occasionally complex notation, and requires a significant number of definitions and accumulated theory to be grasped to understand the literature, and motivated readers may be unsure if the effort to learn this common language will be worthwhile.

Fortunately, recent work has demonstrated the power of computational topology to justify this learning curve. Topological concepts have been applied to a wide range of application areas including shape acquisition for solid modelling using computer aided design, shape representations for interoperability, portability and simplification in computer graphics, mesh generation for physical simulation and finite element analysis, configuration spaces in robotics, molecular biology, computer vision and databases. These application areas all rely on topological algorithms, and collaboration between topologists and computer scientists have analyzed the computational complexity. Typically, planar topological problems are polynomially solvable, problems in \mathbb{R}^3 are exponentially solvable and are thought to be *NP*-complete, while problems in \mathbb{R}^4 and above are known to be undecidable [120]. Such analysis and the wide range of applications, many in high dimensional spaces, highlights the need for further investigation into efficient algorithms for computational topology.

In this section, we summarize a survey on computational topology [119]. This survey focuses on applications and methods, and provides an introductory set of definitions in topology to facilitate the language barrier between topology and computer science. Next, we provide a more in depth review of one application of computational topology to the qualitative analysis of the space of natural images in computer vision [121][3]. This approach uses *persistent homology* to characterize the topology of the space of 3x3 patches in natural images. This is an example application of

computational topology which addresses an approach to suitably preprocess noisy data to enable topological analysis of global structure.

The survey by Dey, Edelsbrunner and Guha [119] provides three main contributions: (i) a description with motivating examples of six application areas for computational topology including image processing, cartography, computer graphics, solid modelling, mesh generation and molecular modelling, (ii) a description of six topological methods and (iii) an appendix with technical definitions to aid the reader without a background in topology. In this section, we focus on intuitive descriptions of the topological applications and methods, then provide an in depth description of one application in section 2.3.4.3.

Methods. Topological methods are theoretical topics in topology, which in this survey paper focus on: decompositions, fixed points, embedding, three-manifolds, and homology computation. Given that definitions and theorems are often their own best summaries, in this section, we will focus on the main ideas for each method rather than reiterating the definitions, theorems and proofs themselves.

Decompositions are the process of decomposing a shape into simple pieces. A classic result for complexes defined by the boundary of polyhedra states that $v - e + f = 2$, where v , e and f are the number of vertices, edges and faces in the polyhedron. More generally, the Euler characteristic of a space is the alternating simplex count, which is determined by counting vertices, edges and triangles in a suitably oriented triangulation. The construction of a decomposition requires considering the cover of a topological space, which is a collection of subsets whose union is the space, and the nerve which is cover with non-empty common intersection. Such theorems have motivated the design of automatic triangulation algorithms for a topological space. Shelling is the process of constructing a complex by adding one cell at a time, and if a complex is shellable, then shelling can be used in such applications as computing the convex hull of a set of n points in $O(\log(n))$ per face, with an initial $O(n^2)$ preprocessing.

Fixed points are those points of a function where the point is its own image. A classical result on contracting maps is that they have a unique fixed point. Continuous Brouwer's fixed point theorem is one of the most basic facts about topological spaces and generalizes the metric fixed point theorem stating that every map $f : \sigma^d \rightarrow \sigma^d$ has a fixed point, where σ^d is a d -simplex homeomorphic to \mathbb{B}^d . This theorem can be used to show the existence of a fixed point on a simplicial map, which is useful

for such applications as finding centerpoints and equipartitions.

An *embedding* of one topological space into another is an injection whose restriction to the image is a homeomorphism. A classic result states that every abstract simplicial complex \mathcal{A} has a special embedding in \mathbb{R}^d called a geometric realization, provided d is large enough, and is always possible if $d = 2k + 1$. For \mathbb{R}^2 , a well studied problem is the embedding of planar graphs, which can always be embedded in \mathbb{R}^3 , but can only be embedded in \mathbb{R}^2 if and only if it does not contain a subgraph homeomorphic to K_5 (complete graph with 5 vertices) or $K_{(3,3)}$ (complete bipartite graph with 6 vertices). Also, a classic result for \mathbb{R}^2 for more general spaces is that a 2-manifold can be embedded in \mathbb{R}^2 iff it is orientable (e.g. projective plane, Klein bottle)

Three-Manifolds are topological spaces that is locally Euclidean (\mathbb{E}^3), and many specialized results exist for 3-manifolds that do not generalize to higher dimensions. Unfortunately, this survey shows its age in that the Poincaré conjecture is described as an open problem, which has since been solved (proved 2002-2003, confirmed in 2006). A *knot* is an embedding of a closed curve in space $K : S^1 \rightarrow \mathbb{R}^3$. A remarkable result due to Seifert states that every knot is the boundary of an orientable 2-manifold embedded in \mathbb{R}^3 .

Homology computation offers a formal algebraic framework for studying and counting holes in a topological space by computing homology groups. Formally, these holes are characterized by Betti numbers, and since Betti numbers are invariant to triangulation, computing the Betti numbers of a simplicial complex is equivalent to computing the Betti number of the underlying space. Betti numbers can be computed for higher dimensional simplices by computing smith normal forms, using incremental algorithms or combinatorial laplacians, or using special cases of solids.

2.3.4.2 Simplicial Homology

In this section, we first provide a brief introduction to those definitions and results in simplicial homology necessary for describing the topological approaches in later sections. For detailed discussion of homology and algebraic topology, see [122, 123, 2].

We begin with general definitions. Practical vision applications are typically limited to dimension ≤ 3 , however we introduce general definitions for completeness.

Definition 2.3.1. A p -dimensional simplex or p -simplex is the convex hull of $p + 1$ affinely inde-

pendent vertices $v \in \mathbb{R}^D$

Definition 2.3.2. A *face* of a p -simplex σ is a non-empty subset of vertices of σ .

Definition 2.3.3. A simplicial complex K is a set of simplices that satisfies the following closure conditions

- (i) Any face of a simplex in K is a simplex in K
- (ii) The intersection of any two simplexes $\sigma_i, \sigma_j \in K$ is a face of both σ_i, σ_j

Let K be a finite simplicial complex of dimension p , such that all simplexes $\sigma \in K$ have dimension at most p . A simplicial k -chain is a finite formal sum of k -simplices

$$\sum_{i=1}^N c_i \sigma_i, \quad c_i \in \mathbb{Z}_2, \quad \sigma_i \in K \quad (2.3)$$

where $c_i \in \{-1, 0, 1\}$ are binary valued coefficients. For each $k \geq 0$, k -chains along with the modulo-2 addition operator form the *chain group* $C_k(K)$.

Definition 2.3.4. The boundary operator $\partial_k: C_k \rightarrow C_{k-1}$ is a homomorphism between chain groups such that

$$\partial_k(\sigma) = \sum_{i=0}^k (-1)^i \langle v^0, \dots, v^{i-1}, \hat{v}^i, v^{i+1}, \dots, v^k \rangle \quad (2.4)$$

The notation \hat{v}^i denotes that the vertex should be dropped. The boundary homomorphism is a linear operator and commutes with addition $\partial_k(c_1 + c_2) = \partial_k(c_1) + \partial_k(c_2) \forall c_1, c_2 \in C_k(K)$. Observe that $\partial_k(\sigma) = \sum c_i \partial_k(\sigma_i)$, and the boundary homomorphism is a map from k -simplices to a sum of its $(k+1)$ faces.

The boundary homomorphism has a unique matrix representation with respect to a choice of basis. Let $\{\sigma_i\}$ and $\{\tau_j\}$ be the sets of k -simplices and $(k-1)$ -simplices of size $|M-1|$ and $|N-1|$ that represent the elementary chain bases for C_k and C_{k-1} . Then, ∂_k is represented as an $M \times N$ boundary matrix with entries $a_{ij} \in \{-1, 0, 1\}$, such that $|a_{ij}| = 1$ if $i \in \tau$ is a face of $j \in \sigma$ with sign determined from (2.4), and zero otherwise. Some authors distinguish the boundary operator ∂_k with the matrix form $[\partial_k]$, however in this review, we assume that the context will make this distinction clear.

Definition 2.3.5. The chain complex

$$C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} C_{k-2} \dots C_0 \xrightarrow{\partial_0} 0$$

is a sequence of chain groups connected by boundary homomorphisms.

The boundary homomorphism has several useful properties, which we state but do not prove [122].

Lemma 2.3.6. Given a boundary homomorphism ∂ ,

- (i) The boundary of a boundary is zero, $\partial_k \partial_{k+1} d = 0$, for every integer k and every $(k+1)$ -chain d .
- (ii) A k -cycle c is a k -chain with zero boundary $\partial_k c = 0$
- (iii) The boundary of every 0-simplex is zero.
- (iv) The cycle group $Z_k = \ker(\partial_k) = \{x \in C_k(K) : \partial_k x = 0\}$
- (v) The boundary group $B_k = \text{im}(\partial_{k+1}) = \{x \in C_k(K) : \exists y \text{ s.t. } x = \partial_{k+1} y\}$.

Elements of the cycle group Z_k are k -chains called k -cycles, elements of the boundary group B_k are k -chains which called k -boundaries, which are boundaries of a $(k+1)$ -chain and are also cycles ($B_k \subset Z_k$).

Definition 2.3.7. An *homology class* is an equivalence class of cycles such that for a fixed representative cycle z_0 , $\{z | z = z_0 + \partial_{k+1} c, c \in C_{k+1}(K)\}$, where equivalent cycles of the same homology class are *homologous* and denoted $c \sim c'$.

Definition 2.3.8. The *homology group* $H_k(K) = Z_k/B_k$ is a quotient group formed on the set of homology classes.

Definition 2.3.9. The k th betti number $\beta_k = \text{rank}(H_k(K)) = \text{rank}(L_k)$ where $L_k = \partial_k^T \partial_k + \partial_{k+1} \partial_{k+1}^T$ is the rank of the k th homology group or equivalently the rank of the k th combinatorial laplacian L_k [124].

Definition 2.3.10. A chain map $M_k : K \rightarrow K'$ is a homomorphism mapping k -simplexes of simplicial

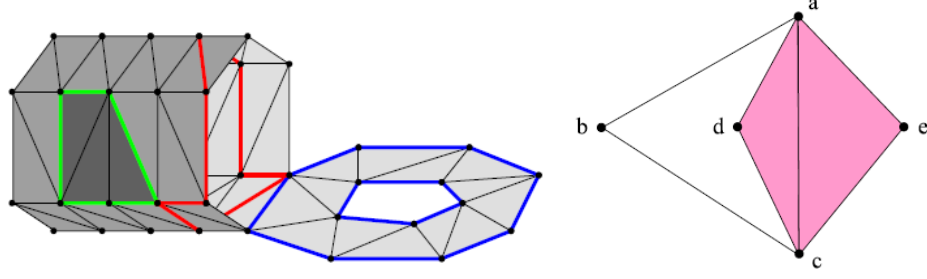


Figure 2.3: Examples of cycle and boundary groups [2]

complexes K and K' . The chain map must satisfy boundary commutativity.

$$\partial'_k \circ M_k = M_{k-1} \circ \partial_k$$

This requirement follows from the requirement $\partial_k \circ \partial_{k+1} = 0$. A chain map between chain complexes maps boundaries to boundaries and cycles to cycles, and induces homomorphisms between homology groups of the two complexes [122].

The definitions for simplicial homology were introduced in general for k -simplexes, however these concepts have intuitive low dimensional interpretations in the context of graph theory. Given a graph $G = (V, E)$, a 0-simplex is a vertex in V , a 1-simplex is an edge in E , a 2-simplex is a triangle which forms a three node clique, a 3-simplex as a tetrahedron or four node clique, and so on. The faces of a edge (1-simplex) are the two incident vertex endpoints (0-simplexes), and trivially the edge itself. The faces of a triangle (2-simplex) are the triangle itself (trivially), three edges (1-simplexes) and three nodes (0-simplexes). The simplicial complex closure constraints in (defn 2.3.3) states intuitively that if two edges are incident on a common vertex, then both edges must contain the vertex as a face. A 1-chain (2.3) is any subset of edges, not necessarily connected. The boundary map ∂_1 (2.4) is the oriented node-edge incidence matrix, and the boundary map ∂_2 is the edge-triangle incidence matrix. For node-edge incidence only, the combinatorial laplacian $L = \partial_1 \partial_1^T = D - W$, which is the classic graph laplacian for unit weights W . Betti numbers β_1 (defn 2.3.3) capture the number of “holes” in the graph, and the homology group H_1 contains equivalence classes such that each homology class is the set of all cycles that differ by a boundary from a cycle surrounding this hole. Finally, traditional graph matching is an estimation of \hat{M}_0 , the permutation matrix between graph nodes (0-simplexes) that preserves edges (1-simplex intersections).

Homology groups (defn 2.3.8) are the key concept in this section, so let's deconstruct it further and give a simple example to build intuition. Figure 2.3 (left) shows a simplicial complex of maximum dimension $k = 2$ that has the shape of a cylinder with a lid. The green edges (c_g) form the boundary of the three 2-simplexes (c) shown as dark grey triangles. Formally, ($c_g = \partial_2 c$) and observe that the “boundary” of the three dark triangles is what one would expect, it is formed from the non-overlapping faces of the triangles. The two blue cycles are homologous since they differ by a boundary, but are not homologous to the red or green cycles.

Figure 2.3 (right) shows a different (smaller) simplicial complex for which we can construct and example to explicitly compute homologous cycles. This simplicial complex contains five 0-simplexes (a, b, c, d, e), seven 1-simplexes ($ab, ac, ad, ae, bc, cd, ce$) and two 2-simplexes (acd, ace) shown in pink. The boundary matrices are constructed using defn 2.3.4 and are given by:

$$\partial_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad \partial_2 = \begin{bmatrix} 0 & 0 \\ -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \quad (2.5)$$

Entries $\partial_1(i, j)$ correspond to whether the i^{th} 0-simplex shares a face with the j^{th} 1-simplex. For example, $\partial_1(1, 1) = 1$ since the edge ab shares a face with point a , $\partial_2(3, 2) = 0$ since triangle ace does not share a face with ad . Observe that the cycle c_{abc} defined by the faces (ab, bc, ac) is represented by a 1-chain $c_{abc} = [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T$ and $\partial_1 c_{abc} = 0$ which satisfies definition 2.3.6 for a 1-cycle. Remember that all addition is modulo-2. An homologous cycle $c_{abcd} \sim c_{abc}$ surrounds the same hole, such that the homologous cycle c_{abcd} differs by a boundary from a representative cycle c_{abc} . In other words, there exists a chain $c \in C_2(K)$ such that

$$c_{abcd} = c_{abc} + \partial_2 c \quad (2.6)$$

Where the boundary of c is $\partial_2 c$ and this boundary is added (modulo-2) to the representative cycle

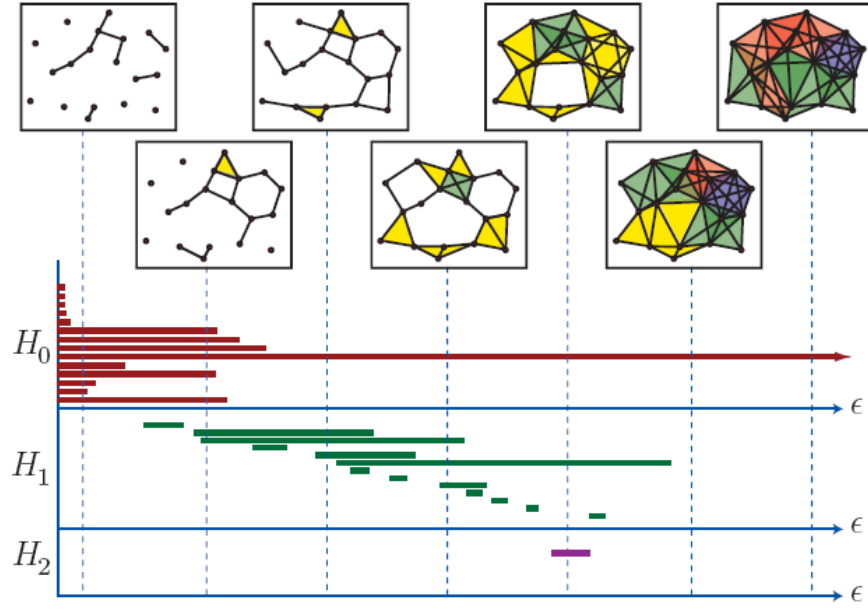


Figure 2.4: Persistent Homology [3]. (top) Rips complexes \mathcal{R}_ϵ for increasing ϵ . Colors correspond to k -simplices. (bottom) “Barcode” representation of persistent homology groups for increasing ϵ . The vertical dotted lines correspond to the Rips complex at a specific ϵ , and the homology groups present with this representation.

to result in the homologous cycle. In this example, $c = [1 \ 0]^T$ is a 2-chain that contains only the 2- simplex acd . The boundary $\partial_2 c = [0 \ 1 \ -1 \ 0 \ 0 \ 1 \ 0]^T$ is the set of faces $\{ac, da, cd\}$, and the modulo-2 addition $c_{abc} = [1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T + [0 \ 1 \ -1 \ 0 \ 0 \ 1 \ 0]^T = [1 \ 0 \ -1 \ 0 \ 1 \ 1 \ 0]^T = c_{abcd}$. Notice that the addition of the cycle and boundary cancelled out the face shared by both, namely ac .

2.3.4.3 Persistent Homology

Persistent homology is a tool from applied algebraic topology that characterizes the homology groups that persist over scale variations for noisy data. Understanding the global properties of a topological space includes characterizing its homeomorphism type, which involves understanding the geometry of the space up to stretching and bending, but not tearing and gluing. Determining the homeomorphism type is difficult, so an alternative is to characterize the *homology* of a discrete approximation of the space, then computing its simplicial homology groups. Informally, the homology of a topological space provides knowledge of number and type of holes in the space, such that the characterization of these holes provides a description of the global topological structure.

A fundamental challenge of computing topological features of real datasets is to determine which features are real (“signal”), which which features are artifacts of the discrete representation (“noise”). For example, given a collection of points x in Euclidean space \mathbb{E}^n , the *Rips complex* \mathcal{R}_ϵ [125] is the abstract simplicial complex whose k -simplices correspond to unordered $(k + 1)$ -tuples of points that are pairwise within distance ϵ . Figure 2.4 (top) shows an example of the Rips complex for a set of points in \mathbb{E}^2 for increasing ϵ . For this abstract simplicial complex, homology generators H_j can be computed, such that the rank of H_0 reflects the number of connected components, the rank of H_1 reflects the number of one dimensional holes and so on. Which is the “right” ϵ for this dataset? In other words, as ϵ increases, holes appear and disappear, which begs the question which holes are “real”? In general, without prior knowledge of the dataset generation, the true ϵ cannot be determined. However, observe that those topological features which persist across large changes in scale ϵ are unlikely to be due to topological noise. So, informally, let the persistent homology be the homology that persists across large changes in ϵ .

The persistent homology is computed as follows. Figure 2.4 (bottom) shows an example for a representation of the persistent homology called a *barcode* [3]. Informally, a barcode can be considered to be the persistence analogue of a Betti number. Recall that a Betti number captures the rank of a homology group, however a homology group is dependent on the abstract simplicial complex, which is dependent on a scale parameter. As the scale increases, some Betti numbers decrease and others increase, which correspond to the changing topology of the abstract simplicial complex. A barcode captures the changes in Betti numbers as a scale or persistence is changed. Figure 2.4 (bottom) shows a scale parameter ϵ as horizontal lines, the homology groups are shown ordered vertically, and the rank of $H_k(\mathcal{R}_{\epsilon_i})$ for the k^{th} homology group H_k for a given Rips complex \mathcal{R}_{ϵ_i} is the number of intervals intersecting the dotted lines. The rank of the homology group H_k is equal to the k^{th} Betti number β_k of the complex, which is a quantitative measure of the global topological structure. For example, β_0 is the number of connected components of the complex. At the second dotted line, the Rips complex contains six connected components, but as ϵ increases at the third dotted line, the Rips complex becomes fully connected. So, the rank of H_0 changes from $\beta_0 = 6$ to $\beta_0 = 1$, as shown by the intersection of the dotted lines with the red intervals. Similarly, at the fifth dotted line, the Rips complex is connected and there is only one hole remaining, so $\beta_0 = 1$ and $\beta_1 = 1$ as shown by the intersection of the dotted line with the red and green intervals.

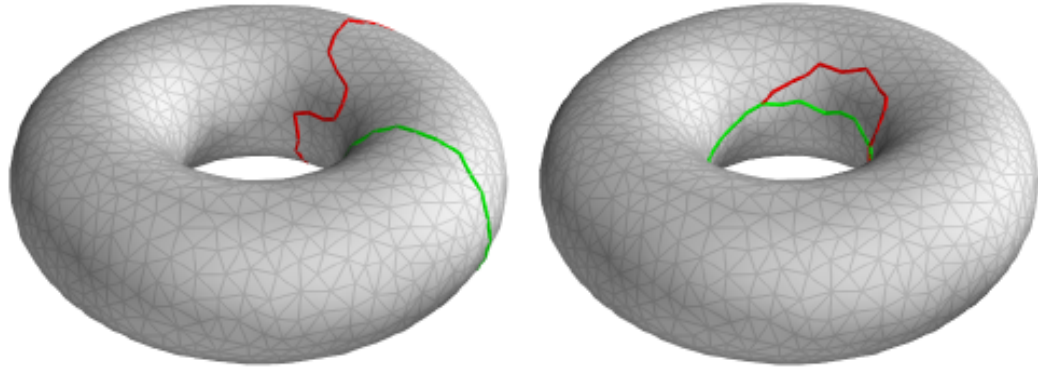


Figure 2.5: Optimal Homologous Cycle Matching [4]. Given a reference cycle (red) for a given homology group, optimal homologous cycle matching finds the minimum weight homologous cycle (green).

One application of persistent homology is the topological analysis of the space of natural images [121][3]. In [121], the authors extended the approach of [126] to characterize the topological space \mathcal{M} of projected 3×3 patches of natural images onto the seven dimensional unit sphere. Understanding the statistics of natural images is motivated by the need to model prior probability distributions of local image patches for use in object recognition, object localization, segmentation, denoising, and compression. For example, in [126] the authors model the full probability distribution of high contrast 3×3 binary patches from natural images, by projecting suitably preprocessed 3×3 patches onto \mathcal{S}^7 , a 7 dimensional unit sphere in \mathbb{R}^8 . Results show that the projected distribution of patches in \mathcal{S}^7 are strongly clustered, and that a majority of the patches are concentrated in submanifolds in the unit sphere. This result shows that natural images are composed of basic image primitives which generate low dimensional non-linear structures where intrinsic dimension of these manifolds is fixed and independent of the embedding space dimension. Such analysis focuses on the probabilistic and geometric model of basic image primitives in natural images. Results of this analysis over a dense subset of points in $\tilde{\mathcal{M}}$ showed that this topological space has $\beta_1 = 5$.

2.3.4.4 Homologous Cycle Matching

Optimal homologous cycle matching is the problem of finding the shortest cycle in the same homology class as a given cycle. Intuitively, in a 2D graph, given a cycle surrounding a hole in the graph, the optimal homologous cycle is the shortest cycle that surrounds the same hole. The pri-

mary result of this paper is that finding optimal cycles in a given homology class can be solved in polynomial time since it can be shown that the integer linear program for homology cycle matching has totally unimodular constraints, and therefore the linear programming formulation provides an integer solution.

In the context of shape matching, homologous cycles and chains have the potential to elegantly capture global matching constraints. These constraints include surrounded, interior, connected which cannot be captured by only local methods. Using these tools from algebraic topology has not been demonstrated convincingly to provide improvements for object detection performance, however it may provide an additional set of tools for characterizing shape to augment geometric methods.

In this section, we will summarize the construction of an integer linear program for optimal homologous cycle matching.

Problem Definition. Optimal homologous cycle matching [4][127] is formulated as follows. Let c be a p -chain in a simplicial complex K , with n simplexes of dimension $p + 1$ and m simplexes of dimension p . The optimal homologous chain problem is to find a p -chain x^* which has the minimum weighted l_1 -norm $\|Wx^*\|_1$ among all chains homologous to c . Equation (2.7) shows the weighted l_1 optimization for homologous chains.

$$\begin{aligned} (x^*, y^*) = \arg \min \quad & \|Wx\|_1 \\ \text{s.t.} \quad & x = c + \partial_{p+1}y \\ & x \in \mathbb{Z}^m, y \in \mathbb{Z}^n \end{aligned} \tag{2.7}$$

In this optimization, $\partial_{p+1} : C_{p+1} \rightarrow C_p$ is a boundary matrix mapping $(p + 1)$ -chains to p -chains and y is a $(p + 1)$ -chain.

The key constraint in this optimization is $x = c + \partial_{p+1}y$. Recall from section 2.3.4.2, that a homology group H_k is defined as $H_k(K) = \{z | z = z_0 + \partial_{k+1}c, c \in C_{k+1}(K)\}$ (defn 2.3.8), where a homologous cycle z differs by a boundary from a fixed representative cycle z_0 . In this case, the known representative cycle is c , and the optimal homologous cycle x must differ by a boundary of a $(p + 1)$ -chain y . Therefore, the resulting p -cycle x is *homologous* to c , and there exists a $(p + 1)$ -chain y such that x and c differ by the boundary of y .

Integer Linear Program. Equation 2.7 includes an l_1 norm in the objective which is non-linear in

x due to the absolute value of the l_1 norm. However, it is well known that an l_1 minimization can be rewritten as an linear program by adding slack variables and additional constraints.

$$\begin{aligned}
(x^+, x^-, y^+, y^-)^* &= \arg \min \sum_i w_i (x_i^+ + x_i^-) \\
\text{s.t. } & x^+ - x^- = c + \partial_{p+1}(y^+ - y^-) \\
& y^+, y^- \geq 0 \\
& x^+, x^- \leq 1 \\
& x^+, x^- \geq 0
\end{aligned} \tag{2.8}$$

Observe that the constraint $x \in \{-1, 0, 1\}$ is equivalent to the relaxed constraint $0 \leq (x^+, x^-) \leq 1$ if x^+ and x^- are restricted to integer solutions. This constrains x to have a natural geometric meaning for chain coefficients without explicitly requiring that $x \in \mathbb{Z}_2$, which leads to an intractable optimization.

The integer linear program in equation 2.8 can be written with linear inequality constraints of the form $Ax \geq b$, such that $x = [x^+ \ x^- \ y^+ \ y^-]^T$ and

$$A = \begin{bmatrix} -I & I & \partial_{p+1} & -\partial_{p+1} \\ I & -I & -\partial_{p+1} & \partial_{p+1} \\ -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \quad b = \begin{bmatrix} -c \\ c \\ -1 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.9}$$

It can be shown that if the boundary matrix ∂_{p+1} is totally unimodular, then the constraint matrix A is totally unimodular, and therefore the linear program in (2.8) has an integer solution and can be solved exactly in polynomial time.

Total Unimodularity and Boundary Matrices. An integer linear program can be solved in time polynomial in the dimensions of the constraint matrix A if and only if the constraint matrix is totally unimodular. The following lemma is proved in [128].

Lemma 2.3.11. *If a matrix A is totally unimodular (TUM) then a matrix A' obtained from A by any of the following operations is also TUM*

- $A' = A^T$
- $A' = [A, I]$
- A' is obtained from gauss jordan pivoting
- Adding one or more rows or columns with all zeros and a single one.
- Removing a row or column from A
- Adding to A one or more rows or columns already in A
- Multiplying a row or column by -1
- Permuting rows or columns

Lemma 2.3.12. *If ∂_{p+1} is totally unimodular, then the constraint matrix A in (2.9) is totally unimodular.*

Proof: The proof uses the properties for totally unimodularity described in lemma 2.3.11. Let B be of size $M \times N$. If B is TUM, then $[B \ B]$ is TUM by applying property five to repeat columns of B . If $[B \ B]$ is TUM, then $[B \ -B]$ is TUM by applying property seven to multiply the first N columns by -1 . If $[B \ -B]$ is TUM then $[I \ I \ B \ -B]$ is TUM by applying property four. If $[I \ I \ B \ -B]$ is TUM then $[-I \ I \ B \ -B]$ is TUM by applying property seven. Observe that this form is the same as the first block row of A in (2.9). Let this block row be A_1 . If A_1 is TUM, then $[A_1^T \ -A_1^T]^T$ is TUM by applying properties one, six and seven. Let this matrix be A_{12} . If A_{12} is TUM, then A is TUM by applying properties four and six. Therefore, since $B = \partial_{p+1}$, if ∂_{p+1} is TUM, then A is TUM. \square

Lemma 2.3.13. *For a finite simplicial complex triangulating a $(p+1)$ -dimensional compact orientable manifold, ∂_{p+1} is totally unimodular.*

Proof: Every p -face is a face of either one or two $(p+1)$ -simplex. For example, in graph theory, every node (0-face) has an endpoint at either one or two edges (1-simplex). Therefore, the boundary matrix ∂_{p+1} will have row entries of at most two non-zeros. Since it is known that a consistent orientation of $(p+1)$ -simplexes always exists for a finite triangulation of a compact orientable manifold, the row entries for ∂_{p+1} with two nonzeros always contain a $+1$ and -1 .

It is known that a matrix A is totally unimodular if rows of A can be partitioned into two disjoint sets A' and A'' such that the following four properties hold (i) every column of A contains at most two nonzeros (ii) every entry is $\{0,1,-1\}$, (iii) If two nonzero entries of A have the same sign, then the row of one is in A' and the other in A'' (iv) If two nonzero entries of A have opposite signs, then both rows are in either A' or A'' . Consider $A = \partial_{p+1}^T$. We have shown (i), (ii) follows from construction of a boundary matrix, we have shown that (iii) for orientable manifolds and (iv) we can construct matrices $A' = A$ and $A'' = \emptyset$ [128]. Therefore, A is TUM, and if A is TUM, then A^T is TUM from property one of lemma 2.3.11. Therefore, ∂_{p+1} is TUM. \square

Finally, lemma 2.3.13 shows that the boundary matrix ∂_{p+1} is totally unimodular, therefore from lemma 2.3.12 the constraint matrix A in (2.9) is totally unimodular, and the linear program in (2.8) provides an integer solution.

Results. The main result is the linear program in equation 2.8, and a characterization of those simplicial complexes for which the boundary matrix is totally unimodular. This characterization enables practical use of this result. An example result is shown in figure 2.5, which shows a reference cycle in red, and an optimal homologous cycle in green for the simplicial complex shown.

The total modularity for boundary matrices that are useful for shape representations were summarized in lemma 2.3.13. The paper also presents extensions for more general cases such as non-orientable manifolds and abstract simplicial complexes, however since these results are not directly relevant for shape representations (shape representations can assume an embedding in \mathbb{R}^2), we do not summarize them here.

2.4 Analysis of Global Shape Representations

In this chapter, we organized shape representations into local and global shape representations, and further decomposed global shape representations into graph based geometric and topological methods. We gave an overview of geometric methods with an organization by graph structure, describing trees, bipartite graphs and general graphs, and graph construction and graph matching approaches for each. We showed graph matching results and described the computational complexity for representational approaches for each graph structure. We described topological methods for shape representation using a unifying framework of simplicial homology, where homology was comput-

ing using persistent homology and matched using homologous cycle matching. We showed how this type of topological representation could be used to capture global topological properties for use in shape representations to complement geometric methods.

In this section, we analyze the performance of the graph based shape representations described so far in both a task independent and task dependent manner. *Task independent* analysis refers to comparing and contrasting the representational power and matching efficiency of different graph based approaches independent of the computer vision task that this representation would be applied to. Section 2.4.1 shows a comparison table and describes the tradeoff of representational power vs. computational tractability inherent with graph based representations. We compare different shape representations using a unifying criteria of structure, attributes, construction and matching, and we compare shape representations using graph structures and matching algorithms, by considering representational power of the graph structure and the efficiency of the matching algorithm.

Task specific analysis considers performance for graph based shape representations in a particular vision task, such as image matching. Unfortunately, the dirty little secret of graph based representations is that they do not perform as well as discriminative template based methods for object detection, recognition or categorization. In fact, all the methods discussed so far when applied to categorization tasks augment graph matching for detection using either (i) a final discriminative classifier, commonly an SVM or randomized forest [129, 130, 33, 25] or (ii) local distance learning [131, 132] to optimize a distance metric for nearest neighbor classification. These classifiers are practical since they result in improved detection performance by reducing the lower false alarm rate for spurious detections on background features. However, they have a highly questionable foundation for shape representation. In section 2.4.4 we will describe why these methods inherit the warts of classical categorization and should be treated with skepticism.

In the remainder of this section, we ask broader questions. We seek to understand if graphs and graph matching are an appropriate shape representation for practical vision tasks such as image matching, or if they have fundamental limitations or if there are theoretical problems with the representational power. Specifically we ask:

- **Is shape similarity geometric?** If we have perfect graph matching that optimally preserves geometry and topology of a reference graph during matching, have we solved the shape match-

| Graph | Matching | Attributes | Explicit? | Example |
|-----------|--------------------|--------------------------|-----------|-----------------|
| star | GHT | prototypes | no | [29, 134] |
| star | DP | discriminative templates | no | [23] |
| star | linear assignment | contours | no | [33] |
| star | GHT | regions | no | [129] |
| bipartite | perfect matching | local descriptors | yes | [31] |
| tree | tree edit distance | weight matrix | yes | [114][113][112] |
| tree | tree isomorphism | inflection points | yes | [32] |
| general | QAP | weight matrix | yes | [107] [109] |
| general | TPS | local descriptors | yes | [30] |
| simplex | Homologous cycles | weight matrix | yes | [4] |

Table 2.1: Comparison of graph based shape representations

ing problem? We show simple counter-examples where graph matching alone does not capture the similarity of two objects. In general, graph matching is an example of a representation based on a *first order isomorphism*, but a more powerful and expressive representation is based on *second order isomorphisms*. We describe work by Shepard [133] and Edelman [1] outlining this idea. This is addressed in section 2.4.2.

- **Is shape similarity topological?** If we have perfect topological shape representations such as simplicial homology and optimal matching of topological representations that preserves topological invariants, have we solved shape matching? Clearly, this answer is no since topological invariants alone do not capture perceptual similarity. We describe a simple counter example, but sketch out how topological invariants can be useful to augment geometric based representations. This is addressed in section 2.4.3.
- **Are discriminative classifiers classical categorization revisited?:** Does the final discriminative classifier step introduced by most graph matching approaches to reduce false alarm rates and handle the dirty little secret come at a cost? Are discriminative classifiers the way forward for shape representations, or does it make an implicit assumption on a discredited theory from cognitive science? In section 2.4.4, we argue that the final discriminative step is on a shaky foundation, and that this is not a credible path forward, even though it looks to improve performance.

2.4.1 Task Independent Comparison

These methods presented in this survey can be compared by considering the following criteria: graph structure, attributes, construction and matching. The graph structure refers to the underlying graphical structure, where in this survey we described trees, bipartite graphs, general graphs and simplicial complexes. The attributes refer to the edge and node features used to describe the local shape of the image at the graph embedding coordinates. Some work considers specific attributes to define the local shape, and other consider a graph structure that requires only a weight matrix to be defined. The graph construction can either be explicit or implicit, where explicit construction means that the graph is embedded prior to matching and implicit means that the construction is coupled with the matching. Finally, the graph matching defines the algorithm used for alignment of the reference graph with the current observed image.

Table 2.1 shows a comparison of graph based shape representations. This table includes the approaches reviewed in this survey, as well as other approaches that were not discussed in detail, but are representative of graph based shape representations.

The primary tradeoff in comparing in graph based representations is representational power versus computational tractability. General graphs encode all relevant pairwise geometry and global topology that capture shape, however matching for a general graph is equivalent to a quadratic assignment problem which is computationally intractable. Section 2.3.3.3 discussed two relaxations of the QAP, a graduated assignment algorithm based on graduated non-convexity, and a spectral relaxation. These algorithms match general graphs, however the final result is a fractional assignment, which must be discretized to a final matching output. The graduated assignment provides an iterative discretization using softassign and annealing, however it is unclear how to choose the annealing schedule in general to avoid being trapped in local minima. The spectral relaxation relaxes the matching constraints to perform a convex quadratic optimization, then enforcing the constraints during the discretization step. The final result is an assignment matrix, but it is not an optimal assignment matrix, since the discretization is a convenient heuristic to recover a discrete solution from a fractional optimization, unrelated to the original objective.

Sections 2.3.3.1 and 2.3.3.2 described bipartite graphs and trees, graph structures which enable efficient matching algorithms. However, these graphs do not capture the full geometry of a shape,

and therefore lack expressive representational power. For example, the tree structure of a star graph cannot capture pairwise geometry between parts, and can only capture the geometry between parts and the centroid. This limits the representational power, since for many objects, such as the legs and torso of a pedestrian, are correlated. The star graph and independence assumption is a convenient fiction to enable efficient matching using the generalized Hough transform. Similarly, a bipartite graph cannot capture any pairwise geometry, and as described in section 2.3.3.2, it is assumed that the overlapping shape context attribute will capture this pairwise interaction rather than having it expressed as geometric constraints during matching. However, these weights are not constraints to be enforced, and the result is that each node is independently matched according to weight, resulting in the same lack of representational power as star graphs.

A similar problem of representational power vs. computational tractability exists in the generative model community. Stuart Geman describes this as the *Markov dilemma*. How do we model constraints on attributes such as poses of pairs eyes while having a Markov network that is modelling eyes as conditionally independent given a face, without giving up computational tractability? [27]. The fact that probabilistic graphical models also share this representational limitation hints that it is universal graph representations, which fundamentally limits the representational power.

2.4.2 Is Shape Similarity Geometric?

Shape matching is the problem of alignment of an exemplar in an input image. Clearly geometry plays a role, as can be shown by dramatic performance improvements of alignment methods that use shape representations that include even weak geometry [15] versus texture-only classification [135]. However, is shape matching just alignment to a reference graph that preserves geometry during matching?

Figure 2.6 shows two examples of where alignment fails to capture perceptual similarity. Figure 2.6 (left) shows two reference images, a dotted square and a triangle, and an observed image of a square. Assume that these reference images and observation have an explicit graph embedding, such that the attribute weights for graph matching are proportional to geometric deformation. In both cases, the reference models will match to 50% of the observation, which from alignment would result in these triangle model and the dotted square as equally similar to the observed square. Clearly, we perceive the dotted square as more similar to the observation than the triangle, so alignment is

not capturing this perceptual similarity. Figure 2.6 (right) again shows two models, a handwritten “O” and a typographic “Q”, and an observation of a typographic “O”. The handwritten “O” does not align anywhere with the observation, while the “Q” aligns everywhere except the most important, yet small, discriminative stroke. Alignment alone would define the “Q” more similar to the “O”, while perceptually we would consider the handwritten “O” more similar. So, clearly there is more to perceptual similarity than geometric alignment alone. However, alignment does provide a cue for similarity, since the dotted square or the handwritten “O” are more similar to the observation than say a cat or a motorcycle.

These examples show that similarity is more than alignment, it requires *relevant alignment*. The same issue of representation has been explored in the categorization literature. For example, Murphy and Medin make the following observation for the use of common attributes for object categorization as described by Edelman [1]:

The number of attributes shared by plums and lawn-mowers could be infinite: both weigh less than 1000 kilograms (and less than 1001 kilograms), both cannot hear well, both have a smell, etc. Any two entities can thus be arbitrarily similar or dissimilar, depending on what is to count as a relevant property.

This observation is an example of the *ugly duckling theorem* which states informally that similarity requires bias, since otherwise is if similarity is judged in terms of number of predicates shared then any two objects are equally similar. The same holds for shape representations. In figure 2.6, the relevant alignments are not all precise deformations of the handwritten O, but rather discriminative properties of the holistic shape. This begs the question, what is important? This seems to imply that discriminative classifiers are necessary, since these classifiers are designed to weight features that are important for classification, but we will see that they also inherit the deficiencies of classical categorization. So, alignment is useful, but it does not tell the whole story.

2.4.3 Is Shape Similarity Topological?

Section 2.3.4 described using homology and homologous cycles for shape representation, but is this sufficient for shape matching? Clearly, topology alone does not capture a representation of objective shape. Topology can be described as qualitative geometry, and the invariants that remain

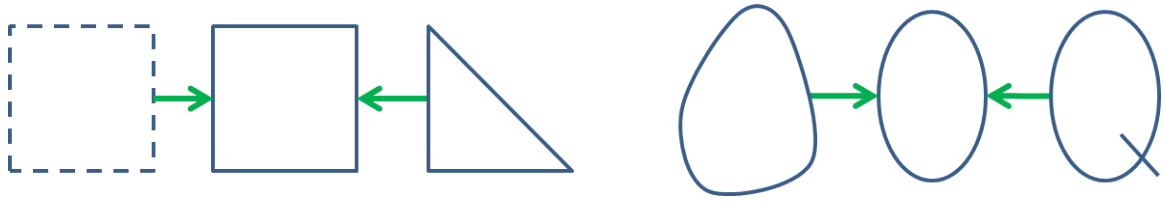


Figure 2.6: Alignment does not capture all similarities

once geometric concepts are removed. However, objective shape is a geometric concept, as shown by our clear perception of bagels and coffee cups as different objects, even though as topological shapes they have the same Betti numbers, and therefore the same topological invariants.

However, homologous cycles and chains do have the potential to elegantly capture global matching constraints in perceptual organization, such as “surrounded”, “interior” and “connected”. These constraints are *global* which capture the holistic properties of a shape which cannot be captured by local node or edge interactions only, and these constraints are independent of geometric deformation. These topological invariants may provide robustness to intraclass variability than geometric attributes only. For example, topological features can be used for recognition of people and objects, such as pedestrians carrying a bag or pushing a stroller may exhibit significant variation in geometric configuration, however the topological invariants remain stable (e.g. the loop of a bag being held, a baby surrounded by and interior to a stroller). These topological attributes provide an invariant shape representation that may provide improved generalization.

So, while use of topology to represent these global properties has not yet been demonstrated convincingly, there is the potential to provide a combined geometric and topological framework that is more powerful than either independently. So, topology may be useful, but it does not tell the whole story.

2.4.4 Are We Revisiting the Classical Theory of Categorization?

The classical theory of categorization states that categories are defined by necessary and sufficient conditions for membership [70, 71]. For example, the category of “triangle” can be defined unambiguously, such that any subset of line segments can be checked for consistency with the definition. So, if a subset of line segments contains three segments, and if each pair of line segment endpoints intersect at a unique point, then the subset is categorized as “triangle”. The conditions are *neces-*

sary in that they must be satisfied to make a categorization decision, and they are *sufficient* in that even if there are other properties present such as the lines colored red or dotted lines, the minimum properties are enough to render a categorization decision. The necessary and sufficient conditions define the category, so this is often called a *definitional* approach to categorization. This approach can be traced back to Aristotle who discussed the essence of a category as those properties shared by all members.

The definitional approach has a number of implied properties. First, an object is either in the category or not. If a subset of line segments satisfies the definition of a triangle, then either it is a triangle or it is not, there are no “in between”. This is often called the property of the excluded middle. Second, there are no distinctions between category members. As long as the necessary and sufficient conditions are satisfied, then the object is assigned membership to a category. There are no “better” or “worse” examples of a category, since all members are equal in satisfying the definition.

The classical view appears intuitive, and it does provide unambiguous categorization for some objects with a definitional nature, but has been widely discredited as a general representation. Consider the category of “dogs”, as described by Murphy [70]:

The definition for dogs...namely things that have four legs, bark have fur, eat meat and sleep is obviously not true. Does something have to have four legs to be a dog? Indeed, there are unfortunate dogs who have lost a leg or two. How about having fur? Although most dogs do have fur, there are hairless varieties like chihuahuas that don't. What about barking? Almost all dogs bark, but I have known a dog that lost its bark as it got older. This kind of argument can go on for some time when trying to arrive at necessary features...Wittgenstein urged his readers not to simply say 'there must be something in common' but to specify the things in common. Indeed, it turns out to be very difficult to specify necessary and sufficient conditions of most real world categories.

This problem is not limited to animals. Murphy describes the practical problem of categorization in material science by as described by distinguished metallurgist Robert Pond [70]

You really don't know what a metal is, and there's a big group of people that don't

know what is a metal is. Do you know what we call them? Metallurgists! ... Here's why metallurgists don't know what a metal is. We know that a metal is an element that has metallic properties. So, we start to enumerate all these properties: electrical, conductivity, thermal conductivity, ductility, malleability, strength, high density. Then you say, how many of these properties does an element have to have to classify as a metal?...We can't get the metallurgists to agree!...So, we just proceed along presuming that we are all talking about the same thing.

These anecdotal examples of the problems facing definitional approach to categorization can be traced back to Wittgenstein, who was the first to question the assumption that categories could be defined. For example, Wittgenstein describes the category of *games* and argues that there are no common definitional properties common to all games. Rather, games share family resemblances such as amusement, rule following or competition. From Lakoff, “Games, like family members, are similar to one another in a wide variety of ways. That, and not a single well defined collection of common properties is what makes game a category.” [71].

Eleanor Rosch and her work on prototype theory provided a set of empirical evidence to challenge and eventually discredit classical categorization [136]. She made two key observations. First, she observed that if categories are definitional, then no member should be any better example of the category than any other. Rosch provided experimental evidence that all categories have best examples called *prototypes*, such as a robin being a prototype of a bird and not an ostrich. These central members are more commonly associated with the category, are more quickly labelled than non-central members, children learn them more quickly, and they exhibit more family resemblances to other members of the category than non-central members. Second, she observed that if category membership is definitional, then there should be no ambiguity as to category membership and categorization should not be observer dependent. However, experimental evidence exists for categories that are ambiguous (is swampwater water or poison?) and categorization that changes from person to person (is a tomato a fruit or vegetable?).

These observations are not explained by classical categorization theory, which puts this theory on shaky foundation. Murphy summarizes the arguments against classical theory as follows [70]:

It has simply ceased to be a serious contender in the psychology of concepts...First,

it has been extremely difficult to find definitions for most natural categories...Second, the phenomenon of typicality and unclear membership are both unpredicted by the classical view...Third, the existence of intransitive category decisions (car seats are chairs, chairs are furniture, but car seats are not furniture).

In short, classical categorization using abstract absolute definitions does not explain experimental evidence for how we perform categorization. It is unlikely that there are “essences” that define objects or “natural kinds” that define animals or plants. As Darwin said, “We shall have to treat species as artificial combinations made for convenience in order to be free from the vain search for the undiscovered and undiscoverable essence of the term ‘species’ ”.

However, even though classical categorization has been discredited, the literature on object categorization implicitly relies on it. Let’s assume that object categorization is limited to basic level categories, then we ask the question, is object categorization equivalent to feature space classification? Feature space classification refers to embedding a labelled training set into a common feature space, then learning a discriminative classifier. Every training image is represented as a point in a common feature space, and a learning framework optimizes a discriminative classification function to partition this space. We will show that this assumption *implicitly assumes the classical view of categorization*, and inherits its warts.

Consider the popular “bag of words” approach to object classification [135]. In this approach, an image is represented by the number of times a set of local prototypes or *visual words* appear in an image, independent of the relative position of these words. Each training image is represented by a histogram which captures the frequency of this unordered “bag” of visual words. Each training image is represented by a histogram, and the set of histograms from the training set serve as observations input to a support vector machine (SVM) classifier. The weights learned by this classifier represent the category. What does this approach say about a category? The category is defined in a *feature space*, such that each training image is represented as a point in this feature space. For bag of words models, the dimensions of the feature space correspond to the frequency of prototypes occurring in a given image. The weights learned for a (one vs. all) linear support vector machine defines an optimal hyperplane in this feature space which separates category members from non-members. Points on one side of the hyperplane are declared category members and points on the

other side are not members. The SVM weights which define the hyperplane parameters define a “rule” that can be used to define the category in terms of the weights in each of the feature space dimensions. For a given observation, the SVM weights define necessary and sufficient conditions for category membership, and either an object is a category member or not. In other words, classical categorization.

This argument is not limited to SVMs. Any discriminative classifier that learns a classification function in a fixed feature space is learning a categorization rule. This is true by construction, since a classification function is a mapping from feature space to label, which encodes a (complicated and opaque and non-obvious) rule for assigning observations to categories. However, as described in section 2.4.4 the classical theory has been widely discredited as a category representation by cognitive scientists. Why then are they so popular? Critics of the classical view claim that the definitional approach is flawed since they themselves cannot write down a definition for a natural category such as dogs. This does not mean that a category definition does not exist, since the definition may be hidden in the data, and can only be uncovered by statistical analysis. However, if experts cannot agree on those features that are needed for categorization, and if a discriminative classifier can capture the essence of a category, then why don’t the experts use a discriminative classifier to settle the debate? Perhaps the classifier cannot capture an optimal statistical rule because a rule does not exist.

So, approaches that use feature space classification appears to be a revisitation of the classical theory of categorization. What are the alternatives? Edelman argues that shape representation should be a *second order isomorphism* [6][1]. An isomorphism refers to a functional mapping that is one to one and onto. Formally, a function f is an isomorphic map if and only if it is a bijection. In the context of shape representations, a first order isomorphism refers to correspondence between a model representing distal properties and an internal representation or proximal representation. For example, 3D reconstruction attempts to create an internal 3D representation for an object that is isomorphic with 3D shape of an object in the scene, or graph matching attempts to create an internal graphical representation of an object that is isomorphic to a reference. In contrast, a second order isomorphism [133] refers to the relationships between representations and not the representations themselves. From Shepard [133]:

Although the internal representation for a square need not itself be square, it should (whatever it is) at least have a closer functional relation to the internal representation for a rectangle than to that, say, for a green flash or the taste of persimmon.

Second order isomorphism and first order isomorphism or representation of similarity instead of representation by similarity. "The idea of second order isomorphism translates to the notion that only certain relations between the objects - not the shape of individual objects themselves - need be represented" [1]. Second order isomorphism neatly bypasses the question of the "right" representation. Instead of trying to find the ideal representation that best captures an object category, second order isomorphism asks for a given representation, do the proximal relations preserve the distal relations? In other words, I may not know how to represent the category of "dog", but for any choice I make for representing shape of an individual dog, the shape of a golden retriever should be more similar to a Doberman than a bus. This focuses on the representation of similarity rather than the representation of an object category. This sentiment is shared in recent work in object categorization on exemplar based similarity [137, 138].

An approach like second order isomorphisms neatly bypass the limitation of classical categorization, since there are no global rules defining categories, there are only similarity relations among data. Similarity computations are delayed until test time, which delays the classification decision until test time, which does not require a predefined category definition learned during training. In essence, the data is it's own definition. In other words, in a complex world, perhaps the best model of the world is the world itself [139].

2.5 Summary

In this chapter, we surveyed graph based shape representations by grouping into geometric methods and topological methods, where each method is organized by invariances of increasing abstraction. For geometric methods, we used the unifying framework of weighted graph matching posed as a quadratic assignment problem as a unifying framework for discussion, and we described invariant properties maintained during various tree, bipartite and general graph matching approximations. For topological methods, we used the unifying framework of simplicial homology, and describe the persistent homology, a technique for recovering the homology given noisy data, and optimal

homologous cycle matching for matching topologically invariant cycles. Finally, we analyzed these methods both in a task independent and task dependent context.

The conclusions are that graph based shape representations based on graph matching to an exemplar, do provide both a measure of geometric and topological similarity. However, they do not provide equivalence to perceptual similarity, they do not embody Edelman's second order isomorphism, and they cannot provide the same performance as a discriminative classifier. However, since discriminative classifiers have their own representational problems, more work is needed to cross the representational gap.

David Weinberger in "Everything is Miscellaneous" [140] observes the same representational problems with knowledge that we outlined with shape representations. The Dewey decimal system for categorizing knowledge is good for searching for books by author, but not for searching for a birthday present. Alphabetization is good for categorizing knowledge in printed encyclopedias, but only if you know the name of what you are looking for. These and other problems of categorization he attributes to the fact that there is no one categorization model for knowledge. He observes, "The world is too diverse for any single classification system to work for everyone in every culture at every time...The best representation depends on the task.". To compensate, he outlines four strategic principles for organizing knowledge: (i) Filter on the way out not on the way in (ii) Put each leaf on as many branches as possible, (iii) Everything is meta-data and everything can be a label and (iv) Give up control. The next generation of shape representations may be down a similar path.

Chapter 3

Nested Shape Descriptors

3.1 Introduction

Local feature descriptors have emerged in the past ten years as the dominant representation for image matching. There exist standard benchmarks for performance evaluation [39, 74, 61], and a zoo of detectors and descriptors [38, 16, 40, 42, 43, 44, 45, 46]. introduced with the trend of faster and faster matching while maintaining approximately equivalent performance to SIFT [35]. Local feature descriptors have been successfully deployed for a wide range of image matching tasks including: stereo, optical flow, structure from motion, egomotion estimation, tracking, geolocation and mapping.

All existing local feature descriptors share a common performance tradeoff between support size and matching selectivity. It is well known that for the task of image matching, descriptors constructed with larger support outperform descriptors with smaller support [42, 43, 44, 45, 46]. Descriptors with large support are constructed with larger image patches that increase the uniqueness of a match and address the aperture problem, however there are diminishing returns for constructing a descriptor too large. For example, there may be arbitrarily large outliers in the descriptor due to occlusions and geometric variation effects far from the descriptor center. So, an ideal descriptor would be as large as possible, while being robust to occlusions.

In this paper, we introduce nested shape descriptors to address this tradeoff. A nested shape descriptor (NSD) is a family of binary local feature descriptors constructed by pooling oriented and scaled gradients over a large geometric structure called an *Hawaiian earring*. An example of

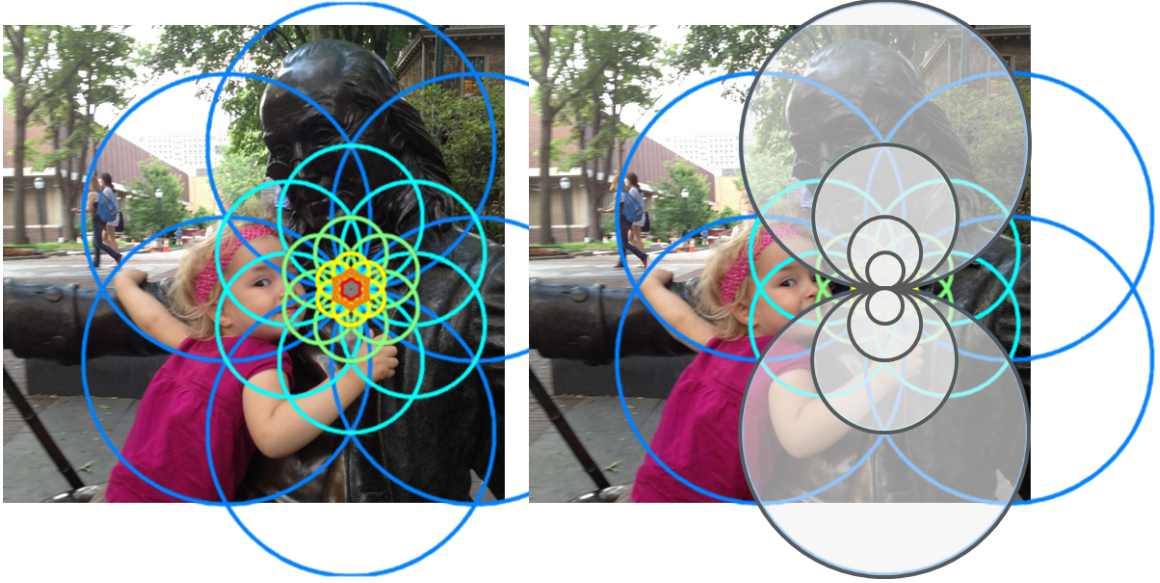


Figure 3.1: Nested shape descriptors pool scaled and oriented gradients over large geometric structures called *Hawaiian earrings*. (left) *Hawaiian earrings* with k -fold rotational symmetry define a member of the nested shape descriptor family called the *seed-of-life descriptor* (right) Two *Hawaiian earrings* substructures in the seed-of-life descriptor are highlighted in grey.

the nested shape descriptor is shown in figure 3.1. Each descriptor has global support covering the entire image, and the structure of the descriptor exhibits fractal self-similarity in scale. This correlated nested structure enables new a robust distance function called the *nesting distance*. The nesting distance uses order statistics for robustness to outliers while maintaining a descriptor with global support.

Nested shape descriptors make four primary contributions.

- **Global support:** Each NSD exhibits support that covers the entire image, which provides improved selectivity for cases exhibiting the aperture problem without sacrificing localization accuracy.
- **Binary:** NSDs are binary, which enables for compact storage and allows the nesting distance to use a fast Hamming distance, without sacrificing matching performance.
- **Robust local distance function:** The nesting distance is a quadratic local distance function that is robust to corruption of the descriptor due to occlusions, geometric variations or lighting.
- **Saliency:** We show that the log-spiral normalization is performing a type of bottom up

saliency computation, which provides a representation of orientation and scaled salient edges. We show how the NSD can be used to provide a saliency map using steerable pyramid reconstruction.

In this paper, we provide sufficient conditions for construction of a nested shape descriptor using key concepts of nested pooling and log spiral normalization. We perform a trade study to determine optimal descriptor parameters for the task of image matching. Finally, we evaluate performance compared to other local feature descriptors on the VGG-Affine image matching benchmark and Photorealistic Virtual City datasets showing measurable performance gains.

3.2 Related Work

There have been many local feature descriptors proposed in the literature in the past ten years. From oldest to newest, the primary developments have been: SIFT [35], PCA-SIFT [36], Shape context [31], local binary patterns [37], SURF [38], GLOH [39], Sparse localized features (SLF) [16], compressed HoG (cHoG) [40], DAISY [41, 42], BRISK [43], BRIEF [44], ORB [45] and FREAK [46].

The trend in local feature descriptor research has been to show comparable performance to SIFT on the VGG-Affine benchmark [39, 74, 61], with ever faster computation. Work has progressed from PCA-SIFT [36] and SURF [38] which show close performance to SIFT with lower dimensionality and faster preprocessing. Recent work has focused on introducing binary features from local comparison tests [44, 43, 45, 46] which enables fast distance metric based on Hamming distance and faster derivatives [75]. These developments have been driven by the need for faster processing to support mobile deployment.

A taxonomy for comparing and contrasting local feature descriptors can be described in terms of five criteria: preprocessing, support, pooling, normalization and descriptor distance. Preprocessing refers to the filtering performed on the input image, support patterns are the geometric structure used for constructing the descriptor and pooling is the aggregation of filter responses over the support structure. Figure 3.2 shows this taxonomy and a comparison of dominant local feature descriptors.

Using this taxonomy, the nested shape descriptor is most closely related to DAISY, BRISK and FREAK. NSD has large support and distance robust to occlusions like DAISY, but it does not

| Descriptor | Preprocessing | Support | Pooling | Normalization | Distance |
|---------------|----------------------------|-----------------------|---|--------------------------|---------------------------|
| SIFT | Oriented gradients | Cartesian grid | Histogram | Truncated norm | L2 |
| SURF | Integral image | Cartesian grid | Histogram | Truncated norm | L2 |
| PCA-SIFT | Oriented gradients | Cartesian grid | histogram, PCA | Truncated norm | L2 |
| Shape Context | Edge detection | Log-polar grid | Sum | Global norm | Bipartite matching |
| GLOH | Oriented gradients | Log-polar grid | Histogram, PCA | PCA | L2 |
| SLF | Laplacian pyramid | Cartesian grid | Max | Global norm | L2 |
| cHoG | Oriented gradients | Trees | Histogram | Global norm | L2 |
| DAISY | Convolved orientation maps | Overlapping log-polar | Patch sampling | Support norm | L2 |
| BRIEF | Gaussian filter | Log-polar patch | Gaussian sampled binary comparisons | None | Hamming |
| BRISK | Gaussian filter | Log-polar patch | Deterministically sampled binary comparison | None | Hamming |
| ORB | Gaussian filter | Log-polar patch | Gaussian sampled binary comparisons | None | Hamming |
| FREAK | Gaussian filter | Log-polar patch | Retinally sampled binary comparisons | None | Hamming |
| NSD | Steerable pyramid | Nested log-polar | Nested pooling | Log-spiral normalization | Hamming, Nesting distance |

Figure 3.2: Taxonomy and comparison of local feature descriptors.

require an iterative optimization framework determine occlusion masks. NSDs are binary with large support like BRISK/FREAK, however NSD support is larger and global relative to the image size. Furthermore, unlike BRISK and FREAK, NSD uses scaled and oriented gradients comparisons rather than pixel comparisons for computing the binary representation.

Finally, local distance functions [132] have been explored for metric learning of exemplar distances for the task of object recognition. However, distance functions for local feature descriptors have been limited to Euclidean, Hamming and Mahalanobis distances, where covariance estimation is typically used only for dimensionality reduction [36][39]. In the taxonomy of [132], the nesting distance is per-exemplar (“where”), online (“when”) using order statistics (“how”) without requiring any offline training.

3.3 Nested Shape Descriptors

In this section, we describe the construction of nested shape descriptors. NSD are constructed by first defining the nested pooling structure (section 3.3.1), which can be decomposed into a sets

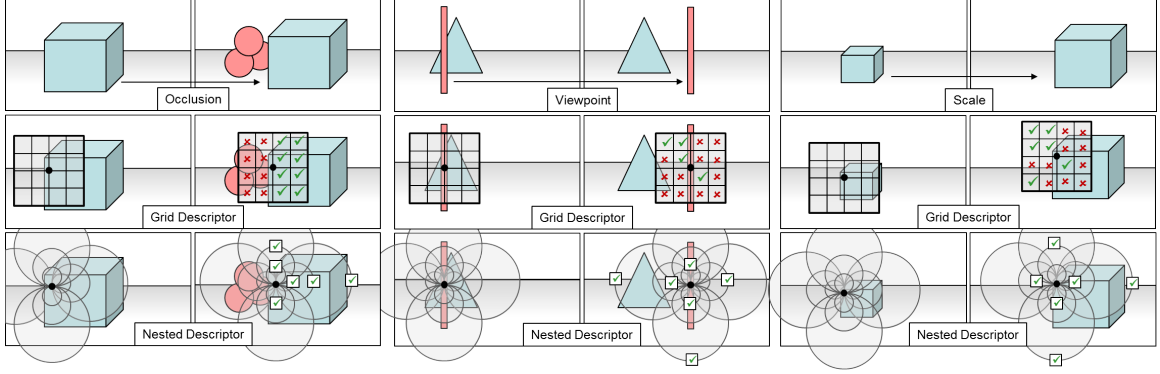


Figure 3.3: Why nesting? (left) Occlusions corrupt half of a generic grid descriptor covering the occluded region (red X’s), while the nesting distance selects the best subset of supports in the nested descriptor that cover only the object (green checkmarks). (middle) Viewpoint changes for long and thin foreground structures introduce errors in grid descriptor matching due to large changes in the background. The nesting distance selects the subset of supports during matching that cover the foreground and are the correct scale to allow for background variation. (right) Scale changes without scale invariant detectors introduce errors in grid descriptor matching due to changes in local support. The nesting distance uses a subset of both large and small scale supports, ignoring intermediate scale supports with corruption.

of “Hawaiian earring” structures. We provide definitions for this construction and show how the nested shape descriptor is constructed from these pieces (section 3.3.2). Furthermore, we define the nesting distance (section 3.3.5), which uses the properties of the nested descriptor to provide robust distance function. Finally, we define a specific member of the nested shape descriptor family called the seed-of-life descriptor (section 3.3.3), constructed using Hawaiian earrings with k -fold rotational symmetry.

What is the intuition behind the nested descriptor? Figure 3.3 shows three cases that motivate the use of nesting. The nested descriptor and nesting distance are compared to a generic grid descriptor (e.g. SIFT, but the same argument holds for log-polar grid descriptors) for three common scene variations: occlusions, viewpoint and scale. The red X’s and green checkmarks show where a grid descriptor is corrupted due to the scene variation, which leads to poor matching performance. For these cases, the NSD and nesting distance are able to select the best subset of supports during matching to provide robustness to these scene variations. See the caption in figure 3.3 for a discussion.

Why the nesting distance? Given a pair of descriptors, the nesting distance computes a weighted sum of the best k coordinate matches. If a coordinate is an outlier (e.g. the worst $n - k$ coordinates,

where n is the dimensionality of the descriptor), then any inliers correlated with this outlier are suspect, and are appropriately downweighted. The nesting distance relies on nesting, such that all supports are linked by exactly one point in the center of the descriptor. We discuss further in section 3.3.5.

3.3.1 Hawaiian Earrings and Nested Pooling

Nested shape descriptors represent shape by pooling of oriented gradients within *Hawaiian earrings*. Figure 3.1 (right) shows an example of the Hawaiian earring substructure formed by a nested set of circles all intersecting at exactly one point at the center. The Hawaiian earring is a nested structure analogous to Matryoshka or Russian nesting dolls, where each smaller doll fits neatly inside the next larger doll. Hawaiian earrings may be combined into sets such that each earring is called a *lobe*. Each lobe exhibits scale symmetry and all earrings intersect at exactly one point in the center.

In the remainder of this section, we formally define the Hawaiian earrings geometric structure. The definitions provide precise construction, however this formality should not obscure the simple intuitive nature of this descriptor. Nested circles of exponentially increasing radius all intersect at exactly one point in the center, and each circle pools oriented gradient responses at a specific scale. Figure 3.1 shows this common center point in red.

3.3.1.1 Formal Definitions

The formal definitions of the Hawaiian earring used to construct the nested shape descriptor are as follows. First, preliminary notation. Let I be an $M \times N$ greyscale image containing pixels $p \in I$ with greyscale value $I(p)$.

Definition 3.3.1. A *support* S at c is $S = \{p \mid p \in I, \|p - c\|_2 \leq r\}$

This defines a *support*. Observe that a support is a set of all pixels within a given radius of a center pixel c . For example, each circle in figure 3.5 is a support.

Definition 3.3.2. A *nested support set* at p is a set of supports $\mathbb{S}_p = \{S_i \mid r_{i-1} < r_i, p \in S_i, i \leq n\}$ and $S_1 = \{p\}$, $S_n = \{I\}$.

This defines a *nested support set*. A nested support set is an ordered set of supports, such that each support contains the element p and smaller supports are contained within larger supports. Formally, inner support region are strict subsets of all outer support regions, $S_1 \subset S_2 \subset \dots \subset S_n$, radii are totally ordered such that $r_1 \leq r_2 \leq r_n$ and p is contained in each support S_i . The set of grey circles shown in figure 3.5 (middle) form a nested support set.

The definition of the nested support set implies two useful properties. First, A nested support set is *precise*. It follows from definition (3.3.2) that the smallest radius $r_1 = 0$ since the innermost support K_1 must contain only p . This definition implies that there exists exactly one point p that is in all supports S_i . This property enables precise pixel level alignment of the nested descriptor for a large support set. Second, a nested support set is *bounded*. The largest support region S_n contains the entire image, which implies that $r_{n-1} < \max(M, N)$ and $r_n \geq \max(M, N)$. This provides a requirement that the largest support must include the entire image to provide global descriptor properties.

Definition 3.3.3. An *Hawaiian earring* $K(\theta)$ is a nested support set \mathbb{S} such that for each support set $S_i \in \mathbb{S}$, $r_i = 2^i$ and $c_i = (2^{i-1}, \theta)$ in polar coordinates.

This defines a specific case of a nested support set called the *Hawaiian earring*. Each support in the Hawaiian earring have exponentially increasing radius, the center of each outer circle is on the boundary of the inner circle and all share exactly one common point at the boundary of all circles. The centers of each support are defined in polar coordinates, such that θ is the orientation of the line intersecting all support centers. For example, figure 3.5 (middle) shows a Hawaiian earring structure in grey, such that the common point is the center of the seed of life structure, and the angle is $K_{\pi/2}$. This structure is fundamental building block of the seed of life and the nested shape descriptor.

Definition 3.3.4. A *seed of life* \mathbb{K}_n is a set of Hawaiian earrings such that $\mathbb{K}_n = \{K_i(\theta_i) \mid \theta_i = \frac{2\pi i}{n}, \forall i \leq n\}$.

This defines the *seed of life*. This geometric structure is a set of Hawaiian earrings such that each is equally spaced in n polar orientations. Figure 3.5 (left) shows the seed of life \mathbb{K}_6 for six quantized orientations. The seed of life defines the pooling structure used in the nested shape descriptor and is the primary construction of this section. Figure 3.7 shows an example of increasing lobes from $\mathbb{K}_1 - \mathbb{K}_{10}$.

In the remaining sections, we will use the following notation to reference the substructures of Hawaiian earrings. The index $\mathbb{K}_n(u)$ refers to the u^{th} of n Hawaiian earrings, also called the u^{th} lobe. The index $\mathbb{K}(u, v)$ refers to the v^{th} support of the Hawaiian earring $K_n(u)$. For example, in figure 3.1 (right), the two lobes highlighted in grey are Hawaiian earrings $\mathbb{K}_6(1)$ and $\mathbb{K}_6(4)$ and the two largest circles are referenced as supports $\mathbb{K}_6(1, 4)$ and $\mathbb{K}_6(4, 4)$.

3.3.2 Nested Shape Descriptors

A *nested shape descriptor* D at interest point p is defined by nested pooling, logarithmic spiral normalization and binarization of oriented gradients B over a nested support \mathbb{K}_n .

$$d(i, j, k) = \sum_{q \in \mathbb{K}(j, k)} B_{ik}(q) \quad (3.1)$$

$$\hat{d}(i, j, k) = d(i, j, k) - d(i, j-1, k-1) \quad (3.2)$$

$$D(i, j, k) = \begin{cases} 1 & \text{if } \hat{d}(i, j, k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

Equation (3.1) is *nested pooling*. Let $B_{rs}(q)$ be a bandpass response at pixel q for subband orientation r at octave scale s [141]. The descriptor $d(i, j, k)$ is the pooled response for orientation subband i , lobe j and lobe scale k . Observe that the bandpass octave scale s is equal to the Hawaiian earring support radius k . In other words, support regions with radius 2^k pool orientation subbands over octave scales k . As the support radius increases, the pooling support contains the next smaller support, resulting in nested pooling within a lobe. Figure 3.4 (left) shows an example of this construction. Equation (3.1) shows sum-pooling, but we also experiment with max-pooling over a support. Pooling strategies will be defined in the experimental results section.

Equation (3.2) is *logarithmic spiral normalization*. A logarithmic spiral is a curve that can be written in polar coordinates as $r = ae^{b\theta}$ for arbitrary positive real constants a and b . A nested support set \mathbb{K}_n exhibits a logarithmic spiral when considering neighboring supports. For example, figure 3.4 (right) shows an example of the logarithmic spiral for \mathcal{K}_6 . Each turn of angle $\theta_i = \frac{2\pi}{6}i$ is a radius of $r_i = 2^i$, which is equivalent to a logarithmic spiral numerically approximated with parameters $a = 1, b = 0.66191$. Figure 3.4 (right) shows a log-spiral and its reflection $r = ae^{-b\theta}$ forming an

elegant flower-like pattern. This pattern encodes the normalization which is a difference of spiral adjacent support, which provides invariance to additive gradient bias.

Equation (3.3) is *binarization*. A nested shape descriptor can be binarized by computing the sign of (3.2). This constructs a nested shape descriptor with binary entries.

Figure 3.9 shows an example of this construction. Nested pooling is equivalent to pooling of fixed radius over scales of a steerable pyramid [141], which is analogous to a “flattening” of a pyramid representation of scaled and oriented gradients. The final nested shape descriptor D is a binary vector of length $(R \times |\mathbb{K}| \times |K|)$ for R orientation bands over $|\mathbb{K}|$ lobes and $|K|$ supports per lobe. For example, for eight orientation subbands, five nested supports, and six lobes has dimensionality $(8 \times 6 \times 5) = 240$.

3.3.3 The Seed-of-Life Descriptor

The nested shape descriptors in section 3.3.2 defines a family of descriptors that share the common properties of nested pooling, log-spiral normalization and binarization. In this section, we define a specific member of this family called the seed-of-life nested shape descriptor or simply the *seed-of-life descriptor*.

The seed-of-life descriptor is a nested shape descriptor such that the nested pooling \mathbb{K}_n is defined using a rotationally symmetric geometric structure called the *seed-of-life*. The seed of life is an ancient geometric symbol formed using Hawaiian earrings with n -fold rotational symmetry. This structure has been discovered as artistic ornamentation in antiquity as far back as the Temple of Osiris in Egypt and Phoenician art from the 9th century BC. It is a central figure in “sacred geometry” where it is a primitive shape used in constructing the “flower of life” and “fruit of life”. An example of the seed-of-life descriptor for \mathbb{K}_6 is shown in figure 3.1 (left).

The seed-of-life descriptor is perhaps the simplest member of the nested shape descriptor family since it exhibits rotational symmetry where Hawaiian earring lobes are spaced uniformly in angle. This descriptor is used for all experiments in section 3.4.

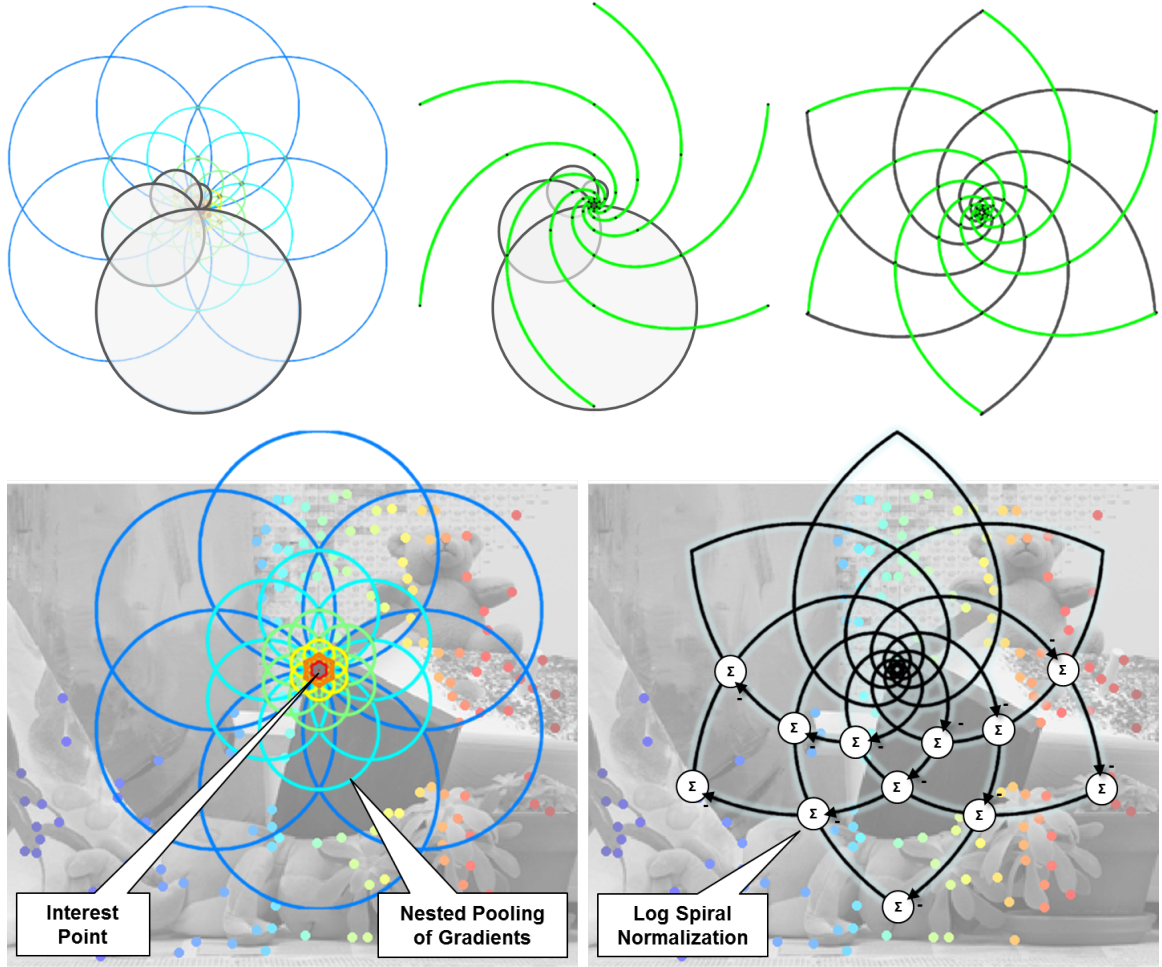


Figure 3.4: (top) Logarithmic spiral property of the nested shape descriptor provides *normalization* and *binarization*. The log-spiral and its reflection shown in grey form an elegant flower-like structure. (bottom) An NSD is formed at each interest point by (left) nested pooling of scaled and oriented gradients and (right) log-spiral difference and binarization.

3.3.4 Seed-of-Life Examples

Figure 3.5 shows an example of the nesting property. A nested shape descriptor exhibits nesting in two ways, Hawaiian earring nesting and cocentric nesting. An *Hawaiian earring* is a geometric structure formed by the nesting of a set of circles that intersect at exactly one point. The gray opacity in this figure shows that the inner circles are fully contained within the outer circles. Cocentric nesting is formed by a set of nested circles that have the same center. These nesting concepts will be used to construct the nested shape descriptor in this section.

Figure 3.6 shows an example of the log-spiral pattern formed by neighboring supports. The sequence of grey circles with centers and radii at left follow the logarithmic spiral shown in green

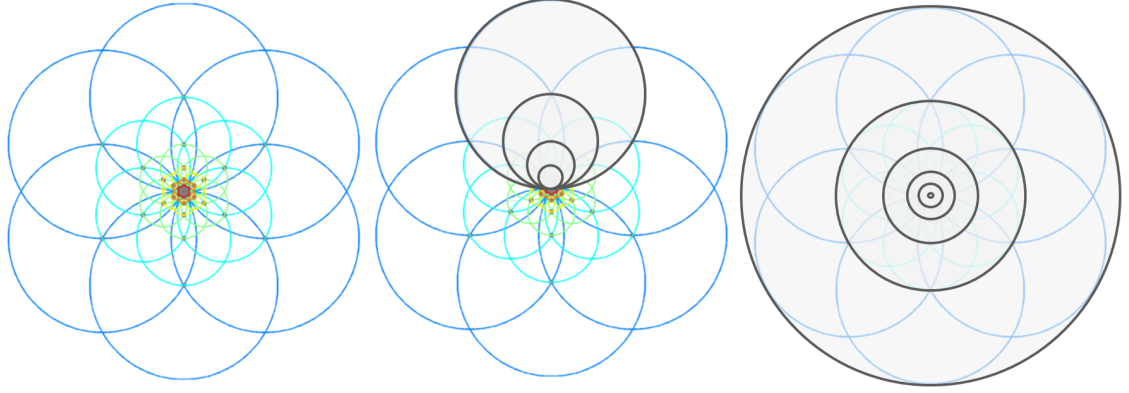


Figure 3.5: Nesting property of the nested shape descriptor. (left) Seed of life, (middle) Hawaiian earring, (right) Cocentric nesting.

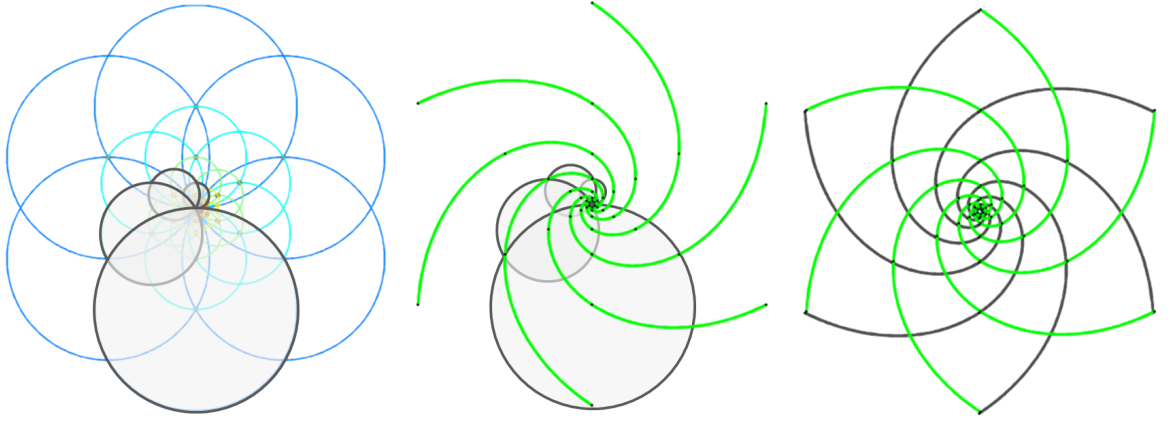


Figure 3.6: Logarithmic spiral property of the nested shape descriptor provides *normalization* and *binarization*. (right) The log-spiral and its reflection shown in grey form an elegant flower-like structure.

in 3.6 (middle). Combining this log-spiral with its reflection (right) forms an elegant flower like structure used for normalization and binarization.

Figure 3.7 shows nested shape descriptors computed for seed-of-life $\mathbb{K}_1 - \mathbb{K}_{10}$. These examples show the rotational symmetry as lobes are added.

3.3.5 Nesting Distance

The nesting distance is a robust quadratic local distance function [132] unique to NSDs based on *order statistics*. Given two nested descriptors p and q , the nesting distance $d(p, q)$ uses order statistics to partition the supports of two nested descriptors into *inliers* and *outliers* by sorting the squared differences up to a given maximum order k . Then, the nesting distance is equivalent to computing

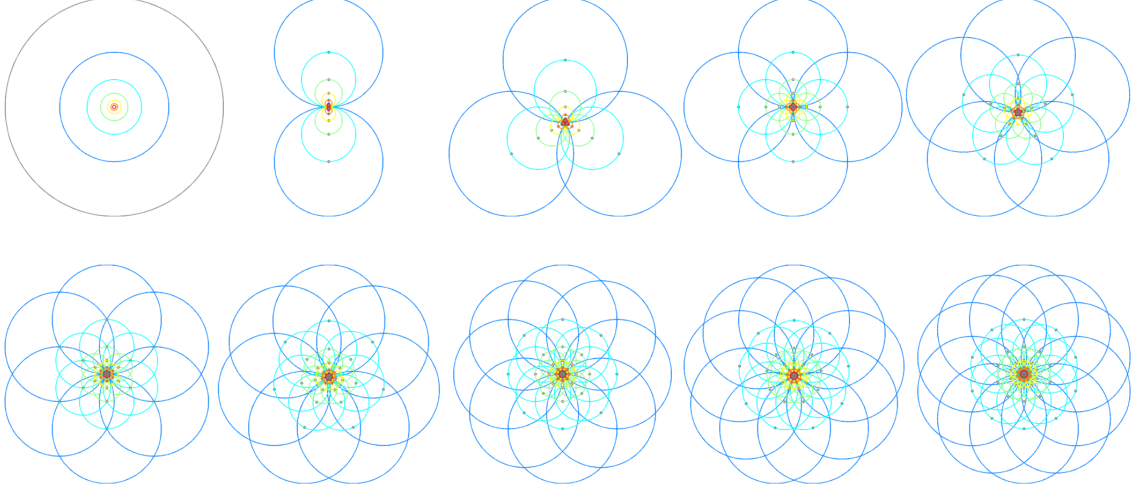


Figure 3.7: Nested shape descriptors with increasing lobes. (top row) $\mathbb{K}_1 - \mathbb{K}_5$, (bottom row) $\mathbb{K}_6 - \mathbb{K}_{10}$.

the conditional Gaussian distribution of inliers given outliers.

First, we introduce order statistics. Order statistics are a partial order of a set of variables $\{x_1, x_2, \dots, x_n\}$ such that $x_{(1)} \leq x_{(k)} \leq x_{(n)}$, where the k^{th} order statistic $x_{(k)}$ is equal to the k^{th} smallest value in the set. Common order statistics include the minimum $x_{(1)}$, maximum $x_{(n)}$ and median $x_{(n/2)}$. To simplify notation, we introduce an $n \times n$ binary diagonal selection matrix $S_{(j,k)}$ which encodes a selection of all variables greater than j -order and less than k -order.

$$S_{(j,k)}(i,i) = \begin{cases} 1 & \text{if } x_{(j)} \leq x_i \leq x_{(k)} \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

Observe that using this notation, $S_{(1,k)} + S_{(k+1,n)} = I$, where I is the identity matrix. Furthermore, if all variables are binary, then order statistics take on a simple form

$$x_{(k)} = \begin{cases} 1 & \text{if } \sum_{i=0}^n x_i \geq k, \quad x_i \in \{0, 1\} \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

which is equivalent to a thresholded Hamming distance.

The nesting distance is defined as follows. Let p and q be two nested descriptors of length n . Consider a partition of all squared differences $(p - q)^2$ for a given maximum k -order statistic. Let

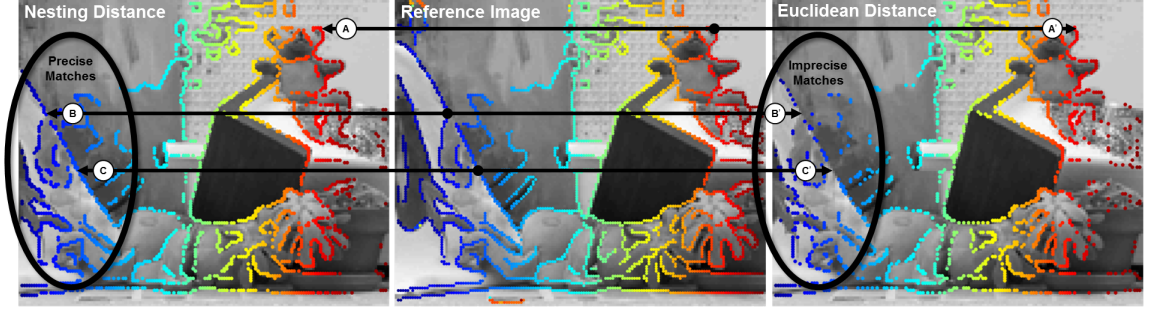


Figure 3.8: Example stereo image matching using the nesting distance and nested shape descriptors. Colors encode corresponding interest points between the reference image (middle) and the observed image using the nesting distance (left) and Euclidean distance (right). The Euclidean distance is affected by occlusions at the image boundary (left ellipse) resulting in local misalignments, while the nested distance is more robust to these occlusion effects.

this partition be represented by selection matrices of *inliers* $S_{(1,k)}$ and *outliers* $S_{(k+1,n)}$. Then, the nesting distance d is

$$d(p, q, \Lambda, k) = (p - q)^T (I - S_{(k+1,n)}) \Lambda S_{(1,k)} (p - q) \quad (3.6)$$

where Λ is an optional quadratic weighting matrix. If Λ is diagonal, then this simplifies

$$d(p, q, \Lambda, k) = (p - q)^T \Lambda S_{(1,k)} (p - q) \quad (3.7)$$

which is simply a sum of k smallest squared differences. Furthermore, if $k = n$ and $\Lambda = I$ then the nesting distance is equivalent to the Euclidean distance.

Lemma 3.3.5. *If the nesting distance is of the form (3.6), then it is equivalent to an unnormalized negative log likelihood of a conditional Gaussian distribution for inliers given outliers.*

Proof. We prove this property formally in section 3.3.6. Informally, consider a Mahalanobis distance $x^T \Lambda x \propto \mathcal{N}(0, \Lambda)$ as a negative log likelihood (unnormalized) of a Gaussian in canonical form. If a subset of variables are observed (e.g. outliers from order statistics), then well known Gaussian identities can update the conditional likelihood of the remaining variables (e.g. inliers) using the precision matrix Λ for distance weighting.

The nesting distance was designed specifically for the structure of the nested shape descriptor.

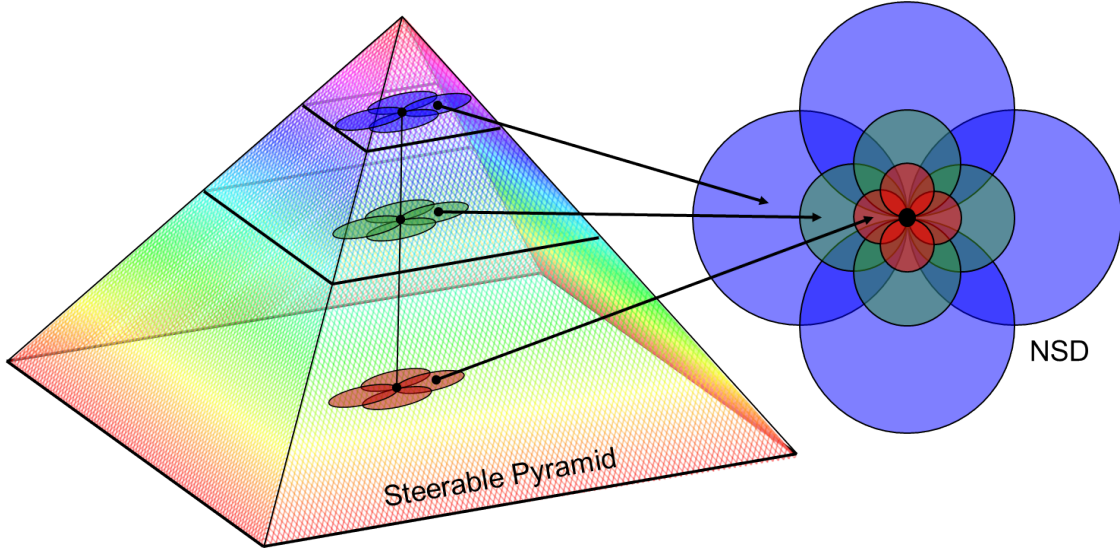


Figure 3.9: Construction of a nested shape descriptor. An NSD can be considered a “flattening” of the steerable pyramid. Supports of fixed sizes at different levels of the pyramid result in exponentially increasing descriptor supports.

First, recall that there exists exactly one point at the center of the NSD *that is in all supports*. Any subset of supports represents the shape of the center pixel at some orientation and scale. Therefore, this enables the use of order statistics to partition the supports into inliers and outliers, since all supports have one point in common.

The nesting distance cannot be used for descriptors with support constructed on a log-polar or Cartesian grid. Figure 3.3 (right) shows a simple counterexample. The majority of the green checkmarks or “good matches” are for supports with no variation on the background far from the center pixel. These matches match the background, are not descriptive for the corner at the center, and *would be the same if we remove the cube altogether*. In contrast, all supports of the NSD include the center pixel due to nesting, so any subset of supports, including large supports, capture the shape of the center pixel.

Figure 3.8 shows an example of the benefits of the nesting distance for image matching. We extract interest points using an edge based detector, compute nested descriptors at each point, then perform greedy minimum distance assignment from the reference to the observation using either the nesting distance or Euclidean distance. This example shows that the nested distance is more robust to occlusions at the image border than the Euclidean distance.

Finally, The nesting distance has two useful properties that are proven in section 3.3.6. First, the

nesting distance is non-metric, since it does not satisfy identity or the triangle inequality properties. This property matches perceptual experimentation as it has been long understood that perceptual distance and similarity functions are non-metric [142]. Second, the nesting distance is robust up to corruption of $n - k$ coordinates.

3.3.6 Proofs

In this section, we provide formal proofs for the lemmas referenced in the main body in chapter 3

Lemma 3.3.6. *If the nesting distance is defined as $d(p, q, \Lambda, k) = (p - q)^T (I - S_{(k+1, n)}) \Lambda S_{(1, k)} (p - q)$, then it is equal to an unnormalized negative log likelihood of a conditional multivariate Gaussian distribution.*

Proof. The proof follows by derivation of the nesting distance to the form of an unnormalized conditional Gaussian distribution.

First, preliminary definitions. A joint Gaussian distribution parameterized in canonical form is given by

$$p(x) = \mathcal{N}^{-1}(h, \Lambda) \quad (3.8)$$

$$f(x) = \frac{1}{2} x^T \Lambda x - h^T x \quad (3.9)$$

for information vector h and precision matrix Λ . The canonical form $\mathcal{N}^{-1}(h, \Lambda)$ is equivalent to the moment form $\mathcal{N}(\mu, \Sigma)$ using the identities $h = \Sigma^{-1} \mu$ and $\Lambda = \Sigma^{-1}$. The quadratic form (3.9) follows from the negative log likelihood of the joint density (3.8), and dropping the constant term.

Let variables x be partitioned into $x = [x_1 \ x_2]$ such that the Gaussian parameters can be partitioned

$$h = [h_1 \ h_2], \quad \Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix} \quad (3.10)$$

The conditional distribution $p(x_1 | x_2)$ can be derived from the joint distribution $p(x_1, x_2)$ using well

known identities.

$$\tilde{h} = h_1 - \Lambda_{12}x_2 \quad (3.11)$$

$$\tilde{\Lambda} = \Lambda_{11} \quad (3.12)$$

where $p(x_1|x_2) = \mathcal{N}^{-1}(\tilde{h}, \tilde{\Lambda})$ is the conditional likelihood of remaining variables x_1 given the observation x_2 [143].

Next, we derive a quadratic function g as the conditional likelihood of remaining variables given an observation. To simplify notation, define a selection matrix S that is a binary diagonal matrix that encodes the partition of variables, where $z_1 = S_1x$, $z_2 = S_2x$. With this notation, observe that $x = z_1 + z_2$, and $S_1 + S_2 = I$.

$$g(x) \propto -\log(p(x_1|x_2)) \quad (3.13)$$

$$g(x) = x_1^T \tilde{\Lambda} x_1 - 2\tilde{h}^T x_1 \quad (3.14)$$

$$= x^T S_1 \Lambda S_1 x - 2(S_1 h - S_1 \Lambda S_2 x)^T x \quad (3.15)$$

$$= x^T S_1 \Lambda S_1 x + 2x^T S_2 \Lambda S_1 x \quad (3.16)$$

$$= x^T (S_1 \Lambda S_1 + 2S_2 \Lambda S_1) x \quad (3.17)$$

$$= x^T ((S_1 + S_2) \Lambda S_1) x + x^T S_2 \Lambda S_1 x \quad (3.18)$$

$$= x^T (I + S_2) \Lambda S_1 x \quad (3.19)$$

This function g is unnormalized negative log likelihood of the conditional distribution, since it drops the constant normalization term.

Finally, the nesting distance d is

$$d(p, q) = (p - q)^T (I - S_{(k+1, n)}) \Lambda S_{(1, k)} (p - q) \quad (3.20)$$

Let the partition $z_1 = S_{(1, k)}x$ be the set of inliers and $z_2 = S_{(k+1, n)}x$ be the set of outliers deter-

mined from order statistics. Then,

$$d(p, q) = (p - q)^T (I - S_1) \Lambda S_2 (p - q) \quad (3.21)$$

$$d(p, q) = g(p - q) \quad (3.22)$$

$$d(p, q) \propto -\log(p(x_1|x_2)) \quad (3.23)$$

Lemma 3.3.7. *The nesting distance is non-metric.*

Proof. We show that the nesting distance satisfies non-negativity and symmetry, but not identity and triangle inequality. Non-negativity $d(P, Q) \geq 0$ is satisfied since all coordinates $(P_i - Q_i)^2$ in the sum are non-negative and real. Symmetry $d(P, Q) = d(Q, P)$ is satisfied since for all coordinates $(P_i - Q_i)^2 = (Q_i - P_i)^2$. Identity $d(p, q) = 0$ iff $p = q$ is not satisfied which can be shown with a simple counterexample. Let $p = [0 \ 0 \ 0]$ and $q = [0 \ 0 \ 1]$, then $d(p, q, \Lambda = I, k = 2) = 0$ but $p \neq q$. Finally, we show a counterexample for the triangle inequality. Let $P = [0 \ 0 \ 0]$, $Q = [0 \ 0 \ 1]$, $R = [1 \ 1 \ 1]$ then $d(P, R, \Lambda = I, k = 2) = 2$, $d(P, Q, \Lambda = I, k = 2) = 0$ and $d(Q, R, \Lambda = I, k = 2) = 1$. Therefore, $d(P, R) \not\leq d(P, Q) + d(Q, R)$ since $2 \not\leq 0 + 1$.

Lemma 3.3.8. *Assuming that P corresponds to Q , and $\Lambda = I$, $d(P, Q, \Lambda, k) = 0$ if and only if $\text{corruption}(Q) < \frac{k}{n}$.*

Proof.

In this section, “corruption” can be anything that distorts a descriptor such as occlusion, view-point, lighting or scale, introducing errors in squared differences in a coordinate during distance computation. Furthermore, a “correspondence” is a true matching of two descriptors P and Q for a given point in a scene.

Let $c = \text{corruption}(Q)$ be a nonzero modification of cN coordinates of Q , where $n = |Q|$. The proof follows from the definition of the nesting distance in that the sum includes the sum of the smallest k squared differences. The largest $n - k$ squared differences can be arbitrarily large without affecting the distance.

(\leftarrow): If $\text{corruption}(\mathbb{Q}) < \frac{k}{n}$, then at least k of the coordinates are uncorrupted. Since $P = Q$, an uncorrupted coordinate i has a squared distance $d(P_i, Q_i) = 0$. The bounds of the sum in the nesting distance are the smallest k squared differences, and since at least k are uncorrupted, and each uncorrupted coordinate has distance zero, the sum $d(p, q, \Lambda, k) = (p - q)^T \Lambda_{S_{(1,k)}}(p - q) = 0$.

(\rightarrow): If $d(P, Q, \Lambda, k) = 0$ then the sum of the smallest k squared differences is zero. Since each squared difference is non-negative, each coordinate of the smallest k squared differences must be zero. Therefore, since corruptions are non-zero modifications, the k coordinates are uncorrupted and $\text{corruption}(\mathbb{Q}) < \frac{k}{n}$.

Lemma 3.3.9. *Assuming that P corresponds to Q and exactly one central pixel q of Q is corrupted, then $\text{corruption}(\mathbb{Q}) = 1.0$ and $d(P, Q, \Lambda = I, k) > 0$ for all $k > 0$.*

Proof. Let $c = \text{corruption}(\mathbb{Q})$ be a nonzero modification of cN coordinates of Q , where $n = |Q|$. The central pixel q of the nested shape descriptor Q is the center of the nested support set as defined in (3.3.2). By construction, the smallest radius of the nested support set $r_1 = 0$ since the innermost support K_1 must contain only q . This implies that there exists exactly one point q that is contained within all supports. Therefore, if q is corrupted, then every support is corrupted. If every support is corrupted, then $\text{corruption}(\mathbb{Q}) = 1.0$, then from lemma 3.3.9 $d(P, Q) \neq 0$, and from the non-negativity property of lemma 3.3.7 $d(P, Q) > 0$.

3.3.7 Rotation Invariance

In this section, we describe an extension of the nested shape descriptor to a rotation invariant representation. Local feature descriptors are traditionally not rotation or scale invariant, rather interest point detectors are used to select salient points in an image with dominant orientation or scale. This dominant orientation and scale is used to normalize the descriptor to the canonical orientation and scale. Rotation invariance is useful for representation of objects with significant pose variation at a single scale, such as objects with in plane rotation.

The nested shape descriptor can be made rotation invariant by pooling over relative rotations. Figure 3.10 shows an overview of this approach. Recall that the NSD pools oriented gradients, where the scale of the circle defines the pooling region and scale. Each circle pools at set of R

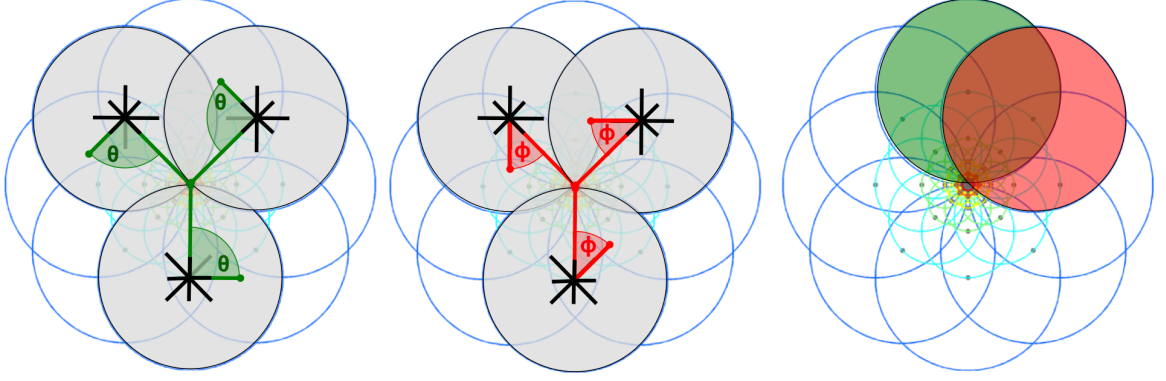


Figure 3.10: Rotation invariant nested shape descriptor. (left) Relative orientations between lobe and gradient orientation are pooled over all lobes and (right) each lobe is summarized with a single pooled relative orientation value.

orientations, shown as the $R = 8$ orientations as a black star in the center of each circle. Each point of this star represents the orientation of the gradients pooled in this region. Furthermore, each circle represents the Hawaiian earring lobe at a given orientation. Consider the *relative orientation* θ between the lobe orientation (θ_{lobe}) and the oriented gradient (θ_{grad}), such that $\theta = \theta_{lobe} - \theta_{grad}$ is constant. The constant relative orientations are pooled and each circle is represented by the pooled relative orientation, shown by the pooled red and green circles at figure 3.10 right. Finally, this pooled result is log-spiral normalized and this result is the final rotation invariant descriptor. By selecting the pooled relative orientations over only specific lobes, we can provide any subset of partial rotation invariance to full rotation invariance. This provides tradeoff between selectivity and rotation invariance based on the amount of rotation invariance desired.

Figure 3.11 shows an example of rotation invariant matching on a synthetically rotated image. This image pair undergoes an in-plane rotation, and we perform greedy image matching using the rotation invariant descriptor. The experimental setup for this result is dense interest point extraction (canny edges), rotation invariant descriptor extraction, exhaustive all pairs distance computation and greedy assignment. Figure 3.11 shows the greedy assignment where colors encode corresponding interest points. Observe that this assignment example does not include any dominant orientation estimation. Interest points can be computed densely without selecting interest points according to dominant orientation.

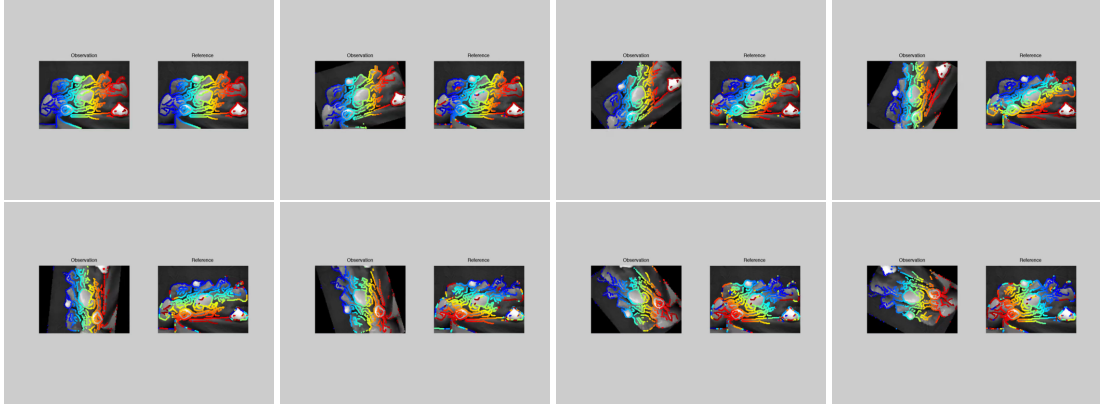


Figure 3.11: Rotation invariant nested shape descriptor.

3.4 Experimental Results

In this section, we provide experimental results for the nested shape descriptor and nesting distance for the task of image matching. First, we perform a trade study using the new experimental protocol of *similarity stereo matching* to determine an optimal set of descriptor parameters for the seed-of-life descriptor. Next, we compare results for the seed-of-life and binary seed-of-life descriptor for the standard VGG-Affine benchmark [74] against SIFT [35] and BRISK [43]. Finally, we show results on a challenging application for which traditional local feature descriptors are not applicable. A Matlab toolbox to reproduce these experiments is available at <https://github.com/jebyrne/seedoflife>.

3.4.1 Experimental System

In this section, we describe the experimental system used to construct seed-of-life descriptors. The subbands B for a nested shape descriptor are scaled and oriented gradients derived from a complex steerable pyramid [141]. The complex steerable pyramid includes steerable filters in a quadrature pair whose magnitude and phase response are useful for representing signed orientations for "black to white" vs. "white to black" transitions. A Matlab toolbox for building and decomposing separable complex steerable pyramids is available at <https://github.com/jebyrne/sepspyr>.

Max-pooling is performed by max-filtering and sampling and steerable pyramid. First, all bands and scales of the steerable pyramid are 7×7 max-filtered. Then, for each interest point p , we

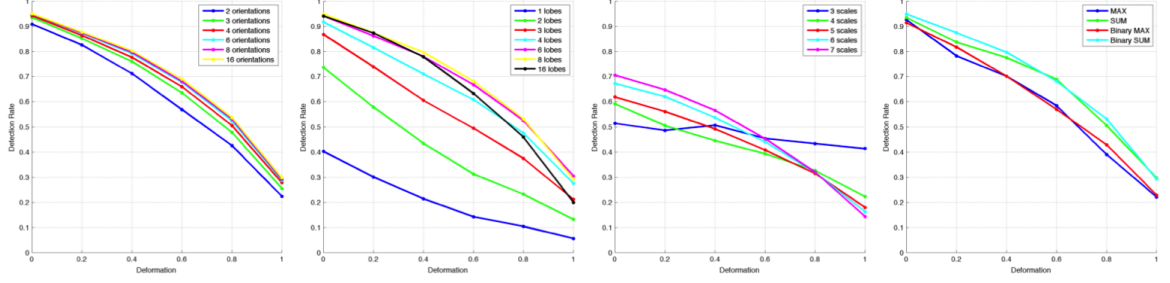


Figure 3.12: Trade study for the seed-of-life descriptor. (left-right) parameter analysis of orientations, lobes, scales and pooling.

construct lobes by uniform polar sampling of each band at n -orientations at a radius of 3 from p . This sampling proceeds cumulatively over scales, and if a lobe is outside the image, then the cumulative pooling simply uses the nearest valid response. Observe that a 7×7 max-filter at scale i is equivalent to a max-pooled support of size $7 * 2^i$ which allows supports to grow exponentially in size without an exponentially increasing number of pixels in each lobe. Sum-pooling is performed by 7×7 max filtering, followed by summing over spatial support to construct a histogram rather than sampling.

A nested shape descriptor can be similarity normalized using a similarity invariant local feature detector. Given a dominant orientation r^* from a feature detector, a normalizing similarity transform is applied to the seed-of-life pooling structure \mathbb{K} for each interest point. Then, orientation bands are circularly shifted and linearly interpolated such that $D(i', j, k) = \hat{D}(i, j, k)$ and $i' = (i - r^*) \bmod (|R|)$. An analogous approach is used for scale normalization.

A Matlab toolbox for constructing seed-of-life nested shape descriptors is available at <https://github.com/jebyrne/seedoflife>.

3.4.2 Middlebury Stereo and Trade Study

In this section, we perform a trade study to determine an optimal set of parameters for the seed-of-life descriptor. The parameters under study were the number of orientation bands, number of lobes, number of scales, pooling strategy, and binary vs. floating point descriptor elements. We performed a set of studies to understand the effect of these parameters on descriptor performance for the task of image matching.

The experimental protocol for performance evaluation is detection rate in *similarity stereo*

matching. We use six images from the Middlebury stereo dataset [144] (teddy, cones, venus, tsukuba, map and sawtooth). Given a stereo pair (I, J) , ground truth disparity D and similarity transform A , we construct a similarity stereo pair (I, J') such that $J' = A(J)$ by applying the similarity transform A to J . Then, corresponding interest points (p, q) in the similarity stereo pair (I, J') satisfy $p = A^{-1}q + D_p$. Correspondences are the composition of stereo disparity and a similarity transform.

The similarity stereo matching uses the repeatability evaluation protocol of [74] for a range of increasing similarity distortions (scale=0.5-1.5, rotation= $\pm \frac{\pi}{16}$). Random similarities are sampled 10 times for each image at the deformation level and the mean detection rate over all six images for each deformation magnitude is shown.

Figure 3.12 shows the results of this study. First, we analyzed the effect of the number of orientations. Increasing the number of orientation subbands offers a modest improvement, up to diminishing returns at eight bands. Second, we analyzed the effect of the number of lobes, and found that increasing significantly improves performance up to eight lobes. Third, we analyzed the effect of scales, and found that scale is inversely correlated with deformation. For small deformations, larger scales perform better, but for larger deformations smaller scales perform better. This result summarizes the known tradeoff between descriptor support and matching performance as was discussed in section 3.1. We selected seven scales. Fourth, we analyzed the pooling strategy and found that sum-pooling (e.g. orientation histograms) had a dramatic improvement over max-pooling for image matching.

The conclusions of this study are a selection of a nominal parameter set. We use eight unsigned orientations, eight lobes, seven scales and sum-pooling. We use these parameters for all experiments in this paper.

3.4.3 VGG-Affine

We show comparative performance for local feature descriptor matching on the VGG-Affine benchmark [74]. This dataset includes images of five distortion classes including blur, viewpoint, scale/rotation, ambient light and JPEG compression. Each distortion class is represented by six images such that the distortion gets progressively worse, and a ground truth homography for performance comparison of local feature descriptors for the task of image matching.

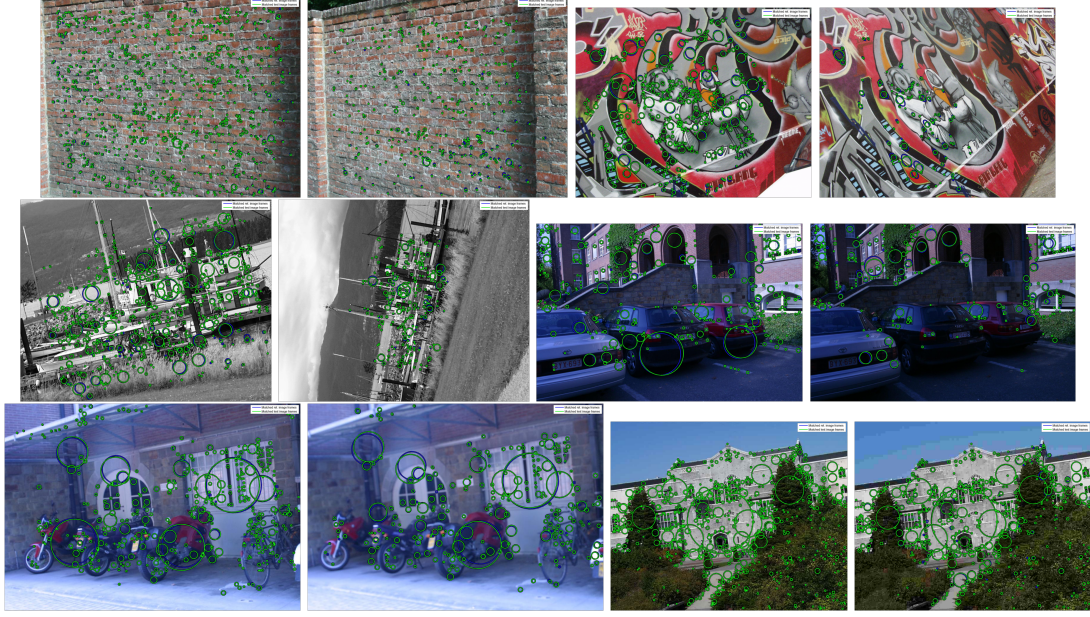


Figure 3.13: Example matching results from the VGG-Affine dataset.(top-bottom) wall, graf, boat, leuven, bikes, ubc

The experimental protocol is as outlined by Mikolajczyk and Schmid [74]. Performance evaluations for local feature descriptors was performed using the matching score criterion (feature match recall). We compare the performance of seed-of-life (SOL) and binary SOL descriptor (section 3.3.3) to SIFT [35] and BRISK [43]. The seed-of-life is identical to the binary seed-of-life but without the final binarization step of eq. (3.3). Both SOL and Binary SOL use the Euclidean (and Hamming) distance, as we evaluate the effect of the nesting distance separately in section 3.4.4. We use a dominant orientation and difference of Gaussians (DoG) scale space feature detector for SIFT and NSD, and the AGAST detector [145] for BRISK. All parameters are defaults provided by the authors, and the parameters for NSD are determined from the trade study in section 3.4.2, with $k = 0.7n$ for nesting distance. We use the analysis tools and software provided by [74][146][43], and we leave out “bark” for consistency with previous work [43]. However, “bark” results are provided separately below for completeness.

Performance results are shown in figure 3.14. These results show that either seed-of-life (SOL) and Binary-SOL outperform SIFT and BRISK for all distortion classes. Furthermore, the binary SOL and SOL descriptor perform equally, which shows that the binarization provides a more compact descriptor without sacrificing performance.

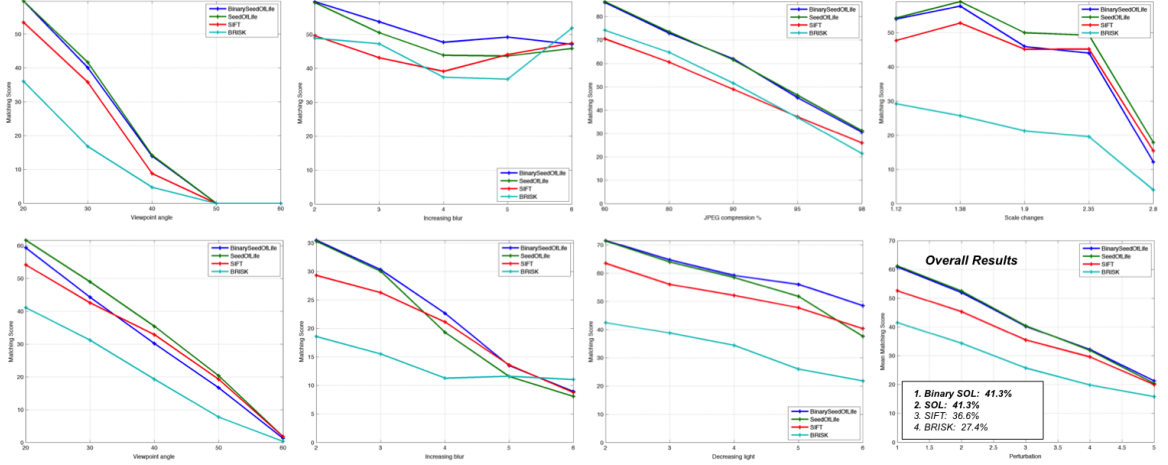


Figure 3.14: VGG-Affine image matching results. (top) “graf”, “bikes”, “ubc”, “boat”, (bottom) “wall”, “trees”, “leuven” and composite. Both SOL and BSOL outperform SIFT and BRISK, and Binary-SOL is the first binary descriptor to outperform SIFT on this benchmark.

Figure 3.13 shows imagery and example feature matching from the VGG-affine dataset. These examples show matched features using NSD and nesting distance for image 2 and image 4 in a subset of distortion classes.

Figure 3.15 shows the matching score for the “bark” example. This example is commonly left out of evaluations of the VGG-Affine dataset as discussed in the main results, since as you can see competing descriptors often perform poorly on this example. However, the results show that the seed-of-life descriptor is competitive with SIFT.

3.4.4 Local Distance Functions

Next, we performed a comparison of the nesting distance vs. the Euclidean distance on the VGG-Affine benchmark. This evaluation was proposed to demonstrate the relative benefit of the nesting distance over the Euclidean distance baseline.

Figure 3.16 shows the results of this study. All distortion classes showed improved performance of the nesting distance over Euclidean. Figure 3.16 shows matching performance plots for the three distortion classes with the largest benefit, blur (“bikes” and “trees”) and decreasing light (“leuven”). The overall performance shows a 4.1% improvement for the nesting distance.

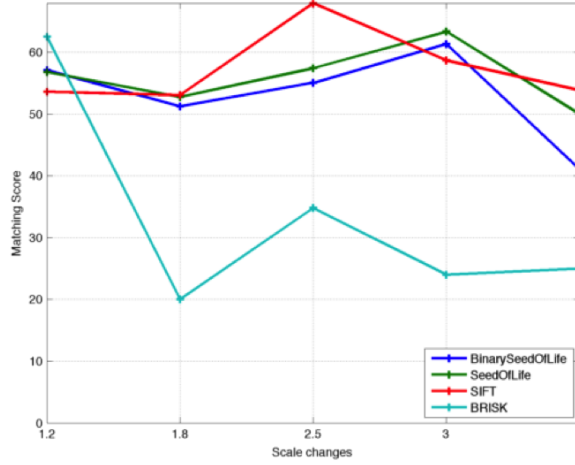


Figure 3.15: Matching score for “bark” in the VGG-Affine dataset

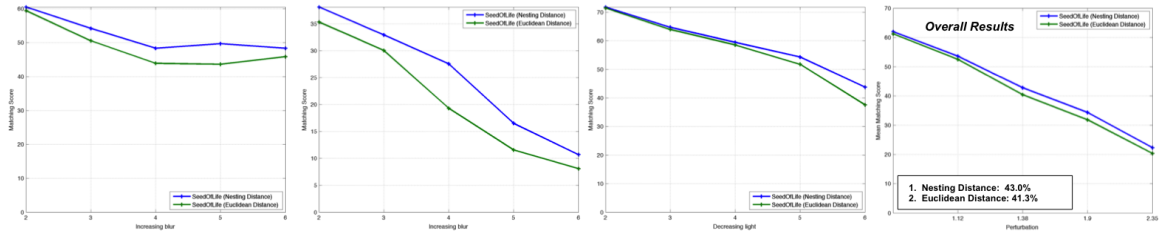


Figure 3.16: Evaluation of the nesting distance on VGG-Affine dataset. See text for a discussion.

3.4.5 Photorealistic Virtual City

Next, we performed a set of experiments using the Photorealistic Virtual City (PVC) dataset [147]. The PVC dataset is a synthetic wide baseline stereo dataset that was designed to study how feature matching performance using local feature descriptors degrade given controlled changes in the lighting, scene, and viewing conditions. Unfortunately, dense ground truth is difficult and expensive to gather for such evaluations in controlled manner. Instead, this dataset uses a photorealistic virtual world to gain complete and repeatable control of the environment in order to evaluate image features. Raytraced rendering is used to study the effects on descriptor performance of controlled changes in viewpoint and illumination. This synthetic dataset has been validated by comparing matching performance on rendered imagery and comparing matching performance to actual imagery of the same scene, and results have shown approximately equivalent performance. This justifies the use of a synthetic dataset to predict performance on natural imagery.

This dataset contains 3000 640x480 color PNG images, over four scenes in an dense urban

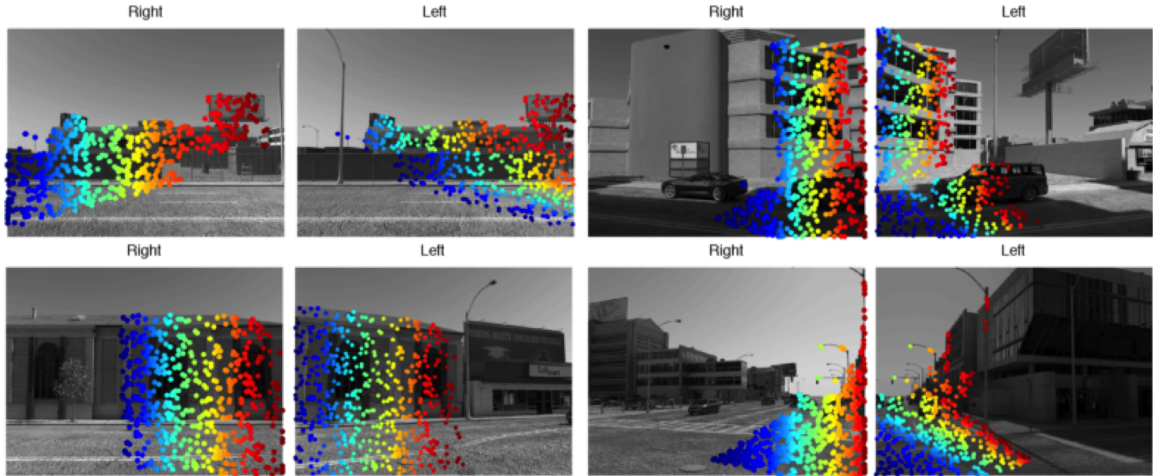


Figure 3.17: Example ground truth correspondence for rendered imagery in the PVC dataset. Shown are a random subset of five hundred pixels such that colors encode corresponding pixels between the right and left images. (top left) camera 1 with translation correspondence (bottom left) camera 2 with rotation correspondence (top right) camera 3 with translation and rotation correspondence (bottom right) camera 4 with translation and rotation correspondence.

environment. Each scene is represented by an image sequence of a translating camera along a city street, such that each image overlaps. At each translation, the camera also rotates by plus and minus 22.5 degrees to provide both orientation change and translation change. Finally, at each camera pose, the lighting is varied by time of day for a sunny august day for five different times at two hour intervals from 9am to 5pm. This provides controlled lighting changes for each scene. No additional noise is added to the rendered imagery, which provides an idealized controlled scenario to evaluate matching using local feature descriptors.

Figure 3.17 shows example imagery from this dataset. We show the ground truth correspondence between a subsampled set of pixels in the right image and the corresponding pixels in the left image, determined from the ground truth range from the virtual city. Correspondences are encoded by color, such that red pixels in the right match red pixels in the left. This image shows examples from each of four cameras, with correspondences consistent with translation only, rotation only and translation and rotation. These images are all shown at the constant time of day of 9am.

The experimental protocol for evaluation on this dataset was greedy matching score given exact correspondence. For each overlapping image pair, we extracted the ground truth correspondence for each pixel in the right image to the corresponding pixel in the left image. We selected a set of 500 correspondences at random, and extracted local feature descriptors for the corresponding

locations in both the left and right image. The descriptors are computed at canonical scale and rotation. Finally, we compute an exhaustive pairwise distance computation, and perform greedy bipartite matching to assign matches from the right to the left. We define a correct match to be a match to within 10 pixels of the ground truth correspondence. The matching score is the total number of correct matches divided by the total number of matches.

This experimental protocol enables isolated analysis of the effects of descriptors only on matching performance. Recall that the VGG-Affine dataset evaluation includes both local feature detectors, which provides affine invariant keypoints with local feature descriptors to compute a final matching score. This score is affected by the quality and accuracy of the keypoint extraction, which can conflate the effects of the matching score with the descriptor performance and the detector performance. In this evaluation, we decouple the detectors and descriptors by including the ground truth correspondences in as "detectors", then compute the descriptors for these ground truth correspondences. Therefore, the matching performance is a function of the descriptors only. This allows conclusions to be drawn about the effect of the descriptors only on matching performance.

In this section, we show the matching performance as a function of translation, rotation or translation and rotation. Furthermore, we show the mean matching performance and the matching performance as a function of time of day. We compare performance of the nested shape descriptor with DAISY [42], SIFT [35], ORB [45], BRISK [43] and FREAK [46]. In this descriptor comparison, DAISY and SIFT are real valued descriptors, while ORB, BRISK, FREAK and NSD are binary valued. The DAISY descriptor was specifically designed and optimized [41] for wide baseline stereo matching. Our results show that NSD outperforms all descriptors in all experiments, which provides a basis of confidence for concluding that the NSD is a state-of-the-art descriptor for wide baseline stereo matching. Our experimental evaluation code is available for download at <https://github.com/jebyrne/PhotorealisticVirtualCity>.

3.4.5.1 Translation Evaluation

First, we performed an evaluation for wide baseline binocular stereo. For each camera, we consider pairs of images that overlap and that are related by a translation only, such as the correspondences shown in figure 3.17 (top left). This scenario models a calibrated and rectified wide baseline stereo configuration such that epipolar lines are aligned with scanlines. Figure 3.18 shows the mean match-

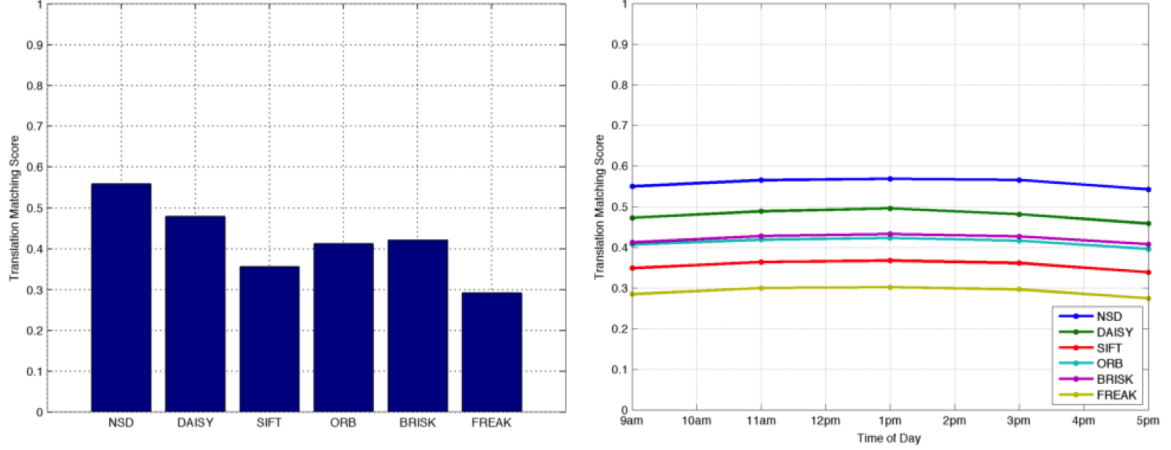


Figure 3.18: Photorealistic Virtual City - Translation only results. (left) Aggregate (right) Time of day

ing score for each descriptor over all cameras and time of day, as well as the mean matching score as a function of time of day. In all cases, our NSD outperforms all local descriptors in all scenarios, including the DAISY descriptor that was specifically designed for wide baseline stereo matching.

Figure 3.19 shows the detailed matching performance for each pair of overlapping images at at given position for each camera in the dataset. The plots in figure 3.18 (left) were constructed by computing the mean over all four plots in this figure. This shows that the NSD is consistently outperforming the other descriptors across all cameras.

3.4.5.2 Rotation Evaluation

Next, we performed an evaluation for a rotational homography. For each camera, we consider pairs of images that are formed by rotating the camera by plus and minus 22.5 degrees in yaw. An example of this rotation scenario is shown in figure 3.17 (top right). Figure 3.20 (left) shows the mean matching score over all cameras for each descriptor, and Figure 3.20 (right) shows the mean matching score as a function of time of day. In this scenario, NSD outperforms all other descriptors, however the performance of DAISY is quite close.

Figure 3.21 shows the matching performance per image for each camera. This shows that the NSD is consistently outperforming DAISY on each image and not just in aggregate performance.

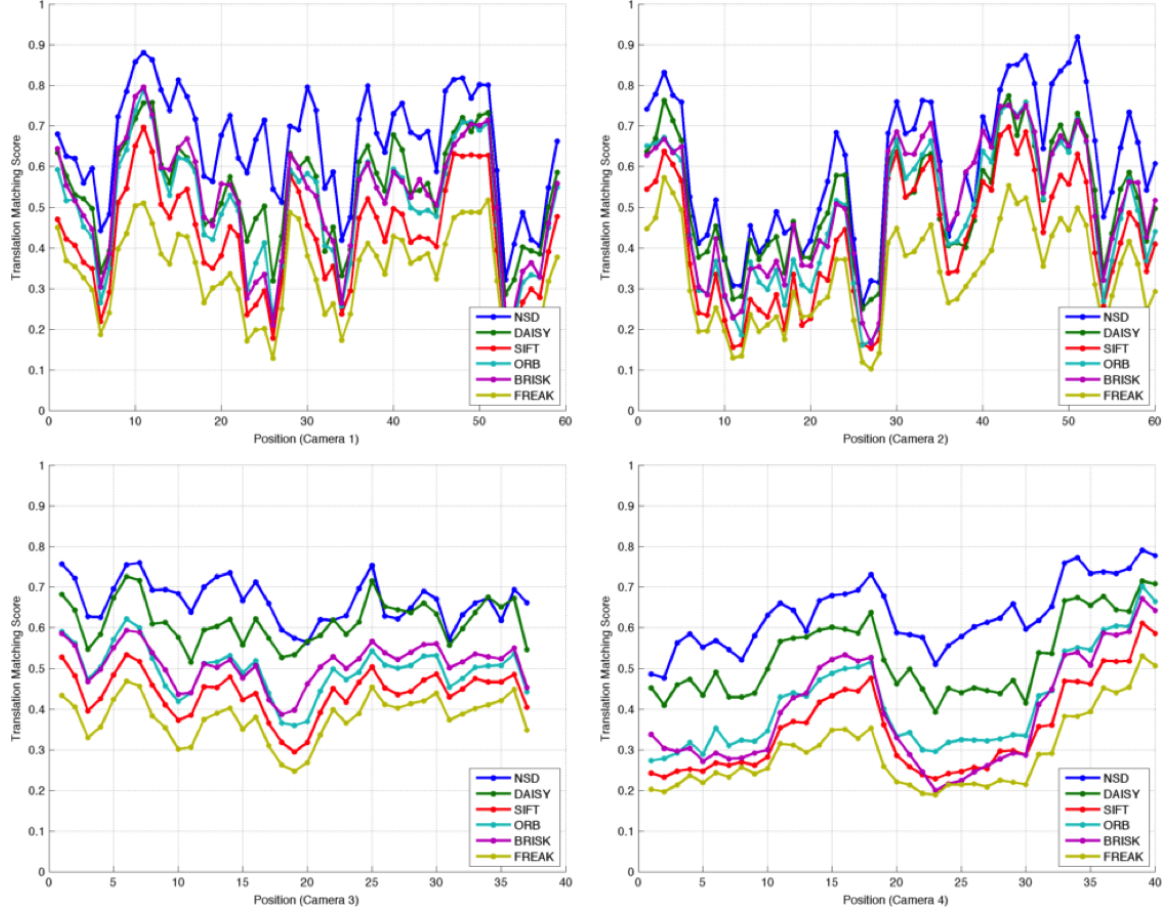


Figure 3.19: Photorealistic Virtual City - Translation only results per location

3.4.5.3 Translation and Rotation

Next, we performed an evaluation for a combined rotational homography and translation. For each camera, we consider pairs of images that are formed by rotationing the camera by plus 22.5 degrees then translating the camera and rotating by -22.5 degrees. This scenarios is the combination of the translation only and rotation only cases evaluated above.

Figure 3.22 (left) shows the mean matching score for each descriptor over all cameras. Figure 3.22 (right) shows the mean matching score for each descriptor as as function of the time of day.

Figure 3.23 shows the detailed results for translation and rotation. This shows that the NSD is consistently outperforming DAISY on each image and not just in aggregate performance.

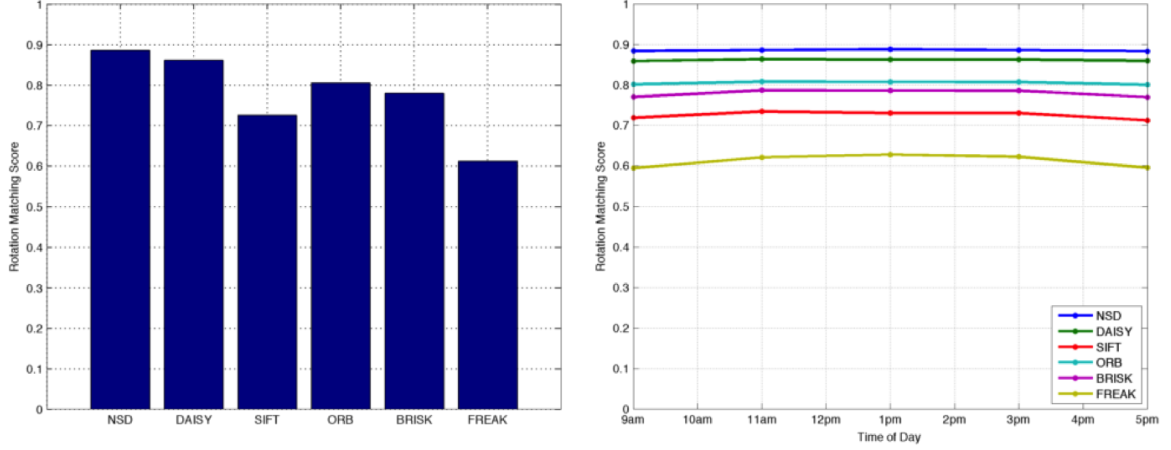


Figure 3.20: Photorealistic Virtual City - Rotation only results. (left) Aggregate (right) Time of day

3.4.6 Storage Weighted Matching

Next, we performed an analysis comparing the matching score as a function of storage requirements for each descriptor. The storage requirements per descriptor are as follows: NSD=64 bytes, DAISY=800 bytes, SIFT=128 bytes, ORB = 32 bytes, BRISK=64 bytes, FREAK=64 bytes. The large storage requirements for DAISY are due to the fact that this descriptor is a real valued 200 dimensional descriptor such that each dimension requires a 32 bit floating point number for storage. We define a *storage weight* as a scale factor relating the storage requirements relative to SIFT, such that a storage weight $w = \exp(-(b - 32)^2 / 128^2)$ is defined for each byte requirement b . This weight is normalized to be in the range $[0, 1]$ such that the minimum storage requirement (ORB) has weight one. This storage weight is used to weigh the matching score for the translation evaluation from section 3.4.5.1. The result is a *storage weighted matching* used to compare the relative performance of the descriptors taking in to account the storage requirements necessary to achieve a given matching score.

Figure 3.24 shows the results for storage weighted matching. Observe that the performance rank for storage weighted matching changed from [1. NSD, 2. DAISY, 3. BRISK, 4. ORB, 5. SIFT, 6. FREAK] to [1. NSD, 2. ORB, 3. BRISK, 4. FREAK, 5. SIFT, 6. DAISY]. The large storage requirements for SIFT and DAISY caused these descriptors to fall in rank to the binary descriptors (ORB, BRISK, FREAK). However, in all cases, the NSD remains the top performing descriptor in both overall matching performance and storage weighted matching performance. This shows that

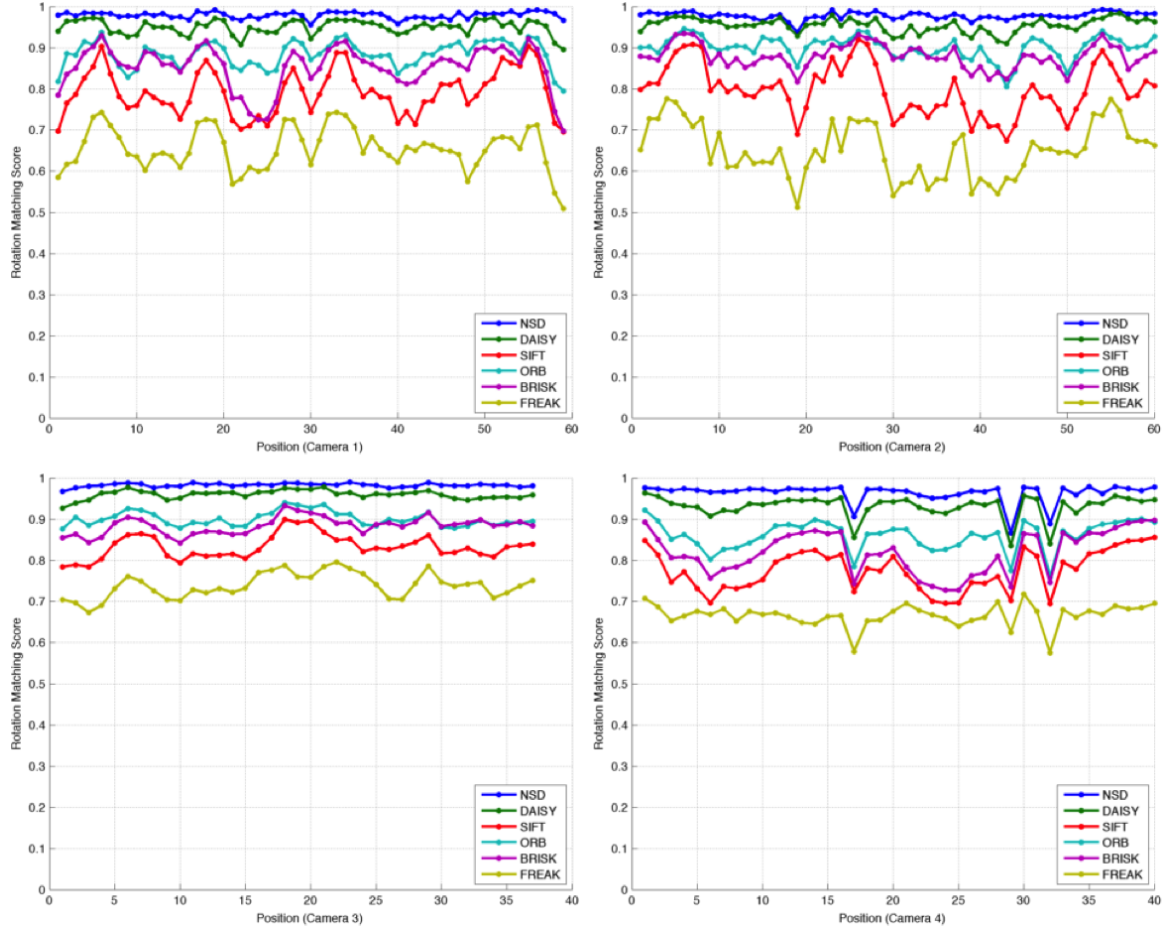


Figure 3.21: Photorealistic Virtual City - Rotation only results per location

NSD is the state-of-the-art *compact* descriptor for image matching.

3.4.7 Dense Strided Descriptors

Finally, we performed an analysis to determine the matching performance under dense descriptor extraction. Dense extraction considers descriptors computed at every pixel location, or at subsampled pixel locations with uniform spacing or *stride*. An example is shown in figure 3.25. Recent analysis has shown [148, 149, 42] that matching performance is significantly improved when the interest point detectors are bypassed and dense interest points on the subsampled image lattice are extracted in both images. Performance is significantly improved performance over methods relying on affine invariant interest point extraction. Therefore, a successful descriptor must be robust to stride variations.

The experimental protocol for this analysis is matching score given dense interest points. We

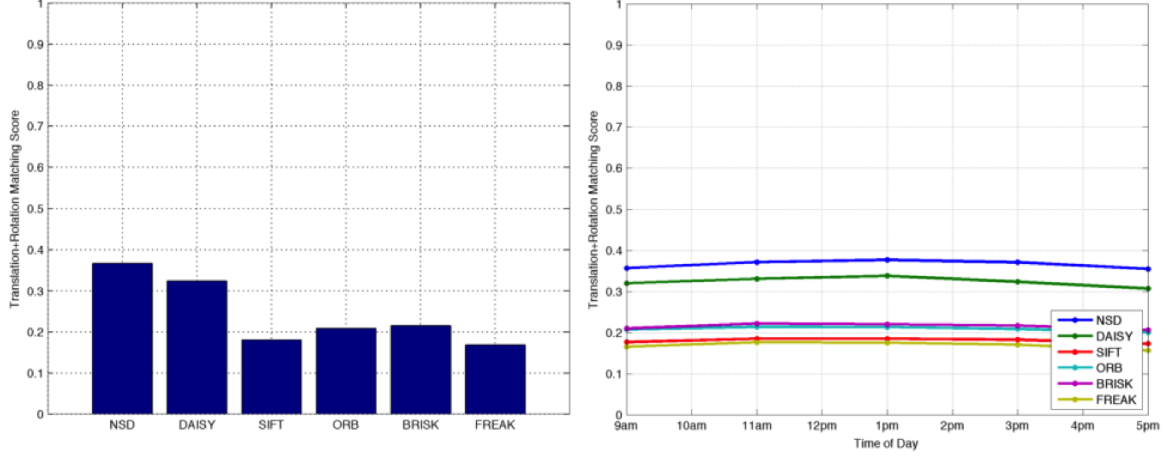


Figure 3.22: Photorealistic Virtual City - Translation and Rotation results. (left) Aggregate, (right) Time of day

extract interest points at strides of 2, 4, 8, 16, 32, 64 in x and y at the same locations in both images. We consider the subset of interest points that are visible in both images, perform descriptor extraction at each interest point, compute a distance for all pairs of descriptors, then compute a greedy bipartite matching. We show the matching score as a function of stride for the translation only case described in section 3.4.5.1.

Figure 3.26 shows the same ranking as displayed in the previous examples with the NSD as the top performer and DAISY as the second best performer. This example shows that the performance of SIFT, BRISK and FREAK have a faster falloff than ORB, DAISY and NSD. Furthermore, as the stride increases the matching performance decreases, up to a stride of 2^4 . At this point, the matching performance for NSD increases faster than DAISY and the others. This shows that the NSD shows better performance as stride increases than the state of the art.

3.4.8 Saliency

The experimental results presented have shown that the nested shape descriptor is a state of the art local feature descriptor. However, can we do a better job in motivating the structure of the descriptor? For example, why do we perform the log-spiral normalization step when constructing the descriptor? Why does this spiral structure work so well? In this section, we show that the log-spiral normalization is computing a center surround difference in scale which is a form of *bottom up saliency*. So, the nested shape descriptor is not simply a pooling of scaled and oriented edges,

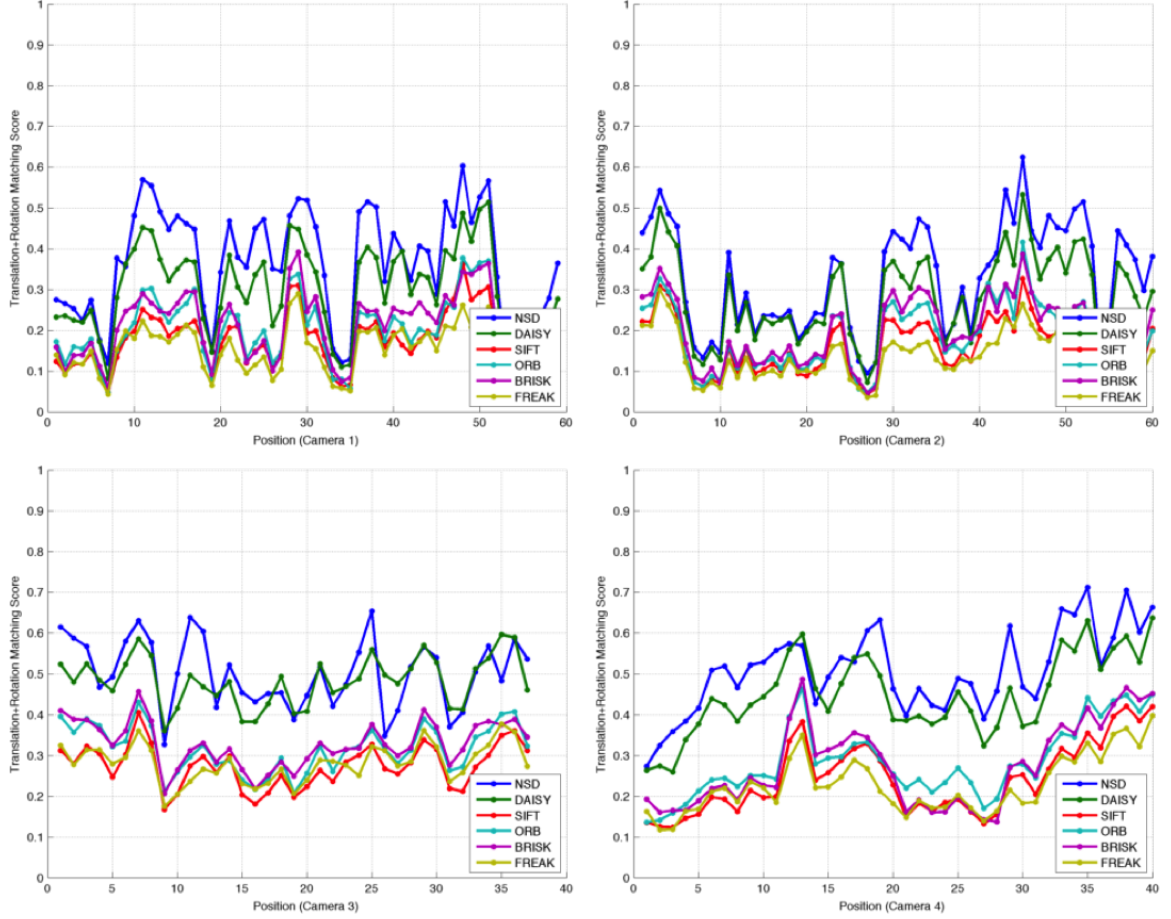


Figure 3.23: Phorealistic Virtual City - Rotation and Translation results per location

but rather it is a representation of *salient edges*. This suggests that the representational power of the nested descriptor is due to representation of saliency, which makes a fundamental connection between the saliency and local feature descriptor literature.

Saliency is a measure of “interesting-ness” or visual features that attract the attention of visual observers. Salient features are said to *pop-out* from the background, and can be used to prioritize candidate object detections to apply finite resources for higher level reasoning. The seminal work of Itti and Koch [150] described salient features in terms of center surround differences in color, grayscale intensity and orientation. A center surround difference in oriented gradients computes derivatives in scale and position of the oriented gradients as a measure of the saliency of a region.

The nested shape descriptor can be motivated in terms of bottom up saliency. The log-spiral normalization step of the nested shape descriptor computes the difference between neighboring

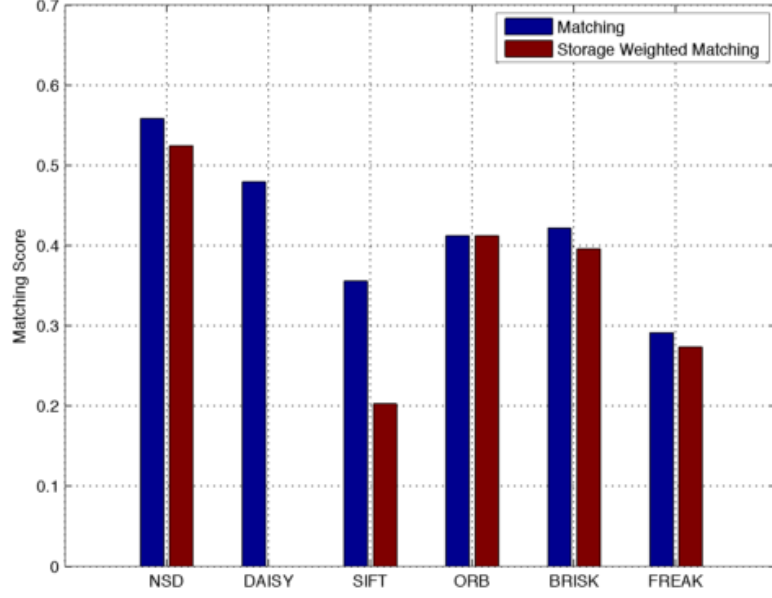


Figure 3.24: Photorealistic Virtual City - Storage Weighted Matching Results

scales and positions in the steerable pyramid.

$$\hat{d}(i, j, k) = d(i, j, k) - d(i, j-1, k-1) \quad (3.24)$$

Equation (3.24) shows this log-spiral normalization, where $d(i, j, k)$ is the pooled response at orientation subband i , lobe j and lobe scale k . The difference is $d(i, j-1, k-1)$ with the smaller scale $k-1$ and neighboring lobe $j-1$. Therefore, this is a difference between neighboring scales, forming a type of center surround difference. Intuitively, this operation highlights *changes* in oriented gradients in scale and position, which is a classic low-level measure of bottom up saliency.

We can demonstrate that the nested shape descriptor is computing salient edges by using it to construct a *saliency map*. A saliency map is a real valued scalar field that encodes the salience of regions in an image. The nested shape descriptor can be used to compute a saliency map in a very simple manner. Recall that the nested shape descriptor requires the construction of a quadrature steerable pyramid to compute multiscale oriented gradients. Given this pyramid, replace the orientation and scale bands with the clipped mean square response of the NSD for each orientation and lobe. Then, replace the low pass response of the steerable pyramid with the squared Laplacian filter response, to implement a center surround difference. Finally, reconstruct the image from this saliency pyramid. In short, *a saliency map is the image reconstructed from the squared response of*

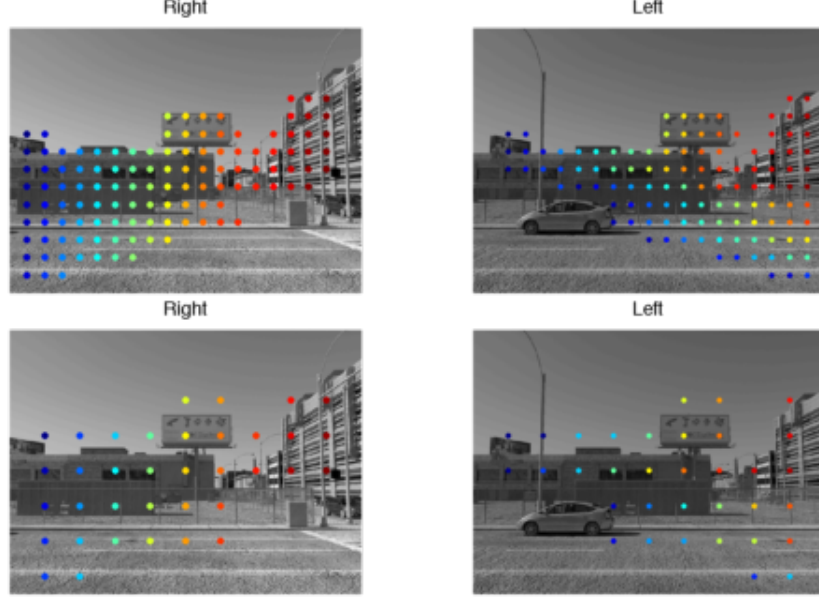


Figure 3.25: Photorealistic Virtual City - Dense Stride. (top) Stride=32, (bottom) Stride=64. Colors encode matching densely extracted interest points for each stride. See text in section 3.4.7 for a discussion.

the nested shape descriptor:

Formally, let a steerable pyramid $B = \{I_0, B_{ij} ; i \leq R, j \leq S\}$ for orientation bands B_{ij} over R orientations i and S scales j and lowpass residual image I_0 . Each band B_{ij} encodes the oriented gradient response at orientation i and scale j . Furthermore, let \hat{d} be a log-spiral normalized nested shape descriptor constructed following eq. 3.1 and 3.2, computed densely at each pixel. Then, let

$$\hat{B}_{ij} = \max(\sum_j \hat{d}(i, j, k)^2, \tau) \quad (3.25)$$

$$\hat{I}_0 = (I_0 * L)^2 \quad (3.26)$$

where L is a 3x3 Laplacian kernel, $*$ is the convolution operation, and τ is a clipping threshold for the maximum squared difference. These are collected as subbands in a steerable pyramid $\hat{B} = \{\hat{I}_0, \hat{B}_{ij}\}$, and these bands are used to reconstruct an image using the standard steerable pyramid reconstruction algorithm, where the filters used for reconstruction are the magnitude of the quadrature pair. This reconstructed image is a saliency map. A Matlab toolbox to construct this saliency map is available at <https://github.com/jebyrne/seedoflife>.

Figure 3.27 shows four examples of the saliency map for a set of classic pop-out images. These

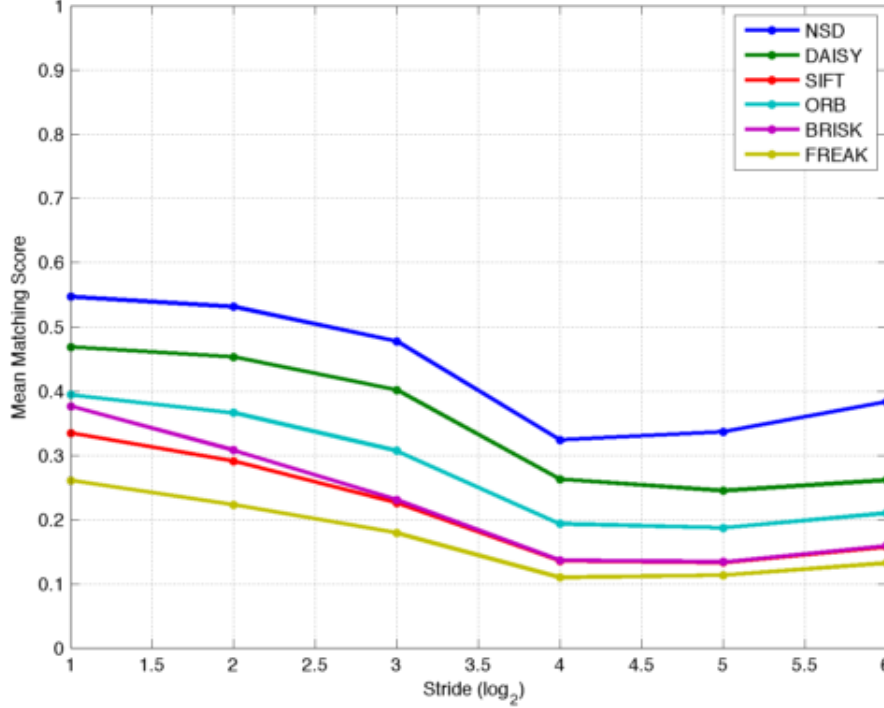


Figure 3.26: Dense Stride evaluation. See text in section 3.4.7 for a discussion.

images show a semi-transparent saliency map overlayed with the original grayscale image, such that red encodes "salient" and blue is "not-salient". The global maximum is shown with a black plus, which encodes the most salient position in the image. These results show that the most salient location is equal to the pop-out location according to convexity, orientation or intensity features. This shows that the descriptor is representing *salient edges*, and not just histograms of oriented gradients.

Figures 3.28 and 3.29 show example saliency maps from two academic saliency datasets. We computed saliency maps for the MIT saliency benchmark [151] and the MSRA Salient Object database [152]. Modern evaluations of saliency include both bottom up and top down context for constructing and evaluating saliency maps. In this section, we are interested in showing the saliency for bottom up features only, so we show qualitative results only rather than compare against the state-of-the-art.

Finally, we observe that the saliency construction methodology described here is unique to the nested shape descriptor. Recall from the construction of the NSD described in figure 3.9, the nested shape descriptor can be considered to be a "flattening" of the steerable pyramid. This structure pro-

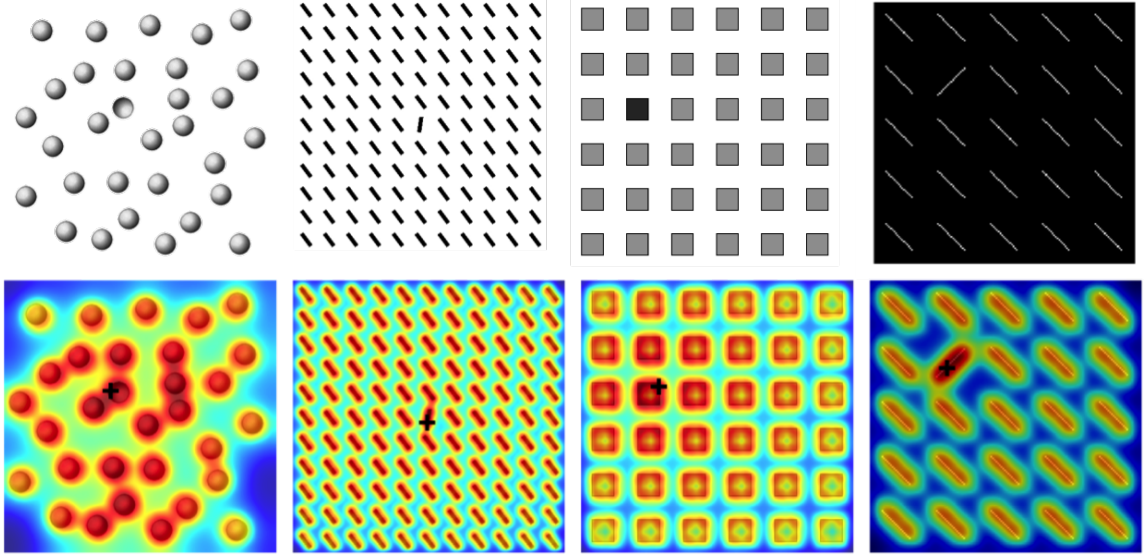


Figure 3.27: Popout examples for NSD saliency map. (left to right) Convexity, orientation, contrast, orientation. (top) input image, (bottom) saliency map where red is high saliency and blue is low saliency, and maximum saliency shown with a black '+’.

vides a mapping between pooling regions of the nested descriptor and scale/orientation coefficients in the steerable pyramid. This mapping provides a means of visualizing the nested shape descriptor by leveraging the reconstruction property of the steerable pyramid, but replacing the pyramid coefficients with NSD pooled and log-spiral normalized coefficients then performing reconstruction.

3.5 Summary

In this chapter, we introduced the nested shape descriptor family and the associated nesting distance, and showed performance of the seed-of-life descriptor for the task of image matching. Results show that this is the *first binary descriptor* to outperform SIFT on the standard VGG-Affine benchmark. Furthermore, the NSD binary descriptor significantly outperforms BRISK, a state-of-the-art binary descriptor and DAISY a state of the art wide baseline stereo matching descriptor. Future work includes exploring other members of the NSD family such as the flower-of-life or fruit-of-life for improved performance.

Acknowledgements and Disclaimer. This material is based upon work supported in part by DARPA under Contract No. W31P4Q-09-C-0051 with Kitware Inc. and NAVAIR contract No. N68335-12-C-0069 with Scientific Systems Company, Inc.

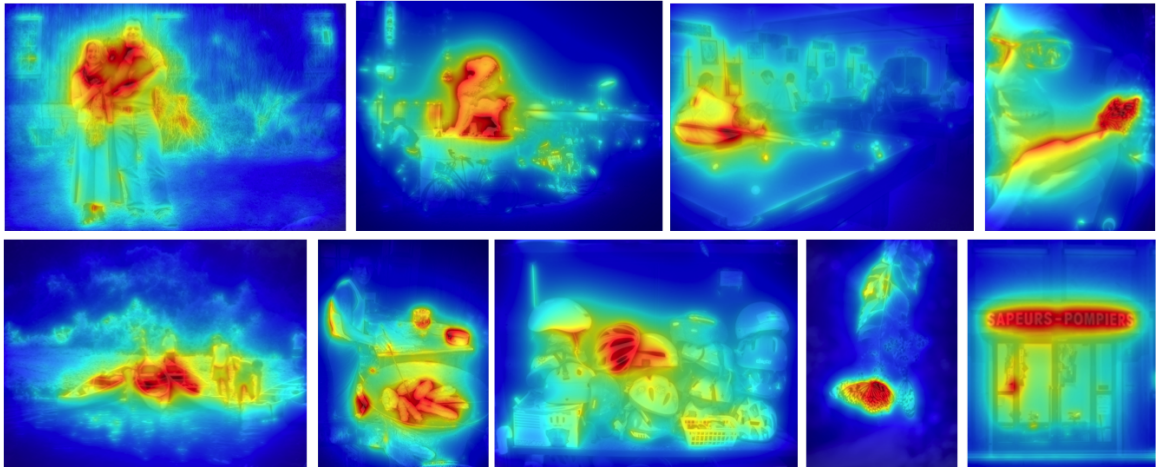


Figure 3.28: Qualitative saliency results from the MIT Saliency Benchmark.

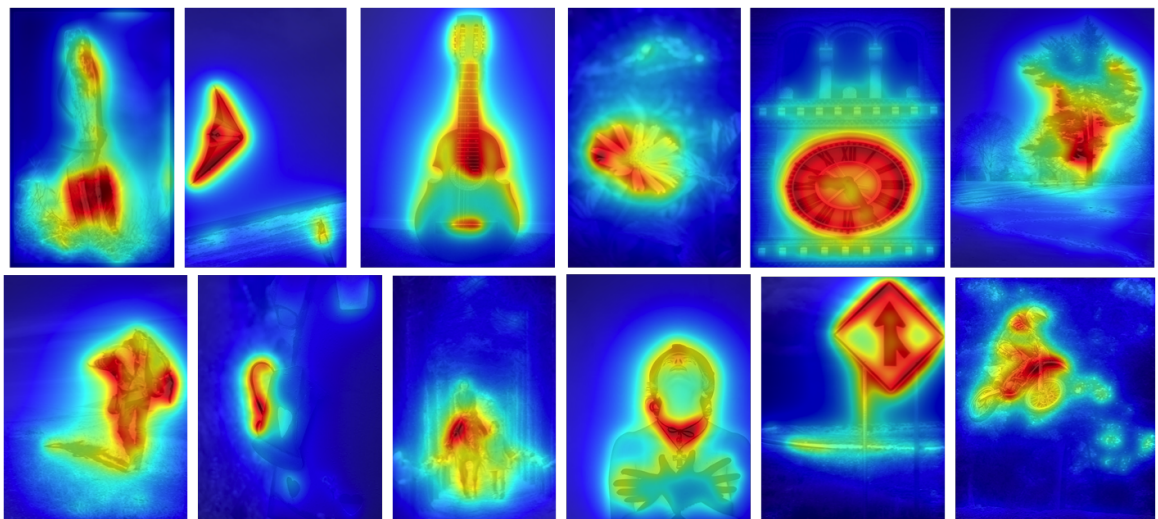


Figure 3.29: Qualitative saliency results from the MSRA salient object database.

Approved by DARPA for public release; distribution unlimited. NAVAIR Public Release 2013-389. Distribution Statement A - Approved for public release; distribution is unlimited. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of DARPA, NAVAIR or the U.S. Government.

Chapter 4

Nested Motion Descriptors

4.1 Introduction

The problem of activity recognition is a central problem in video understanding. This problem is concerned with detecting actions in a subsequence of images, and assigning this detected activity a unique semantic label. The core problem of activity recognition is concerned with the representation of *motion*, such that the motion representation captures the informative or meaningful properties of the activity, and discards irrelevant motions due to camera or background clutter.

A key challenge of activity recognition is motion representation in *unconstrained video*. Classic activity recognition datasets [78] focused on tens of actions collected with a static camera of actors performing scripted activities, however the state-of-the-art has moved to recognition of hundreds of activities captured with moving cameras of "activities in the wild" [79][80][81]. Moving cameras exhibit unconstrained translation, rotation and zoom, which introduces motion at every pixel in addition to pixel motion due to the foreground activity. The motion due to camera movement is not informative for the activity, and has been shown to strongly affect the overall activity representation performance [93].

Recent work has focused on motion descriptors that are invariant to camera motion [96, 97, 98, 93, 94, 95, 99, 100, 101]. Local spatiotemporal descriptors such as, such as HOG-HOF [86, 87] or HOG-3D [90], have shown to be a useful motion representation for activity recognition. However, these local descriptors are not invariant to dominant camera motion. Recent work has focused on aggregating these local motion descriptors into *dense trajectories*, where optical flow techniques are

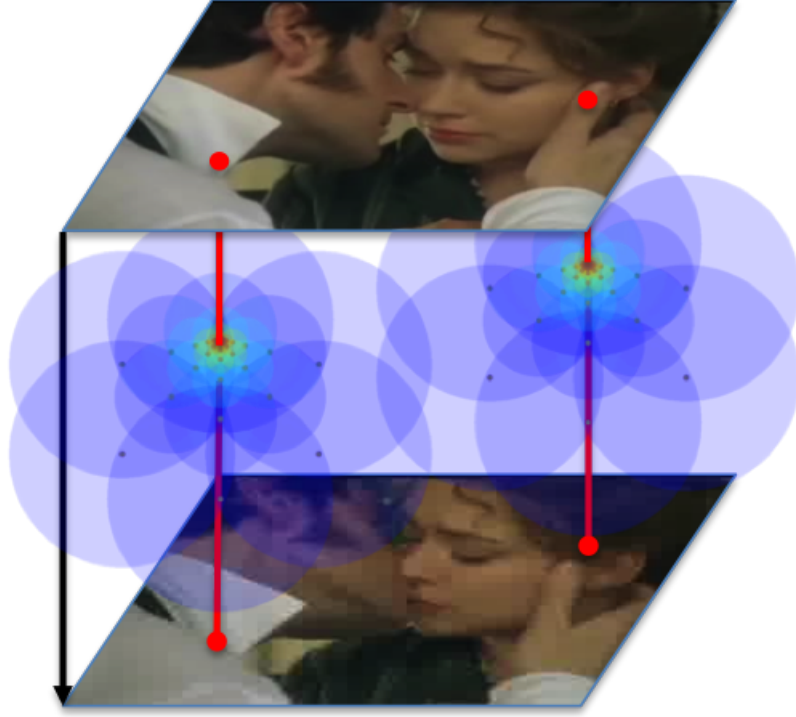


Figure 4.1: Nested motion descriptors

used to provide local tracking of each pixel. Then, the local motion descriptors are constructed using differences in the flow field, and then are concatenated along a trajectory for invariance to global motion. However, these approaches all rely on estimation of the motion field using optical flow techniques, which have shown to introduce artifacts into a video stream due to an early commitment to motion or over-regularization of the motion field, which can corrupts the motion representation.

In this paper, we propose a new family of binary local motion descriptors called *nested motion descriptors*. This descriptor provides a representation of *salient motion* that is invariant to global camera motion, without requiring an explicit optical flow estimate. The key new idea underlying this descriptor is that appropriate sampling of scaled and oriented gradients in the complex steerable pyramid exhibits a *phase shift* due to camera motion. This phase shift can be removed by a technique called a *log-spiral normalization*, which computes a phase difference in neighboring scales and positions, resulting in a relative phase where the absolute global image motion has been removed. This approach is inspired by phase constancy [47], component velocity [48] and motion without movement [49, 50], which uses phase shifts as a correction for translation without an

explicit motion field estimate. The nested motion descriptor is an extension of the nested shape descriptors introduced in [51]. The nested shape descriptor is a state-of-the-art binary local feature descriptor, which we extend to motion representation in this paper. This descriptor uses log-spiral normalization to represent salient edges, therefore the nested motion descriptor represents *salient motion*. Figure 4.1 shows the 3D pooling structure of this descriptor, and figure 4.3 shows the phase correction procedure.

4.2 Related Work

Activity recognition has a long history in the computer vision literature. Recent surveys of action recognition capturing the state of the art are available [53, 76] and a critical review of action recognition benchmarks [77]. Classic activity recognition datasets [78] focused on tens of actions collected with a static camera of actors performing scripted activities, however the state-of-the-art has moved to recognition of hundreds of activities captured with moving cameras and poor quality video of "activities in the wild" [79][80][81].

The literature on motion representation can be decomposed into approaches focused on local motion descriptors, mid-level motion descriptors or global activity descriptors. Higher level motion representations are typically focused on representing semantic activity categories, and learning mid-level representations suitable for action recognition. Examples include discriminative mid level features [82], actemes [83], motionlets [84], motion atoms and phrases [85]. In general, these higher level representations build upon local motion representations to extract activity specific discriminative motion patterns. In this section, we will focus on local motion representations only, which are most relevant to the nested motion descriptor.

A *local motion descriptor* is a representation of the local movement in a scene centered at a single interest point in a video. Examples of local motion descriptors include HOG-HOF [86, 87], cuboid [88], extended SURF [89] and HOG-3D [90]. These descriptors construct spatiotemporal oriented gradient histograms over small spatial and temporal support, typically limited to tens of pixels spatially, and a few frames temporally. HOG-HOF includes a histogram of optical flow [86, 87], computed over a similar sized spatiotemporal support. Furthermore, recent evaluations have shown that activity recognition performance is significantly improved by considering dense

regular sampling of descriptors [91][92], rather than sparse extraction at detected interest points, such as spatio-temporal interest points (STIP) [54].

An interesting recent development has been the development of local motion descriptors that are invariant to dominant camera motion. A translating, rotating or zooming camera introduces global pixel motion that is irrelevant to the motion of the foreground object. Research has observed that this camera motion introduces a global translation, divergence or curl into the optical flow field [93], and removing the effect of this global motion significantly improves the representation of foreground motion for activity recognition. The motion boundary histogram [86, 94, 95] computes a global motion field from optical flow, then computes local histograms of derivatives of the flow field. This representation is sensitive to local changes in the flow field, and insensitive to global flow. Motion interchange patterns [96, 97, 98] compute a patch based local correspondence to recover the motion of a pixel, followed by a trinary representation of the relative motion of neighboring patches. Finally, dense trajectories [94, 95, 99] concatenate HOG-HOF (or more recently co-occurrence HOG [100], or first order differential motion patterns [93]), and motion boundary histograms for a tracked sequence of interest points forming a long term trajectory descriptor. The improved dense trajectories [99] with fisher vector encoding is the current state-of-the-art on large datasets for action recognition [101].

4.3 Nested Motion Descriptors

A nested motion descriptor is a representation of salient motion in a video that is invariant to camera motion. The nested motion descriptor is an extension of the nested shape descriptor [51] to the representation of motion. Figure 4.2 shows that while the nested shape descriptor pools the magnitude of edges, the nested motion descriptor pools phase gradients which captures translation of edges in a video. In section, 4.3.2 we discuss the use of the complex steerable pyramid to compute relative phase. In section 4.3.3, we derive the relationship between phase gradients and component velocity, which provides a connection between relative phase and motion in a video. This component velocity captures all motion in an image, however a local motion descriptor should be representative of the motion of the foreground only and not influenced by the global motion of the camera. In section 4.3.7, we show that by pooling component velocity, we can remove the effect of camera

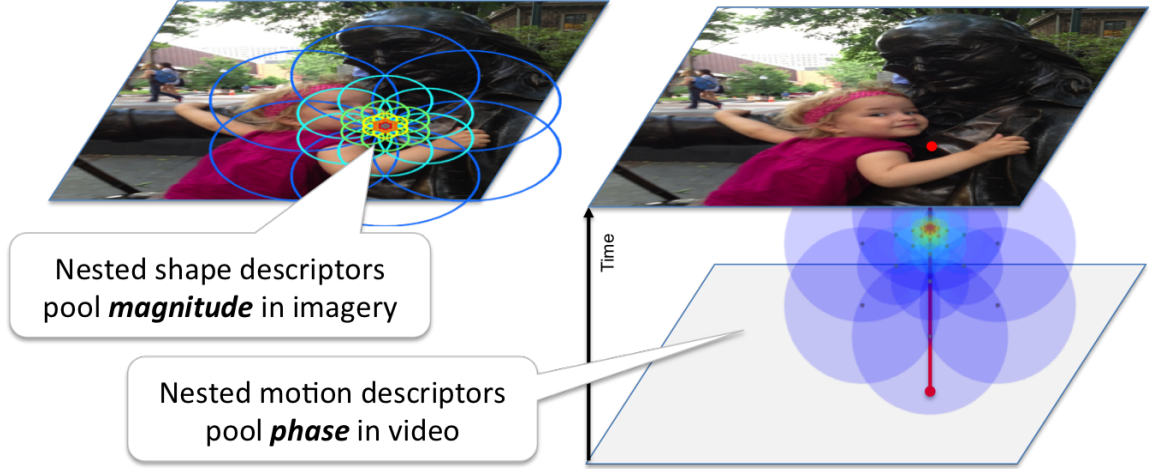


Figure 4.2: From nested shape descriptors to nested motion descriptors. Nested shape descriptors pool oriented and scaled gradients magnitude which captures the contrast of an edge in an image. Nested motion descriptors pool *relative phase* which captures *translation* of an edge. Projecting the structure of the nested motion descriptor onto a single image (“collapsing” the descriptor) will form the structure of the nested shape descriptor.

motion by computing a difference between neighboring positions and scales. This difference removes the constant velocity from the component velocity and provides an estimate of acceleration. This approach does not require an explicit optical flow estimate, and is inspired by work on phase based optical flow [153, 154, 48, 47] and “motion without movement” [49, 50] which leverages the relationship between phase shifts in the frequency domain and translation in the spatial domain. Finally, in section 4.3.6, we describe the overall construction of the nested motion descriptor, and show how this descriptor can be used to visualize salient motion in section 4.3.8. This visualization demonstrates that the NMD captures *salient motion* due to the foreground and not global motion due to the camera.

4.3.1 Overview

Figure 4.3 provides an overview of the construction of the nested motion descriptor. This procedure is summarized as a three step process: bandpass filtering, spatio-temporal phase pooling and log-spiral normalization. First, *bandpass filtering* is performed to decompose each image in a video into a set of orientation and scale selective subbands using the complex steerable pyramid [155, 141, 156]. The complex steerable pyramid includes basis filters in quadrature pairs, which allows for magnitude and phase estimation for each subband. We compute the relative magnitude and relative

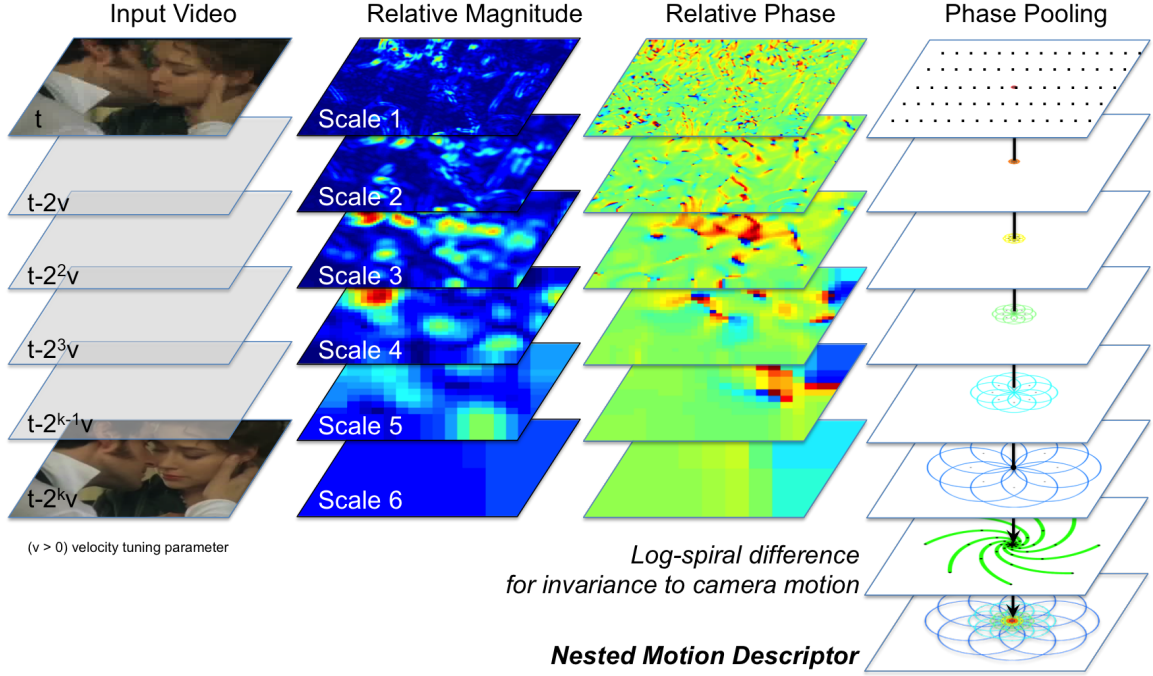


Figure 4.3: Nested Motion Descriptors (NMD). (left) An input video is decomposed into a set of frames of length $2^k v$, where k is the number of scales in the pyramid decomposition and v is a fixed velocity tuning parameter. (middle) The relative magnitude and phase is computed for each orientation and scale subband in a steerable pyramid decomposition from the first frame to subsequent frames on a log-scale. Frames further away in time are represented with a large scale coarse motion, and frames close in time are represented with a small scale fine motion. Shown is the 0° orientation subband only. (right) For each dense interest point in the current frame t , we pool the robust component velocity derived from relative phase in a set of circular pooling regions all intersecting at the center interest point. Log-spiral normalization computes the difference between phases in neighboring scales and positions along a log-spiral curve. The phase pooling aggregates component velocities, so this difference computes an acceleration which is invariant to constant velocity of the camera. The result is a nested motion descriptor at this interest point that is invariant to camera motion.

phase for each subband from a current frame to a past frame. This relative bandpass response is visualized in figure 4.3. We compute relative magnitude and phase for scales following a log scale, so that we compute a large scale bandpass response for frames further away in time. This encodes a fixed velocity tuning for a velocity parameter v .

Relative magnitude and phase provide measurements of *speed and direction of motion* in a video. An example is shown in figure 4.3 (middle) of a kiss from the human motion database [79]. In this example, the man on the left tilts his head and moves in towards the woman on the right. Observe that there is small scale motion of the man's sideburns and ear, medium scale motion of the

collar and woman’s eyes, and large scale motion of the two heads moving towards each other. The relative magnitude over various scales captures this motion. Similarly, the relative phase encodes a spatial translation from frame t to $t - k$. The relative phase is shown on the scale $[-\pi, \pi]$ where zero phase is green, negative phase is blue and positive phase is red. The phase of the mid and large scale motions encode the movement of the faces. Furthermore, at the largest scale, observe that there are two motions present, of the two heads moving towards each other.

Second, we perform *phase pooling*. We derive the relationship between phase gradients and component velocity, such that pooling component velocity is equivalent to pooling phase gradients. Furthermore, we derive a robust form of the component velocity using phase stability, to provide robust measurements of component velocities in regions of unstable phase. We define a set of pooling regions to pool the component velocity in neighboring spatial and temporal regions, to provide invariance to local geometric transformations. Each of the pooling regions is centered at an interest point, and the pooling regions are uniformly distributed in angle around the interest point. Each pooling region is represented by a single component velocity, and all orientations and scales are concatenated into a single nested motion descriptor for the interest point. This is visualized in figure 4.3 by the “collapsing” of the descriptor across scales into a combined descriptor at the bottom of the figure. This pooling and sampling of subband component velocity is the primary construction of the nested motion descriptor.

Third, we perform *log-spiral normalization*. Relative phase or *phase gradients* are proportional to the motion in an image. This motion could be due to the salient motion of a foreground object, or due to the global motion of the camera. Observe that the global motion of the camera introduces pixel motion that is a composition of global translation, rotation and scale. In these cases, the motion field in a local patch is uniformly offset, so that all vectors in the motion field in this patch are offset by a fixed bias due to the camera motion. The relative phase is also offset by a fixed constant. We can remove this constant by computing a phase difference with neighbors in position and scale. This is the goal of the log-spiral normalization, which computes a phase difference to remove this fixed bias due to camera motion. The log-spiral normalization procedure is outlined in figure 4.3 (bottom right), with the spiral like arrangement showing the differences to be computed along this spiral.

The nested motion descriptor is a spatiotemporal extension of the nested shape descriptor for video. Figure 4.2 shows a comparison of the nested shape descriptor (NSD) and the nested motion

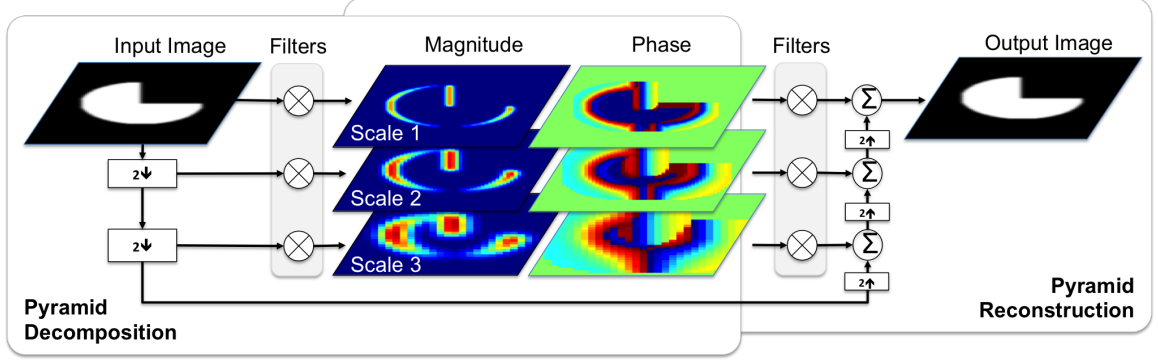


Figure 4.4: Pyramid decomposition and reconstruction with the complex steerable pyramid.

descriptor (NMD). Observe that the NSD is constructed by pooling oriented and scaled gradient magnitudes in a single image, while the NMD pools scaled and oriented phase over frames of a video. However, the fundamental nested pooling structure is the same, and the NMD extends the pooling to 3D spatiotemporal pooling regions. Observe that "collapsing" the NMD over time results in the same structure of the NSD. This allows the tools of log-spiral normalization and saliency visualization developed for the nested shape descriptor to be applied to the nested motion descriptor.

In the remaining section, we describe each of these stages of processing in more detail.

4.3.2 Complex Steerable Pyramid

The complex steerable pyramid [155, 141, 156] is an overcomplete decomposition of an image into orientation and scale selective subbands. The orientation subbands exhibit a steerability property such that the response to an arbitrary orientation is a linear combination of basis subbands. Furthermore, a complex steerable pyramid includes basis filters in quadrature pairs, such that each basis filter is further decomposed into an oriented filter and its Hilbert transform, forming an in-phase and quadrature component shifted by 90° in phase.

The complex steerable pyramid is computed using a recursive pyramid decomposition [156]. Given a set of steerable basis filters G and Hilbert transform H , let a basis filter F be represented in complex form by $F = G + H * i$. Each filter is tuned to a bandpass response in frequency ω and orientation θ forming a set of complex steerable filters $F_{\omega,\theta}$. The bandpass response $B_{\omega,\theta} = I \otimes F_{\omega,\theta}$ is formed by convolution of an image I with the complex filter. The pyramid decomposition is

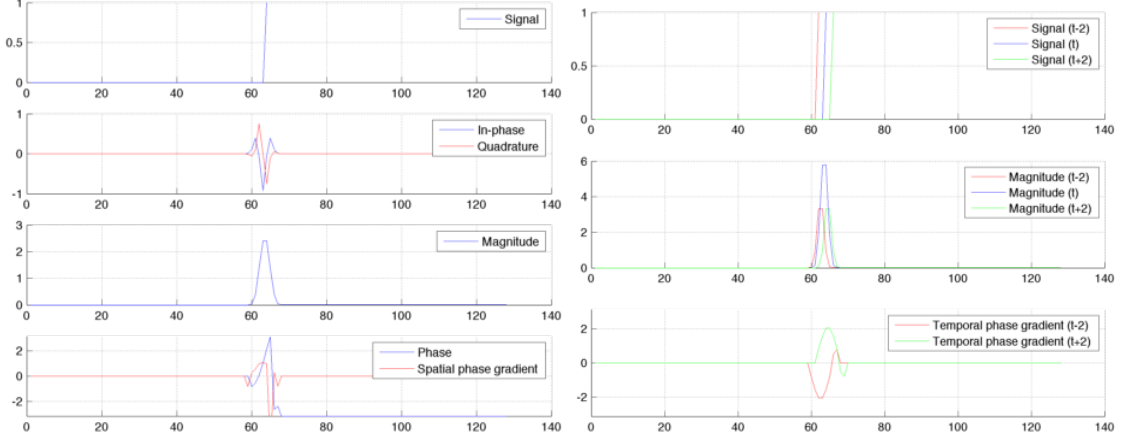


Figure 4.5: An example of the magnitude and phase response of a complex filter to a translating 1D step edge signal. (left column, top to bottom) (a) step edge signal (b) impulse response of 1D quadrature filters (c) magnitude of complex filter response to step edge (d) phase and spatial phase gradient ($|\vec{\phi}|$) of complex filter response showing linearity of phase. (right column, top to bottom) (e) A step edge translating left to right. (f) the magnitude response (g) the temporal phase gradient (ϕ_t). Observe that at the edge, the spatial phase gradient $|\vec{\phi}| = 1$ and the temporal phase gradient is $\phi_t = \pm 2$, which measures a spatial shift of $\frac{\phi_t}{|\vec{\phi}|} = \pm 2$.

formed by recursively convolving an image I with a lowpass filter F_0 , downsampling the image by a factor of 2, then computing the bandpass response B . This pyramid decomposition procedure is shown in figure 4.4 This decomposition can be made faster by considering separable kernels for the lowpass and complex steerable filters forming a separable quadrature steerable pyramid [141].

The complex steerable pyramid provides a measurement of the magnitude and phase of oriented and scaled edges. Following pyramid decomposition, complex valued bandpass coefficients can be decomposed into a real component representing the in-phase response, and the imaginary component representing the quadrature response. Let a coefficient $c_{\omega,\theta}(u,v) = x + iy$ be the complex valued coefficient for subband with orientation θ and scale ω for pixel (u,v) with real component x and imaginary component y . Then, the magnitude and phase of this coefficient is $|c| = \sqrt{x^2 + y^2}$ and phase of $\angle c = \text{atan2}(y,x)$. Intuitively, the magnitude is proportional to the contrast of an edge at the tuned orientation and scale at (u,v) , and the phase is proportional to the shift in the direction of the tuned filter orientation to the dominant edge. In other words, phase encodes a *spatial offset* to an edge.

Figure 4.5 (a-d) shows an example of the magnitude and phase response of a complex quadrature filter for a 1D step edge. The impulse response of this real and imaginary component of this

quadrature pair is shown in the second plot. Observe that these filters form a quadrature pair such that the quadrature component is shifted by $+\frac{\pi}{2}$ relative to the in-phase component. The phase plot show that the phase exhibits a *linear* response near the step edge (modulo π where the phase wraps from $+\pi$ to $-\pi$). Furthermore, the phase gradient is *constant* in this region and equal to one. This linearity of phase is exploited to estimate velocity in the next section.

4.3.3 Phase Gradients and Component Velocity

In general, the relationship between phase, translation and velocity is summarized in the *Fourier shift theorem*. This classic theorem states that a translation in the spatial domain is equivalent to a phase shift in the frequency domain. In this section, we derive the relationship between phase and phase gradients to derive a measurement of velocity.

An interesting property of the complex steerable pyramid is the ability to introduce motion without changing position simply by varying the local phase. This phenomenon has been described as "Motion without Movement" [49], such that continuously varying the local phase of a bandpass response induces the visual phenomenology of global motion. This relationship between phase and motion has been used in phase based optical flow methods [48, 47] to enforce the *phase constancy* constraint [48], such that feasible optical flow solutions are constraint to lie on contours of constant phase. This constraint has shown to be more stable than the more common brightness constancy constraint [154, 47] over ranges of shape deformation and lighting. Recent work has exploited this relationship between phase and motion to amplify small changes in phase to visualize of microscopic motion at macro scale [50]. This approach multiplies small changes in phase by a large constant, then each image is reconstructed by collapsing the steerable pyramid, introducing a local image translation due to the local phase shift.

The phase constancy constraint is defined as follows [48]. Let a complex bandpass response B tuned to an orientation and scale be given by:

$$B(x, t) = \rho(x, t)e^{i\phi(x, t)} \quad (4.1)$$

The magnitude ρ and phase ϕ of this complex valued spatiotemporal function are also are spatiotemporal functions that evolve in space and time. Next, consider a moving point at x_0 . This moving

point evolves according to the *motion field*, a spatiotemporal vector field that defines the movement of each pixel through time. The motion field is encoded as a function $x_0(t)$ which defines the spatial position of x_0 as a function of time. Fleet and Jepson in their seminal work on phase based optical flow [153, 154, 48, 47] hypothesized that the temporal evolution of spatial contours of constant phase provides a better approximation to the motion field than do contours of constant amplitude. This *phase contour assumption* states that the motion field must satisfy

$$\phi(x_0(t), t) = c \quad (4.2)$$

where c is a real valued constant. A point $x_0(t)$ propagating as a function of time according to the motion field is constrained to fall on a contour of constant phase $\phi(x_0(t), t)$. Intuitively, this states that phase is coherent and is preserved as a point propagates through time.

The phase contour assumption can be used to construct the *phase constancy constraint*. Differentiating the phase contour constraint, we obtain:

$$\nabla \phi(x, t) \bullet \vec{v} = 0 \quad (4.3)$$

where $\nabla \phi(x, t) = [\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial t}]^T$ is the *phase gradient* and $\vec{v} = [\frac{\partial x_0}{\partial t}, \frac{\partial y_0}{\partial t}, 1]^T$ is the *component velocity* at point (x_0, y_0) . Rearranging terms

$$\frac{\partial \phi}{\partial x} v_x + \frac{\partial \phi}{\partial y} v_y = - \frac{\partial \phi}{\partial t} \quad (4.4)$$

where we use the shorthand notation $\vec{v} = [v_x, v_y, 1]$ for the partial derivatives of component velocity and similarly $\nabla \phi(x, t) = [\phi_x, \phi_y, \phi_t]^T$ for the phase gradient. The phase constancy constraint states that the projection of the component velocity onto the spatial phase gradient is equal to the negative temporal phase gradient. This is identical to the classic brightness constancy constraint, using local phase instead of local brightness. Observe that the dot product in (4.3) shows that the velocity cannot be determined normal to the phase gradient, which provides a constraint only on the component of velocity tuned to the orientation of the filter B . The phase constancy constraint in (4.4) shows the explicit relationship between the phase gradient and velocity.

This method can be used to estimate the component velocity for each tuned orientation and

scale $B_{\omega,\theta}$. We use the notation $\vec{\phi} = [\phi_x, \phi_y]^T$ to denote the spatial phase gradient, then the spatial phase gradient defines a unit vector $\hat{n} = [\frac{\phi_x}{|\vec{\phi}|}, \frac{\phi_y}{|\vec{\phi}|}]^T$. The unit vector constraints the direction of the component velocity, due to the dot product in the phase constancy constraint. The velocity magnitude α can be determined directly from (4.4):

$$\alpha = \frac{-\phi_t}{|\vec{\phi}|} \quad (4.5)$$

where $\vec{\phi} = [\phi_x, \phi_y]$ is the spatial phase gradient. This is a single equation in a single unknown for the velocity scale α . Given the observed phase gradient, the component velocity is estimated $\vec{v} = \alpha\hat{n}$. Fleet and Jepson further proposed that the component velocities can be used as an overcomplete set of measurements to estimate the optical flow v using regularized least squares optimization. This can provide an estimate of pixel velocity or *optical flow* from measurements of component velocity, which is the foundation of phase based optical flow methods [153, 154, 48, 47].

The component velocity (4.5) is a function of only phase gradients which can be computed efficiently from the complex steerable pyramid. The bandpass response in the complex steerable pyramid for a given tuned orientation and scale at time t is denoted $B_{\omega,\theta}^t$. To simplify notation, when the bandpass orientation and scale (ω, θ) is implied, let this bandpass response be written as $B_{\omega,\theta}^t = B_t$. The phase gradient is given by

$$\nabla \phi = \frac{Im(B^* \Delta B)}{|B|^2} \quad (4.6)$$

where $Im(z)$ is the imaginary component of the complex number z , and B^* is the complex conjugate of the complex valued bandpass response [48]. This identity for the phase gradient depends only on the complex bandpass response, and avoids an explicit computation of the phase angle using a trigonometric function.

Figure 4.5 (d-f) shows an example of the phase gradient and component velocity estimate. In this example, a 1D step edge is translating by two pixels from left to right. Figure 4.5 (e) shows the magnitude response of this translation, and (f) shows the temporal phase gradient computed using (4.6). The spatial phase gradient is shown in figure 4.5 (d). Using the measured phase gradients, we can use (4.5) to compute the velocity magnitude $\alpha = \frac{\pm 2}{1}$, which shows that the phase gradients

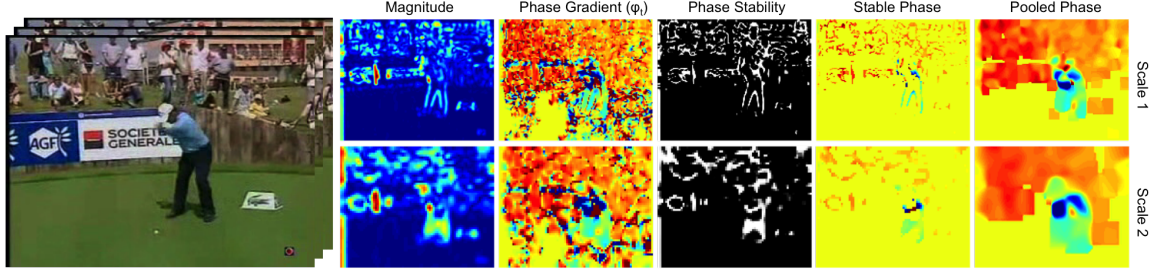


Figure 4.6: Robust Phase Pooling. The temporal phase gradient is noisy due to the measurement of phase in regions where phase is unstable, such as the region on the grass and in the crowd. The phase stability measure provides an estimate of locations of stable phase. Only the stable phase is used for pooling, resulting in pooled phase that captures the motion of the background and foreground of the golfer in the scene. This pooled phase is used to construct the nested motion descriptor.

provide a measurement of shift of the translating step edge.

4.3.4 Robust Component Velocity

It is important to discuss the *stability* of phase based component velocity estimation. Fleet and Jepson [154, 47] suggest a threshold on a function of the magnitude response to discard regions with poor phase stability. They show that a sufficient statistic for a robust phase estimate is the ratio between the spatial derivative of magnitude and the absolute magnitude. In other words, we require a small change in magnitude relative to the absolute magnitude in order to have stable phase estimate.

$$P = \{q \mid \frac{|\rho_x(q)|}{\rho(q)} < \tau, q \in I\} \quad (4.7)$$

The set P is a set of interest points in an image I such that each interest point satisfies the constraint for phase stability. A feasible interest point is one that has a small spatial change in magnitude (e.g. a local maxima of magnitude, at the phase zero crossing) and has a large edge magnitude. This constraint discards regions of low contrast (small denominator) and non-maximum edges (large numerator), leaving interest points that have sufficiently stable phase characteristics for computing component velocity.

The stability constraint in (4.7) be combined with the phase gradient (4.6) into a single mea-

surement of *robust phase gradient* $\nabla\hat{\phi}$

$$f(\rho, \tau, \beta) = \frac{1}{1 + \exp(-\beta(\tau - \frac{|\rho_x|}{\rho}))} \quad (4.8)$$

$$\nabla\hat{\phi} = f(\rho, \tau, \beta) \nabla\phi = \frac{\nabla\phi}{1 + \exp(-\beta(\tau - \frac{|\rho_x|}{\rho}))} \quad (4.9)$$

The logistic function in (4.8) provides a soft threshold for the stability constraint. The robust phase gradient is equal to $\nabla\phi$ when $\frac{|\rho_x|}{\rho} \ll \tau$, and smoothly transitions to zero as $\frac{|\rho_x|}{\rho} \gg \tau$. The parameter β encodes the sharpness of the transition of the logistic function from zero to one.

This estimate of robust phase gradient can be used to define a *robust component velocity*. Following the definition of component velocity in (4.5), and replacing the phase gradients with the robust phase gradients in (4.9), we define the robust component velocity as

$$\hat{\alpha} = \frac{-\hat{\phi}_t}{|\hat{\phi}|} \quad (4.10)$$

Intuitively, this function provides a measurement of component velocity that is equal to the observed velocity if the magnitude is sufficient. However, if the magnitude is not sufficient and the phase is unstable, such as a region of low contrast, then the function will provide a measurement of zero velocity. This formulation of robust component velocity is a new contribution of this work.

Figure 4.6 shows an example of the phase stability and robust phase gradient. In this example, a golfer is in the middle of the backswing and the camera is panning from left to right to begin following the ball. We show the magnitude and phase for an oriented bandpass response tuned to two octave scales and 0° orientation. The observed temporal phase gradient is very noisy due to regions of poor stability where the magnitude is small or non-maximum. The phase stability in (4.7) can be used to identify the regions in the imagery with stable phase, which is shown in the grayscale image such that white pixels are stable, and black are unstable. Finally, the robust phase gradient is computed using this stability constraint as in (4.9) resulting in stable phase measurements. The figure shows that the stable phase gradient is much less noisy and clearly reflects the true motion of the background and the swing of the golfer. In the next section, we discuss aggregation of this stable phase using spatial and temporal phase pooling, as shown in the final column.

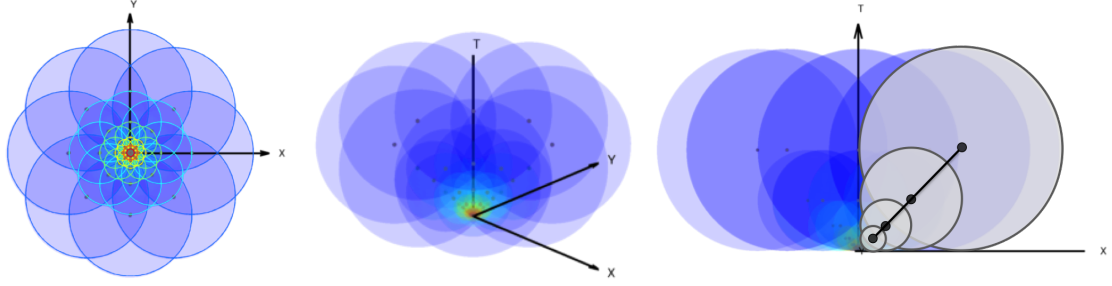


Figure 4.7: Perspective views of the spatiotemporal pooling regions of the nested motion descriptor. (left) $az=90^\circ$, $el=90^\circ$, the temporal axis is pointed into the page. We overlay the nested shape descriptor onto this view, which shows that the NMD has an equivalent pooling structure to the NSD (middle) $az=45^\circ$, $el=25^\circ$, with the temporal axis pointed into the page, (right) $az=90^\circ$, $el=0^\circ$, with the Y axis pointed out of the page. This view shows that the temporal pooling regions increase proportionally to spatial scale. The slope of the line connecting the centers is determined by the velocity tuning of the descriptor. A video visualization of this descriptor is available at <http://youtu.be/RfJJHmXnRAw>.

4.3.5 Robust Phase Pooling

Spatiotemporal phase pooling refers to the aggregation or accumulation of phase gradients over neighboring positions and times. The pooling regions over which the accumulation occurs are represented as spheres in a 3D spatiotemporal volumes (x, y, t) where (x, y) are spatial image support in pixels and t is the temporal support in frames of a video. The radius of the sphere defines the spatial and temporal support of the aggregation. Figure 4.7 shows perspective views of the spatiotemporal pooling regions for the nested motion descriptor.

Spatiotemporal pooling in the nested motion descriptor is constrained such that the temporal projection of pooling regions is equivalent to the nested shape descriptor [51]. Figure 4.7 (left) shows an example of this spatiotemporal pooling constraint. Furthermore, as the spatial scale of the pooling region increases, the temporal scale also increases and the center of the pooling region shifts in time. This is shown in the perspective view in figure 4.7 (right). The intuition for this pooling strategy is that motions far away in time should be measured at coarser scale while motions close in time should be measured at a finer scale. Figure 4.7 (right) shows that the projection of the spheres onto the (x, y) plane will result in a set of circles that intersect at exactly one point at the origin. The highlighted set of spheres in grey form an *Hawaiian earring* structure when projected onto the (x, y) plane.

Formally, the spatiotemporal nested pooling is defined as follows. We will use the notation

and conventions defined in [51], where sets of spheres are grouped into lobes forming an Hawaiian earring when projected onto the (x,y) plane. The descriptor exhibits n -fold rotational symmetry so that there are n lobes equally spaced in angle. The notation $\mathbb{K}_n(i, j)$ refers to the sphere in the i^{th} lobe at j^{th} scale, with center $c_{ij} = [2^j \cos(i\frac{2\pi}{n}), 2^j \sin(i\frac{2\pi}{n}), 2^j \mathbf{v}]^T$ and radius $r_{ij} = [2^j, 2^j, 2^j \mathbf{v}]^T$ in (x, y, t) spatiotemporal volume. The parameter \mathbf{v} is the velocity tuning of the NMD, which "squashes" the descriptor temporally to tune to faster or slower motion.

For example, Figure 4.7 shows an NMD with 8-fold rotational symmetry, such that there are 8 lobes each containing a nested set of spheres over five scales. The set of all pooling regions in this NMD is \mathbb{K}_8 . Figure 4.7 (right) shows a set of spheres highlighted in grey all which are in the same lobe. These spheres are referenced as $\mathbb{K}_8(0, j)$ for the $i = 0$ lobe, and each lobe is referenced by scale j . As the scale increases, both the center and radius of each pooling sphere increases exponentially.

Finally, we perform pooling of robust phase gradients within these spherical pooling regions. Recall that the definition of the robust phase gradients uses the fact that some regions of the image are unstable, and do not provide reliable phase estimates. So, phase cannot just be accumulated over each pooling region, as there may be different number of stable phase estimates in each region. To compensate, we pool robust phase gradients, but normalize by the total phase stability measure in the pooling region. This phase pooling is equivalent to the mean robust phase gradient within the pooling region. Figure 4.6 shows an example of this pooling in the final column. This phase pooling is used to construct the robust component velocity and the nested motion descriptor.

4.3.6 Construction of the Nested Motion Descriptor

Finally, we can pull together the results from the previous sections to construct a nested motion descriptor at an interest point as follows. Let $B'_{\omega, \theta}$ be a bandpass response at scale ω and orientation θ at time t , for each frame in a video clip as computed in section 4.3.2. Next, compute the phase gradients for each bandpass response following (4.6), and compute the robust phase gradient following (4.9). This stable phase is pooled using the spatiotemporal pooling in section 4.3.5 for a given spatiotemporal pooling support \mathbb{K}_n , such as the visualization in figure 4.7. Finally, the robust component velocity is computed as in (4.10) for the pooled phase gradients. Let the robust component velocity be indexed $\hat{\alpha}_{ij}^t(q)$ for orientation i and scale j at pixel q , where the phase gradient is computed using the current frame and frame t . Then, the nested motion descriptor is constructed

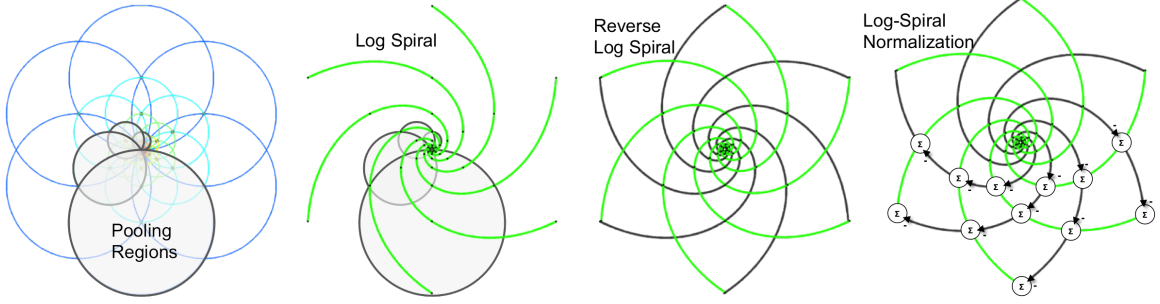


Figure 4.8: (top) Logarithmic spiral property of the nested motion descriptor provides *normalization* and *binarization*. The log-spiral and its reflection shown in grey form an elegant flower-like structure. (bottom) An NMD is formed at each interest point by (left) nested pooling of scaled and oriented gradients and (right) log-spiral difference and binarization.

from pooled robust component velocities, normalized by the stability constraint:

$$d(i, j, k, t) = \frac{\sum_{q \in \mathbb{K}_n(j, k)} \hat{\alpha}_{ik}^t(q)}{\sum_{q \in \mathbb{K}_n(j, k)} f(q)} \quad (4.11)$$

$$\hat{d}(i, j, k) = d(i, j, k, t - 2^k \mathbf{v}) - d(i, j - 1, k - 1, t - 2^k \mathbf{v}) \quad (4.12)$$

$$D(i, j, k) = \begin{cases} 1 & \text{if } \hat{d}(i, j, k) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.13)$$

Equation (4.11) is *robust component velocity pooling*. The descriptor $d(i, j, k, t)$ is the pooled component velocity for orientation subband i , lobe j and lobe scale k at frame t . Observe that the bandpass scale k is equal to the pooling support radius k . In other words, support regions with radius 2^k pool orientation subbands over octave scales k , so we pool coarser gradients over larger supports. Furthermore, the normalization constant is the pooled phase stability constraint in (4.8). This provides a weighted mean component velocity within the pooling region, where the weight is provided by the phase stability.

Equation (4.12) is *logarithmic spiral normalization*. This log-spiral normalization computes the difference between component velocities at neighboring scales and positions within the same frame. Observe that there is a coupling between the frame offset, pooling scale and bandpass scale, since all depend on k . This results in pooling coarser velocities over larger supports. We discuss in the next section how this normalization provides invariance to camera motion.

Figure 4.8 shows an example of this log-spiral normalization. In general, a *logarithmic spiral* is

a curve that can be written in polar coordinates as $r = ae^{b\theta}$ for arbitrary positive real constants a and b . A nested support set \mathbb{K}_n exhibits a logarithmic spiral when considering neighboring supports. For example, figure 4.8 (right) shows an example of the logarithmic spiral for \mathcal{K}_6 . Each turn of angle $\theta_i = \frac{2\pi}{6}i$ is a radius of $r_i = 2^i$, which is equivalent to a logarithmic spiral numerically approximated with parameters $a = 1, b = 0.66191$. Figure 4.8 (right) shows a log-spiral and its reflection $r = ae^{-b\theta}$ forming an elegant flower-like pattern. The sequence of grey circles with centers and radii at left follow the logarithmic spiral shown in green in 4.8 (middle). Combining this log-spiral with its reflection (right) forms an elegant flower like structure used for normalization and binarization. Figure 4.8 (right) shows an example of the log-spiral normalization procedure. This pattern encodes the normalization which is a difference of spiral adjacent support, which provides invariance to camera motion. Intuitively, the log-spiral difference is a difference in component velocities between neighboring positions and scales. If these component velocities are the same (due to global camera motion) then the difference will remove this motion. We discuss this further in section 4.3.7.

Finally, equation (4.13) is *binarization*. A nested motion descriptor can be binarized by computing the sign of (4.12). This constructs a nested motion descriptor with binary entries. This is an optional step which can be used to provide compact representation.

The final nested motion descriptor D from (4.13) is a binary vector of length $(R \times |\mathbb{K}| \times |K|)$ for R orientation bands over $|\mathbb{K}|$ lobes and $|K|$ supports per lobe. For example, for eight orientation subbands, five nested supports, and six lobes has dimensionality $(8 \times 6 \times 5) = 240$. The nested motion descriptor can also be real valued using (4.12), without the final binarization step.

4.3.7 Invariance to Camera Motion

In this section, we describe how the log-spiral normalization of the nested motion descriptor provides invariance to global camera motion. The key intuition for this procedure is that each dimension of the NMD encodes the robust component velocity of estimated at a specific orientation and scale. The log-spiral normalization computes a difference between neighboring scales and positions in the NMD, within the same frame. If both of these dimensions are moving with the same velocity, due to the global camera motion, then the difference will remove this effect. Basically, the log-spiral difference is computing an *local acceleration* or second order derivative between neighboring velocities pooled in the nested motion descriptor. Acceleration is invariant to constant velocity, so if

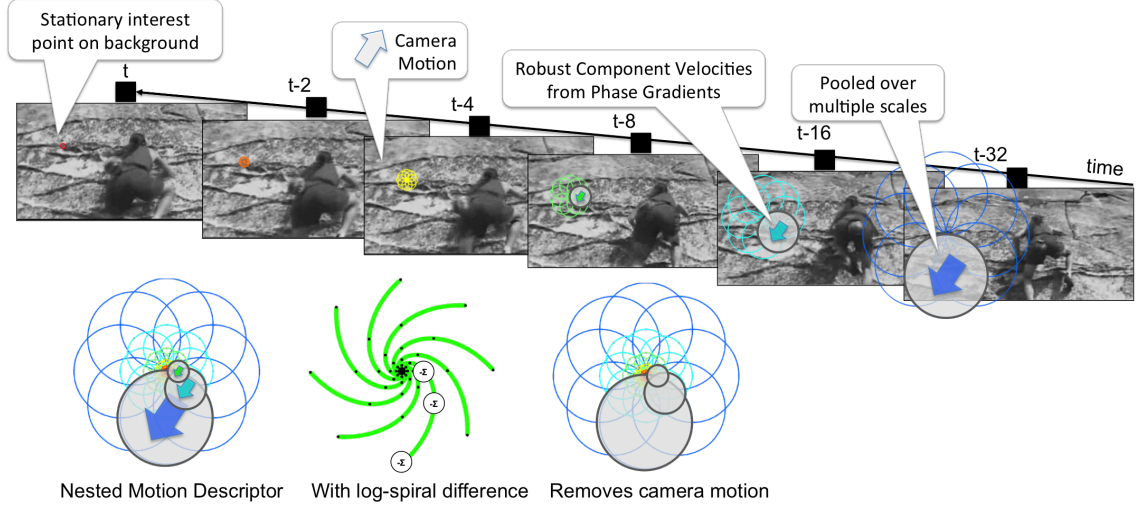


Figure 4.9: The nested motion descriptor is invariant to global camera motion. (top) A video sequence of a rock climber where the camera is following the climber up the rock face. For a given fixed interest point on the background, we compute the nested motion descriptor. Observe that the robust component velocities for this interest points are the same. (bottom) When computing the log-spiral difference, the constant velocity due to the camera motion is removed, leaving only *acceleration*.

the camera is translating with a constant velocity, the descriptor will be invariant to this motion.

Figure 4.9 shows an example of the invariance to the dominant camera motion. This figure shows a video sequence of a rock climber where the camera is following the climber up the rock face. This introduces constant velocity motion in the background due to the camera motion. We show a single interest point on the background to show that this effect of the motion from the camera is removed. We compute the robust component velocities using the nested motion descriptor construction in the previous section. Observe that each pooling region on this background interest point result in the same component velocity. This is the same due to the global motion of the camera. When we compute the log-spiral difference, this constant velocity is removed, resulting in robust component velocities of zero.

Finally, it is important to note that this approach to invariance to camera motion does not require an estimate of the dominant camera motion, or an estimate the optical flow field. Both of these alternative techniques require a commitment to a motion estimate, which if incorrect can introduce errors or smoothing artifacts into the motion representation. The NMD does not require these assumptions.

4.3.8 Motion Saliency

We can visualize salient motion using the steerable pyramid reconstruction. In chapter 3, we showed that there exists a mapping from the dimensions of the nested shape descriptor and the coefficients of steerable pyramid, and that we can use this mapping to provide a saliency map by pyramid reconstruction. In this chapter, we have shown that there exists a mapping from the nested motion descriptor to the nested shape descriptor, which means that we can reuse the same technique for saliency map construction,

A saliency map is a real valued scalar field that encodes the salience of regions in an image. The nested motion descriptor can be used to compute a saliency map in a very simple manner. Recall that the nested motion descriptor requires the construction of a quadrature steerable pyramid to compute multiscale oriented gradients. Given this pyramid, replace the orientation and scale bands with the clipped mean square response of the NMD for each orientation and lobe. Then, replace the low pass response of the steerable pyramid with the squared Laplacian filter response, to implement a center surround difference. Finally, reconstruct the image from this saliency pyramid. In short, a motion saliency map is the image reconstructed from the squared response of the nested motion descriptor.

Formally, let a steerable pyramid $B = \{I_0, B_{ij} ; i \leq R, j \leq S\}$ for orientation bands B_{ij} over R orientations i and S scales j and lowpass residual image I_0 . Each band B_{ij} encodes the oriented gradient response at orientation i and scale j . Furthermore, let \hat{d} be a log-spiral normalized nested motion descriptor constructed following eq. 3.1 and 3.2, computed densely at each pixel. Then, let

$$\hat{B}_{ij} = \max(\sum_j \hat{d}(i, j, k)^2, \tau) \quad (4.14)$$

$$\hat{I}_0 = (I_0 * L)^2 \quad (4.15)$$

where L is a 3x3 Laplacian kernel, $*$ is the convolution operation, and τ is a clipping threshold for the maximum squared difference. These are collected as subbands in a steerable pyramid $\hat{B} = \{\hat{I}_0, \hat{B}_{ij}\}$, and these bands are used to reconstruct an image using the standard steerable pyramid reconstruction algorithm, where the filters used for reconstruction are the magnitude of the quadrature pair. This reconstructed image is a saliency map. Finally, a saliency video is encoded from the set of saliency maps computed from the video, and rescaled so that the maximum saliency response is encoded as



Figure 4.10: The nested motion descriptor represents salient motion in video. We show a semitransparent saliency map for motion overlaid on each frame of video. This saliency map shows salient responses in red and non-salient in blue. The salient responses show the foreground motion of the basketball dribbling and suppresses the motion of the camera in the background.

red.

The final saliency map encodes the unoriented motion saliency, of motion in any direction. Colorization to encode oriented motion saliency is straightforward since these orientations are already computed in the nested motion descriptor. However, we leave this visualization extension as future work.

Figure 4.10 shows an example of salient motion constructed using this technique. This example shows four frames from a short clip of dribbling a basketball from the human motion database [79]. This clip contains large scale and small scale motion of the body and hands of the player, as well as global camera motion down and to the left. The colors encode the saliency map such that red is salient and blue is not-salient. Observe that the salient motion extracted using this technique highlight the small motions of dribbling the basketball and not the large motions due to the camera. In section 4.4.3 we show a comparison of this motion saliency with and without the log-spiral normalization to demonstrate the representational power of this approach.

4.4 Experimental Results

In this section, we show results for applying nested motion descriptor to the task of activity recognition. We focus on three datasets, and compare results for a simple bag-of-words classification framework, to highlight the performance differences due to motion descriptors only.

The goal of our experimental evaluation is demonstrating of *relative performance* of local motion descriptors for the task of activity recognition. This experimental evaluation does not attempt to achieve the state of the art in activity recognition on any one dataset. For example, the current state of the art uses higher level activity representations using improved dense trajectories and Fisher

vector encoding of activities [101]. Instead, we are interested in determining the relative effect of only the local motion descriptors, in order to determine the relative benefit of this representation for this task. As a result, we consider only the relative performance of classification using a simple and well understood activity representation based on bag-of-words. This will not achieve state of the art, but the relative ranking is insightful for the performance of the descriptors only. These descriptors could then be used to improve the performance of dense trajectories to further push the state of the art. This evaluation strategy was used for baseline comparisons of local motion descriptors in activity recognition evaluations in [91, 92], and we follow the same approach.

We compare performance of the nested motion descriptors to HOG-HOF [54] and HOG-3D [90]. As described in the related work, there are many other motion descriptors including motion boundary histograms, motion interchange patterns and variants of dense trajectories. However, all of these descriptors are non-local. They focus on optical flow to aggregate local descriptors by tracking points through a long trajectory, which is a form of a global representation. In fact, dense trajectories define their representation as set of HOG-HOF descriptors extracted along a trajectory. The nested motion descriptor is local to a specific interest point, rather than capturing the properties of a trajectory. Therefore, we compare to other local motion descriptors. The evaluation in [91] showed that HOG-HOF and HOG-3D outperformed cuboid and dense SURF, so we limit our evaluation to these two descriptors. Furthermore, the improved dense trajectories consider HOG-HOF as the local motion descriptor extracted along the trajectory, so we use this as our baseline.

The datasets chosen for this evaluation span the complexity representative of classic and modern activity recognition problems. The KTH actions dataset [78] (2004), is representative of classic activity recognition dataset, with six classes and unmoving and zooming cameras. The UCF sports actions dataset [157] (2008) has nine activity classes, but these videos are collected in unconstrained television footage. Finally, the human motion database (HMDB) [79] (2011) is representative of a modern dataset with over fifty actions in unconstrained video.

The state of the art for activity recognition has moved to larger and more diverse datasets [80][158] with hundreds of activity classes, however since our focus is on relative performance of descriptors, we focus on classic datasets that span the complexity rather than pushing the absolute classification accuracy performance. Furthermore, classification performance has saturated on the KTH actions dataset to near perfect classification results, due primarily to the fact that the camera

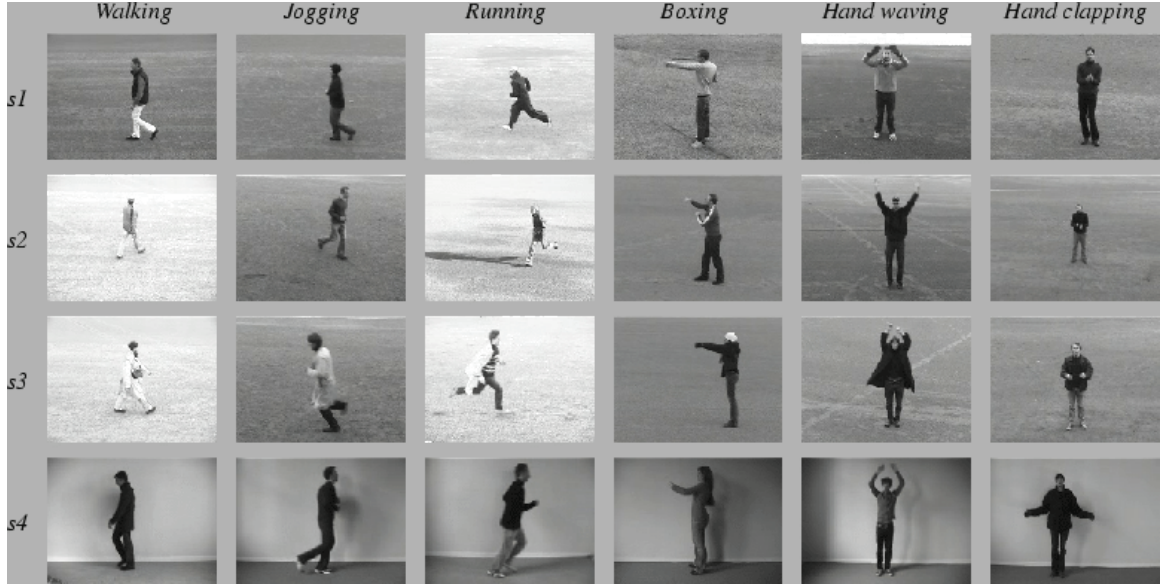


Figure 4.11: Examples frames from the six activity classes in the KTH actions dataset.

is not moving. However, remember that our analysis is focused on demonstrating the relative performance benefit of the local motion descriptors, and not the absolute classification performance of the activity recognition framework. So, these datasets remain informative for this relative analysis task.

4.4.1 Experimental System

The experimental system we consider for evaluation of nested motion descriptor performance is activity recognition using a bag of words representation.

For each observation, we densely extract local motion descriptors from each frame in the video, with the given spatiotemporal stride. We use all descriptors from a random sample of 30 videos to perform vector quantization to learn the K words in the vocabulary. Then, for each video, we construct a bag-of-words representation by assigning each densely extracted descriptor to the closest word, and creating a normalized histogram of word occurrence. Finally, classification is performed by training a one-vs-rest linear SVM classifier for each class, then selecting the maximum likelihood class for each observation. We report results in classification rate or mean average precision across all classes for each dataset.

We compare to the baseline of [54] and HOG-3D [90] local motion descriptors. We use the

public implementations available from the author’s websites, and initialize these descriptors to the parameters listed below.

Finally, we use the following parameters in all experiments, in addition to the default parameters recommended by the original authors.

- **Resolution:** We downsample frames so that the maximum dimension is 160 pixels.
- **Visual words:** 600 words in the vocabulary, trained from a random sample of 10,000 descriptors from 30 videos.
- **Stride:** $dx=5$, $dy=5$ spatially, $dt=5$ temporally
- **NMD parameters:** $scales=5$, $orientations=8$, $lobes=8$, real valued (without binarization), with log-spiral normalization
- **Dataset size per class:** 30 training videos, 65 testing videos.

Training and testing splits follow the recommendations from the dataset authors, unless otherwise noted. For KTH actions, we follow the recommended training and testing splits where we divide the test set into nine subjects (2, 3, 5, 6, 7, 8, 9, 10, and 22) and the training set into the remaining subjects. For HMDB, we use the unstabilized HMDB videos and limit the training and testing to the listed number of videos per class above. For UCF sports, we perform leave one out cross validation due to the limited number of videos available per class and report only confusion matrix and mean classification rate results.

4.4.2 Experimental Datasets

KTH actions is a classic activity recognition dataset [78]. This dataset contains six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed several times by 25 subjects in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes and indoors. This dataset contains 2391 sequences, such that all sequences were taken over homogeneous backgrounds with a static camera with 25 Hz frame rate. The sequences were downsampled to the spatial resolution of 160x120 pixels and have an average length of four seconds. Figure 4.11 shows example frames for all six activity categories in the KTH actions dataset.

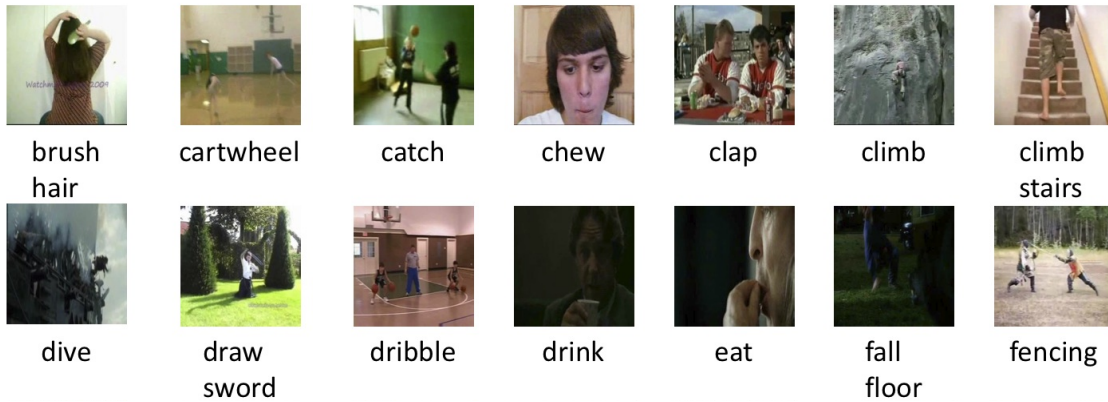


Figure 4.12: Examples frames from 14/51 activity classes in the Human Motion Database (HMDB).

The **UCF sports actions dataset** [157] consists of a set of nine actions collected from various sports typically featured on broadcast television channels such as the BBC and ESPN. The video sequences were obtained from a wide range of stock footage websites including BBC Motion gallery, and Getty Images. This dataset contains close to 200 video sequences at a resolution of 720x480. The collection represents a natural pool of actions featured in a wide range of scenes and view-points. Actions in this data set include: Diving (16 videos), Golf swinging (25 videos), Kicking (25 videos), Lifting (15 videos), Horseback riding (14 videos), Running (15 videos), Skating (15 videos), Swinging (35 videos) and Walking (22 videos).

The **Human Motion DataBase (HMDB)** is a recent activity dataset containing a large number of activities in the wild [79]. HMDB is an activity recognition dataset collected from various sources, mostly from movies, and a small proportion from public databases such as the Prelinger archive, YouTube and Google videos. The dataset contains 6849 clips divided into 51 action categories, each containing a minimum of 101 clips. The categories can be grouped in five types:

- General facial actions such as smile, laugh, chew, talk.
- Facial actions with object manipulation such as smoke, eat, drink.
- General body movements: cartwheel, clap hands, climb
- Body movements with object interaction: brush hair, catch, draw sword,
- Body movements for human interaction: fencing, hug, kiss

Figure 4.12 shows examples from fourteen activity classes in this dataset.

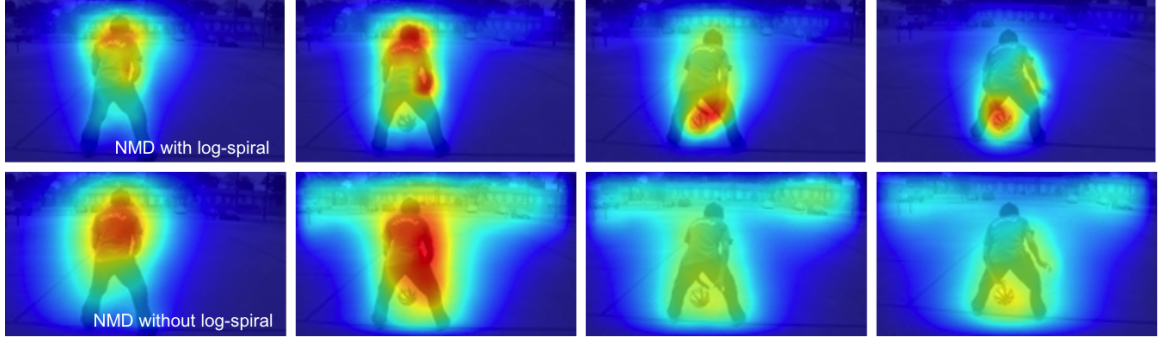


Figure 4.13: Motion saliency for basketball dribbling. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization suppresses the camera motion and highlights the salient motion of the basketball dribbling in the scene. A video visualization of this motion saliency is available at <http://youtu.be/t6D1c6M98aE>.

4.4.3 Motion Saliency

In this section, we show results applying the visualization of salient motion captured by the NMD as described in section 4.3.8. Recall that a motion saliency map is the image reconstructed from the squared response of the nested motion descriptor. We show results for a sampling of videos from the KTH actions and HMDB datasets, and compare qualitative results with and without the log-spiral normalization. These results demonstrate the effectiveness of the log-spiral normalization in representation of salient motion and suppressing the effect of camera motion.

Video saliency is a emerging field of investigation with datasets and approaches recently being developed for evaluation [159]. However, like image based saliency, quantitative performance evaluation typically considers such metrics as human gaze prediction, which requires cultural and contextual biases such as high level information of human faces and center bias. Instead, we show qualitative results for bottom up motion saliency, to show that the dominant camera motion can be suppressed in these videos. These results could be integrated into a larger system for video saliency that includes both bottom up and top down information for a more quantitative analysis.

Figure 4.13 shows an example of basketball dribbling from HMDB. The top row shows the output of the motion saliency using the NMD, and the bottom row shows the same output using the NMD without the log spiral normalization. This video includes a dominant camera motion down and to the left, and this manifests in the bottom row as motion in the background. This motion is shown as a motion on the horizon where there is sufficient texture to satisfy the phase stability

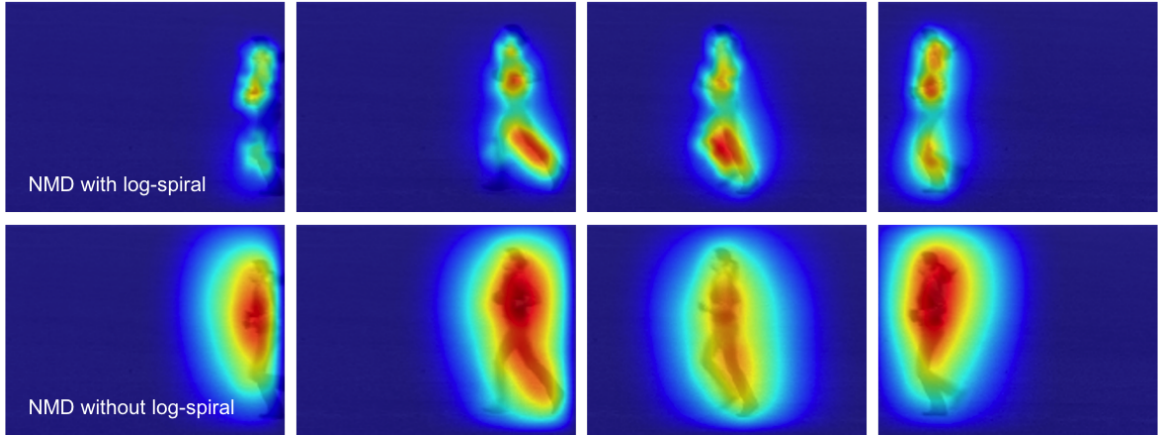


Figure 4.14: Motion saliency for KTH jogging. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization highlights the salient motion of the runners legs and arms, while the motion without log-spiral normalization saturates with the motion of the mean velocity of the body. A video visualization of this motion saliency is available at <http://youtu.be/zzhos4lj-QE>

requirements. Furthermore, the motion of the body of the player dominates the motion without the log-spiral, but when this is included in the representation, then the salient motion of the basketball and the head relative to the motion of the body pops out.

Figure 4.14 shows an example of jogging from the KTH actions dataset. This example considers a static camera, so there is zero motion in the background due to camera motion. The bottom row shows the motion visualization without the log-spiral normalization, and this shows that the motion is dominated by the overall movement of the jogger from right to left. The top row shows the effect of the log-spiral normalization which causes the motion of the legs and pumping of the arms to pop out.

Figure 4.15 shows an example of rock climbing from the HMDB. In this example, two rock climbers are racing to the top of an indoor rock climbing wall and the camera follows the climbers up the wall introducing large camera motion up and to the right. The bottom row shows that without the log-spiral normalization, the background motion tends to dominate the motion representation which manifests as motion everywhere in the scene. The top row shows that the log-spiral normalization is able to suppress this dominant motion so that the motion of the climbers pops out from the background.

Figure 4.16 shows an example of hug from the HMDB. This example also includes a camera motion panning from left to right as the two people converge to a hug. Without the log-spiral

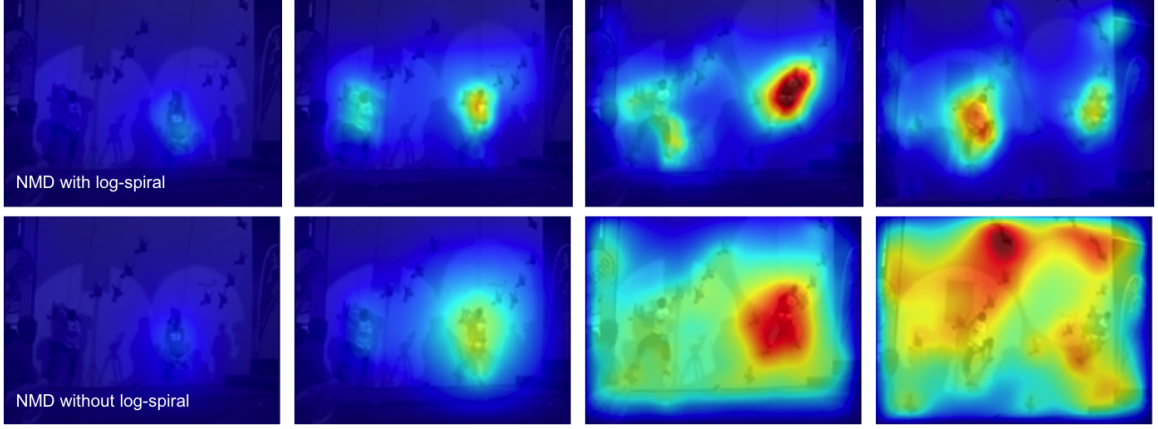


Figure 4.15: Motion saliency for HMDB rock climbing. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization suppresses the significant camera motion in the scene focusing on the salient motion of the rock climbers only. A video visualization of this motion saliency is available at <http://youtu.be/MShHPa15KsU>.

| Descriptor | KTH Actions | UCF Sports Actions | HMDB |
|------------|-------------|--------------------|------|
| HOG-HOF | 0.81 | 0.62 | 0.23 |
| HOG-3D | 0.86 | 0.75 | 0.24 |
| NMD | 0.87 | 0.77 | 0.25 |

Table 4.1: Mean average precision (mAP) results for activity recognition. Results show that the nested motion descriptor (NMD) outperforms the baseline on all classes.

normalization, this camera motion dominates, reducing the scene to a single motion blob. With the log-spiral normalization, the salient motion of the hands and head as two enter the hug.

4.4.4 Activity Recognition

The overall results are shown in table 4.1. We report mean classification rate results over all activity classes for activity recognition using the experimental framework in section 4.4.1.

Results show that the nested motion descriptor (NMD) outperforms the baseline on all datasets. These results are consistent with reported results in the literature using bag-of-words framework, albeit at a lower overall classification rate. We believe this is due to the smaller total vocabulary size (600 vs. 4000 in [91]), however the relative performance change across the dataset is consistent. The best performance is on the KTH actions dataset which does not contain any global camera motion, the second best is on UCF sports which contains camera motion but a limited number of object classes. The worst performance is on unstabilized HMDB, due to the large number of

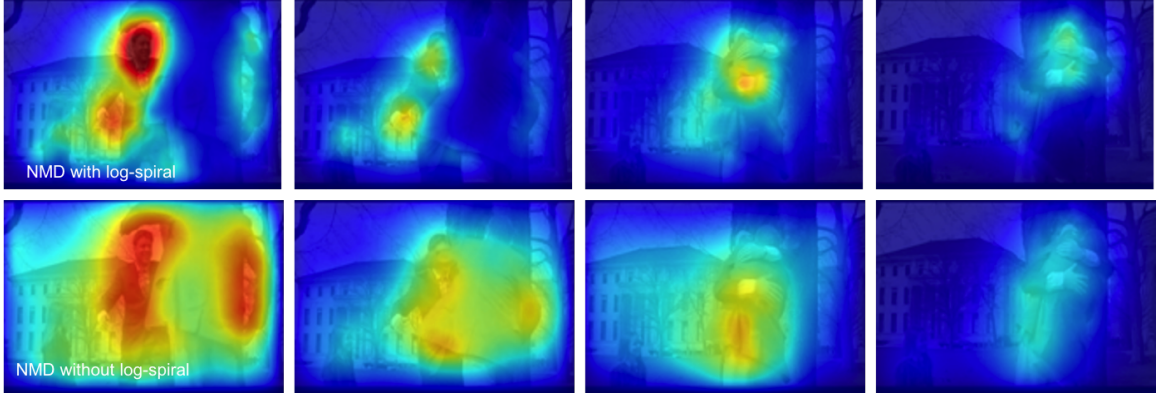


Figure 4.16: Motion saliency for HMDB hug. (top) Salient motion using NMD with log spiral normalization. (bottom row) NMD without log-spiral normalization. The log-spiral normalization focuses on the subtle hand movements that form a hug and suppresses the background motion of the camera. A video visualization of this motion saliency is available at <http://youtu.be/yxlg9rvXvuQ>.

classes. However, we observe that the NMD does still provide improved performance over the baseline descriptors.

Figure 4.17 shows detailed classification results on the UCF sports actions dataset. Recall that this dataset requires leave one out cross validation results due to the limited number of training examples per class. We observed that this dataset includes a significant background context that affects the results for comparing motion descriptor. Specifically, the "Kicking-Front" and "Kicking-Side" classes contains wide open grass fields with strong field line markers. Observe that the HOG-3D descriptor confuses only kicking-front and kicking-side, while the NMD performs poorly on this class but better on all other classes. We hypothesize that this is due to the context of the large football fields on which this action takes place, rather than the motion of the foreground itself. The NMD suppresses the motion on the ground due to the dominant camera motion, while the HOG-3D descriptor leverages this context that is unique to these two classes. If we remove these biased classes from the aggregate scores, we see that the NMD outperforms the HOG-3D using motion only on the remaining classes, and these are the score reported. However, this result does highlight the need for a composite descriptor that can leverage features from many different sources, including the surrounding context of the background.

Figure 4.18 shows detailed classification results on the KTH actions dataset. This dataset has a large number of training examples per class which allows for evaluation using precision-recall

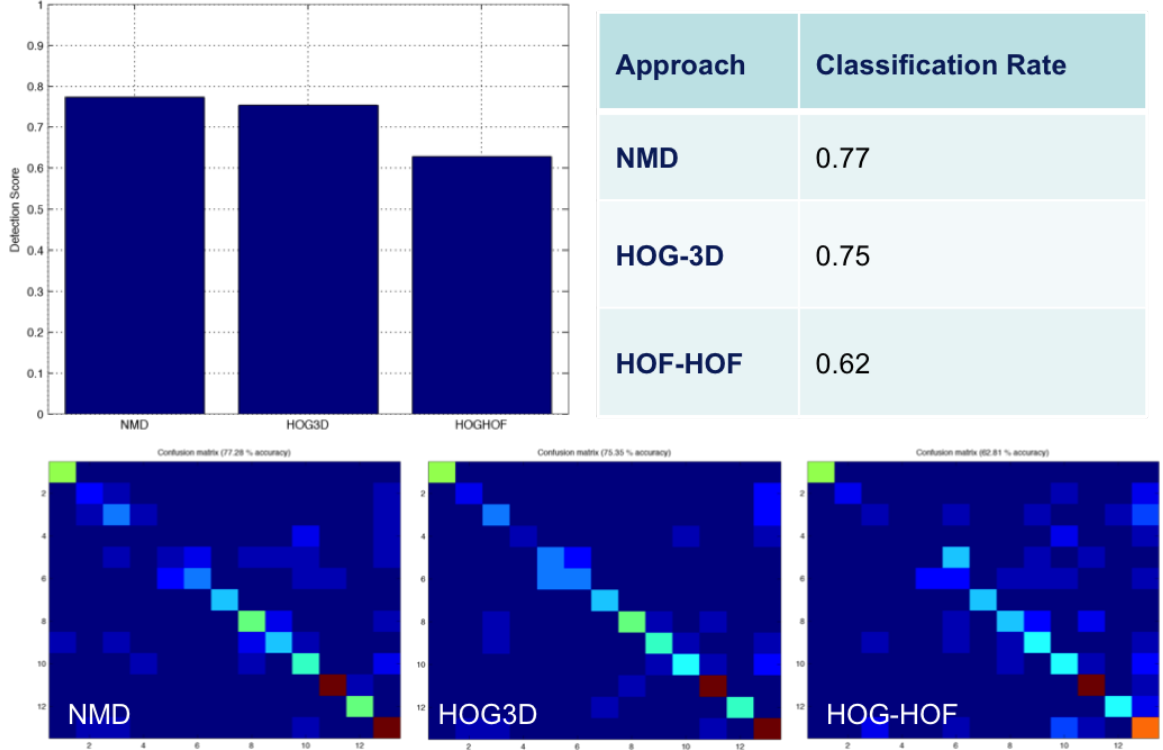


Figure 4.17: Activity recognition results on UCF sports actions. (top) Classification rate, (bottom) confusion matrices. The class index for each result: 'Diving-Side', 'Golf-Swing-Back', 'Golf-Swing-Front', 'Golf-Swing-Side', 'Kicking-Front', 'Kicking-Side', 'Lifting', 'Riding-Horse', 'Run-Side', 'SkateBoarding-Front', 'Swing-Bench', 'Swing-SideAngle', 'Walk-Front'. Class confusion results show that HOG-3D is inflating the classification results for 'kicking' by leveraging the background context of the football pitch, while the NMD is penalized for suppressing this background due to camera motion. See text for a discussion.

curves in addition to the confusion matrices and classification rates. This result shows that the NMD exhibits significantly improved average precision for boxing and handwaving, but is worse on jogging. An analysis of the confusion matrix for the NMD shows that performance on jogging is confused with running and walking. This suggests that the absolute velocity is a discriminative feature for this class, and the log-spiral normalization discards this information when computing the invariance to camera motion. It is interesting to note that in some cases, the dominant motion in the scene is informative for classification. This highlights the need for a composition of various descriptors for accurate activity classification.

4.5 Summary

In this chapter, we introduced the nested motion descriptor for representation of salient motion for activity recognition. We motivated the construction of this descriptor using phase based optical flow, we described the construction of the descriptor, we described how the use of the log-spiral normalization provides invariance to dominant camera motion. Furthermore, we showed example motion saliency results for videos with large camera motions, and demonstrated the performance of this descriptor for activity recognition.

The results show that there is a slight improvement for the NMD over HOG-3D and a significant improvement over HOG-HOF for all datasets considered. Furthermore, results show that the nested motion descriptor is suppressing dominant camera motion, however this suppression can have a negative affect on activity recognition. There are activity classes for which the absolute velocity is a discriminative feature such as jogging vs. walking. Any feature that suppresses this absolute velocity and considers only changes in velocity will suppress this discriminative feature and the result will be worse recognition performance for this class. We observed this in the KTH actions dataset, where all other classes showed improved performance over the baseline, but jogging was worse. Furthermore, there are classes for which there exist background biases such as the football pitch present for "kicking" in the UCF sports actions. The NMD suppresses this background motion as being non-informative motion of the camera. However, this field was present only in these two classes, resulting in a dataset bias that can be exploited. We saw that HOG3D exploited this background context when confusing Kicking-Side and Kicking-Front, and artificially inflating the classification rate. When these classes were removed, the performance of the NMD was shown to be superior as shown in the figure, however including these classes results in a classification rate of NMD=0.67 and HOG3D=0.71. This result highlights the fact that sometimes the camera motion is informative for classification.

These results suggest that a successful representation for activity recognition should include a composite of descriptors that capture a wide range of features. This is the strategy taken by the current state of the art in activity recognition (improved dense trajectories [101]). So, we expect that introducing the NMD into this composite framework should provide improved performance.

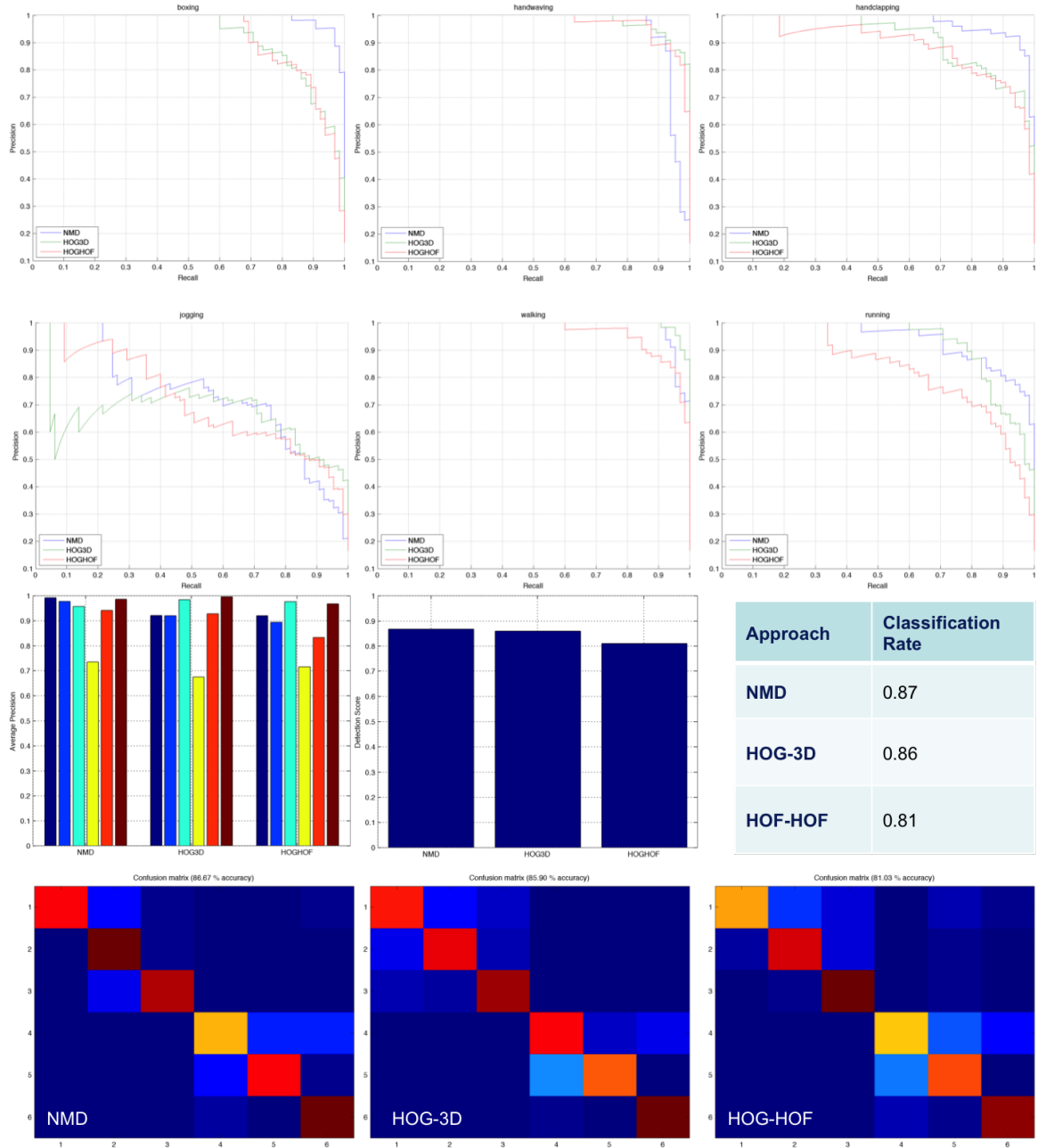


Figure 4.18: Activity classification results on KTH actions (top) Precision-recall curves for each of six activity classes (middle) average precision per class, mean classification rate, (bottom) confusion matrices. For all results, the class indexes are ordered: boxing, handclapping, handwaving, jogging, running, walking. NMD results are improved for boxing and handclapping, but worse for jogging. See text body for a discussion.

Chapter 5

Nested Pooling

5.1 Introduction

Human-scene interactions are the interplay of humans with static objects or *functional scene elements* over a wide area. For example, figure 5.1 shows a terrestrial surveillance video of an urban scene that contains the functional scene elements (FSE) bike rack, newspaper box, trashcan, subway entrance, crosswalk, road, parking space and sidewalk. A pedestrian may enter the scene, lock a bike to a bike rack, get a newspaper, discard trash and enter the subway. FSEs exhibit large spatiotemporal variations in both appearance and usage, however usage patterns are generally more consistent than appearance. For example, a bike rack may be anything from a metal bar to a tree trunk, but usage generally includes some combination of insert-bike-into-rack, remove-helmet and lock -bike. The goal of recognition of human-scene interaction is the recognition of such functional scene elements using patterns of activities executed during usages over extended time periods.

Human-scene interaction is a growing area of investigation that is driven by Wide Area Motion Imagery (WAMI) data collections. WAMI data is collected from a high resolution, low frame rate electro-optical (EO) camera from a high altitude aerial platform. WAMI data collections over urban areas can enable such new applications as functional building recognition, empty parking space detection or FSE surveillance. However, WAMI data presents a unique challenge for human-scene recognition algorithms due to low frame rate, wide field of view, stabilization, occlusions and large intraclass variations. Furthermore, due to the cost of aerial data collections, there are limited WAMI datasets available for evaluation, and due to privacy concerns, almost all WAMI data collects are

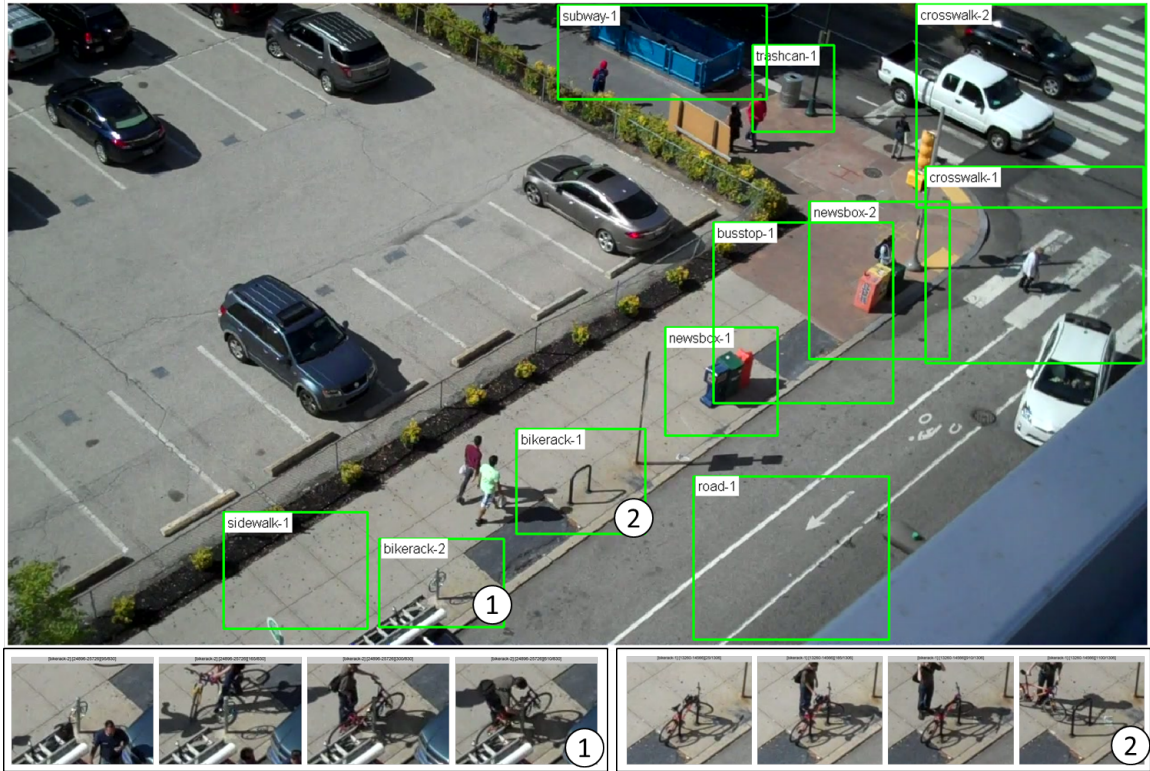


Figure 5.1: Recognition of human-scene interactions is the classification of functional scene elements such as bike racks, newspaper boxes or trashcans using only activities performed during usage. FSEs such as bike racks (1) and (2) may vary widely in appearance, but exhibit similar *weakly causal* usage patterns over time which can be used for classification.

restricted and distribution limited. Those datasets that are publically available, such as CLIF-07 over Columbus OH, GREENE-07 over Beavercreek OH or webcam surveillance datasets [160, 161] are either too short in duration or they are focused on tracking or functional object recognition rather than curation for evaluation of functional scene element recognition.

In this paper, we make two primary contributions. First, we introduce a new dataset that has been collected and curated for functional scene element recognition called the *Penn-FSE* dataset. The Penn-FSE dataset is static, terrestrial surveillance video of urban scenes, capturing hundreds of annotated functional usages of eleven object classes over 8 hours of video. This dataset is described in section 5.4.

Second, we describe a new pooling strategy for representation of functional scene elements called *nested pooling*. Bag-of-words based representations of activities rely on spatiotemporal pooling regions to perform max-pooling of learned prototypes to construct prototype histograms based representation of an activity. Nested pooling represents an activity as a bag-of-words model, however instead of pooling over a uniform region as in traditional bag of words models [135], or spatial pyramid based pooling as in spatial pyramid matching [162], the pooling regions are *nested*. This representation is inspired by a general class of local feature descriptors called *nested shape descriptors* [51]. We show that this nested pooling is well suited for modeling weakly causal activities commonly found with functional scene elements. This approach can be considered a middle ground in single level representations of human activities [53] between spatiotemporal feature based representations which ignore causality [54, 55] and sequence or graphical model based activity representations [56, 57] which represent causality by computationally expensive optimization of sequence alignments or probabilistic inference of optimal activity states. Nested pooling combines the best properties of these two approaches, which enables a representation of *weak* causality while maintaining the fast exemplar based recognition of unordered representations. This approach is described in section 5.3.

Finally, we show classification results using the nested pooling on the Penn-FSE dataset. Results show that the nesting property provides a demonstrable improvement over a non-causal baseline for recognition of usages of functional scene elements. Results are described in section 5.5.

5.2 Related Work

Human-scene interaction is related to human-*object* interaction [163, 164, 165, 57] which is focused on interactions of humans with objects such as instruments or sports equipment over small time scales. These approaches use both appearance and object specific motion during usage for classification. In contrast, human-scene interactions with functional scene elements [166, 167] assumes that the scene is static (or stabilized) and large scale, and the human is interacting with many static functional scene elements within a large scale operational area. Functional scene elements are more closely related to functional objects [168] which are moving objects in a scene with a specific purpose such as a postman or delivery truck. Unlike human-object interactions, functional objects and FSEs are defined primarily by their usage and not by appearance.

Human activity recognition has a long history [53], and we touch only those approaches most closely related to the proposed approach. Activity representations using spatiotemporal templates [169, 170, 171] can be used to capture causality over small time scales. However, these representations are not broadly selective to deformations and do not capture causality over large time scales. Activity representations using spatiotemporal features [54, 172, 173, 174] provide a compact motion representation that are tolerant to clutter, occlusions and scale changes. A key component of an activity representation is a binning or pooling strategy for aggregating feature responses, where strategies explored include histograms of spatiotemporal interest points [54], spatial pyramids [175, 15, 176] or non-overlapping horizontal, vertical and temporal grids [173, 87]. These approaches are sensitive to the grid spacing since an optimal spacing depends on the activity. In contrast, nested pooling provides a new strategy that uses *overlapping* nested aggregation regions to represent causality without grid assumptions.

This approach is most closely related to recent work in activity representation [175, 162] which uses pyramid max pooling of a set of either trained action classifiers or object classifiers over a temporal support. In contrast, this work evaluates nested pooling strategies instead of pyramid pooling strategies, and we show that this has a demonstrable benefit for the class of functional scene element recognition.

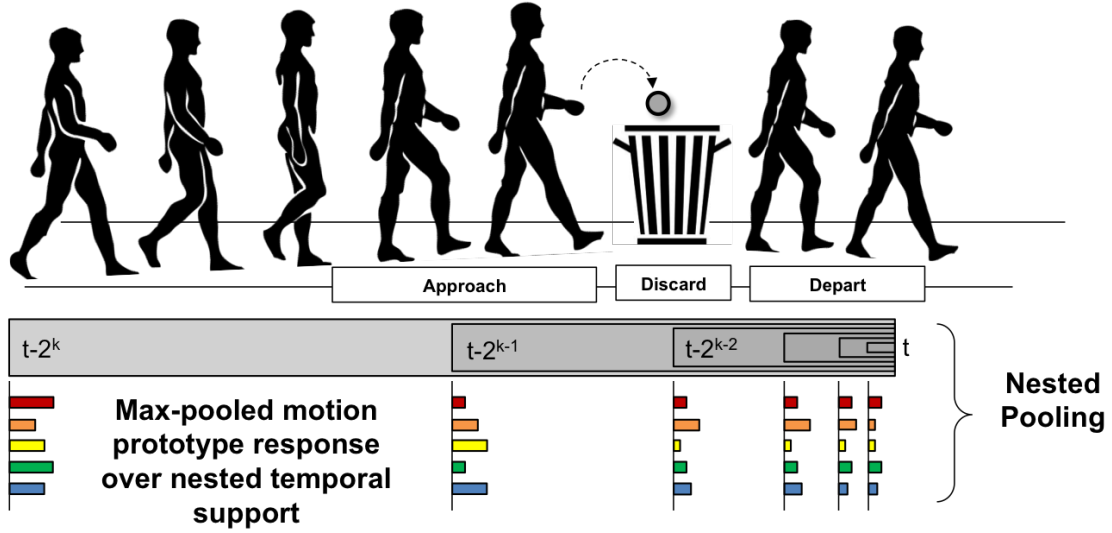


Figure 5.2: Nested pooling is the encoding of an activity as a max pooled set of motion prototypes in a nested set of support regions centered at object usage event, where each support region increases on a log scale. Each gray region is a pooling region which is represented by a histogram of prototype responses within the pooling region. Observe that the inner support regions are fully contained within the outer support regions, forming *nested pooling*.

5.3 Nested Pooling

Nested pooling is a new activity representation well suited for recognition of functional scene elements. Figure 5.2 shows an example of nested pooling. Activities can be described as a set of motions that are temporally linked into a causal sequence to form the activity. Nested pooling represents this spatiotemporal sequence by max-pooling motion prototypes over nested support regions centered at a given spatiotemporal point p . Each region is represented as an unordered bag-of-words, and each histogram from each nested pooling region is concatenated into an overall representation for the activity.

Nested pooling is a representation that captures weak causality in activity representation. *Weak causality* is defined as a partial order of temporal activities. Nested supports provides a total order of temporal supports, however within each support the max pooling provides a locally unordered representation of activities. This type of representation is suitable for functional scene elements usages which have large temporal variations on usage patterns.

Figure 5.3 shows why nested pooling captures weak causality. In this example, two usages of a bike rack differ in the local order of remove-helmet and lock-bike action prototypes. The



Figure 5.3: Why nesting for activity representation? Nested pooling preserves partial order for locally unordered action prototypes (helmet/lock) and unknown temporal scale variations (loiter/depart).

pyramid pooling cannot capture this partial order since the relative temporal position of these actions is unknown and they fall in different bins. So, pyramid pooling can only represent the actions unordered in the largest bin $(1,2,3)$, which is non-causal. In contrast, nested pooling can capture the order that approach comes before either helmet or lock $(1 \prec (2,3))$, since even if there are temporal variations, at least one nested support will capture $(2,3)$ without (1) . So, nesting preserves weak causality.

Nested pooling is straightforward to construct. Assume that a given frame at time t_0 is given as the reference frame. Nested pooling considers the set of K temporal pooling regions $P = \{P_1, P_2, \dots, P_k\}$ such that $P_i = \{t | t \geq t_0 - 2^i, t \leq t_0\}$. Intuitively, P_i is the set of frames from time $t_0 - 2^i$ up to time t_0 . Observe that each pooling region exhibits nesting, such that $P_1 \subset P_2 \subset \dots \subset P_k$. Finally, given a set of K motion prototypes, constructed using vector quantization of local motion descriptors, nested pooling is used to compute a histogram of words for each pooling region. Each histogram is independently normalized to sum to one, and concatenated into a nested pooling representation of an activity.

Nested pooling can be compared to a classic bag of features representation [135]. If a nested pooling was (i) constructed with only the largest support region P_k then nested pooling would be equivalent to a bag of features histogram. However, the nested pooling is more expressive than a bag

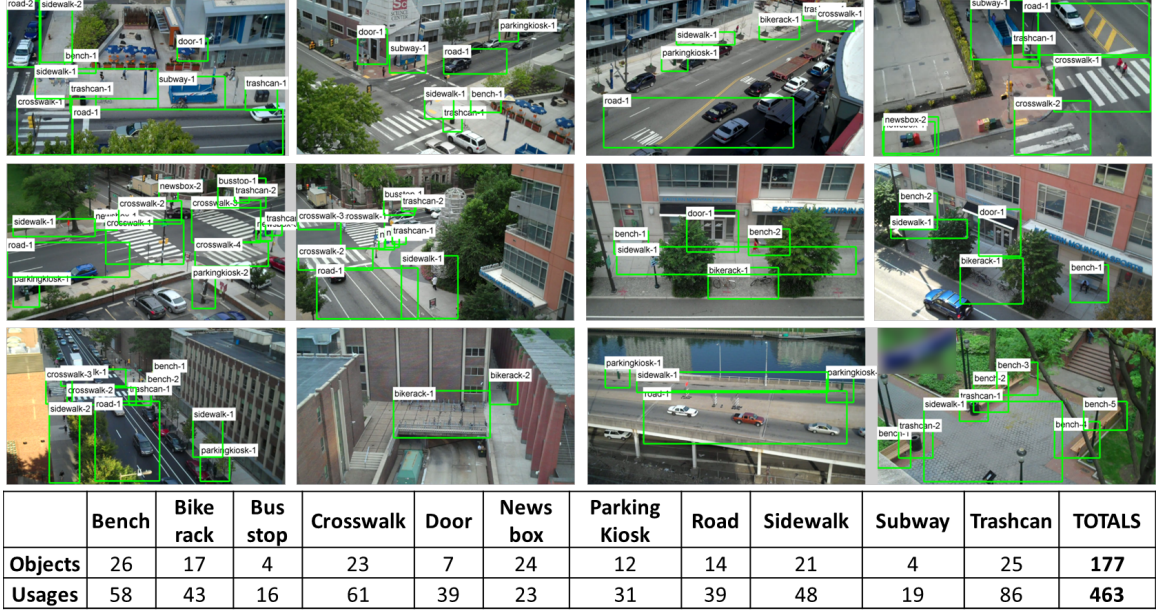


Figure 5.4: Penn Functional Scene Element (Penn-FSE) dataset. Shown are representative frames from 12 of 24 videos in the full dataset. The dataset includes annotations for 177 functional scene elements and 463 usage annotations in over 8 hours of video.

of features, since the nesting property captures weak causality through the ordering of the nested supports. This will be demonstrated experimentally in section 5.5.

5.4 Penn Functional Scene Element (Penn-FSE) Dataset

The Penn Functional Scene Element (Penn-FSE) dataset is a new dataset collected to provide functional scene element recognition that is longer, cheaper, more extensible and unrestricted than WAMI video. This dataset contains over 8 hours of annotated video collected in downtown urban scenes. These scenes include the 11 most common static functional scene elements that are found in a typical city street in downtown Philadelphia: benches, bike racks, bus stops, crosswalks, doorways, newspaper boxes, parking kiosks/parking meters, roads, sidewalks, subway entrances and trashcans. The videos are collected from a static surveillance camera mounted on buildings looking down on the street, typically from the top floor of large, open air parking garages. Twenty four videos were collected, each approximately a half hour long, and each contains multiple objects and usages performed by both volunteer actors and random pedestrians.

Figure 5.4 shows example frames and annotations from each of videos in the dataset. The

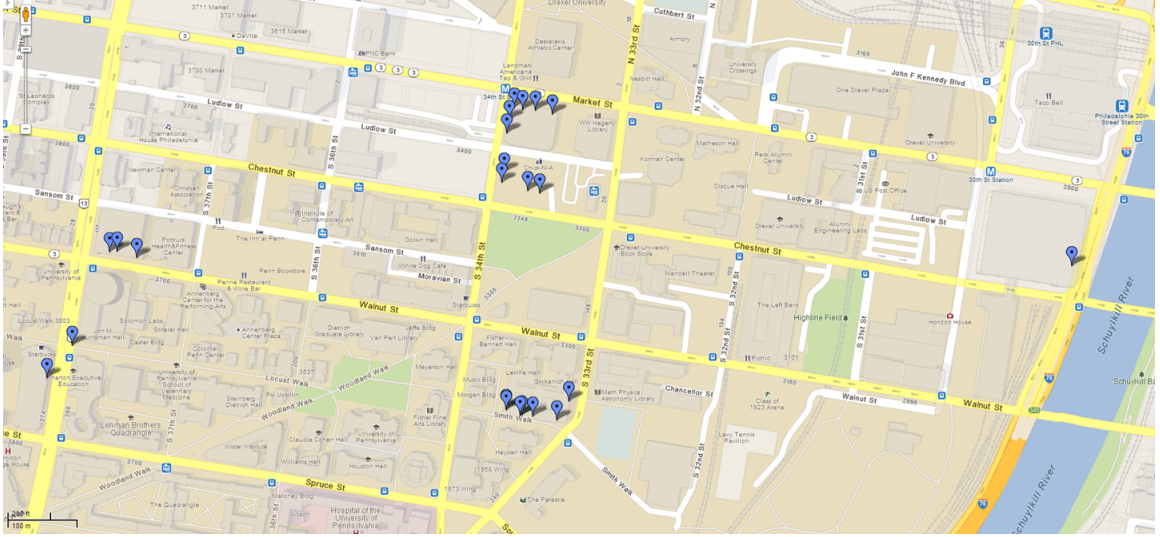


Figure 5.5: Data collection locations for the Penn Functional Scene Element dataset. The dataset contains 24 data collection sites in western Philadelphia.

dataset includes scale variations, pose variations and complex urban activities. Usage statistics for each of the functional scene elements is shown in figure 5.4. The dataset includes hundreds of annotated usages of each object, which allows for statistically significant performance evaluation of the classification algorithm.

Figure 5.5 shows the data collection locations for this dataset. The dataset includes 24 different data collection sites in western Philadelphia. These sites were selected to provide open air visibility of complex urban scenes, and typically were collected from large open air parking structures in the city.

The dataset has been curated for public release and is available for download [URL redacted]. This dataset release is 31GB, and includes twenty four H.264 encoded videos, 860850 exported color JPEG images at 1280x720 resolution, Matlab annotation tools and an annotation spreadsheet describing 177 FSE and 463 usages. This spreadsheet describes all metadata for the videos including date and time collected, total usages, total number of objects, functional scene element bounding boxes and usage annotations for temporal bounding box containing a usage of each scene element.

Figure 5.1 shows an example of the annotation tools. Unlike existing annotation tools that focus on bounding boxes or polygons in static imagery, functional scene element annotation requires *video bounding boxes* to capture usages of each element. The Matlab annotation tools include a Matlab GUI that makes it efficient to play frame sequences to temporally localize a bounding box containing

usages. The visualization tools include the ability to display any video annotation and display all of the bounding boxes for a video. The output of this annotation is in an excel spreadsheet format that is importable into Matlab, and other cross platform tools for analysis. Finally, we annotated “hard” and “easy” usages of objects, where hard cases are those with large occlusions, background clutter or significant scale variation. The user can optionally choose to include these during evaluation.

5.4.1 Penn-FSE Dataset Examples

In this section, we show example imagery from the Penn-FSE dataset showing usages for each functional scene element. The dataset overview in section 5.4 shows a wide field of view image of an urban scene, with labeled bounding boxes containing a functional scene element. Figures 5.6 and 5.7 show the cropped bounding box for one instance of each functional scene element class. Each row is a set of ten frames from the bounding box which shows a “video clip” usage of the functional scene element. The full set of frames in the bounding box were used for computing the nested pooling, however we limit to ten frames for visualization.

Figures 5.6, 5.7 and 5.8 show that there are significant variability in camera pose, object appearance, object scale and usage patterns. For example,

- Figure 5.6 (row one), shows a bench being used by two people, such that one person joins the other and then wanders around left and right while talking.
- Figure 5.6 (row two) contains a pedestrian unlocking a bike from a tree which is being used functionally as a bike rack.
- Figure 5.6 (row three) contains a crosswalk that includes both pedestrians and bikes.
- Figure 5.7 (row one) shows a parking kiosk being used by two people simultaneously, one helping the other.

As these examples show, no attempt was made to engineer “easy” usages of these functional scene elements. The scene contains background clutter and natural usage patterns which are often subtle and partially occluded. For example, figure 5.8 shows additional usages that highlight some of this variability.

- Figure 5.8 (row one,two), shows the appearance variability of trashcans that can be found in an urban environment. These examples show that the difference between a passing pedestrian

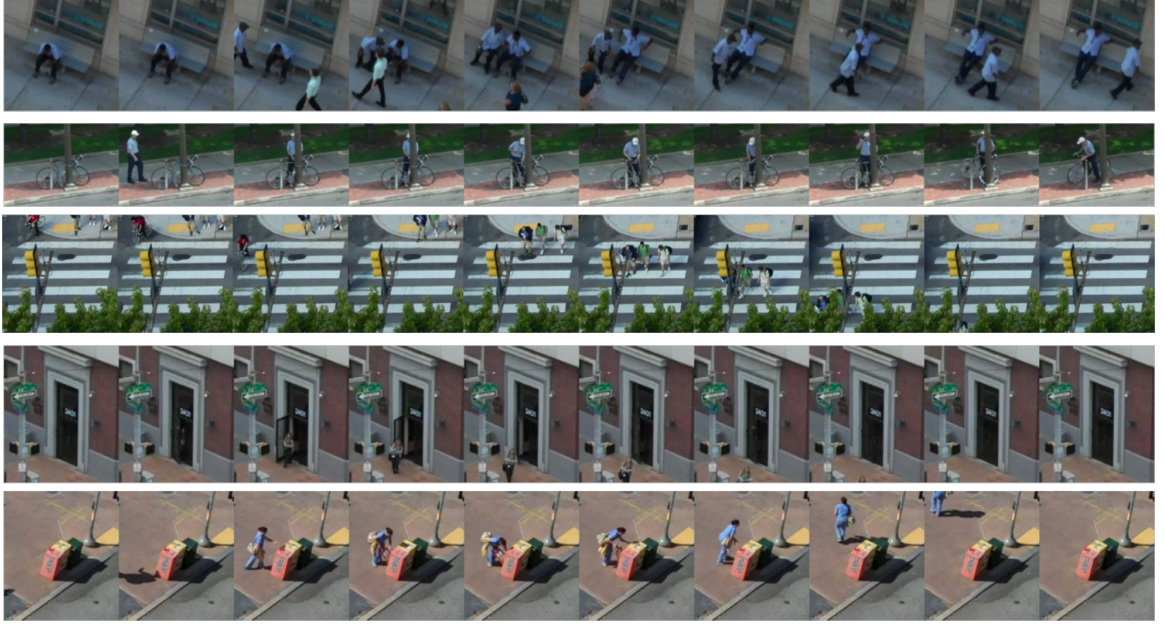


Figure 5.6: Examples of functional scene element usages in Penn-FSE. (rows) bench, bikerack, crosswalk, door, newsbox

and a usage of a trashcan is often small and subtle motion of the arm making it challenging to recognize.

- Figure 5.8 (row three) shows a low concrete wall that is being used as a bench by two women waiting for the bus.
- Figure 5.8 (row four) shows the pose variations for the subway, with pedestrians entering, exiting and loitering.
- Figure 5.8 (row five) shows a parking kiosk being used at a small scale with an occlusion by a passing truck and partial occlusion by the kiosk.
- Figure 5.8 (row six) shows a sidewalk with significant shadowing due to time of day.

These examples highlight the challenge and unique nature of this dataset.

5.5 Experimental Results

The objective of the experimental evaluation is to determine the benefit of the nested pooling for the task of recognition of functional scene elements (FSE). In this investigation we seek to show the *relative* performance improvement due to nesting as compared to two baseline pooling strategies:

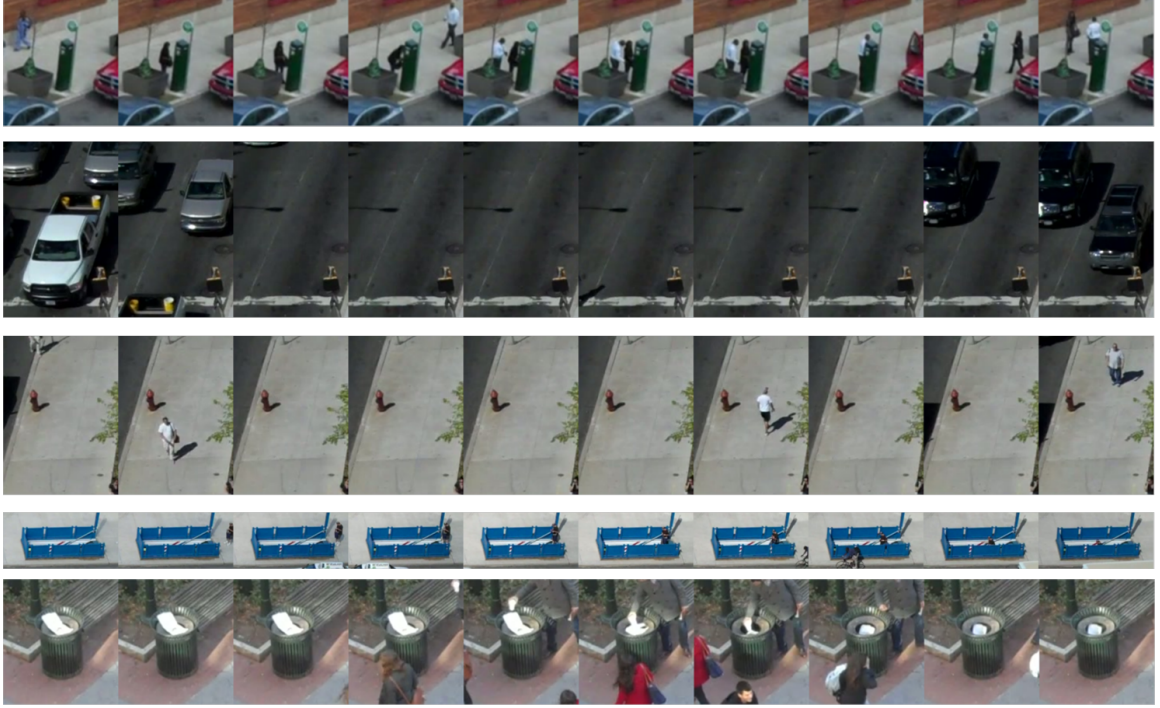


Figure 5.7: Examples of functional scene element usages in Penn-FSE. (rows) parking kiosk, road, sidewalk, subway, trashcan

bagging and temporal pyramid pooling. Specifically, this investigation was designed to answer the following three questions:

- What is the average precision for each FSE using nesting (weakly causal) as compared to a bagged (non-causal) and pyramid (weakly causal, pyramid gridded) baseline , and does nesting provide a demonstrable performance improvement?
- Which FSE classes does nesting provide the biggest relative benefit over the baseline and why?
- What is the overall mean classification error for nested pooling on Penn-FSE?

To answer these questions, we performed analysis of the nested pooling classifier on the Penn-FSE dataset. Results are shown in this section. Chapter 6 shows additional application of this approach on functional scene element recognition in WAMI video.

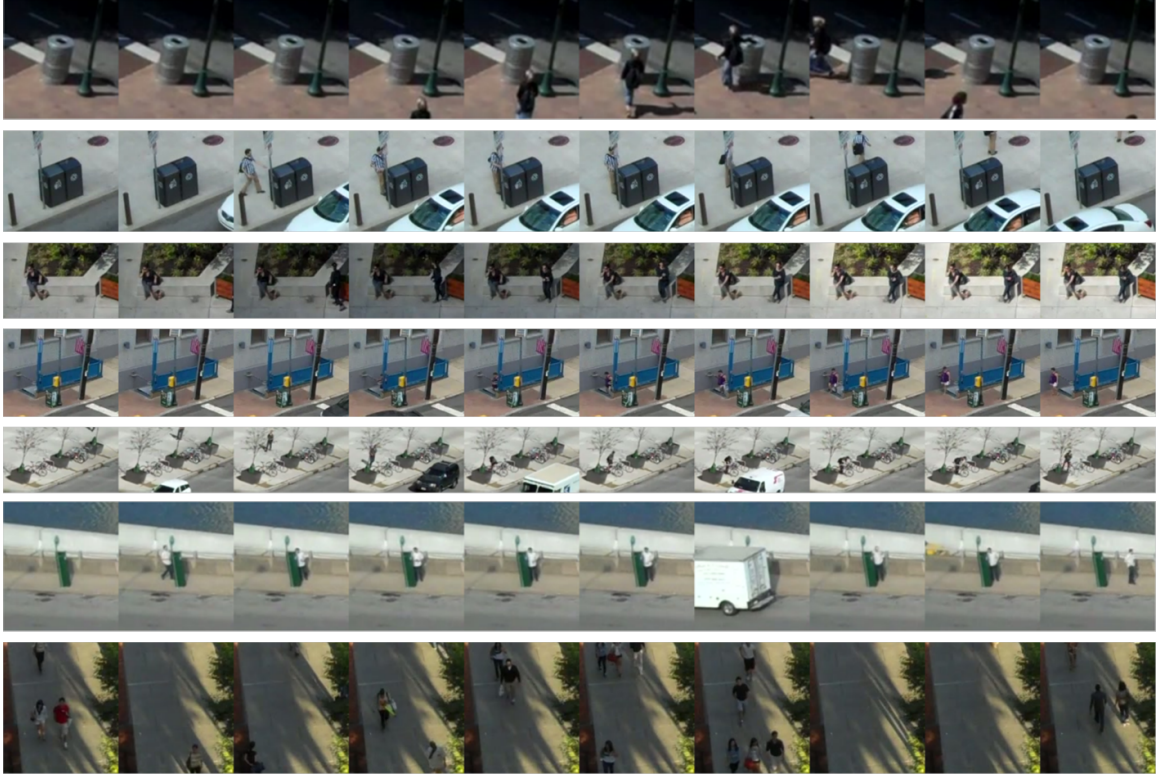


Figure 5.8: Examples of variability of functional scene element usages in Penn-FSE. (rows) trashcan, trashcan, bench, subway, bike rack, parking kiosk, sidewalk

5.5.1 Experimental System

The experimental regime evaluated in this effort is leave one out cross validation for supervised classification of stabilized spatiotemporal bounding boxes. We assume that we are given a labeled dataset of N bounding boxes each representing one of K classes from stabilized video. Each bounding box has a fixed position and spatial image support over a given number of frames in a video. This spatiotemporal bounding box contains exactly one functional scene element, and the temporal support contains at least one usage of the functional scene element. Key performance metrics are average precision computed from precision-recall curves and mean classification error accumulated over all $N - 1$ leave-one-out cross validation rounds.

The experimental system architecture used for evaluation is as follows. Given labeled bounding boxes in the training set, bounding boxes were normalized to a uniform scale such that the minimum dimension was normalized to 128 pixels. For each image sequence, we compute spatiotemporal interest points [54] then compute a HOG-HOF local motion descriptor [54] for each interest point.

Motion prototypes were computed using k-means clustering of a sampled subset of HOG-HOF descriptors, where $k=1024$. We computed nested pooling using $k=13$ nested support regions, which covers approximately five minutes of 30Hz video. Finally, nested pooling was densely sampled at every fifth frame within the spatiotemporal bounding box. Finally, we use a local naive Bayes nearest neighbor (LNBNN) classifier [177][178], which is a data driven classifier using local distance function.

Finally, we compare results of the experimental system to three baseline approaches. The baselines are constructed using a bag-of-words model [135], such that vector quantization is used to compute visual prototypes, and these prototypes are pooled over finite pooling regions to construct a histogram representation. In this comparison, each baseline differs in the pooling support used for matching of prototypes. First, we consider a non-causal baseline using *bagged pooling*. This is the default bag-of-words model using a single temporal pooling region to construct histograms. Second, we consider pyramid pooling. This performs max-pooling over supports defined by a three level temporal pyramid, consistent with [162]. Note that we do not consider gridded pooling, as it was determined in [87] that bagged outperform gridded pooling, and since the lowest level of the pyramid is a 4×1 temporal grid, the grid pooling is a subset of pyramid pooling and is redundant. Finally, we consider nested pooling which replaces the pyramid pooling with a nested pooling structure. This baseline was included to determine the effects of the temporal pooling only to capture weak causality.

5.5.2 Penn-FSE Results

We performed a leave one out cross validation on the Penn-FSE dataset and generated performance evaluation results for the nested pooling based classifier. For each round, we ran the experimental system and baseline and accumulated key metrics.

Figure 5.9 shows the confusion matrices for the experimental system as well as an overlaid precision recall curve for the experimental system for all classes. The overall mean classification rate is 0.45 for the nested pooling, 0.37 for the bagged baseline and 0.28 for the pyramid baseline. Nesting provides a 22% improvement over bagging and 61% improvement over temporal pyramid. Figure 5.10 shows separate precision-recall curves for each of ten classes. Each figure has an associated relative average precision which is the difference ($AP_{exp} - AP_{baseline}$), where a positive relative

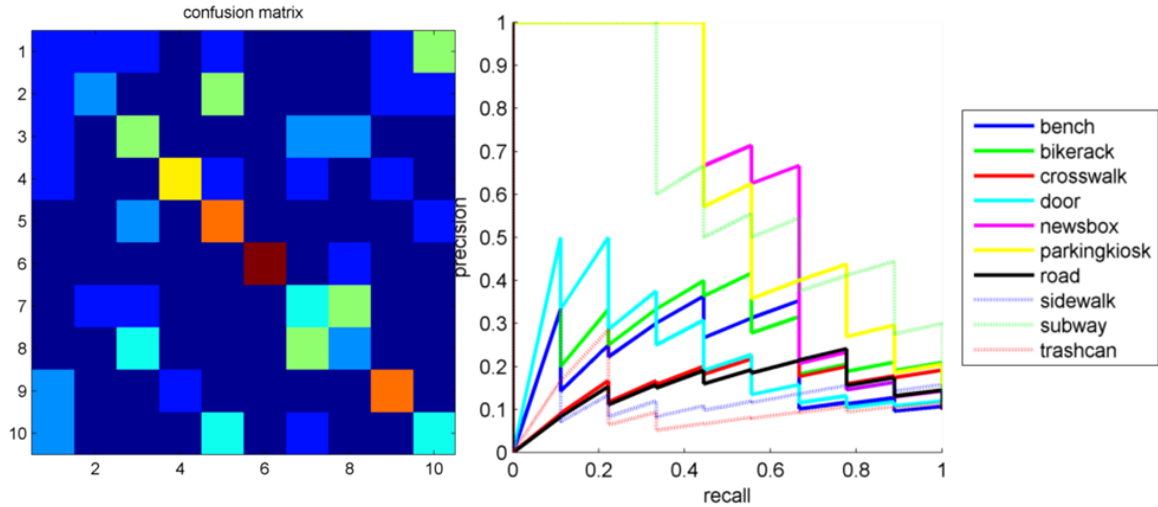


Figure 5.9: Summary results. (left) Confusion matrix for nested pooling with mean classification rate 0.45, (right) Precision recall for each class. Class indexes for confusion matrix are in the order of the legend in the precision-recall curve.

AP means that the nested experimental system is performing better than the bagged baseline.

Some observations regarding the relative performance of the experimental system and baseline system.

- The best relative performance is on crosswalk and parking kiosk with a relative average precision of +0.33. These are both classes that have weak causal usage patterns. For example, crosswalks have many pedestrians walking in an unordered manner, then cars driving.
- The worst relative performance is on subway entrance with a relative AP of -0.13. In many instances, the subway entrances are viewed in profile, so the downward stairs are not visible. In these cases, the training set exemplars appear to have the pedestrians slowly disappearing into the sidewalk. The sum accumulation of bagging captures this phenomenon better than the max-pooling.
- The pyramid pooling performs the worst overall, which is consistent with the conclusions in [87] regarding poor performance of gridded pooling.

Some observations regarding the absolute performance of the experimental system.

- The best performance is on “subway” and “newsbox”, which exhibit weakly causal usages.
- Roads are most commonly confused with crosswalks.
- Sidewalks are most commonly confused with roads.

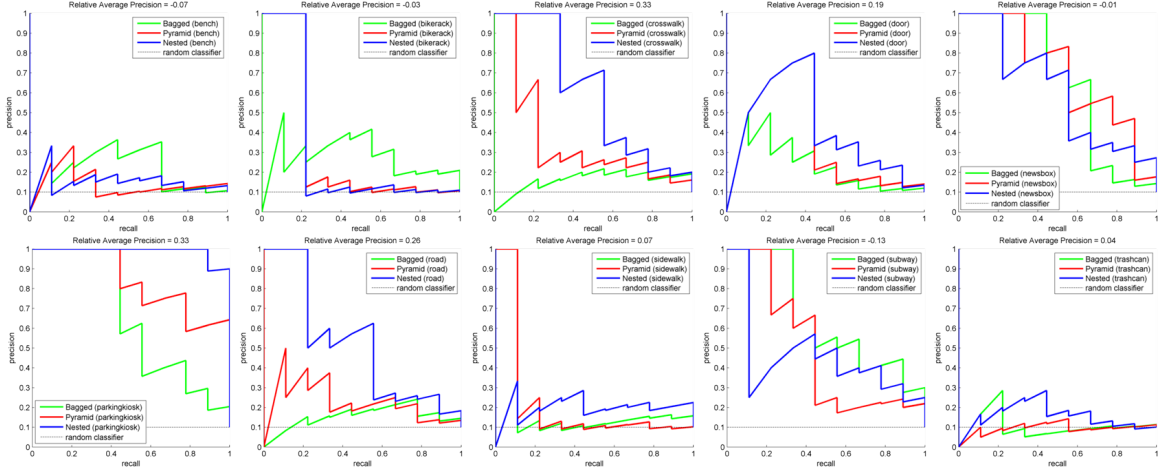


Figure 5.10: Precision recall and average precision on Penn-FSE dataset for nested pooling compared to baseline pyramid pooling and bagged pooling. (rowwise) Class $[AP_{exp}, AP_{baseline}]$: (1) bench [0.26,0.33], (2) bikerack [0.36,0.39], (3) crosswalk [0.61,0.28], (4) door [0.52,0.32], (5) newsbox [0.63,0.64], (6) parkingkiosk [0.98,0.66], (7) road [0.55,0.29], (8) sidewalk [0.32,0.25], (9) subway [0.53,0.66], (10) trashcan [0.27,0.23]. Mean classification rate over all classes shows that nesting provides a 22% improvement over bagging and 61% improvement over temporal pyramid.

- Parking kiosks are confused with newsboxes
- Trashcans are commonly confused with benches since in many of the exemplars, trashcans are located next to benches.
- The worst performance is on “road” which is due to the spatial variation of the orientation of the road not being normalized prior to classification..
- For some classes, such as sidewalk, the top scoring test samples are incorrect which results in a spike to zero of the precision. This effect requires further analysis.

5.6 Summary

We have described a new pooling strategy for representing weakly causal activities called *nested pooling*. We evaluated performance of this representation for functional scene element recognition in a new functional scene element dataset. Results show that the nesting provides a marginal improvement over bagged and pyramid pooled baseline systems for representation of weak causality.

The conclusions that we reach for nested pooling are that this approach provides at best a marginal improvement over other pooling strategies. We collected a challenging new dataset for

functional scene element recognition, and evaluated the nested pooling strategy using a baseline bag of words framework. However, while the nested pooling did provide an improvement over the baseline pyramid and bagged pooling strategies, it was not significant enough to make a claim that this would generalize. Furthermore, there were object classes for which the nested pooling was distinctly worse, even though in aggregate the results were a net improvement. Since these results were not dramatically better on this dataset, we make the claim that while results did improve, we use this as a negative result. The pooling strategy does not have as dramatic affect on the performance of activity recognition as compared to the local descriptors used to construct the representation. So, we do not recommend further analysis along this path.

Acknowledgement. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Nos. W31P4Q-09-C-0051. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the U.S. Government.

Disclaimer. The Penn-FSE data set was approved for use by the University of Pennsylvania where it was collected and institutional review board (IRB) approval was granted. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressly or implied, of the Defense Advanced Research Projects Agency or the U.S. Government. Approved for public release; distribution unlimited.

Chapter 6

Applications to Perception for Unmanned Aerial Systems

In this chapter, we describe two applications of nested descriptors to problems in perception for unmanned aerial systems (UAS). First, we describe the problem of deck pose estimation for landing of an autonomous rotorcraft. We describe the algorithm used to generate the results in section 6.1.1 and show results in section 6.1. Next, we describe an application of the nested pooling to classification of functional scene elements in wide area motion imagery data. We show results on the LAIR dataset in section 6.2.

6.1 Shipboard Landing using Nested Descriptors

In this section, we describe an application of the nested shape descriptors to the problem of visual landing of a rotary wing platform. NSD are used to estimate the position and orientation of a candidate landing zone over a wide range of scales during the approach and landing.

Visual pose estimation for landing is the problem of estimating the 6-DOF position and orientation of a moving landing zone relative to a vehicle with suitable accuracy for safe landing. Given correspondences between an observed image and a known metric markings on the landing zone, we can recover pose using well known techniques of robust homography estimation and decomposition [179].

The primary challenge of this problem is the standardized markings in a landing zone. Figure

6.1 shows short wave infrared (SWIR) imagery collected during a nominal daytime flight showing typical landing zone markings. Observe that the standard markings are composed of a white outer circle, solid inner circle and bisecting line on a grey background. Commonly used feature detectors that rely on corners or scale-space extrema do not provide enough features for robust homography estimation. The nested shape descriptors provide broadly selective response to scale variations and can use edge based detectors, which provides a larger set of interest points for homography estimation.

Performance results are shown in figure 6.1. We collected four landing approaches of 10Hz 2456x2048 color video and 30Hz 640x512 SWIR video of a manned helicopter approaching a static landing zone during midday. We collected differential GPS ground truth position of both the landing zone and the air vehicle with 1σ accuracy of 5-15cm. We manually estimated correspondences between the observed imagery and the reference landing markings to recover the ground truth camera orientation. Next, we processed the video to estimate nested shape descriptors at edge detections, performed greedy assignment, and passed these matches to a robust homography estimation and decomposition to estimate the landing zone position relative to the camera. We compared the estimated landing zone position to differential GPS ground truth and results show that the nested shape descriptors achieve 2σ position errors in X, Y and Z of less than 1ft during the descent and landing.

6.1.1 Deck Pose Estimation

The shipboard landing problem can be defined as follows. Given an image of planar landing deck with known markings, recover the camera pose (translation and rotation) relative to the deck within accuracy requirements suitable for safe landing during all terminal descent stages.

In this section, we describe the computer vision algorithm used to estimate the deck pose from an input image. Our approach is sparse feature based matching using nested descriptors, a least median of squares robust homography estimation [180] with preconditioning [181], bagging [182] and nested shape reprojection error. The novel contribution is the use of the nested shape descriptors during reprojection for increased accuracy and the use of nested covariance for homography estimation to compensate for uncertain correspondence given circular deck markings.

This section is organized as follows. First, we will describe the nested descriptors. These are a local shape representation of an image that is useful for precise correspondence between an input

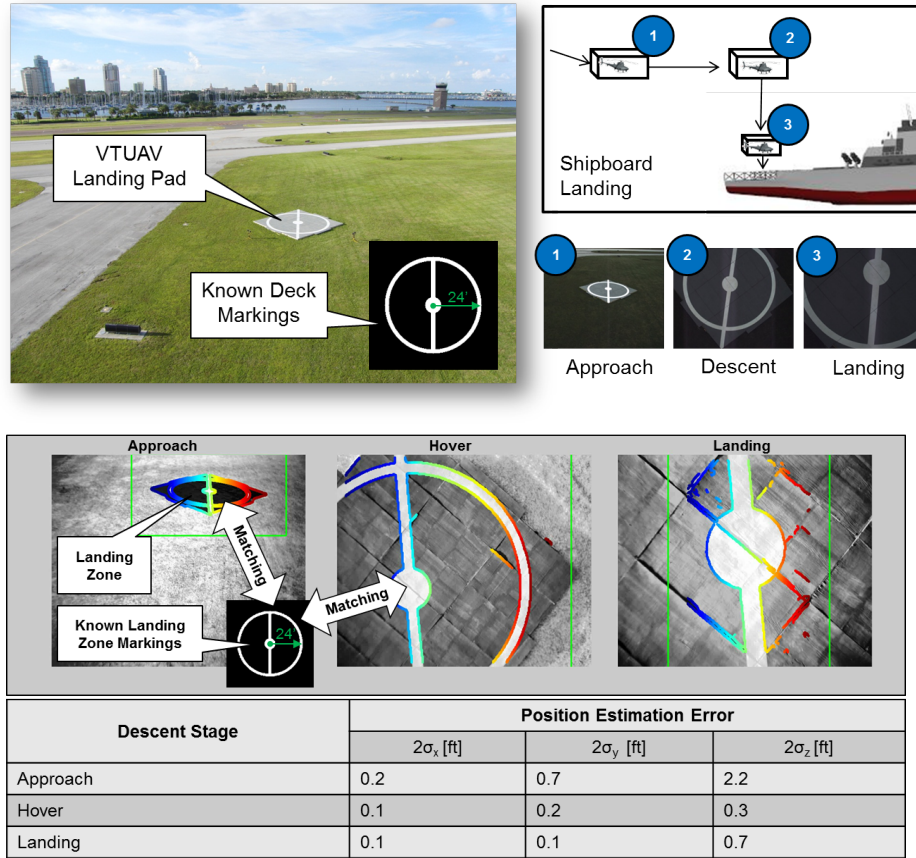


Figure 6.1: Application of the nested shape descriptors to visual landing zone pose estimation. Colors encode the matching of the observed landing zone with the known markings (red=right, blue=left), and the green square encodes the detected position of the landing zone. Nested shape descriptors provide broadly selective scale matching without requiring scale invariant interest points.

image and a reference template. Next, we will derive the optimization for estimating the planar homography between corresponding planar points. This homography is the projective transformation relating the camera and the deck, such that this homography can be decomposed to provide an estimate of the deck pose suitable for shipboard landing. Finally, we will describe the full deck pose estimation algorithm in detail.

6.1.2 Nested Shape Descriptors

Refer to Chapter 3 for a detailed discussion.

6.1.3 Planar Homography

Homography estimation is a well understood problem with a long history [179, 183]. Given a set of projective points x_i in a plane and a corresponding set of projective points x'_i in a second plane, an homography is a projective transformation H such that $x'_i = Hx_i$ for each x_i . Given a camera C with image plane Π' and pixel x' corresponding to a point x in a world plane Π , an homography captures the relationship $x' = Hx$ mapping corresponding points.

In this section, we derive the planar homography from the perspective projection and derive the decomposition of the homography into scene relative position and orientation.

Before introducing homography estimation, we first provide preliminary notation and definitions. Assume that all points p and lines l are in projective (homogeneous) points in \mathbb{P}^2 , unless otherwise noted as Cartesian coordinates in \mathbb{R}^2 by a hatted variable such that projective and Cartesian points are related by homogenization $\hat{p} = [(p_1/p_3) \ (p_2/p_3)]^T$ for elements p_i of p . Let q be a point in pixel coordinates, \tilde{q} be a point in retinal coordinates such that $\tilde{q} = K^{-1}q$ for an intrinsic camera calibration matrix K . Let there be a scene or world coordinate frame W , such that x^W is a point \mathbb{R}^3 relative to the world frame. Similarly, let there be a camera coordinate frame C in the scene such that x^C is a point in \mathbb{R}^3 relative to the camera frame.

The *perspective projection matrix* Π_W is the imaging model from the scene frame to the camera retinal frame.

$$\Pi_W = K \begin{bmatrix} R_C^W & T_W^C \end{bmatrix} \quad (6.1)$$

For a 3D point x in scene coordinates, $\hat{p} = \Pi_W x$ is the corresponding 2D point p in retinal coordinates in the image. The perspective projection Π_W encodes the perspective projection for image formation, and is decomposed into *camera pose* parameters. The camera pose is defined as a rotation R_C^W of the scene frame in the camera frame and the translation T_W^C for the position of the scene origin in the camera frame.

A *homography* is a perspective projection when the scene points are constrained to be planar. Assume that there exists a planar scene, and without loss of generality assume that the scene plane

is at $Z = 0$. The perspective projection reduces to a projective transform or homography H .

$$H = [\Pi_W]_{(1,2,4)} \quad (6.2)$$

where the notation $[A]_{(i,j,k)}$ refers to forming a matrix using only the columns (i, j, k) of A . Observe that since $Z = 0$ (or a linear transform applied to the planar scene points to enforce $Z=0$) the third column corresponding to the Z coordinate can be dropped, reducing the 3×4 perspective projection to a 3×3 projective transformation.

The homography H can be decomposed into a parameters (R, T) which capture the camera pose in terms of the rotation R and translation T of the scene frame in the camera. The rotation and translation have a trivial decomposition using columns of H , which follows directly from (6.1) and (6.2)

$$T_W^C = H_{(3)} \quad (6.3)$$

$$R_C^W = [H_{(1)} \ H_{(2)} \ (H_{(1)} \times H_{(2)})] \quad (6.4)$$

where the notation $u \times v$ is the vector cross product to recover an orthogonal rotation matrix R . The translation T_W^C is the position of the world frame in the camera frame, and provides a camera relative position of the deck in the camera.

6.1.4 Direct Linear Transform

The direct linear transform (DLT) [179] is a classic technique for homography estimation which provides a least squares optimal homography using singular value decomposition. In this section, we derive the direct linear transformation.

The direct linear transform for homography estimation can be described as follows. Given $n \geq 4$ corresponding points (p, q) , for points p in the image corresponding to points q in the scene plane, an optimal homography H^* minimizes the least square error

$$H^* = \arg \min_{H \in \mathcal{H}} \sum_{i \leq n} \|\hat{q}_i - \widehat{HKp_i}\|^2 \quad (6.5)$$

The constrained minimization is performed over the space of feasible rank-8 constrained homogra-

phies \mathcal{H} .

Observe that the norm domain in (6.5) uses *Cartesian* points (\hat{p}, \hat{q}) , however the homography to estimate is a projective transform which is defined up to a scale. These projective points can be *homogenized* to result in Cartesian points \hat{q} where

$$\hat{q}_1 = \frac{H_1 K p}{H_3 K p} \quad (6.6)$$

$$\hat{q}_2 = \frac{H_2 K p}{H_3 K p} \quad (6.7)$$

where the notation H_j refers to the j^{th} row of H . This homogenized notation can be rearranged to an objective $(0 - A_i h)$ such that

$$A_i = \begin{bmatrix} -q_3 p^T K^T & 0^T & q_1 p^T K^T \\ 0^T & -q_3 p^T K^T & q_2 p^T K^T \end{bmatrix} \quad (6.8)$$

where q_j refers to the j^{th} element of vector q . In this formulation, the 3×3 matrix unknown H are reshaped to a 9×1 vector unknown h , such that the (homogenized) linear system $\widehat{H}p = \hat{q}$ is equivalent to $A_i h = 0$. This is repeated for all corresponding points, such that the submatrices A_i are concatenated into an $2n \times 9$ observation matrix A

$$A = \begin{bmatrix} A_1 \\ \vdots \\ A_n \end{bmatrix} \quad (6.9)$$

Finally, the least squares solution for the overdetermined linear system $Ah = 0$ is

$$h^* = \arg \min_{h \in \mathcal{H}} \|Ah\|^2 \quad (6.10)$$

where the constrained minimization is over feasible homography matrices of rank 8, which is equivalent to a norm constraint $\|h\| = 1$ on the vector form. Observe that the matrix optimization (6.5) is equivalent to the vector optimization (6.10).

The constrained objective in (6.10) is minimized by the right singular vector corresponding to the smallest singular value of A , following a standard Rayleigh quotient argument [179]. In the case

with $n = 4$, the vector h spans the nullspace of A and the residual errors are zero. The 9×1 vector h^* is then reshaped rowwise to a 3×3 matrix and normalized by the second singular value λ_2 of A [179]

$$H^* = \lambda_2^{-1} \begin{bmatrix} h_1^* & h_2^* & h_3^* \\ h_4^* & h_5^* & h_6^* \\ h_7^* & h_8^* & h_9^* \end{bmatrix} \quad (6.11)$$

resulting in the least squares homography H^* that minimizes (6.5).

6.1.5 Nested Shape Reprojection Error

A *reprojection error* is an error metric used to score the accuracy of an homography H estimated from a small representative sample using all correspondences. Classic reprojection error considers the *geometric* error of the reprojected points. For example, Hartley and Zisserman’s “Gold Standard” reprojection [179] is

$$E_{\text{geom}}(p, q) = \|\hat{q} - \widehat{Hp}\|^2 \quad (6.12)$$

However, this geometric reprojection error does not take into account the appearance or shape at the reprojected points which can results in fine registration errors. Furthermore, this reprojection does not take into account the uncertainties of correspondences.

In contrast, a shape reprojection error

$$E_{\text{nd}}(p, q) = \|X_T(q) - X_T(Hp)\|_F^2 \quad (6.13)$$

takes into account the local appearance of the image to provide a more precise local distance function that is shape aware. The norm $\|\bullet\|_F$ in (6.13) is the Frobenius matrix norm.

6.1.6 Deck Pose Estimation Algorithm

In this section, we describe a feature based approach to deck pose estimation. Our approach is sparse feature based matching using nested descriptors, a least median of squares robust homography estimation [180] with preconditioning [181], bagging [182] and nested shape reprojection error.

Algorithm 1: Deck Pose Estimation

Input: $K, I, R, p, q, H_D^R, \theta_b, k$ **Output:** \hat{t} $(p_i, q_i) \leftarrow \forall_i \arg \min_j \|X_I(p_i) - X_R(q_j)\|^2$ // Greedy nested shape assignment
 $leastMedian \leftarrow \infty$ **for** $i \leftarrow 1$ **to** $maxiter$ **do** $(\tilde{p}, \tilde{q}) \leftarrow \theta_b(p, q, k)$ // Bagged sample $H_R^I \leftarrow f_{dlr}(K^{-1}\tilde{p}, K^{-1}\tilde{q})$ // Linear Homography $z \leftarrow \|X_I(KH_R^I K^{-1}q) - X_R(q)\|_F^2$ // Nested shape reprojection error (6.13) **if** $median(z) < leastMedian$ **then** $leastMedian = median(z)$ // Least median of squares $\hat{t} = [(KH_R^I K^{-1})H_D^R]_{(3)}$ // Position of deck origin in camera frame

(6.3)

The novel contribution is the use of the nested shape descriptors during reprojection for increased accuracy to compensate for uncertain correspondence given circular deck markings.

The algorithm for deck pose estimation which uses the nested shape descriptors and robust homography estimation is summarized in algorithm 1. This algorithm assumes as input

- K : a calibrated intrinsic calibration matrix determined from an offline calibration procedure to determine focal length, principal point, and lens distortion [184].
- I : an input grayscale image of size $M \times N$
- R : a reference image of size $M' \times N'$. This reference template is determined offline from reference drawings of the deck. An example reference template is the black and white circle shown in figure 6.1. This is the standard deck marking for shipboard landing for the Firescout UAS on air capable ships.
- p : n interest points in I , as determined from Canny edge positions.
- q : m interest points in R as determined from Canny edge positions.
- H_D^R : a presurveyed homography from the deck to reference template determined from a manual correspondence between the corners of reference template and a manually metric survey of the deck, and these correspondences are input to the DLT algorithm in section 6.1.4.
- θ_b : a bagged sample function with sample regions defined to capture the unique structures in the template, such as the center circle and the upper and lower corners.
- k : The number of correspondences to use in the homography s, such that $k \geq 4$.

- *maxiter*: The maximum number of iterations of the least median of squares randomized homography search.

Algorithm 1 proceeds as follows. First, we compute an initial correspondence using greedy assignment of nested descriptors. $X_I(p)$ is the nested descriptor computed from image I at interest points p , and $X_R(q)$ is the nested descriptor from reference template R at interest points q . For each descriptor in the observed image using the minimum pairwise Euclidean distance to all nested shape descriptors in the template. This provides a list of n initial correspondences (\hat{p}, \hat{q}) .

Next, we perform bagged sampling [182]. The bagged sample function θ_b is a function that selects k correspondences at random from the initial correspondence list, such that samples are chosen in a round robin selection phase from one of b *sample bags*. A sample bag is a regions of interest in the reference template, such that a correspondence (p, q) drawn from bag j guarantees that point q is in sample region j . In this application, the bagged sample function defines five sampling regions from the template that capture the circle corners and lines as groups. k samples are chosen round robin from each of these bags in order. The bagged sampling function provides efficiency in the random sample so that non-degenerate and unambiguous correspondences are chosen. For example, if all samples are drawn from only bags 1 and 2 there exists a rotational symmetry in the outer circle that will results in an incorrect alignment. Similarly, if all of the samples are chosen from bag 5, then since the samples are closely spaced, the resulting homography is likely to be numerically unstable. By performing round-robin random sampling from each bag in order, we maintain well separated non-degenerate and non-ambiguous correspondences for use in homography estimation.

Next, we perform robust homography estimation using a least median of squares approach [180]. We randomly select k correspondences using the bagged sampling, then we compute the DLT homography from this random sample. The points are first preconditioned for improved numerical stability [181]. The nested covariance is important to provide an uncertainty of the correspondences since the majority of points on a deck template are ambiguous, such as points on the inner line or the outer circle. Next, we compute the nested shape reprojection error (6.13) for the remaining points, and compute the median of this reprojection error. If the median is less than the least median so far, we update the best solution and iterate. The algorithm terminates after *maxiter* iterations.

The output of the algorithm is \hat{t} , the best position of the deck in the camera frame as estimated

from the homography decomposition (6.3). This uses the transitivity property of homographies $H_D^I = H_T^I H_D^T$, such that a homography from deck to image (H_D^I) is given by the product of presurveyed homography from deck to reference template (H_D^R) and estimated homography from template to image (D_R^I).

6.2 Classification in Aerial Imagery using Nested Pooling

In this section, we show classification results applying the nested pooling from chapter 5 to the LAIR 2010 dataset.

The LAIR 2010 dataset is a wide area motion imagery (WAMI) dataset containing seven minutes of high altitude, high resolution imagery of an urban environment. This WAMI dataset was annotated with polygonal bounding boxes capturing 523 annotations of 28 classes of functional scene elements and buildings. These classes include roads, intersections, crosswalks, check points, parking lots, banks, gas stations, churches and restaurants. These annotations serve as a labeled training set for evaluation of a functional scene element classification task. This dataset is distribution limited, therefore example imagery cannot be shown in this thesis.

The experimental system for this evaluation is as follows An analyst extracts a bounding box in a large WAMI image and asks "what is in this bounding box?". The experimental system performs preprocessing to stabilize the WAMI data, then the subsystem shown in the green box performs classification of this bounding box. The processing chain includes four steps: spatiotemporal interest point extraction [54], hog-hof descriptor computation, nested pooling construction, local naive bayes nearest neighbor classification [178]. The final classification uses an offline training set constructed by user annotation.

The experimental protocol for evaluation on this dataset was leave one out cross validation over 378 rounds. We generated precision recall curves for each class by considering all LOOCV rounds for the class under test.

The precision recall performance on a subset of this labeled dataset is shown in figure 6.2 and 6.3. Confusion matrix and summary results are shown in figure 6.4 which shows that the overall classification rate is 0.33 and a mean average precision of 0.24.

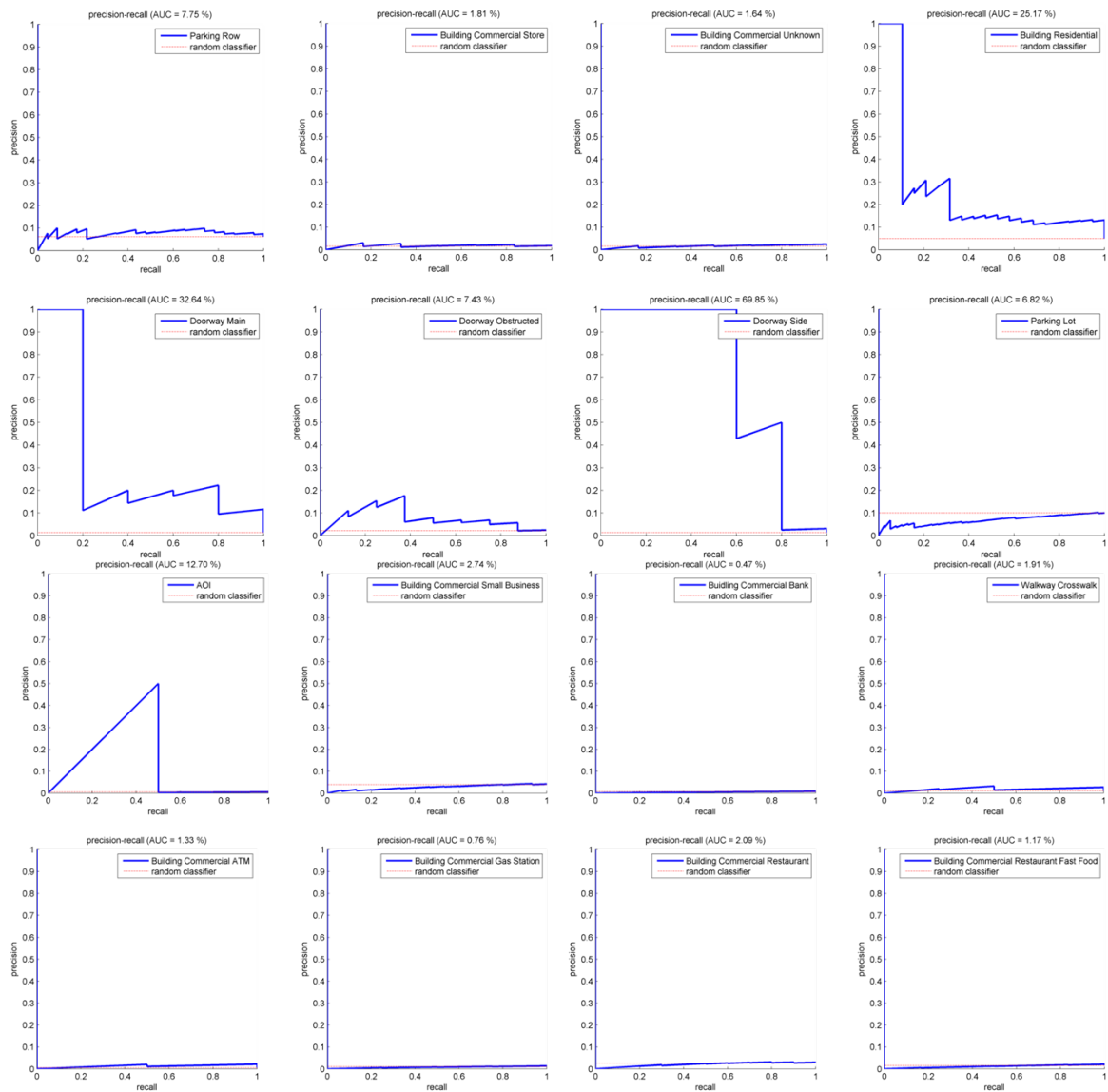


Figure 6.2: LAIR precision-recall performance evaluation

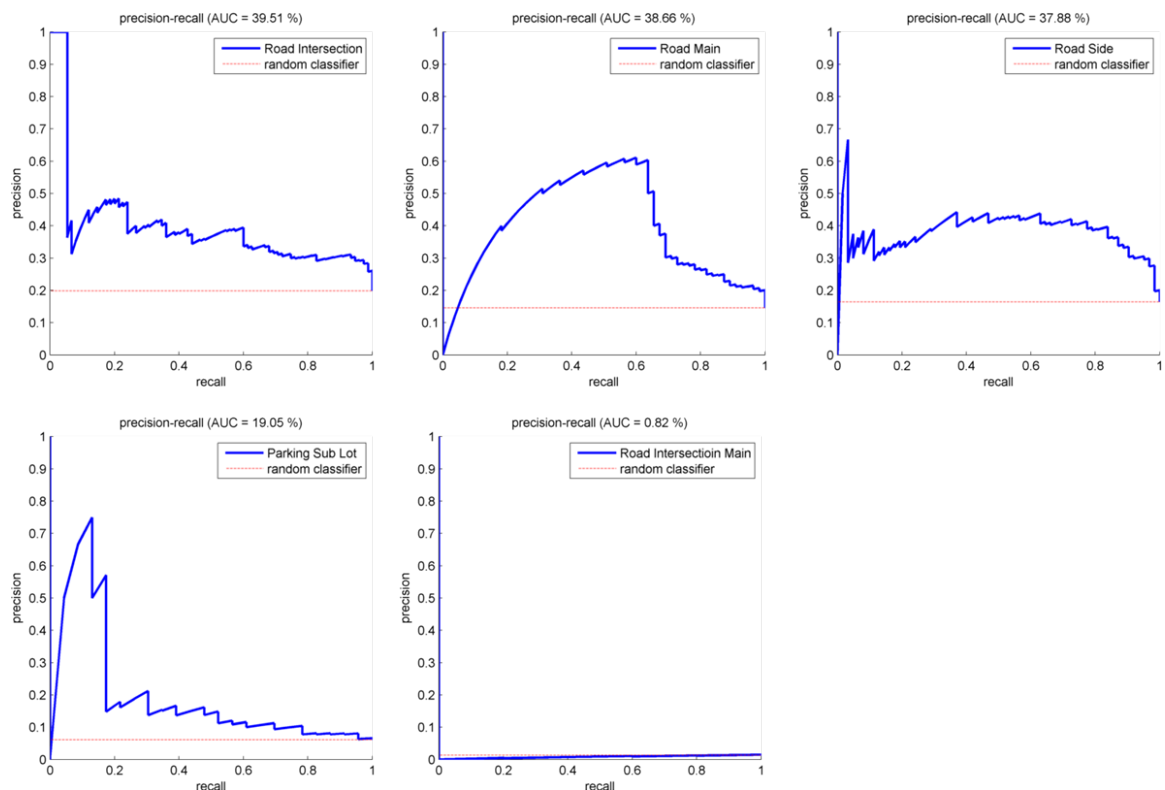


Figure 6.3: LAIR precision-recall performance evaluation (continued)

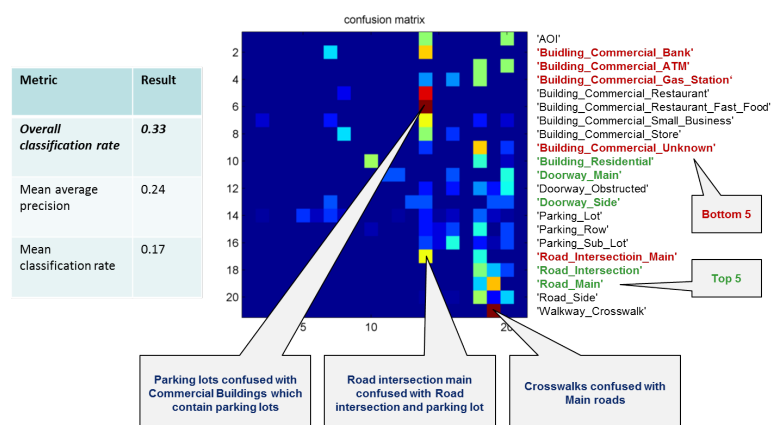


Figure 6.4: LAIR performance evaluation summary. Shown are summary statistics and confusion matrix result for classification performance. The summary also includes annotations for the best and worst classes, and highlights those classes that are most often confused.

Chapter 7

Conclusions

In this thesis, we introduced shape representations using *nested descriptors*. This thesis made three primary contributions: nested shape descriptors, nested motion descriptors and nested pooling. We described the place for these descriptors in the larger framework of global vs. local shape representations, we provided the theoretical foundation for each of these approaches, discussed related work and showed state of the art results for each of these approaches for representing shape in imagery or video.

The conclusions we reach are that shape representation using nested shape descriptors and nested motion descriptors provide a significant improvement over the state-of-the-art for representation of local shape. These descriptors outperformed the state of the art, and can serve as a new foundation for local shape when constructing global shape representations. Furthermore, we demonstrated a connection between the nested descriptors and salient edges and motion that suggest that the representational power of these descriptors is due to the representation not of pooled spatiotemporal edges, but rather pooled salient spatiotemporal edges. Finally, we show an intuitive visualization of these descriptors in imagery and video that provides an natural way of displaying the descriptor contents to a user for debugging a higher level shape representation.

The conclusions that we reach for nested pooling are that this approach provides at best a marginal improvement over other pooling strategies. We collected a challenging new dataset for functional scene element recognition, and evaluated the nested pooling strategy using a baseline bag of words framework. However, while the nested pooling did provide an improvement over the baseline pyramid and bagged pooling strategies, it was not significant enough to make a claim that

this would generalize. Furthermore, there were object classes for which the nested pooling was distinctly worse, even though in aggregate the results were a net improvement. Since these results were not dramatically better on this dataset, we make the claim that while results did improve, we use this as a negative result. The pooling strategy does not have as dramatic affect on the performance of activity recognition as compared to the local descriptors used to construct the representation. So, we do not recommend further analysis along this path.

The nested descriptors suggest extensions that can be further evaluated as future work. The approaches for shape representation presented in this thesis focus on representations that are globally local. The related work discusses the tradeoffs between local, global and globally local representations, and suggests that using a globally local representation as attributes to construct a global representation suggests an improved representation. We believe that the nested descriptors provide a solid foundation for further exploration of such higher level global shape representations.

Finally, we observe the connection between nested descriptors and *second order isomorphism*. In the related work, we discussed the global shape representation theorems based on Shepard's classic work on second order isomorphism [133, 1], which proposes that representations of shape is not the shape itself, but rather relationships between representations and not the representations themselves. We have shown that part of the representational power of the nested descriptors comes from the log-spiral normalization, which is in effect a difference of differences, or a second order shape representation. This form of representation is a fundamental departure from the classic first order representations of shapes in terms of templates, and suggests that future research should focus on shape representations with similar second order effects.

Bibliography

- [1] S. Edelman, *Representation and Recognition in Vision*, The MIT Press, 1999. vi, 1, 14, 15, 16, 43, 46, 51, 52, 154
- [2] D. Freedman and C. Chen, “Algebraic topology for computer vision,” *Nova Science*, 2011. vi, 30, 33
- [3] R. Ghrist, “Barcodes: the persistent topology of data,” *Amer. Math. Soc. Current Events Bulletin*, Jan. 2007. Revised version in *Bull. Amer. Math. Soc.*, vol. 45, pp. 61–75, 2008. vi, 28, 35, 36, 37
- [4] T. K. Dey, A. Hirani, and B. Krishnamoorthy, “Optimal homologous cycles, total unimodularity, and linear programming,” in *42nd ACM Sympos. Comput. Theory (STOC 2010)*, 2010, pp. 221–230. vi, 37, 38, 43
- [5] S. Palmer, *Vision Science: Photons to Phenomenology*, MIT Press, 1999. 1, 2, 25
- [6] S. Edelman, “Representation, similarity, and the chorus of prototypes,” *Minds and Machines*, pp. 45–68, 1995. 1, 15, 51
- [7] S. Edelman, “Representation is representation of similarities,” *Behavioral and Brain Sciences*, vol. 21, no. 4, pp. 449–467, 1998. 1, 15
- [8] S.J. Dickinson, A. Leonardis, B. Schiele, and M. Tarr, *Object Categorization*, chapter The Evolution of Object Categorization and the Challenge of Image Abstraction, Cambridge University Press, 2009. 1, 14, 16, 25
- [9] S. Palmer, *Object perception: Structure and Process*, chapter Reference frames in the perception of shape and orientation, pp. 121–163, Lawrence Erlbaum Associates, 1989. 1

- [10] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*, MIT Press, 1982. 2, 15, 22
- [11] I.L. Dryden and K.V. Mardia, *Statistical Shape Analysis*, Wiley, 1998. 2
- [12] T. F. Cootes, D. H. Cooper, C. J. Taylor, and J. Graham, “A trainable method of parametric shape description,” *Image and Vision Computing*, vol. 10, no. 5, pp. 289–294, June 1992. 3
- [13] T. F. Cootes, G. J. Edwards, and C. J. Taylor, “Active appearance models,” *IEEE TPAMI*, vol. 23, no. 6, pp. 681–685, 2001. 3
- [14] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *In CVPR*, 2005, pp. 886–893. 3
- [15] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *In CVPR*, 2006, pp. 2169–2178. 3, 10, 45, 128
- [16] J. Mutch and D. Lowe, “Object class recognition and localization using sparse features with limited receptive fields,” *International Journal of Computer Vision (IJCV)*, vol. 80, no. 1, pp. 45–57, October 2008. 3, 5, 17, 18, 54, 56
- [17] T. Serre, L. Wolf, and T. Poggio, “Object recognition with features inspired by visual cortex,” in *CVPR*, 2005. 3
- [18] P. Viola and M. Jones, “Robust real-time object detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2001. 3
- [19] U. Grenander and M. Miller, *Pattern Theory: From Representation to Inference*, chapter , Oxford University Press, 2007. 3
- [20] N. Pinto, Y. Barhomi, DD Cox, and JJ DiCarlo, “Comparing state-of-the-art visual features on invariant object recognition tasks,” in *EEE Workshop on Applications of Computer Vision (WACV 2011)*, 2011. 3
- [21] L. Zhu, Y. Chen, and A. Yuille, “Learning a hierarchical deformable template for rapid deformable object parsing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 32, no. 6, pp. 1029–1043, 2010. 3, 15

- [22] Y. Amit and P. Troune, “Patchwork of parts models for object recognition,” *IJCV*, vol. 75, no. 2, November 2007. 3
- [23] P.F. Felzenszwalb and D.P. Huttenlocher, “Pictorial structures for object recognition,” *IJCV*, vol. 61, pp. 2005, 2003. 3, 4, 5, 43
- [24] S. Fidler and M. Boben, *Object Categorization: Computer and Human Vision Perspectives*, chapter Learning Hierarchical Compositional Representations of Object Structure, Cambridge University Press, 2009. 3, 4, 5, 15
- [25] P. Felzenszwalb, D. McAllester, and D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *CVPR*, 2008. 3, 4, 5, 42
- [26] S.C. Zhu and D. Mumford, “A stochastic grammar of images,” *Found. Trends. Comput. Graph. Vis.*, vol. 2, pp. 259–362, January 2006. 3, 4, 5, 15
- [27] Y. Jin and S. Geman, “Context and hierarchy in a probabilistic image model,” in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, 2006, pp. 2145–2152. 3, 4, 5, 15, 45
- [28] L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille, “Part and appearance sharing: Recursive compositional models for multi-view multi-object detection,” in *CVPR*, 2010. 3, 4, 5, 15
- [29] B. Leibe, A. Leonardis, and B. Schiele, “Combined object categorization and segmentation with an implicit shape model,” in *In ECCV workshop on statistical learning in computer vision*, 2004, pp. 17–32. 3, 4, 5, 43
- [30] A. C. Berg, T. L. Berg, and J. Malik, “Shape matching and object recognition using low distortion correspondences,” in *CVPR*, 2005. 3, 4, 5, 24, 43
- [31] 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. 24, pp. 509–521, April 2002. 3, 4, 5, 17, 18, 24, 26, 43, 56

- [32] K. Siddiqi, A. Shokoufandeh, S. Dickinson, and S. Zucker, “Shock graphs and shape matching,” *International Journal of Computer Vision*, vol. 35, no. 1, pp. 13–32, 1999. 3, 4, 5, 43
- [33] P. Srinivasan, Q. Zhu, and J. Shi, “Many-to-one contour matching for describing and discriminating object shape,” in *CVPR*, 2010. 3, 4, 5, 42, 43
- [34] A. Lehmann, B. Leibe, and L. Van Gool, “Prism: Principled implicit shape model,” in *BMVC*, 2009. 3, 4, 5
- [35] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004. 5, 17, 18, 54, 56, 73, 76, 80
- [36] Y. Ke and R. Sukthankar, “Pca-sift: A more distinctive representation for local image descriptors,” in *CVPR*, 2004. 5, 17, 18, 56, 57
- [37] T. Ojala, M. Pietikainen, and T. Maenpää, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002. 5, 17, 18, 56
- [38] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, “SURF: Speeded up robust features,” *Computer Vision and Image Understanding*, vol. 10, pp. 346–359, 2008. 5, 17, 18, 54, 56
- [39] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 27, no. 10, pp. 1615–1630, 2004. 5, 13, 17, 18, 54, 56, 57
- [40] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, “Chog: Compressed histogram of gradients - a low bit rate feature descriptor,” in *CVPR*, 2009. 5, 17, 18, 54, 56
- [41] S. Winder, Gang Hua, and M. Brown, “Picking the best daisy,” in *CVPR*, 2009. 5, 17, 56, 80
- [42] E. Tola, V. Lepetit, and P. Fua, “Daisy: An efficient dense descriptor applied to wide-baseline stereo,” *PAMI*, vol. 32, no. 5, pp. 815–830, 2010. 5, 17, 18, 54, 56, 80, 85

- [43] S. Leutenegger, M. Chli, and R. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *ICCV*, 2011. 5, 17, 18, 54, 56, 73, 76, 80
- [44] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *ECCV*, 2010. 5, 17, 18, 54, 56
- [45] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *ICCV*, 2011. 5, 17, 18, 54, 56, 80
- [46] A. Alahi and R. Ortiz and P. Vandergheynst, “FREAK: Fast retina keypoint,” in *CVPR*, 2012. 5, 17, 18, 54, 56, 80
- [47] D. Fleet and A. Jepson, “Stability of phase information,” *IEEE Trans on Pattern Anal. and Mach. Intell. (PAMI)*, vol. 15, no. 12, pp. 1253–1268, 1993. 9, 94, 97, 102, 103, 104, 105
- [48] D. Fleet and A. Jepson, “Computation of component image velocity from local phase information,” *International Journal of Computer Vision*, vol. 5, no. 1, pp. 77–104, 1990. 9, 94, 97, 102, 103, 104
- [49] W. Freeman, E. Adelson, and D. Heeger, “Motion without movement,” *ACM Computer Graphics, (SIGGRAPH’91)*, vol. 25, no. 4, pp. 27–30, July 1991. 9, 94, 97, 102
- [50] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman, “Phase-based video motion processing,” *ACM Trans. Graph. (Proceedings SIGGRAPH 2013)*, vol. 32, no. 4, 2013. 9, 94, 97, 102
- [51] J. Byrne and J. Shi, “Nested shape descriptors,” in *ICCV*, 2013. 9, 10, 95, 96, 107, 108, 127
- [52] L. Fei-Fei, R. Fergus, and P. Perona, “Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories,” in *IEEE CVPR04 Workshop on Generative-Model Based Vision*, 2004. 10, 15
- [53] J. Aggarwal and M. Ryoo, “Human activity analysis: A review,” *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, April 2011. 11, 19, 95, 127, 128
- [54] I. Laptev, “On space-time interest points,” *IJCV*, 2005. 11, 20, 96, 114, 115, 127, 128, 136, 150

- [55] H. Wang, M. M. Ullah, A. Klser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in *BMVC*, 2009. 11, 127
- [56] T. Darrell and A. Pentland, “Space-time gestures,” in *CVPR*, 1993. 11, 127
- [57] A. Prest, C. Schmid, and V. Ferrari, “Weakly supervised learning of interactions between humans and objects,” *PAMI*, vol. 34, no. 3, pp. 601–614, March 2012. 11, 127, 128
- [58] S. Loncaric, “A survey of shape analysis techniques,” *Pattern Recognition*, vol. 31, no. 8, pp. 983–1001, August 1998. 13
- [59] R. Veltkamp and M. Hagedoorn, ,” in *Principles of visual information retrieval*, Michael S. Lew, Ed., chapter State of the art in shape matching, pp. 87–119. Springer-Verlag, 2001. 13
- [60] B. Munsell, P. Dalal, and S. Wang, “Evaluating shape correspondence for statistical shape analysis: A benchmark study,” *PAMI*, vol. 30, no. 11, pp. 2023–2039, November 2008. 13
- [61] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors - survey,” *CVG*, vol. 3, no. 1, pp. 1–110, 2008. 13, 18, 54, 56
- [62] S.V.N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt, “Graph kernels,” *Journal of Machine Learning Research*, vol. 11, pp. 12011242, April 2010. 14
- [63] S. Ullman, *High Level Vision*, MIT Press, 1996. 15
- [64] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. L. Yuille, “Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion,” in *ECCV*, 2008. 15
- [65] A. Leonardis, “Evaluating multi-class learning strategies in a generative hierarchical framework for object detection,” in *NIPS*, 2009. 15
- [66] B. Ommer and J. Buhmann, “Learning the compositional nature of visual object categories for recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 99, 2010. 15
- [67] G. Hinton, S. Osindero, and Y. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 1527-1554, 2006. 15

- [68] I. Kokkinos and A.L. Yuille, “Hop: Hierarchical object parsing,” in *CVPR*, June 2009. 15
- [69] K. Grauman and B. Leibe, “Visual object recognition,” *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 5, no. 2, pp. 1–181, April 2011. 16
- [70] G. L. Murphy, *The Big Book of Concepts*, MIT Press, 2002. 16, 47, 48, 49
- [71] G. Lakoff, *Women, Fire and Dangerous Things: What categories reveal about the mind*, University of Chicago Press, 1987. 16, 47, 49
- [72] M. Yang, D. Kriegman, and N. Ahuja, “Detecting faces in images: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 1, pp. 34–58, 2002. 16
- [73] N. Werghi, “Segmentation and modeling of full human body shape from 3-d scan data: A survey,” *IEEE transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2007. 16
- [74] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Gool, “A comparison of affine region detectors,” *IJCV*, vol. 65, no. 1, pp. 43–72, 2005. 18, 54, 56, 73, 75, 76
- [75] M. Muja and D. Lowe, “Fast matching of binary features,” in *CRV*, 2012. 18, 56
- [76] J. K. Aggarwal and M. S. Ryoo, “Human activity analysis: A review,” *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, April 2011. 19, 95
- [77] T. Hassner, “A critical review of action recognition benchmarks,” in *CVPR*, 2013. 19, 95
- [78] Christian Schuldt, Ivan Laptev, and Barbara Caputo, “Recognizing human actions: A local svm approach,” in *ICPR*, 2004. 19, 93, 95, 114, 116
- [79] H. Kuhne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “Hmdb: A large video database for human motion recognition,” in *ICCV*, 2011. 19, 93, 95, 98, 113, 114, 117
- [80] Kishore K. Reddy and Mubarak Shah, “Recognizing 50 human action categories of web videos,” *Machine Vision and Applications Journal (MVAP)*, 2012. 19, 93, 95, 114

- [81] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid, “Actions in context,” in *IEEE Conference on Computer Vision & Pattern Recognition*, 2009. 19, 93, 95
- [82] Philip H.S. Torr Michael Sapienza, Fabio Cuzzolin, “Learning discriminative space-time action parts from weakly labelled videos,” *International Journal of Computer Vision*, 2013. 20, 95
- [83] W. Zhang, M. Zhu, and K. Derpanis, “From actemes to action: A strongly-supervised representation for detailed action understanding,” in *ICCV*, 2013. 20, 95
- [84] L.M. Wang, Y. Qiao, and X. Tang, “Motionlets: Mid-level 3d parts for human motion recognition,” in *CVPR*, 2013. 20, 95
- [85] L.M. Wang, Y. Qiao, and X. Tang, “Mining motion atoms and phrases for complex action recognition,” in *ICCV*, 2013. 20, 95
- [86] Navneet Dalal, Bill Triggs, and Cordelia Schmid, “Human detection using oriented histograms of flow and appearance,” in *ECCV*, 2006. 20, 93, 95, 96
- [87] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *CVPR*, 2008. 20, 93, 95, 128, 137, 138
- [88] Garrison Cottrell Piotr Dollr, Vincent Rabaud and Serge Belongie, “Behavior recognition via sparse spatio-temporal features,” in *ICCV VS-PETS 2005*, 2005. 20, 95
- [89] G. Willems, T. Tuytelaars, and L. Van Gool, “An efficient dense and scale-invariant spatio-temporal interest point detector,” in *ECCV*, 2008. 20, 95
- [90] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid, “A spatio-temporal descriptor based on 3d-gradients,” in *BMVC*, 2008. 20, 93, 95, 114, 115
- [91] H. Wang, M. M. Ullah, A. Klser, I. Laptev, and C. Schmid, “Evaluation of local spatio-temporal features for action recognition,” in *BMVC*, 2009. 20, 96, 114, 120
- [92] P. Bilinski and F. Bremond, “Evaluation of local descriptors for action recognition in videos,” in *International Conference on Computer Vision Systems*, Sophia Antipolis, France, 2011. 20, 96, 114

- [93] M. Jain, H. Jegou, and P. Bouthemy, “Better exploiting motion for better action recognition,” in *CVPR*, 2013. 20, 93, 96
- [94] H. Wang, A. Klser, C. Schmid, and L. Cheng-Lin, “Action recognition by dense trajectories,” in *CVPR*, 2011. 20, 93, 96
- [95] H. Wang, A. Klaeser, C. Schmid, and C-L Liu, “Dense trajectories and motion boundary descriptors for action recognition,” *IJCV*, 2013. 20, 93, 96
- [96] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, “Motion interchange patterns for action recognition in unconstrained videos,” in *ECCV*, 2012. 20, 93, 96
- [97] Y. Hanani, N. Levy, and Lior Wolf, “Evaluating new variants of motion interchange patterns,” in *CVPR workshop on action similarity in unconstrained video*, 2013. 20, 93, 96
- [98] L. Yeffet and L. Wolf, “Local trinary patterns for human action recognition,” in *ICCV*, 2009. 20, 93, 96
- [99] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *ICCV*, 2013. 20, 93, 96
- [100] X Peng, Y Qiao, Q Peng, and X Qi, “Exploring motion boundary based sampling and spatial-temporal context descriptors for action recognition,” in *BMVC*, 2013. 20, 93, 96
- [101] H. Weng and C. Schmid, “Lear-inria submission for the thumos workshop,” in *THUMOS: The First International Workshop on Action Recognition with a Large Number of Classes, in conjunction with ICCV ’13, Sydney, Australia.*, 2013. 20, 93, 96, 114, 123
- [102] R. Zass and A. Shashua, “Probabilistic graph and hypergraph matching,” in *CVPR*, June 2008. 21, 24
- [103] R. Diestel, *Graph Theory*, Springer-Verlag, 2010. 22, 26
- [104] H. Bunke, “Graph matching: Theoretical foundations, algorithms, and applications,” in *In Proc. Vision Interface 2000*, 2000, pp. 82–88. 22
- [105] D. Conte, P. Foggia, C. Sansone, and M. Vento, “Thirty years of graph matching in pattern recognition,” *IJPRAI*, vol. 18, no. 3, pp. 265–298, May 2004. 22

- [106] J. R. Ullman, “An algorithm for subgraph isomorphism,” *J. Assoc. Comput. Mach.*, vol. 31, no. 42, 1976. 23
- [107] S. Gold and A. Rangarajan, “A graduated assignment algorithm for graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, 1996. 23, 43
- [108] M. Leordeanu and M. Hebert, “A spectral technique for correspondence problems using pairwise constraints,” in *CVPR*, 2005. 23
- [109] T. Cour, P. Srinivasan, and J. Shi, “Balanced graph matching,” in *Advances in Neural Information Processing Systems (NIPS)*, 2006. 23, 43
- [110] C. Schellewald and C. Schnorr, “Probabilistic subgraph matching based on convex relaxation,” in *In Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005. 23
- [111] H. Bunke, “Error correcting graph matching: On the influence of the underlying cost function,” *IEEE Trans. PAMI*, vol. 21, 1999. 23, 24
- [112] P. Bille, “A survey on tree edit distance and related problems,” *Theoretical Computer Science*, vol. 337, pp. 217–239, 2005. 24, 43
- [113] D. Shasha and K. Zhang, “Simple fast algorithms for the editing distance between trees and related problems,” *SIAM J. Comput.*, vol. 16, no. 6, pp. 1245–1262, 1989. 24, 43
- [114] P.N. Klein, “Computing the edit-distance between unrooted ordered trees,” in *In Proceedings of the 6th annual European Symposium on Algorithms (ESA)*. 1998, pp. 91–102, Springer-Verlag. 24, 43
- [115] X. Gao, B. Xiao, D. Tao, and X. Li, “A survey of graph edit distance,” *Pattern Analysis and Applications*, vol. 13, January 2010. 24
- [116] H. Jiang, S. X. Yu, and D. R. Martin, “Linear scale and rotation invariant matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010. 24
- [117] H. Li, E. Kim, X. Huang, and L. He, “Object matching with a locally affine-invariant constraint,” in *CVPR*, 2010. 24

- [118] J. Kleinberg and E. Tardos, *Algorithm Design*, Addison Wesley, 2005. 26
- [119] T. Dey, H. Edelsbrunner, and S. Guha, “Computational topology,” *Advances in Discrete and Computational Geometry*, pp. 109–143, 1999. 27, 28, 29
- [120] Bern et. al, “Emerging challenges in computational topology,” in *Results of the NFS Workshop on Computational Topology*, 1999. 27, 28
- [121] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian, “on the local behavior of spaces of natural images,” *International Journal of Computer Vision*, vol. 76, pp. 1–12, January 2008. 28, 37
- [122] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 2002. 30, 32, 33
- [123] H. Edelsbrunner and J. Harer, “Persistent homology a survey,” *AMS*, 2007. 30
- [124] T. E. Goldberg, “Combinatorial laplacians of simplicial complexes,” Tech. Rep., Bard University, 2002. 32
- [125] A. Zomorodian, “Fast construction of the vietoris-rips complex,” in *Computers and Graphics, Shape Modeling International, Aix-en-Provence, France*, 2010. 36
- [126] A. Lee, K. Pedersen, and D. Mumford, “The nonlinear statistics of high-contrast patches in natural images,” *International Journal of Computer Vision*, vol. 54, pp. 83–103, 2003. 37
- [127] A. Tahbaz-Salehi and A. Jadbabaie, “Distributed coverage verification algorithms in sensor networks without location information,” *IEEE Transactions on Automatic Control*, vol. 55, 2010. 38
- [128] A. Schrijver, *Theory of linear and integer programming*, Wiley-Interscience series in discrete mathematics and optimization, 1986. 39, 41
- [129] C. Gu, J. Lim, P. Arbellez, and J. Malik, “Recognition using regions,” in *CVPR*, 2009. 42, 43
- [130] S. Maji and J. Malik, “Object detection using a max-margin hough transform,” in *CVPR*, 2009, pp. 1038–1045. 42

- [131] A. Frome, Y. Singer, F. Sha, and J. Malik, “Learning globally-consistent local distance functions for shape-based image retrieval and classification,” in *ICCV*, 2007. 42
- [132] D. Ramanan and S. Baker, “Local distance functions: A taxonomy, new algorithms, and an evaluation,” *IEEE Pattern Analysis and Machine Intelligence (PAMI)*, vol. 33, no. 4, pp. 794–806, April 2011. 42, 57, 64
- [133] R. Shepard and S. Chipman, “Second order isomorphism of internal representations: Shapes of states,” *Cognitive Psychology*, vol. 1, no. 1, pp. 1–17, 1970. 43, 51, 154
- [134] B. Leibe, A. Leonardis, and B. Schiele, “Robust object detection with interleaved categorization and segmentation,” *International Journal of Computer Vision Special Issue on Learning for Recognition and Recognition for Learning*, vol. 77, pp. 259–289, 2008. 43
- [135] L. Fei-Fei and P. Perona, “A bayesian hierarchical model for learning natural scene categories,” in *CVPR*, 2005. 45, 50, 127, 130, 137
- [136] E. Rosch, *Cognition and Categorization*, chapter Principles of Categorization, pp. 27–48, Erlbaum, Hillsdale, NJ, 1978. 49
- [137] T. Malisiewicz, A. Gupta, and A. Efros, “Ensemble of exemplar-svms for object detection and beyond,” in *ICCV*, 2011. 52
- [138] T. Malisiewicz and A. Efros, “Beyond categories: The visual memex model for reasoning about object relationships,” in *NIPS*, 2009. 52
- [139] R. Brooks, “Intelligence without representation,” *Artificial Intelligence*, vol. 47, pp. 139–159, 1991. 52
- [140] D. Weinberger, *Everything is Miscellaneous: The Power of the new digital disorder*, Times Books Henry Holt and Company, 2007. 53
- [141] E.P. Simoncelli and W.T. Freeman, “The steerable pyramid: A flexible architecture for multi-scale derivative computation,” in *IEEE Second Int’l Conf on Image Processing*, 1995. 61, 62, 73, 97, 100, 101
- [142] A. Tversky, “Features of similarity,” *Psychological Review*, vol. 84, pp. 327–352, 1977. 68

- [143] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009. 69
- [144] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *IJCV*, vol. 47, no. 1-3, pp. 7–42, April-June 2002. 75
- [145] E. Mair, G. Hager, D. Burschka, M. Suppa, and G. Hirzinger, “Adaptive and generic corner detection based on the accelerated segment test,” in *ECCV*, 2010. 76
- [146] K. Lenc, V. Gulshan, and A. Vedaldi, “Vlbenchmarks-1.0-beta,” <http://www.vlfeat.org/benchmarks/>, 2012. 76
- [147] Biliana Kaneva, Antonio Torralba, and William T. Freeman, “Evaluating image features using a photorealistic virtual world,” in *IEEE International Conference on Computer Vision*, 2011. 78
- [148] C. Liu, J. Yuen, and A. Torralba, “Sift flow: Dense correspondence across scenes and its applications,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, 2011. 85
- [149] T. Tuytelaars, “Dense interest points,” in *CVPR*, 2010. 85
- [150] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, November 1998. 87
- [151] T. Judd, F. Durand, and A. Torralba, “A benchmark of computational models of saliency to predict human fixations,” Tech. Rep., MIT, January 2012. 90
- [152] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum, “Learning to detect a salient object,” in *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Minneapolis, Minnesota, 2007. 90
- [153] D. Fleet, *Measurement of Image Velocity*, Kluwer Academic Press, 1992. 97, 103, 104
- [154] A. Jepson and D. Fleet, “Phase singularities in scale-space,” *Image and Vision Computing Journal*, vol. 9, no. 5, pp. 338–343, 1991. 97, 102, 103, 104, 105

- [155] E. Simoncelli, W. Freeman, E. Adelson, and D. Heeger, “Shiftable multi-scale transforms,” *IEEE Trans. Info. Theory*, vol. 2, no. 38, pp. 587–607, 1992. 97, 100
- [156] J Portilla and E. Simoncelli, “A parametric texture model based on joint statistics of complex wavelet coefficients,” *International Journal of Computer Vision*, 2000. 97, 100
- [157] M. Rodriguez, J. Ahmed, and M. Shah, “Action mach: A spatio-temporal maximum average correlation height filter for action recognition,” in *CVPR*, 2008. 114, 117
- [158] K. Soomro, A. Roshan, and M. Shah, “Ucf101: A dataset of 101 human action classes from videos in the wild,” Tech. Rep. CRCV-TR-12-01, UCF, November 2012. 114
- [159] D. Rudoy, D.B Goldman, E. Shechtman, and L. Zelnik-Manor, “Learning video saliency from human gaze using candidate selection,” in *CVPR*, 2013. 118
- [160] S. Oh and et al, “A large-scale benchmark dataset for event recognition in surveillance video,” in *CVPR*, 2011. 127
- [161] A. Abrams, J. Tucek, N. Jacobs, and R. Pless, “Lost: Longterm observation of scenes (with tracks),” in *Workshop on Applications of Computer Vision (WACV)*, 2012. 127
- [162] S. Sadanand and J. Corso, “Action bank: A high-level representation of activity in video,” in *CVPR*, 2012. 127, 128, 137
- [163] A. Gupta, A. Kembhavi, and L. Davis, “Observing human-object interactions: Using spatial and functional compatibility for recognition,” *PAMI*, vol. 31, no. 10, pp. 1775 – 1789, 2009. 128
- [164] B. Yao and L. Fei-Fei, “Modeling mutual context of object and human pose in human-object interaction activities,” in *CVPR*, 2010. 128
- [165] N. Ikizler-Cinbis and S. Sclaroff, “Object, scene and actions: Combining multiple features for human action recognition,” in *ECCV*, 2010. 128
- [166] R. Collins M. Turek, A. Hoogs, “Unsupervised learning of functional categories in video scenes,” in *ECCV*, 2010. 128

- [167] E. Swears and A. Hoogs, “Functional scene element recognition for video scene analysis,” in *Workshop on Motion and Video Computing*, December 2009. 128
- [168] S. Oh, A. Hoogs, M. Turek, and R. Collins, “Content-based retrieval of functional objects in video using scene context,” in *ECCV*, 2010. 128
- [169] A. Bobick and J. Davis, “The recognition of human movement using temporal templates,” *PAMI*, vol. 23, no. 3, 2001. 128
- [170] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as space-time shapes,” in *ICCV*, 2005. 128
- [171] A. Efros, A. Berg, G. Mori, and J. Malik, “Recognizing action at a distance,” in *ICCV*, 2003. 128
- [172] A. Yilmaz and M. Shah, “Recognizing human actions in videos acquired by uncalibrated moving cameras,” in *ICCV*, 2005. 128
- [173] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, “Behavior recognition via sparse spatiotemporal features,” in *VS-PETS*, 2005. 128
- [174] Y. Ke, R. Sukthankar, and M. Hebert, “Spatiotemporal shape and flow correlation for action recognition,” in *CVPR*, 2007. 128
- [175] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *CVPR*, 2012. 128
- [176] L. Zelnik-Manor and M. Irani, “Event-based analysis of video,” in *CVPR*, 2001. 128
- [177] O. Boiman, E. Shechtman, and M. Irani, “In defense of nearest-neighbor based image classification,” in *CVPR*, 2008. 137
- [178] S. McCann and D. Lowe, “Local naive bayes nearest neighbor for image classification,” in *CVPR*, 2012. 137, 150
- [179] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, June 2000. 141, 144, 145, 146, 147

- [180] P. Rousseeuw, “Least median of squares regression,” *Journal of the American Statistics Association*, vol. 79, no. 388, pp. 871–880, December 1984. 142, 147, 149
- [181] R. Hartley, “In defense of the eight-point algorithm,” *PAMI*, vol. 19, no. 6, pp. 580–593, June 1997. 142, 147, 149
- [182] Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong, “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry,” *Artificial Intelligence*, vol. 78, no. 1-2, pp. 87–119, October 1995. 142, 147, 149
- [183] Y. Ma, S. Soatto, J. Kosecka, and S. Sastry, *An Invitation to 3-D Vision*, Springer, November 2003. 144
- [184] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *ICCV*, 1999. 148