Shape Matching and Object Recognition

by

Alexander Christiansen Berg

B.A. Mathematics (Johns Hopkins University) 1994
M.A. Mathematics (Johns Hopkins University) 1995

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Jitendra Malik, Chair
Professor David A. Forsyth
Professor Peter J. Bickel

Fall 2005

The dissertation of Alexander Christiansen Berg is approved:

Professor Jitendra Malik, Chair — Date

Professor David A. Forsyth — Date

Professor Peter J. Bickel — Date

University of California, Berkeley

Fall 2005

Shape Matching and Object Recognition

by

Alexander Christiansen Berg

Abstract

Shape Matching and Object Recognition

by

Alexander Christiansen Berg

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor Jitendra Malik, Chair

We address comparing related, but not identical shapes in images following a deformable template strategy. At the heart of this is the notion of an alignment between the shapes to be matched. The transformation necessary for alignment and the remaining differences after alignment are then used to make a comparison.

A model determines what kind of deformations or alignments are acceptable, and what variation in appearance should remain after alignment. This ties strongly with the idea that the difference in shape is the residual difference, after some family of transformations has been applied for alignment.

Finding an alignment of a model to a novel object involves search through the space of possible alignments. In many settings this search is quite difficult. This work shows that the search can be approximated by an easier discrete matching problem between key points on a model and a novel object. This is a departure from traditional approaches to deformable template matching that concentrate on analyzing differential models. This thesis presents theories and experiments on searching for, identifying, and using alignments found via discrete matchings.

In particular we present a mathematical and ecological motivation for a medium scale descriptor of shape, geometric blur. Geometric blur is an average over transfor-

mations of a sparse signal or feature channel, and can be computed using a spatially varying convolution. The resulting shape descriptors are useful for evaluating local shape similarity. Experiments demonstrate their efficacy for image classification and shape correspondence.

Finding alignments between shapes is formulated as an optimization problem over discrete matchings between feature points in images. Similarity between putative correspondences is measured using geometric blur, and the deformation in the configuration of points is measured by summing over deformations in pairwise relationships. The matching problem is formulated as an integer quadratic programming problem and approximated with a simple technique. Experimental results indicate that this generic model of local shape and deformation is applicable across a wide variety of object categories, providing good (currently the best known) performance for object recognition and localization on a difficult object recognition benchmark.

Furthermore this generic object alignment strategy can be used to model variation in images of an object category, identifying the repeated object structures and providing automatic localization of the objects.

Professor Jitendra Malik, Chair Date

Acknowledgments

Berkeley would only be a beautiful place to live, having pleasant weather and gorgeous views, if it were not for the people that make it truly wonderful. I will list some of those people for some of the contributions they have made to my work and life during graduate school. It would be impossible to thank everyone, so here is the abbreviated version:

Thanks to Tamara for everything always.

Thanks to Jitendra for being an advisor, a researcher, a collaborator and a co-author — for helping me, "get to know every pixel" and for boundless enthusiasm for and about computer vision.

Thanks to my many and varied co-authors: To Ziv Bar-Yossef for being "hyper-dynamic", to Steve Chien always thoughtful, to Jittat Fakcharoenphol for constant good cheer, to Dror Weitz for constant skepticism, to Jitendra Malik for inspiration, to Alyosha Efros for keeping the diminutive a little too long while never outgrowing it and for helping to set me on this path, to Greg Mori for being subject and researcher in one, to Tamara Berg for too much always, to Jaety Edwards for script hacking, to Michael Maire and Ryan White for being the next generation, to Yee-Whye Teh and Erik Miller for a little complexity, to David Forsyth for spirited discussion and a bit of red ink, and to Xiaofeng Ren for clarity, albeit blurry.

Thanks to the rest of the vision group at Berkeley over the years: Jianbo Shi, Serge Belongie, Yair Weiss, Jana Kosecka, Charless Fowlkes, Hao Zhang, Andrea Frome, Deva Ramanan, Ashley Eden, Slav Petrov, and the many others.

To all those that made Berkeley wonderful, both listed and not, I am grateful and thank you.

—Alex Berg, Berkeley

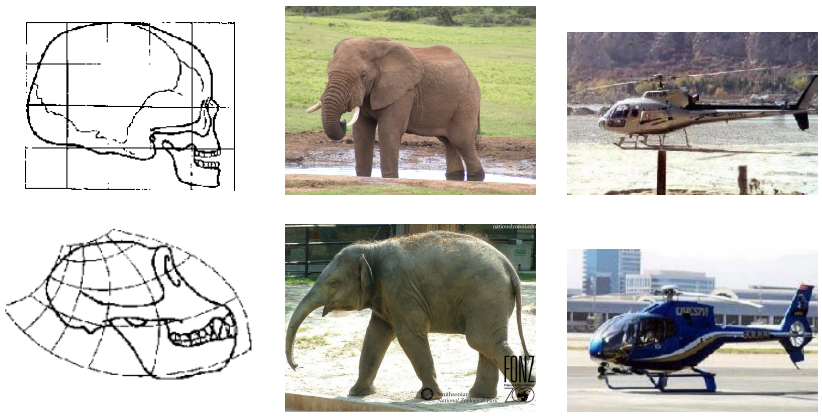
to Patricia and Raymond

Contents

Chapter 1

Introduction

1.1 Motivation



We address comparing related, but not identical shapes in images. As concrete examples, consider the images above. How similar are the two skulls? Are the two elephants shown from the same species? How similar are the helicopters in the two scenes? At the heart of all these questions is the notion of an alignment between the shapes to be matched. The transformation necessary for alignment and the remaining differences after alignment are then used to make a comparison.

The question, "Are these similar?" has occupied philosophers for millennia. This thesis focuses on measuring the similarity of objects in images. Human perception of similarity has been studied by psychologists since the beginning of modern psychology around the 1870s. Particularly relevant from that era is the slogan of the Gestalt movement started by Max Wertheimer, "The whole is different than the sum of the parts." The parallel here is the goal of finding an alignment between objects in order to compare them. The alignment itself is found by considering the aspects of the whole object together.

Philosophers, psychologists, and naturalists[1] have all considered the problem of comparing shapes in images, but it was not until the advent of practical computers around the late 1960s and early 1970s that these theories could begin to be effectively tested on real images. At least three different groups working in different communities initiated related approaches to the problem: in computer vision, Fischler and Elschlager [Fischler and Elschlager, 1973], in statistical image analysis, Grenander [Grenander *et al.*, 1991] (and earlier), and in neural networks, von der Malsburg [Lades *et al.*, 1993] (and earlier). We will use a deformable template framework similar to that from statistical pattern theory.

Our goal is to localize and categorize objects in images. This is accomplished by building models for object appearance and evaluating how well these models fit parts of images. In particular the models we construct will be deformable templates. Deformable templates are models parameterized by a deformation, referred to earlier as an alignment. The model determines what kind of deformations or alignments are acceptable, and what variation in appearance should remain after alignment. This ties strongly with the idea that shape is the residual difference, after some family of transformations is applied for alignment.

[1] D'Arcy Thompson (1860-1948), a naturalist, analyzed variation in biological forms before the formal development of mathematics in this area. Two of his drawings form the left column of the first figure.

Aligning a model to a novel object involves search through the space of possible alignments. In many settings this search is quite difficult. This work shows that the search can be approximated by an easier discrete matching problem between key points on a model and a novel object. This is a departure from traditional approaches to deformable template matching that concentrate on analyzing differential models. This thesis presents theories and experiments on searching for, identifying, and using alignments found via discrete matchings.

We break the problem of matching shapes into computing measures of local shape similarity and finding a low distortion correspondence between keypoints that have similar local shapes.

In particular we present a mathematical and ecological motivation for a medium scale descriptor of shape, geometric blur. Geometric blur is an average over transformations of a sparse signal or feature channel, and can be computed using a spatially varying convolution. The resulting shape descriptors are useful for evaluating local shape similarity. Experiments demonstrate their efficacy for image classification and shape correspondence.

Finding alignments between shapes is formulated as an optimization problem over discrete matchings between feature points in images. Similarity between putative correspondences is measured using geometric blur, and the deformation in the configuration of points is measured by summing over deformations in pairwise relationships. The matching problem is formulated as an integer quadratic programming problem and approximated with a simple technique. Experimental results indicate that this generic model of local shape and deformation is applicable across a wide variety of object categories, providing good (currently the best known) performance for object recognition and localization on a difficult object recognition benchmark.

Furthermore this generic object alignment strategy can be used to model variation in images of an object category, identifying the repeated object structures and

providing automatic localization of the objects.

1.2 Outline

Chapter 2 develops a simple method for comparing signals taking into account an idea of what geometric distortions are expected. The result is a soft similarity measure between patches called geometric blur. The motivation is to make an estimate of whether an alignment of objects that brings together two patches has a chance of being a good overall alignment by computing some measure of similarity between the patches. The approach presented attacks the problem in two ways. First by looking at paches that are small with respect to the aligning transform a local approximation of the aligning transform as affine is appropriate. Second instead of considering all possible affine transformations explicitly, geometric blur is developed as a way to estimate the average quality of matches over a range of transformations. The mathematical development allows a great deal of generality – later geometric blur is used in settings other than simply finding alignments between objects. Experiments on synthetic data are presented to highlight the features of geometric blur as a similarity measure.

Next in Chapter 3 we begin the empirical study of correspondences in real images. Given a known correspondence between images of the same object, we study the covariance of corresponding patches. In some settings this covariance is well modeled by a simple geometric blur. This result motivates geometric blur as a basis on which to build descriptors for wide-baseline stereo matching of images of the same object.

Chapter 4 extends the experimental motivation for geometric blur to the problem of identifying similar bits of shape in images of objects from a variety of categories. Here the goal is not simply recognizing different views of the same object but recognizing different instances of a category of object. We begin to address the question

of how to discretize an alignment problem. A simple descriptor of local shape is developed by sub-sampling geometric blur computed from oriented edge channels. Chapter 4 also introduces the dataset used for multiple experiments in the remainder of the thesis.

Using geometric blur to estimate the quality of a matching locally, Chapter 5 presents a framework for evaluating and finding matchings based on pairwise relationships between the individual keypoint matches. Modeling pairwise interactions results in an integer quadratic programming problem that is potentially quite difficult. It turns out that the instances observed in practice are relatively simple. This chapter presents matching results between objects in different images. In addition experiments show that the quality of these matchings can be used as a similarity metric resulting in good performance for nearest neighbor recognition of object categories. *The key result is that the general shape of objects is matched in real images of many categories of object at a level of fidelity that is useful for recognition.*

The recognition experiments in Chapter 5 use a generic model for the variation of exemplars from many different categories of object. This same generic alignment model can be used to bootstrap learning category specific models for variation and alignment. Chapter 6 presents results on building models for variation of object categories, and automatically segmenting objects from their background using this approach.

Before beginning the technical discussion we go through some recent history of the study of visual recognition.

1.3 Some Connections to Psychology

"Esse est percipi"

–George Berkeley

George Berkeley's quote translates as, "to be is to be perceived" and is but one of the manifold theories from Philosophy and Psychology concerning visual perception from function to physical instantiation. Most relevant to this work are results on perception of shape, and the recognition of objects — many of which have influenced the approach taken here.

Computer vision has dual goals: using visual data to construct accurate models of the world, and understanding/mimicking human abilities to do so. In fact psychologists[2] have sometimes confused these goals as being the same. More recent work has demonstrated that context and prior experience have a great influence on how humans perceive visual stimuli. This thesis reflects the dual nature of computer vision. The mathematical development of geometric blur is motivated by an explicit model for the variation in signals due to geometric distortion, and attempts to measure this variation. However, when applying the matching framework developed in this work to object categorization, success is measured by the ability to agree with human categorizations for objects, and many design choices are made with an eye toward accomplishing this goal.

Using the location and orientation of edge-like structures in images is consistent with studies showing that the location of an edge, or phase, contains a great deal of the information in an image, at least as far as humans are concerned [Piotrowski and Campbell, 1982]. In fact using edge-like features and the spatially varying geometric blur fits well with what is known about the log polar structure of retinotopic maps in the first part of the visual cortex (V1) [Tootell *et al.*, 1982]. The fact that deriving geometric blur with the objective of making signal comparison robust to small geometric distortion results in a structure very similar to the log polar retinotopic map

[2]James J Gibson (1904-1979), a psychologist, proposed that perception was "invariant" dependent only on external physical stimuli. Despite ignoring what was later understood to be the strong influence of context on perception, he introduced ideas of ecological optics, the statistics and properties of the physical world that humans can observe, that are still relevant and studied today.

points in a direction for further study.

Studies of "shape constancy" indicate that small variations in viewing angle are usually discounted by human observers [Thouless, 1931] [Slater and Morison, 1985]. This relates to the objective for geometric blur presented in Chapter 2, robustness to small affine transformations.

At the level of object recognition, the ideas of recognition by prototypes [Rosch, 1973] (and later) and of prototypical views [Palmer *et al.*, 1981] are consistent with some of the simple recognition strategies presented in this work based on exemplar images. A novel image is classified by using a similarity measure to compare it to previously observed images of objects, combining the idea of recognition by prototypical exemplars and the idea of canonical two dimensional projections of objects. Also studies of the ambiguity of parts without context support an approach where multiple parts are combined in recognition [Palmer, 1975]. These all tie into the Gestalt notion that recognition depends jointly on the parts together. In computer science terms, this implies that the discrete optimization we will encounter might be difficult.

Despite the many connections between work in psychology and the algorithms presented here, there is an enormous amount of theory and speculation about human perception yet untapped by computer vision researchers. Even accepting that only some small percentage of that work is important and relevant for constructing machine vision systems, a great deal remains unexplored.

As a counter point: despite the fact that humans have worked to understand vision for as long as philosophy has been around, it was the advent of computers in the mid 1900s that allowed computational models for vision to be tested, and resulted in the conclusion by psychologists that,

> *"... vision is extremely difficult."*
>
> –Stephen E. Palmer

Chapter 2

Geometric Blur

2.1 Introduction

The introduction motivates geometric blur with the goal of estimating the quality of an alignment that brings one key point of a model together with one key point of a novel instance. This chapter expands that motivation, describes a mathematical theory for geometric blur, and presents results on simple synthetic patterns to illustrate its properties. Chapter 3 presents results on learning optimal transformations for matching images of objects, viewed from different camera angles, that shows similar structure to the developments in this chapter. Chapter 4 develops a descriptor based on geometric blur and presents experiments on classifying images containing instances of object categories.

The two helicopters shown in Figure 2.1 are easily recognizable as helicopters and a young child could indicate positions for the nose and tail for each. The crops below indicate the difficulty faced by a computer. Analogous structures in the images are only very roughly similar. In order to find a correspondence and then an alignment between the two objects it is necessary to find some way to get at this rough similarity.

Figure 2.1: In the **top row** are two images showing similar objects, helicopters. The **bottom row** shows that the local structure of the objects is only very roughly similar. Geometric blur is motivated by the goal of identifying rough similarity between bits of shapes.

One similarity between the two cropped parts of the helicopter is the smooth protruding shape. Although not close to being the "same" shape they are nevertheless similar. This is not a coincidence as helicopters fly through the air, often frontwards, and must be somewhat aerodynamic. Another similarity is the specularity on the top portion of the object. In both cases this seems to result from the sun reflecting off of the windows. Again this is not a coincidence, helicopters are aerodynamic, smooth, often shiny, often outside, often right side up, and it is often sunny outside. Making either the notion of smooth protruding shape or specular reflection resulting from the sun on the windshield precise is difficult, and perhaps impossible without a great deal more high level knowledge than is currently available to machine vision systems. Instead we note that these similarities in shape, lighting, and reflectance result in somewhat repeatable edge-like features in the image. There is a roughly vertical bit of edge near the tip of the nose and slanting up and to the right above

it. In fact the design of cockpit windows and the resulting specularities, give rise to a number of edges sweeping up and to the right from the nose. There are also more level edges sweeping slightly down and to the right from the nose. It is this rough sketch of edges that we will exploit. Although the edges are artifacts produced by a complex combination of geometry, pose, lighting, and surface reflectance, none of which are actually the same for the two examples shown, they nevertheless show a rough consistency.

We will develop a mathematical framework for finding similar locations on objects by looking for similar configurations of discrete features nearby. These discrete features should result from repeatable phenomena, and could be based on color, texture, or a number of other cues. Later, in Chapter 5 these local similarities will be combined to find an alignment between shapes.

Section 2.2 gives a mathematical motivation and derivation for geometric blur. This is followed by Section 2.3 showing results of experiments on synthetic data. Chapter 3 presents empirical results on real images indicating that for images of the same object the geometric blur model may be appropriate. Chapter 4 presents a feature descriptor based on geometric blur and describes experiments using geometric blur for localizing parts of objects in images and a bag of features model for recognizing images of object categories. This is extended to alignment in Chapter 5

2.2 Derivation of Geometric Blur

We motivate geometric blur as an estimate for a robust similarity measure which requires computing an average over distorted versions of a signal. This average can be computed by convolving the signal with a spatially varying kernel. Furthermore we show that in a reasonable setting this can be done very efficiently. This chapter concludes with test results using geometric blur in a recognition task on synthetic

signals.

2.2.1 An Example as a Way Point

Before beginning this development a simple example of comparing distorted signals is presented to make concrete some of the mathematics to follow.

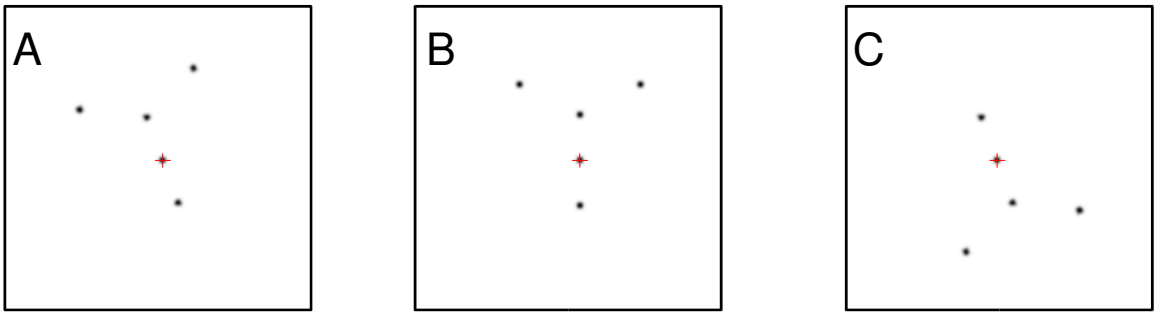


Figure 2.2: Three similar signals composed of impulses. The goal is to recognize that a small transformation brings **A** and **B** into alignment, but not so for **B** and **C**.

In Figure 2.2, which signal, A or C, is most similar to the signal B? The question is ambiguous, and we need to take into consideration some type of accepted variation, say small affine transformations. Note that here we mean spatial affine transformations, not transforms in intensity. Making robust comparison of signals with variation in intensity is rather better studied than the variation in signals due to distortions in geometry. Even with this added information, the correlation between either the left (A & B) or right (B & C) pair of signals is low and quite similar, providing no information about which are more similar. This can be seen in the first row of Figure 2.3 where the insets show the point-wise products of the signals on either side. Note that smoothing the signals with a uniform Gaussian does not quite solve the problem, as can be seen in the second row of the Figure 2.3. After blurring the signals with a uniform Gaussian the correlation between either pair of signals is similar, missing the clear differences. The basic idea of geometric blur is to

blur or average the signals over the range of acceptable transformations (small affine transformations in this case), as shown in the third row of Figure 2.3. This will turn out to be mathematically equivalent to convolving the signal with a spatially varying kernel. Roughly speaking, parts of the signal farther from the center are blurred more because they have the opportunity to move more. After this type of blur, correlation can correctly identify the more similar pair.

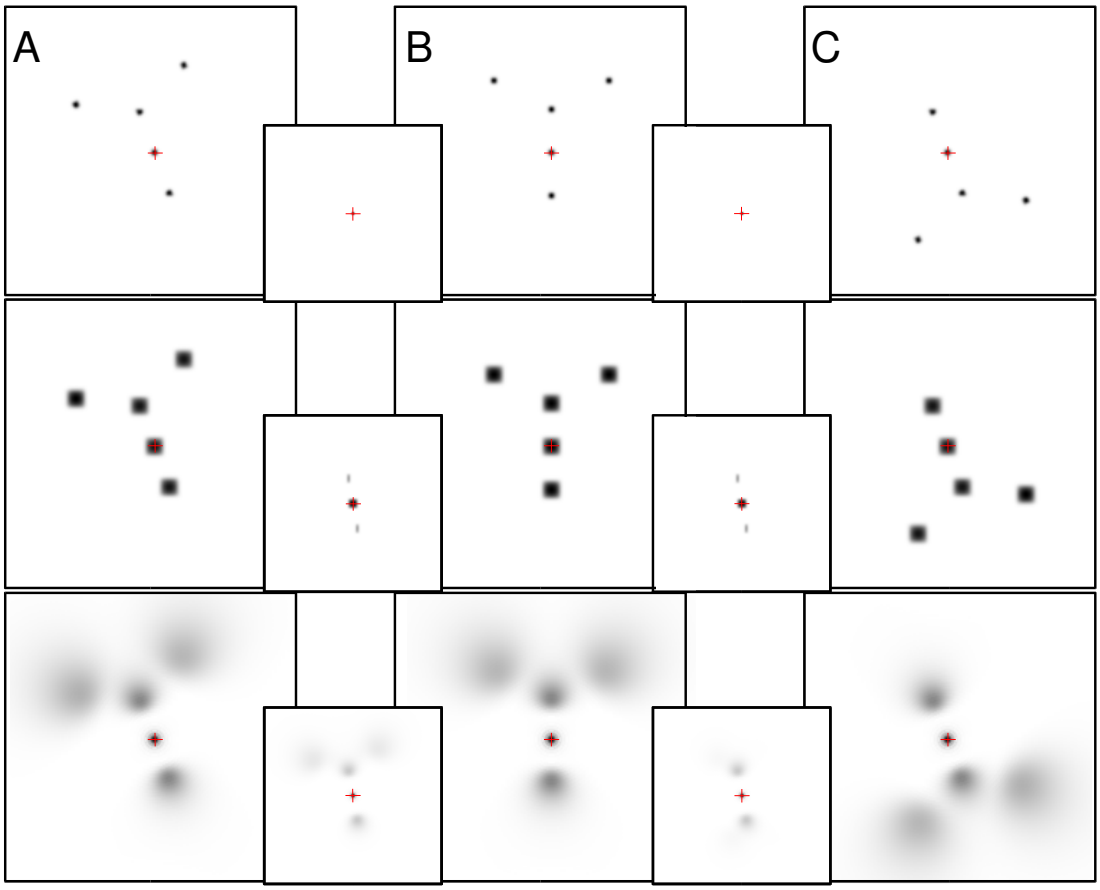


Figure 2.3: The **top row** shows three signals, *A*, *B*, and *C*. The **top row insets** show the point-wise product of the signals on either side, each results in correlation 0.2. the **second row** shows the result of applying a Gaussian blur to the signals. Note that more context is not included, but the correlations are still equal (0.22). The **third row** shows the result of applying geometric blur, a spatially varying blur replicating the effect of averaging over small affine transforms of the signal. Now the insets indicate a difference between the correlations: 0.63 for the correct match versus 0.4 for the incorrect match.

2.2.2 Mathematical Motivation

There are two basic approaches for making robust or invariant comparisons between images of objects: finding transformational alignments or computing invariant features. In order to make these concrete, consider two signals, A and B in the space of all signals I and let T be a transform on signals in the space of transforms $\mathcal{T}$. Suppose $S(A, B)$ is a similarity function on signals then an invariant or robust comparison can be constructed in one of two ways:

1. For a transformation T, let $S_T(A, B) = S(A \circ T, B)$. Then the average or maximum of S_T over transforms is a robust or invariant similarity between A and B. A prior on T can be introduced and used to form a weighted average of the S_T.

2. Define features $f : I \to O$, that are invariant to transformations, so $f(A) = f(A \circ T)$ for any $T \in \mathcal{T}$. Then $\hat{S}(A, B) = s(f(A), f(B))$ is an invariant similarity, where s is a similarity on the output of the invariant features.

The drawback with the first approach is usually the computation time to find the best transformation T, or to integrate over a range of transformations. The drawback of the second is usually a loss of discriminative power. It is difficult to construct invariant functions that maintain relevant information about the signal. Our approach is motivated by approximating the first approach. Instead of finding transforms that align two signals maximally, we look at how well the signals are aligned on average by a range of transforms. This average is weighted more heavily toward transforms that are considered more likely; for instance the identity.

In order to compute the average similarity over a range of transformations we need to compute the following integral:

$$\int_T S(A \circ T, B) d\mu \tag{2.1}$$

where μ is a measure over the space of transformations. The goal now is to *approximate* this calculation. One realization[1] using normalized correlation as the similarity function looks like:

$$\int_T \frac{1}{|A||B|} \int_x (A \circ T)(x) B(x) dx d\mu \tag{2.2}$$

The first approximation is to drop the normalization factors $|A|$ and $|B|$ for now, they will return in a slightly different form in Section 2.2.4. Then reverse the order of integration to obtain:

$$\int_x \int_T (A \circ T)(x) B(x) d\mu dx = \int_x B(x) \left(\int_T (A \circ T)(x) d\mu \right) dx \tag{2.3}$$

The expression in parenthesis on the right hand side of Equation 2.3 is an average over transformed versions of the signal A. Geometric blur refers to computing this average of geometric transformations of a signal.

2.2.3 Geometric Blur Definition

The geometric blur $GB_I(x)$ of a signal $I(x)$ over coordinate x is defined to be the integral over a range of distorted versions of the signal:

$$GB_I(x) = \int_T I(T(x)) d\mu \tag{2.4}$$

Where T are spatial transforms and μ is a measure on the space of transforms. If

[1] Appendix 2.A shows that other choices for similarity end up requiring similar calculations.

the transforms are parameterized by $\mathbb{R}^k$, and μ is absolutely continuous with respect to the Lebesgue measure on $\mathbb{R}^k$, then by the Radon-Nikodym theorem there is a density ρ such that:

$$GB_I(x) = \int_p I(T_p(x))\rho\left(T_p\right) dp \tag{2.5}$$

Where T_p is a transform specified by parameters p in $\mathbb{R}^k$ and the integral is computed with respect to the Lebesgue measure on $\mathbb{R}^k$. The density ρ determines the measure on transforms. In order to reduce notational clutter we will usually drop the subscript p and assume that the transform T is parameterized by p.

Equation 2.4 is an integration over warped versions of the signal. Rewriting this to integrate over the range (spatial coordinates of I) of the transforms, using multiple iterations of the Fubini-Tonelli theorem gives:

$$\begin{aligned}
GB_I(x) &= \int_p I(T(x))\rho\left(T\right) dp && (2.6)\\
&= \int_{p,z} I(T(x))\rho\left(T\right)\chi(T(x) == z)\left(dp \times dz\right) && (2.7)\\
&= \int_z \int_T I(T(x))\rho\left(T\right)\chi(T(x) == z) dp dz && (2.8)\\
&= \int_z I(z) \int_{T:T(x)==z} \rho\left(T\right) d\tilde{p} dz && (2.9)
\end{aligned}$$

Where $p \in \mathbb{R}^k$ parameterizes the transforms T with the the Lebesgue measure on $\mathbb{R}^k$ and $z \in \mathbb{R}^l$ parameterizes the (bounded) range of the transforms with the Lebesgue measure on $\mathbb{R}^l$ divided by the area of the range of the transforms (in order to ensure that the transition from Equation 2.6 to 2.7 holds). Here $d\tilde{p}$ indicates integration with respect to the measure on the "slice", in this case, $\{T : T(x) == z\}$.

A change of variables puts this in the form of a convolution with a spatially varying

kernel $K_x(y) = \int_{T:(x-T(x))==y} \rho\,(T) d\tilde{p}$, where the slice is now, $\{T : (x - T(x)) == y\}$,

$$\begin{aligned} GB_I(x) &= \int_z I(z) \int_{T:T(x)==z} \rho\,(T)\, d\tilde{p} dz & (2.10) \\ &= \int_y I(x-y) \int_{T:(x-T(x))==y} \rho\,(T)\, d\tilde{p} dy & (2.11) \\ &= \int_y I(x-y) K_x(y) dy & (2.12) \end{aligned}$$

2.2.4 Comparing Signals Using Geometric Blur

In practice the motivating Equation 2.1, repeated here,

$$\int_T S(A \circ T, B) d\mu \tag{2.13}$$

is not quite appropriate for comparing signals. We often encounter two observations of a shape and would like to evaluate their similarity or likelihood of being the same. This is better expressed by the following:

$$\int_{T_a} \int_{T_b} S(A \circ T_a, B \circ T_b) d\mu d\mu \tag{2.14}$$

Using normalized correlation and applying the results of the previous section we obtain

$$\int_{T_a} \int_{T_b} S(A \circ T_a, B \circ T_b) d\mu d\mu = \int_x \int_{T_a} A \circ T_a \int_y B(x-y) K_x(y) dy d\mu dx \tag{2.15}$$

here we make a broad approximation[2] and separate the integrals:

$$\int_x \left(\int_{T_a} A \circ T_a d\mu \right) \left(\int_y B(x-y) K_x(y) dy \right) dx \tag{2.16}$$

Apply the result of the previous section again to obtain:

$$\int_x \left(\int_y A(x-y) K_x(y) dy \right) \left(\int_y B(x-y) K_x(y) dy \right) dx \tag{2.17}$$

which is simply the correlation between the geometric blur of each signal. Here we note that the geometric blur signals are normalized for the correlation.

The key point is that geometric blur is computed for each signal, and then compared afterward using a simple normalized correlation.

2.2.5 Fast Computation

If the spatially varying kernel $K_x(y)$ is simple enough the computation in Equation 2.12 becomes quite easy. The two conditions required are that $K_x(y)$ is depends only on $|x|$ and $|y|$. Our most often used kernel is in fact a shaped like a Gaussian with varying standard deviation, $K_x(y) = f(\alpha|x| + \beta) G_{\alpha|x|+\beta}(y)$, where f is a scaling factor. It is clear that this kernel satisfies both conditions. Given these conditions we can rewrite Equation 2.12 as follows:

$$GB_I(x) = \int_y I(x-y) K_x(y) dy \tag{2.18}$$

$$= \int_y I(x-y) K_{|x|}(y) dy \tag{2.19}$$

[2]The geometric blur computation itself is exact, however the motivation presented here for geometric blur is an approximation. The two approximations made when motivating geometric blur by correlation are the one mentioned above and the rearrangement of the normalization.

Here $K_{|x|}(y)$ is simply $K_{x_0}(y)$ for any x_0 with $|x_0| = |x|$. Picking a discrete set of values $r_1 \dots r_k \in \mathbb{R}$ we can approximate 2.18 as:

$$GB_I(x) \approx \sum_{i \in \{1 \dots k\}} \text{ind}(x, i) \int_y I(x - y) K_{r_i}(y) dy \tag{2.20}$$

where

$$\text{ind}(x, a) = \begin{cases} 1 & \text{if } a = \text{argmin}_{i \in \{1 \dots k\}} \, ||x| - r_i| \text{ and;} \\ 0 & \text{otherwise.} \end{cases}$$

is simply an indicator function that "chooses" the correct blur level. These turn out to be concentric annular regions. The computational cost of computing geometric blur is then just the cost of computing k convolutions and using the indicator function to trim out the appropriate sections. Chapter 4 shows how a descriptor is created in practice. Furthermore if the $K_x(y)$ are separable, as they are in most of the examples, then the convolutions themselves can be computed in one dimension and are very efficient.

2.3 Behavior on Synthetic Signals

We present some simple examples to illustrate the properties of geometric blur, then consider a recognition experiment on synthetic data.

2.3.0.1 Example 1

Returning to the example signals in Figures 2.2 and 2.3. We now consider comparing signal **B** to rotations of itself, and rotations of its vertical mirror image. The green dashed lines in Figure 2.4 show the correlation between **B** and rotated versions of itself, and the red dashed line shows correlations between **B** and rotated versions of its vertical mirror image. As a reference, the signals shown in Figure 2.2 would

correspond to the signals used for a rotation of 0.35 radians as shown on the far right of Figure 2.4.

In this and all other examples in this section the kernel function is $K_x(y) = f(\alpha|x| + \beta)G_{\alpha|x|+\beta}(y)$, where G is a Gaussian with the specified standard deviation, and f is a normalization factor so that the K_x is L^2 normalized.

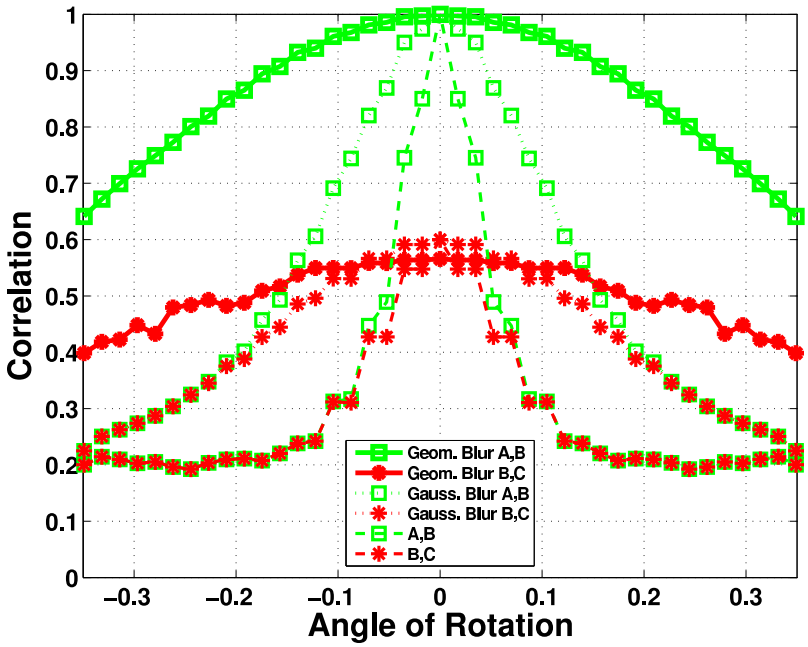


Figure 2.4: The far right end of the graph, rotation by 0.34 radians corresponds to the signals shown in Figure 2.3.

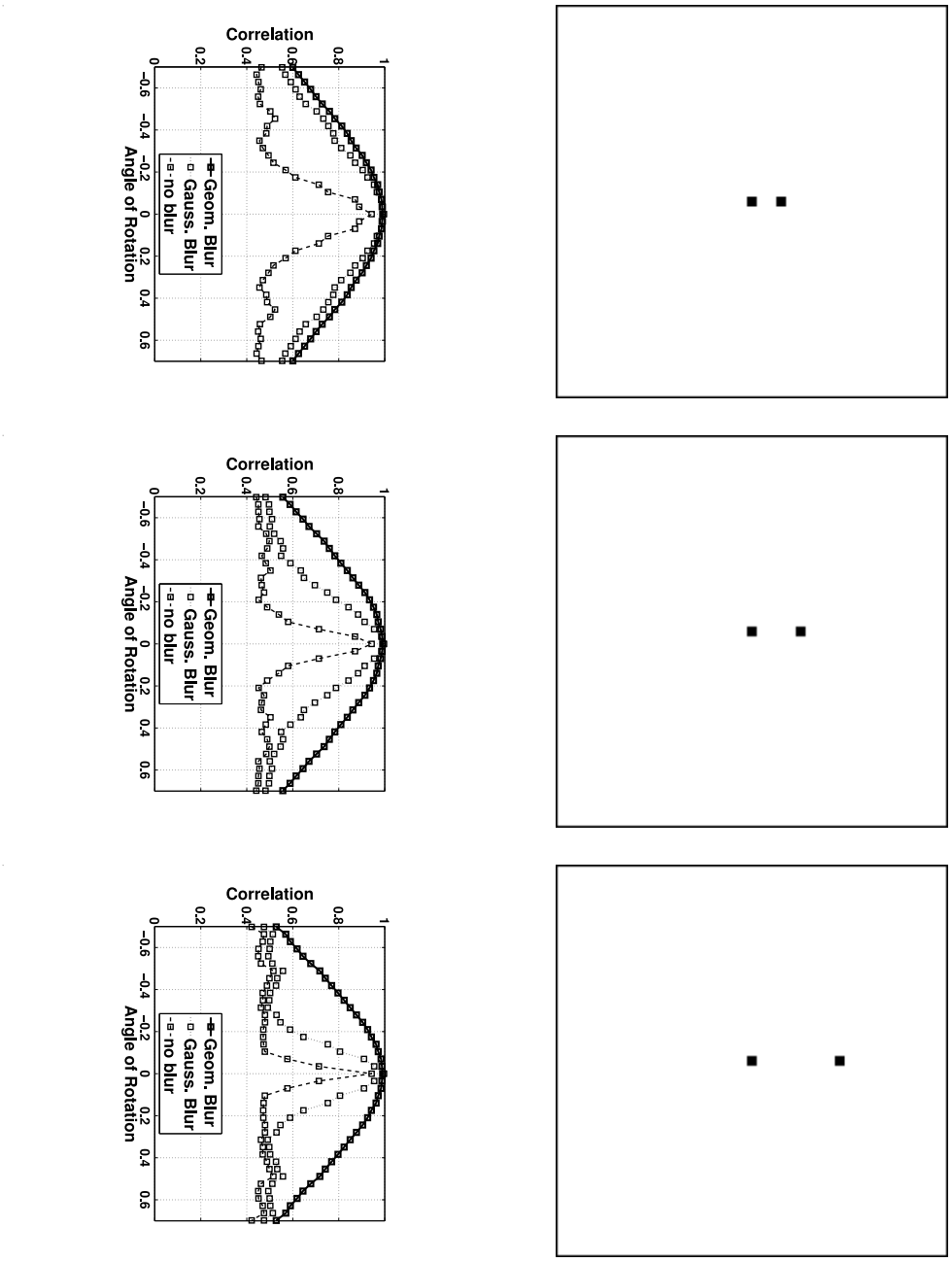


Figure 2.5: Each plot on the left shows the correlation of the signal, to its right, with a rotated version of itself as the amount of rotation changes. Without any blur the correlation drops of rapidly. This is mitigated by a Gaussian blur, but the effect depend on the scale of the signal. Geometric blur increases farther from the origin and results in correlation close to independent of the signal, when the signal is sparse.

2.3.0.2 Example 2

One of the key features of geometric blur is that for sufficiently sparse signals the correlation between a signal and a transformed version of itself depends only on the transformation, not on the signal. This is illustrated with three different signals composed of two spikes as shown in the top row of Figure 2.5. In each case there is one spike at the center and a second spike at various distances away from the center. Comparing the signals to rotated versions of themselves results in the correlation plots shown below in Figure 2.5. These plots show the correlation using no blur, a uniform Gaussian blur, and a geometric blur. Note that for a particular choice of variance for the Gaussian points close to the center are blurred too much and point farther away blurred not enough. Geometric blur results in a fall-off in correlation that is close to independent of the actual position of the spikes. As noted earlier this property only holds for sufficiently sparse signals.

2.3.0.3 Example 3

In order to see that geometric blur helps for discrimination in the presence of distortion we performed a discrimination task using 200 test patterns. Rotated versions of the test patterns were compared to the original test patterns. Both the original test patterns and the rotated versions were blurred by either geometric blur or a uniform Gaussian blur. For geometric blur, a spatially varying kernel $K_x(y) = G_{\alpha|x|}(y)$, where $G_\sigma(y)$ is a Gaussian with standard deviation σ, was applied. For uniform Gaussian blur the kernel $G_\sigma(y)$ was applied. Then each blurred rotated pattern was compared to all the blurred original patterns using normalized correlation and matched to the closest one. The test patterns used in this example were random with each pixel in a disc of radius 25 pixels independently being turned on with probability 5%. Figures 2.6 shows the the misclassification rate as the amount of blur, α or σ is

varied. Geometric blur has much better discriminative power, and manages to be general enough to handle large rotation somewhat more effectively than uniform blur.

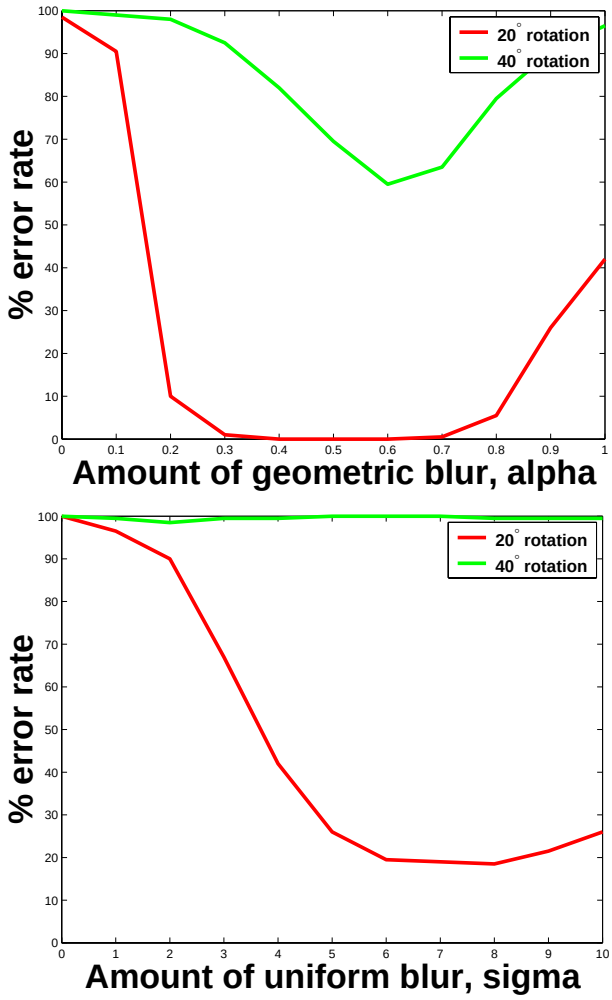


Figure 2.6: Identifying 200 random test images after rotation, using various amounts (α) of geometric blur on the **top** plot, and varying amounts of a uniform Gaussian blur on the **bottom** plot.

2.4 Conclusion

This chapter introduced geometric blur, rewriting the average over distorted versions of a signal as a convolution with a spatially varying kernel:

$$\int_T I(T(x))d\mu = \int_y I(x-y)K_x(y)dy$$

This is motivated by approximating robust similarity measures for geometrically distorted signals. We originally introduced geometric blur in [Berg and Malik, 2001], and these ideas have been applied in [Efros *et al.*, 2003] (in a temporal instead of spatial domain) and [Berg *et al.*, 2005] and others ([Ren *et al.*, 2005]). In addition the mathematical development showing that geometric blur is a method to obtain robustness to small affine transforms has been applied to understanding of a shape contexts, and has inspired some of the generalization of shape contexts [Mori *et al.*, 2005].

Appendix 2.A Alternate Motivations

As an alternative to motivating geometric blur by approximating Equation 2.2 consider a generative model:

$$J = I \circ T + \eta$$

for observations, J, of a base signal, I. There is some probability $p(T)$ for each transform, and a noise model $p(\eta)$. The probability of observing J is then

$$\int_T p(\eta == I \circ T - J)p(T)dT$$

If we assume a simple Gaussian noise model then this becomes:

$$\int_T e^{-\frac{1}{a}\int_x \frac{1}{b}(I\circ T(x)-J(x))^2 dx} p(T)dT$$

writing out the series for e yields:

$$\int_T \sum_{n=0\ldots\infty} \frac{1}{n!}\left(-\frac{1}{a}\int_x \frac{1}{b}\left(I\circ T(x)-J(x)\right)^2 dx\right)^n p(T)dT$$

after swapping the sum and the integral,

$$\sum_{n=0\ldots\infty} \int_T \frac{1}{n!}\left(-\frac{1}{a}\int_x \frac{1}{b}\left(I\circ T(x)-J(x)\right)^2 dx\right)^n p(T)dT$$

The first term ($n=0$) is constant. We will will write the second ($n=1$) term:

$$-\frac{1}{a}\int_x\int_T \frac{1}{b}\left(I\circ T(x)-J(x)\right)^2 p(T)dTdx$$

that expands to

$$-\frac{1}{a}\int_x \left(\int_T \frac{1}{b}I^2(T(x))p(T)dT - 2J(x)\int_T \frac{1}{b}I(T(x))p(T)dT + J^2(x)\right)dx$$

The two integrals over T are then the geometric blur of I^2 and I.

Chapter 3

Ecologcal Study of Blur

3.1 Introduction

This chapter presents empirical results showing that the covariance of regions around corresponding points in images is well fit by a simple geometric blur. Given a "correct" correspondence between points on images, the covariance of corresponding image regions can be studied. We consider correspondences between images varying with respect to factors such as: viewing direction, viewing orientation, focus, etc. Each set of images considered in this chapter is of the same object while Chapter 4 shows results on images of many different objects from a number of object categories.

In order to discuss corresponding points in images, it is first necessary to identify points in images. Generally a region of interest operator will be applied to images and produce feature centers and support regions, sometimes in addition to a direction associated with the region. In a sense this is the second approach discussed in Section 2.2.2, finding features that are invariant to transforms; the idea is that the region of interest operator commutes with transformations of the image. It is difficult to do this well, and in practice this invariant feature approach is combined with a

robust comparison of the residual difference. It is a subtle point, but worth keeping in mind that this is a combination of the two techniques discussed in Section 2.2.2.

Consider the two views of a tree shown in Figure 3.1. Given a correspondence between the images we can consider the similarity between corresponding regions. In the rest of this chapter we use known transformations between the images together with various region of interest operators to find corresponding pairs of image regions. The covariance of these regions is then studied, as well as the covariance of edge-like features associated with the regions.

Figure 3.1: A pair of images of a tree showing change in viewpoint and viewing direction.

3.2 Images and Correspondences

The images used in this chapter come from work by K. Mikolajczyk and C. Schmid [Mikolajczyk and Schmid., 2003] on region of interest operators and descriptors for wide-baseline matching. There are 6 sets of images of an object or scene. Each set exhibits different types of variation. We break these 6 sets of images into two groups as exemplified in Figure 3.2. One group showing variation in viewing direction, and the other not showing variation in viewing direction, but exhibiting variation in other parameters such as focus and illumination.

Figure 3.2: **Top**: Two images from a set exhibiting significant change in viewpoint, **Bottom**: Two images from a set showing change in focus, with little to no change in viewpoint.

3.2.1 Region of Interest Operator

The two region of interest operators considered are the best performing operators from [Mikolajczyk and Schmid., 2003]. The region of interest operators are described in more detail below:

1. **Harris Affine:** Regions of interest are found at local maximum of an operator on scale and transform space. This operator responds to corners and edges. The region of interest is an ellipse or rectangle. In order to compute a descriptor, the region is transformed into a canonical circle or square. This region of interest operator is appropriate when there is significant variation in viewing direction.

One major drawback is that the local maxima of the operator are somewhat unstable.

2. **Maximally Stable Extremal Regions:** This region of interest operator is based on finding extended regions with small variation in intensity and high contrast boundaries. When they exist these regions are quite stable across views of the same object, and are appropriate for use under a wide range of viewing changes. The main drawback of this region of interest operator is the relatively sparse response on images resulting in relatively few output regions. Additionally in some cases changes in viewing direction or lighting may change the relative intensity (eg shadows or relief on surfaces) resulting in instability or lack of repeatability of the regions.

Figure 3.3 shows a sampling of feature point locations found by the first two methods with some regions of interest indicated. Figures 3.4, 3.5, 3.6, and 3.7 all show pairs of regions chosen by the respective region of interest operators. Note that only corresponding regions that agree closely with the known alignments between images are considered and a subset of these is shown here.

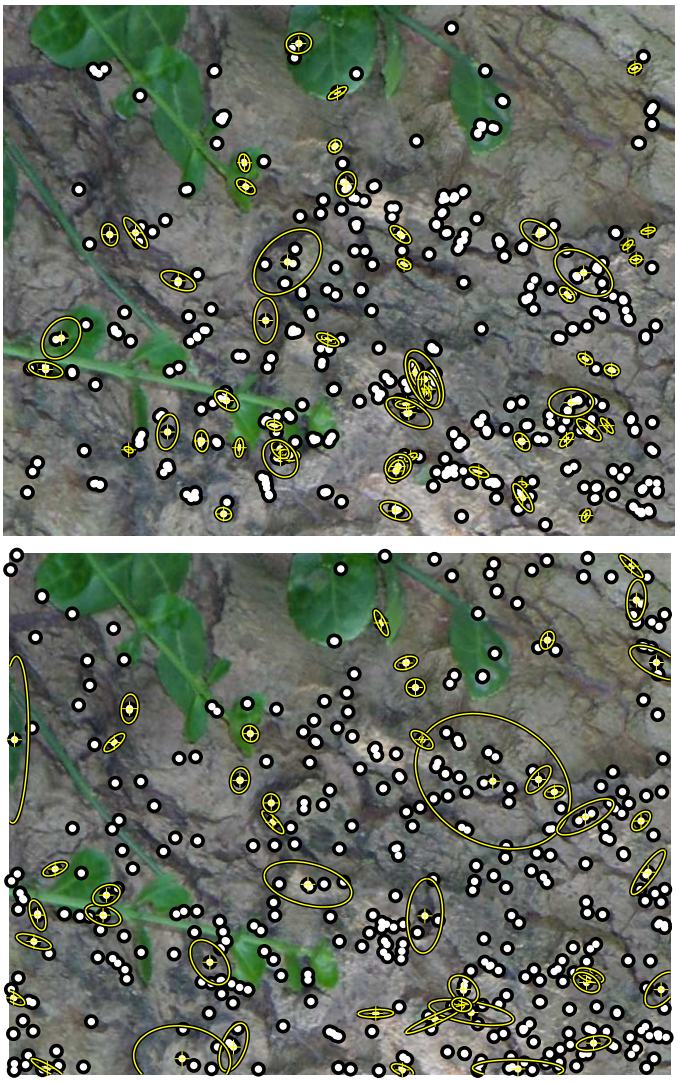


Figure 3.3: **Top Left**: Interest points found by the Harris-Affine operator. The support for a subset of the features is shown with ellipses. **Right**: Interest points found by the Maximally Stable Extremal Region (MSER) detector. The support for a subset of the features is shown with ellipses.

Figure 3.4: Rectified paired patches found by the Harris-Affine detector from image sets showing significant change in viewing direction. Note that the centers of the patches are usually on edges or at corners, and that the orientations and scales of the patches are often slightly incorrect.

Figure 3.5: Rectified paired patches found by the Maximally Stable Region detector from image sets showing significant change in viewing direction. Note that the centers of the patches are usually on constant blobs, and there are often variations in the orientations and shapes of the structure surrounding the blob.

Figure 3.6: Rectified paired patches found by the Harris-Affine detector from image sets with fixed viewpoint but varying focus and illumination. Note that although these patches are more consistent than the ones found in Figure 3.4, there is still variation in orientation and scale.

Figure 3.7: Rectified paired patches found by the Maximally Stable Region detector from image sets with fixed viewpoint but varying focus and illumination. Note that although these patches are more consistent than the ones found in Figure 3.5, there is occasionally variation in orientation and scale.

3.3 Observations of Covariance

Each experiment uses a region of interest operator, either Harris-Affine or MSER, combined with a known correspondence between images to produce a pair of matching patches. The actual values in a patch are either the gray-scale values of the image in the region of interest, or the values from an edge detector[1]. Each patch is reshaped to form a vector of values and the mean value subtracted. All of these vectors for the patches are made into a matrix, and another matrix is formed with the corresponding patches. Each patch occurs in both matrices.

A covariance matrix between gray-scale patches obtained using the Harris-Affine interest point operator on the set of images with varying viewpoint is shown in Figure 3.8. This figure is somewhat difficult to interpret directly because the patches were reshaped into vectors. In order to visualize the result we reshape the covariance as shown in Figure 3.9. Each small block represents the covariance of all the pixels in one patch with respect to a particular pixel in the corresponding patch. The location of the small block specifies the pixel in the corresponding patch. For example the block at the lower right of the image shows the covariance of the all the pixels in a patch with the pixel in the lower right of the corresponding patch.

Here we can see that the middle pixel varies with pixels in a relatively tight radius around the middle of the corresponding patch, and that pixels in the periphery vary with pixels over a wider range in the corresponding patch. Figure 3.10 shows the results of fitting the standard deviation for a Gaussian to the pattern in each subplot. The estimated standard deviation is plotted against the distance from the center of the patch. In this case note the clear linear structure.

While the pattern shown is consistent with a geometric blur model, we are more interested in the covariance of edge responses as these are expected to generalize

[1]The derivative of Gaussian edge detector used in the next section.

better than pixel intensities. While intensity is probably useful for the image sets here (each set is of a single object or scene) in the next chapter we will deal with intraclass variation in objects. Then we will rely on the consistency of edge features instead of intensity.

3.3.1 Discussion

For image sets with varying viewpoint and view direction using the Harris-Affine region of interest operator there is a clear increase in the amount of blur moving away from the center of the patch. Figures 3.10 and 3.13 show that this is true for the gray-scale patches and for edge patches. For the image sets without change in viewpoint the rate of increase of blur appears less as can be seen in Figure 3.16. This indicates that the blur pattern is not simply a result of error in the interest point operator.

Interestingly when using the MSER interest point operator the covariance is quite different, as can be seen in Figure 3.17 and 3.18. Looking back to the patches in Figure 3.5 the prominent feature is a blob at the center of each patch. This results in a consistently small gradient in the central region and explains the structure. If the size of the patches is doubled relative to the size determined by the MSER operator, then the resulting covariance shown in Figure 3.19 and Figure 3.20 shows that the linear increase in blur actually starts outside the central patch. This shown explicitly in Figure 3.21.

3.4 Conclusion

This chapter presents observations about the covariance of corresponding patches in different images of the same object. In some settings the structure of these covariances

shows a linear increase in blur consistent with the geometric blur model. In addition the amount of blur depends on how much change in viewing direction there is between images even using region of interest operators designed to offset this change. As a corollary it seems that the geometric blur model might be applicable to constructing feature descriptors for use with the region of interest operators discussed. This is the first study of this structure, and may be useful for future design and analysis of region of interest operators and descriptors.

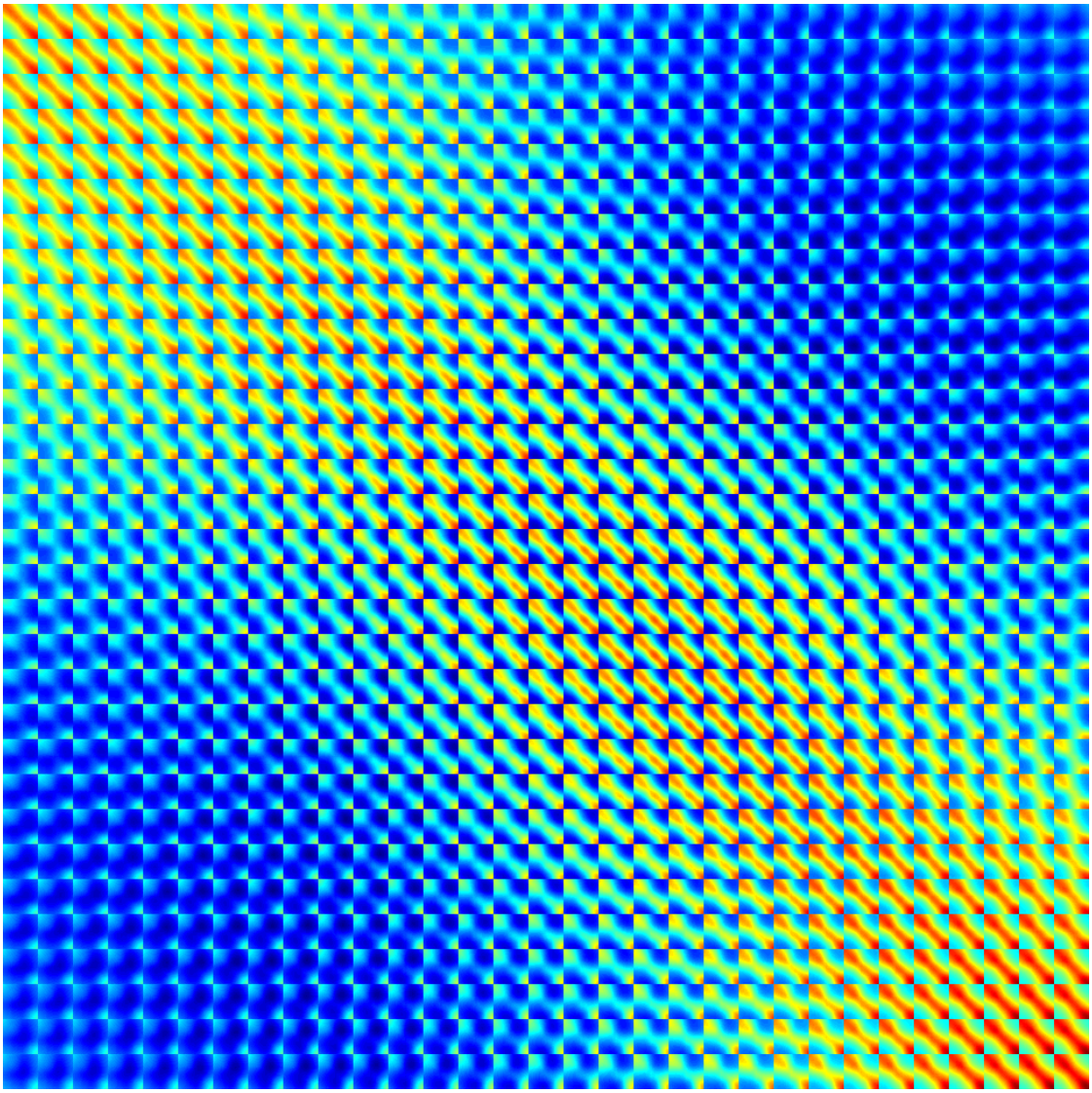


Figure 3.8: Covariance of intensities between corresponding patches of intensities using Harris-Affine detector on image sets 1,3, and 4, which show variation in viewpoint.

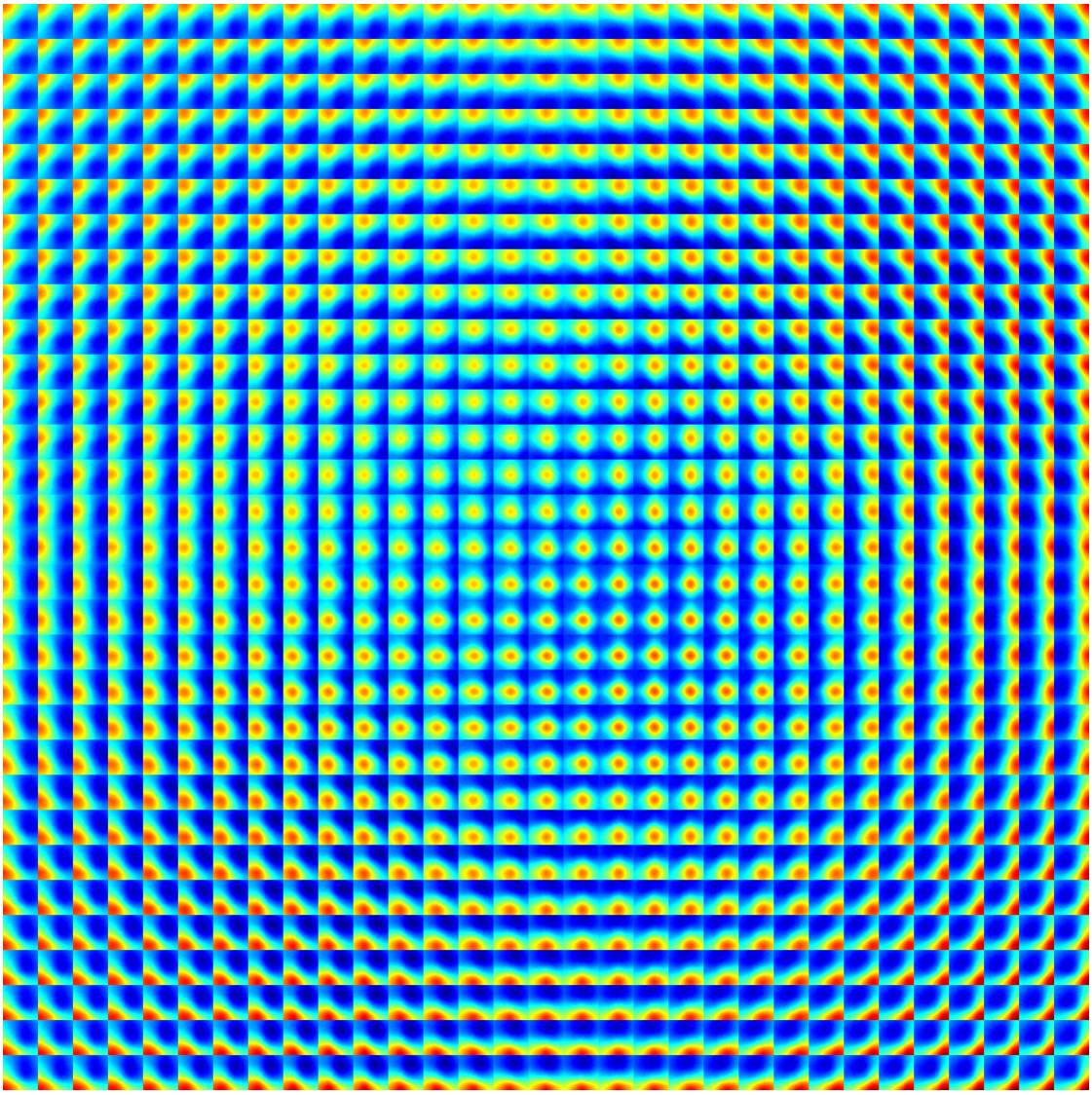


Figure 3.9: Covariance of intensities between corresponding patches using Harris-Affine detector on image sets 1,3, and 4, which show variation in viewpoint. These have been reshaped so that each small block represents the covariance of all the pixels in one patch with respect to a particular pixel in the corresponding patch. The location of the small block specifies the pixel in the corresponding patch. For example the block at the lower right of the image shows the covariance of the all the pixels in a patch with the pixel in the lower right of the corresponding patch.

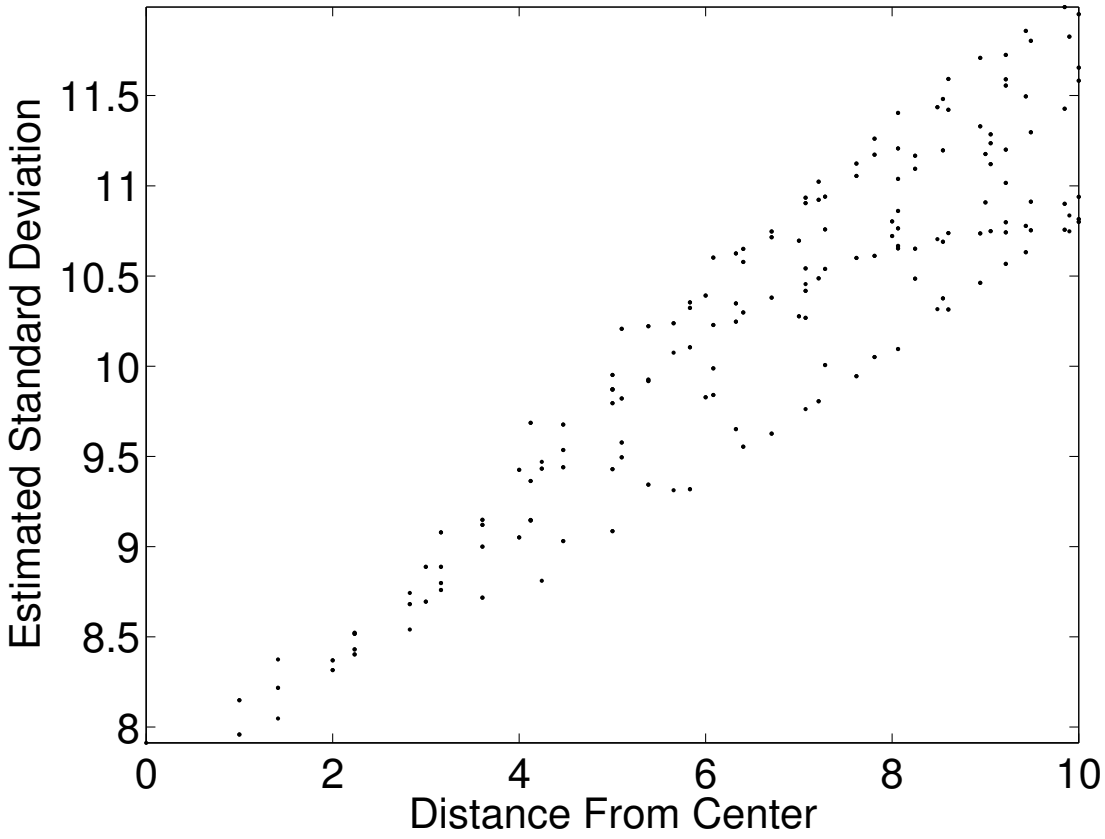


Figure 3.10: Results of fitting Gaussians to the blur patterns shown in Figure 3.9 of covariance of intensities between corresponding patches using Harris-Affine detector on image sets 1,3, and 4, which show variation in viewpoint. The estimated standard deviation is plotted against the distance from the center. The amount of blur in the covariance increases almost linearly.

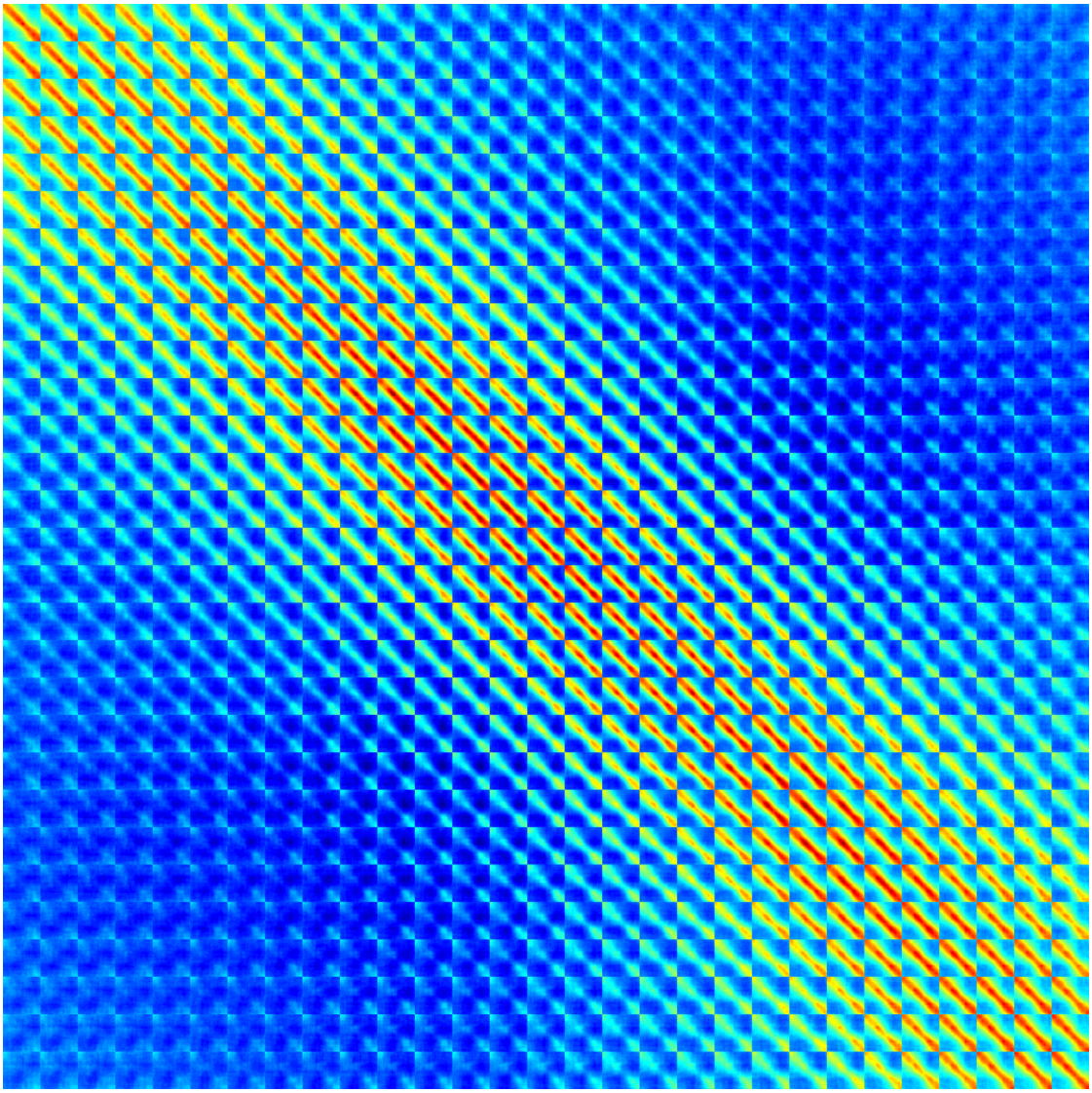


Figure 3.11: Covariance of edge response between corresponding patches using Harris-Affine detector on image sets 1,3, and 4, which show variation in viewpoint.

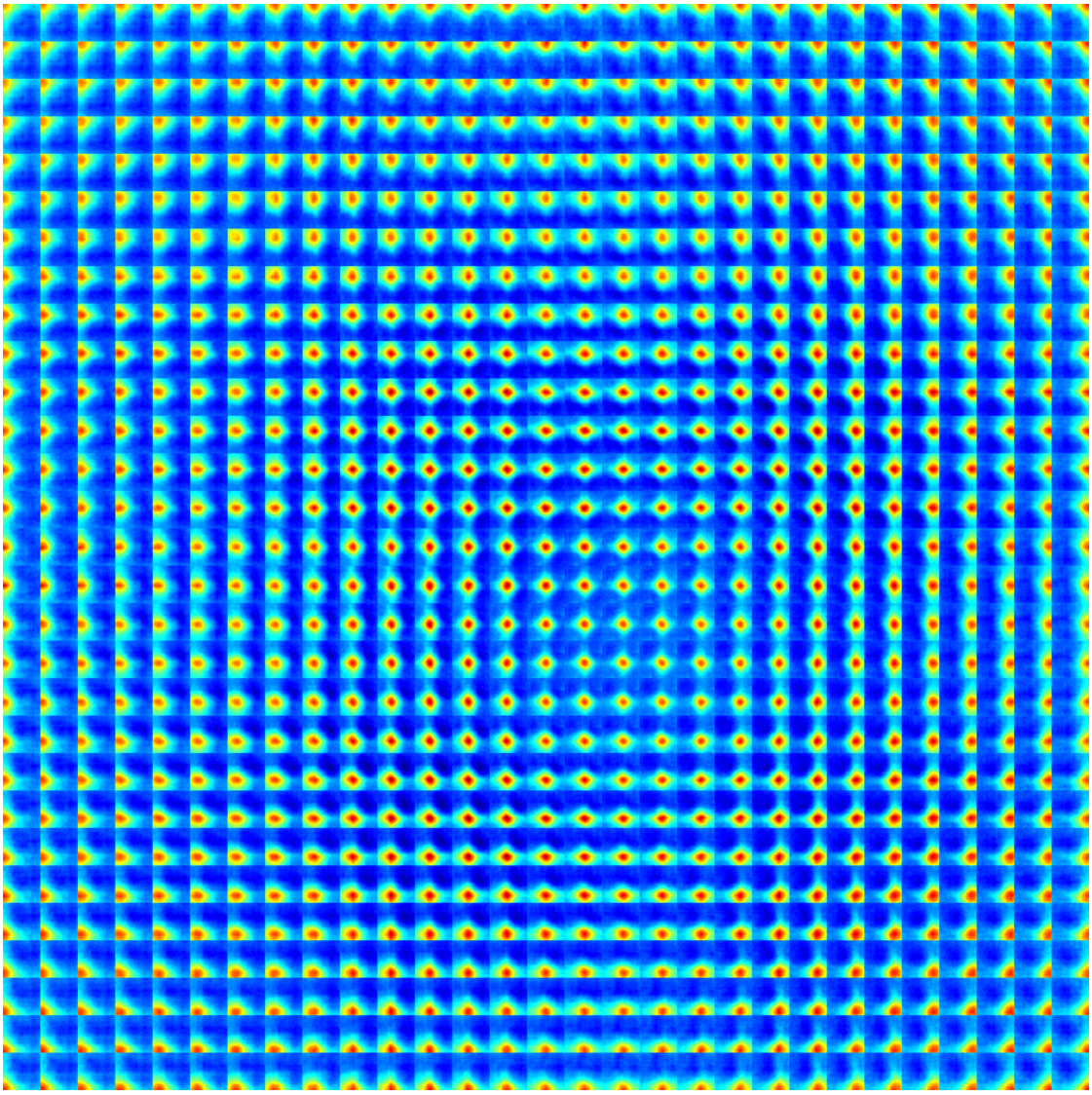


Figure 3.12: Covariance of edge response between corresponding patches using Harris-Affine detector on image sets 1,3, and 4, which show variation in viewpoint. These have been reshaped so that each small block represents the covariance of all the pixels in one patch with respect to a particular pixel in the corresponding patch. The location of the small block specifies the pixel in the corresponding patch. For example the block at the lower right of the image shows the covariance of the all the pixels in a patch with the pixel in the lower right of the corresponding patch.

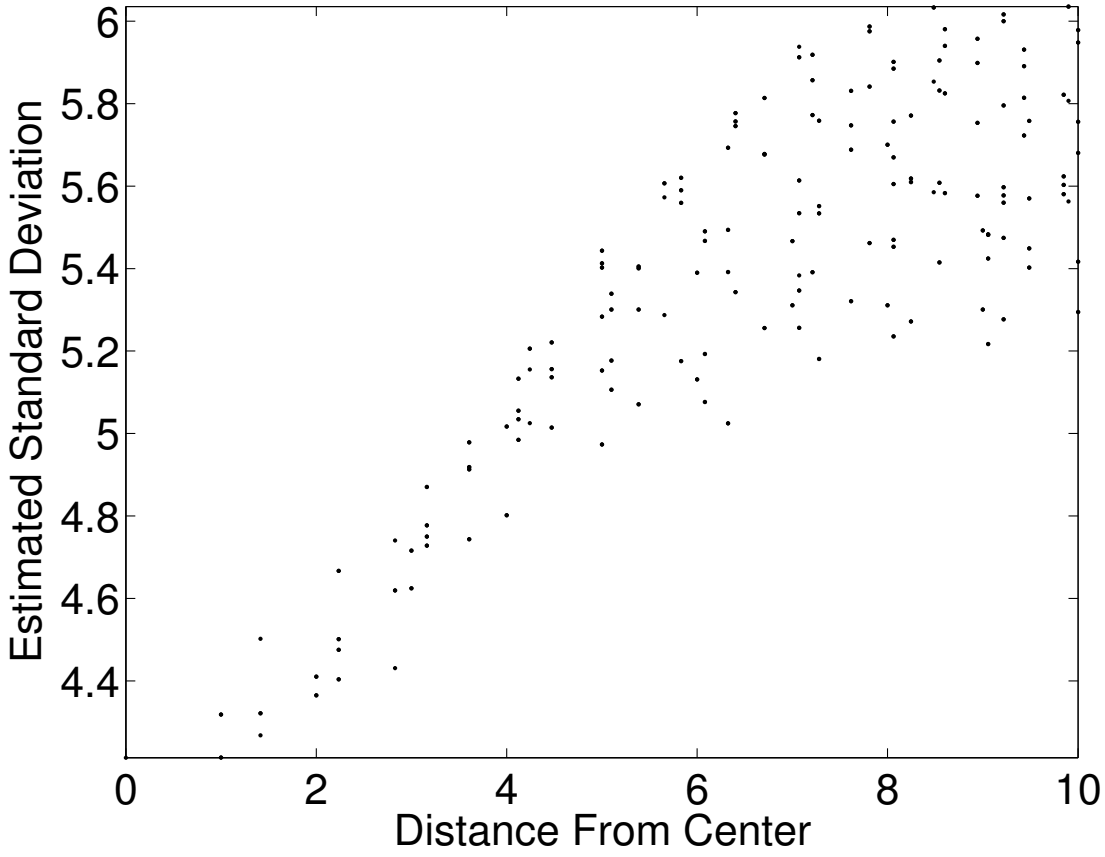


Figure 3.13: Results of fitting Gaussians to the blur patterns shown in Figure 3.12 of covariance of edge response between corresponding patches using Harris-Affine detector on image sets 1,3, and 4, which show variation in viewpoint. The estimated standard deviation is plotted against the distance from the center. The amount of blur in the covariance increases almost linearly.

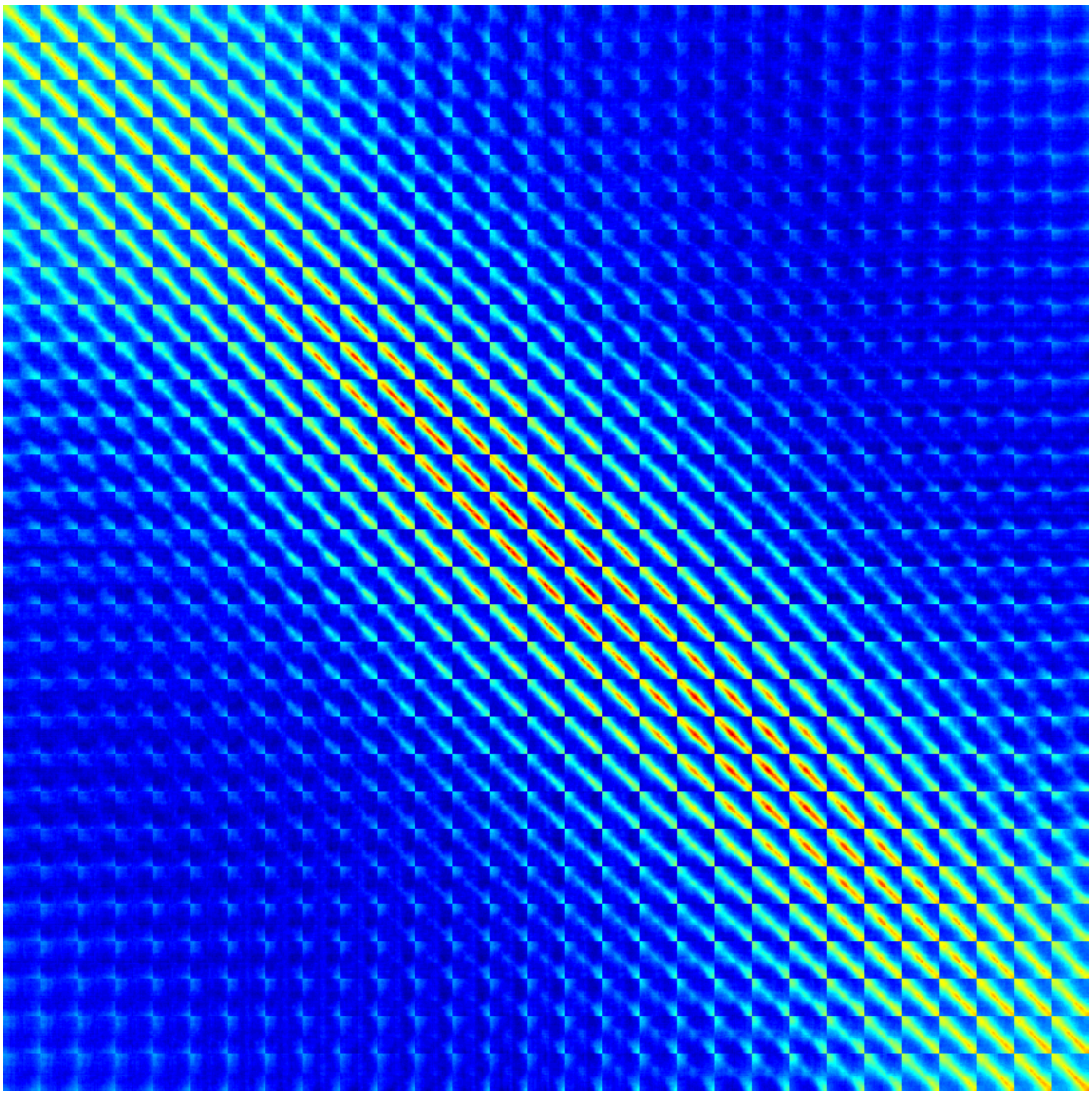


Figure 3.14: Covariance of edge response between corresponding patches using Harris-Affine detector on image sets 2,5, and 6, which do not show variation in viewpoint, but do vary focus and illumination.

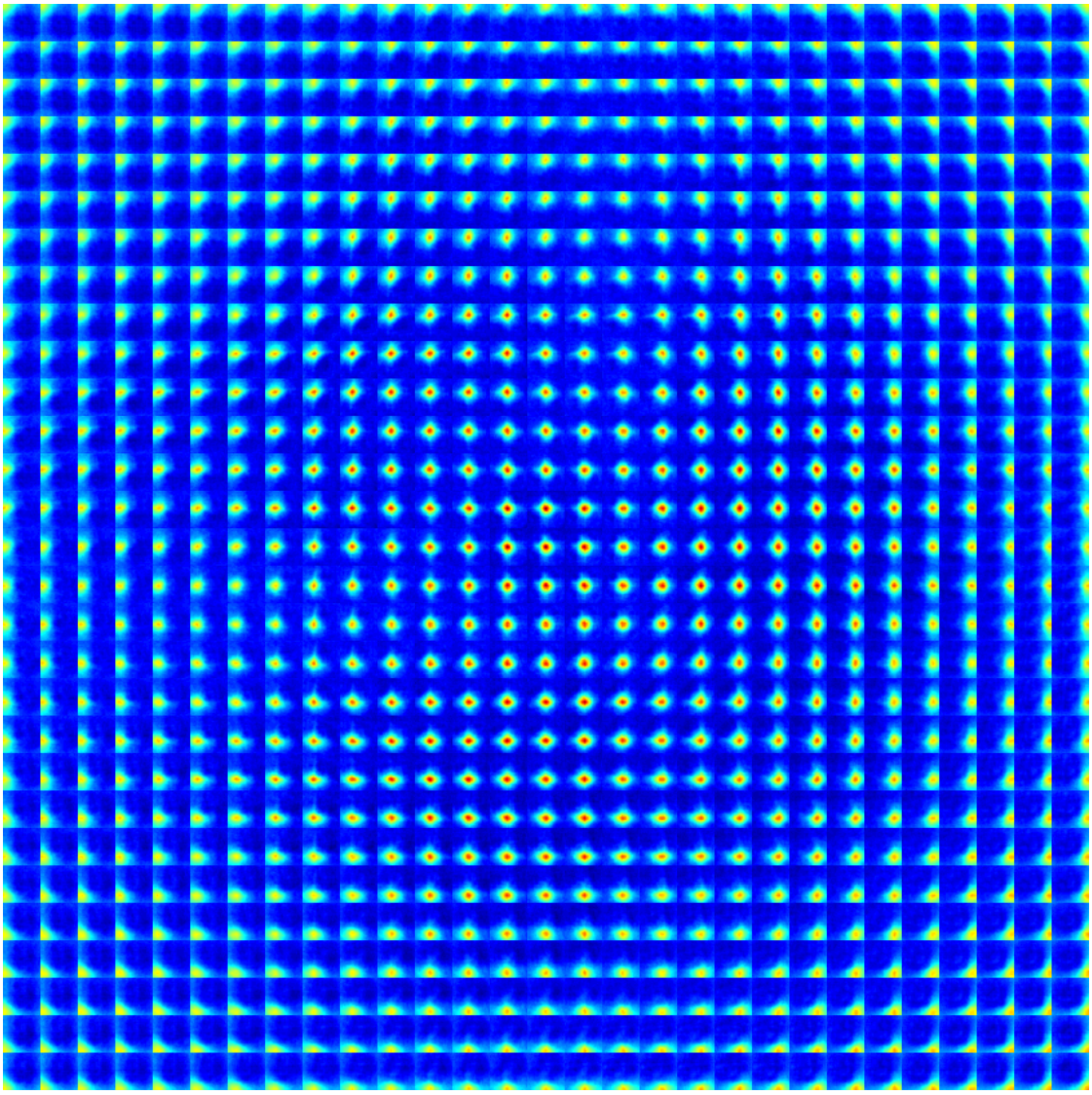


Figure 3.15: Covariance of edge response between corresponding patches using Harris-Affine detector on image sets 2,5, and 6, which do not show variation in viewpoint, but do vary focus and illumination. These have been reshaped so that each small block represents the covariance of all the pixels in one patch with respect to a particular pixel in the corresponding patch. The location of the small block specifies the pixel in the corresponding patch. For example the block at the lower right of the image shows the covariance of the all the pixels in a patch with the pixel in the lower right of the corresponding patch.

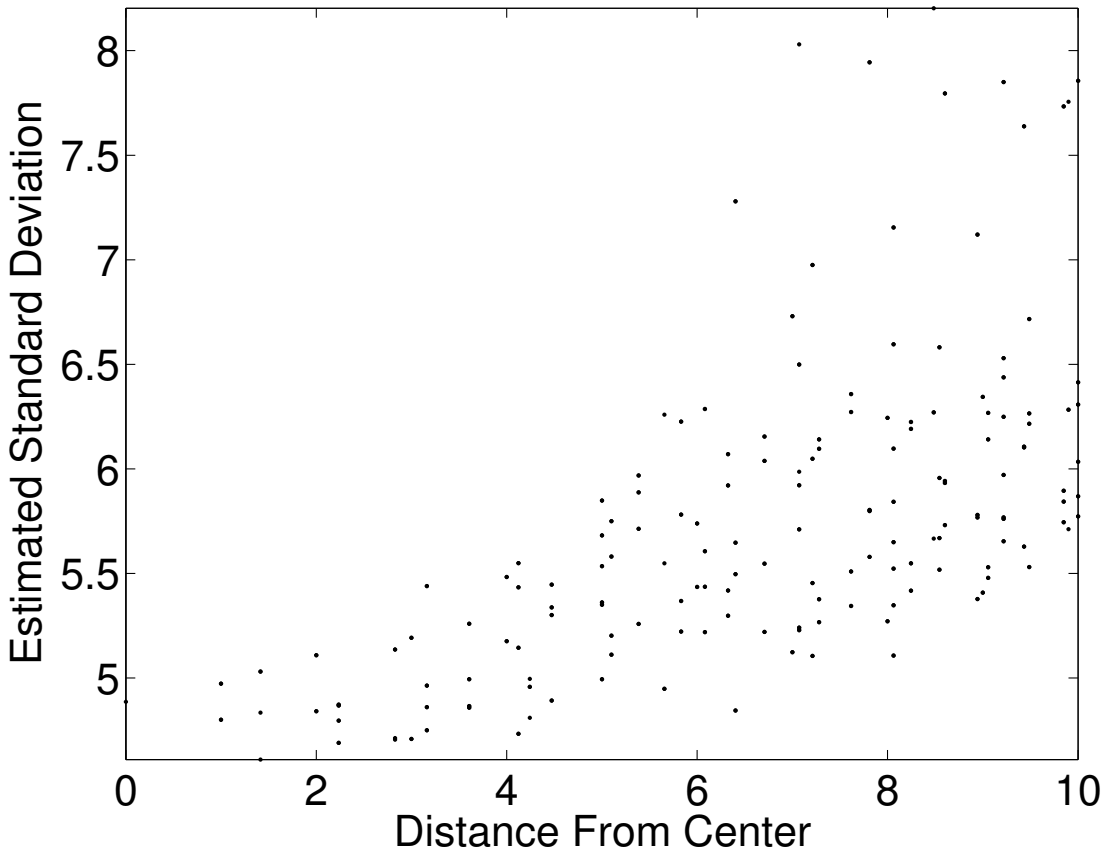


Figure 3.16: Results of fitting Gaussians to the blur patterns shown in Figure 3.15 of covariance of edge response between corresponding patches using Harris-Affine detector on image sets 2,5, and 6, which do not show variation in viewpoint, but do vary focus and illumination. The estimated standard deviation is plotted against the distance from the center. The amount of blur in the covariance increases almost linearly but with a smaller slope than for the image sets with varying viewpoint as in Figure 3.13.

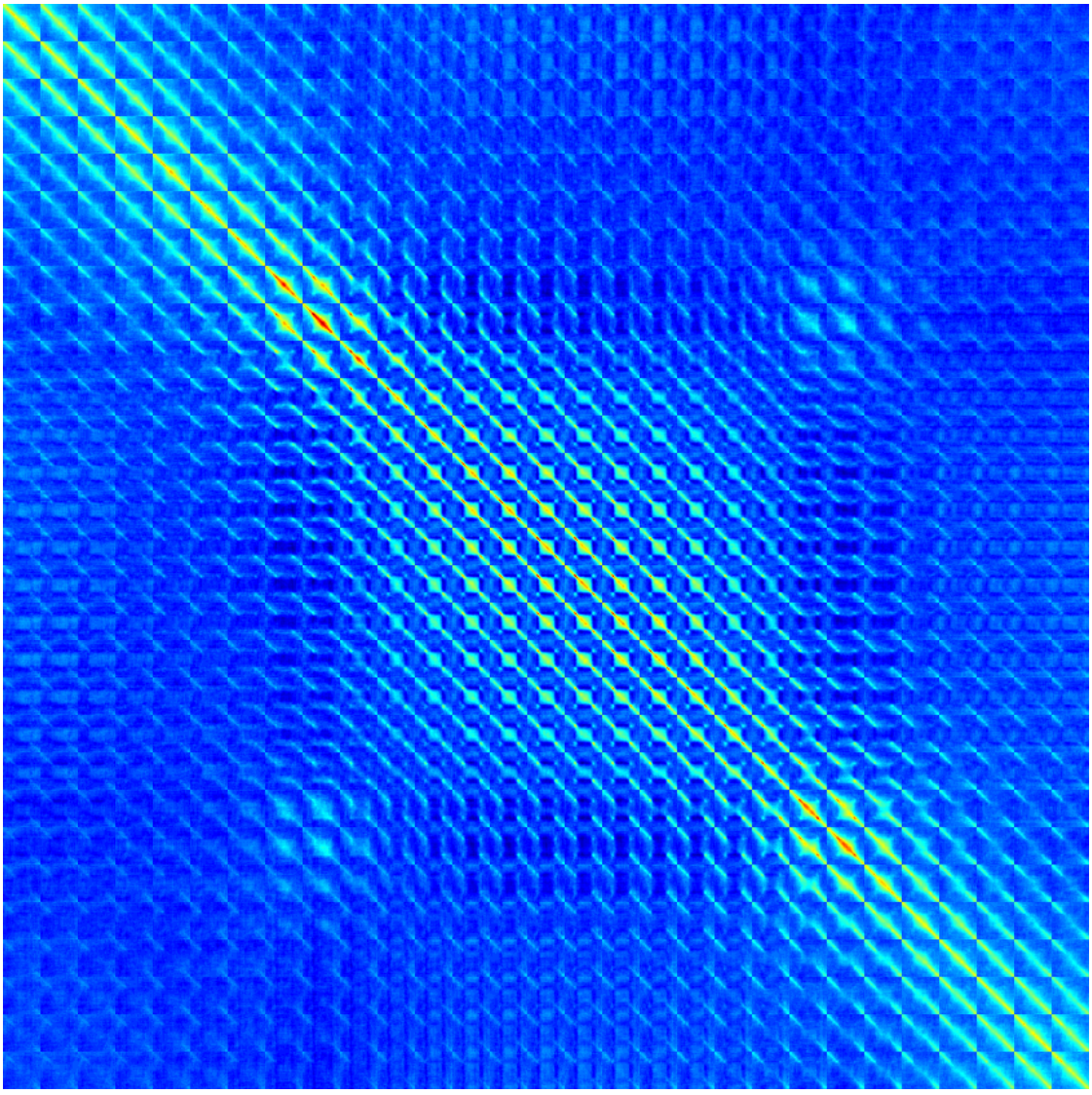


Figure 3.17: Covariance of edge response between corresponding patches using he MSER detector on image sets 1,3, and 4, which show variation in viewpoint.

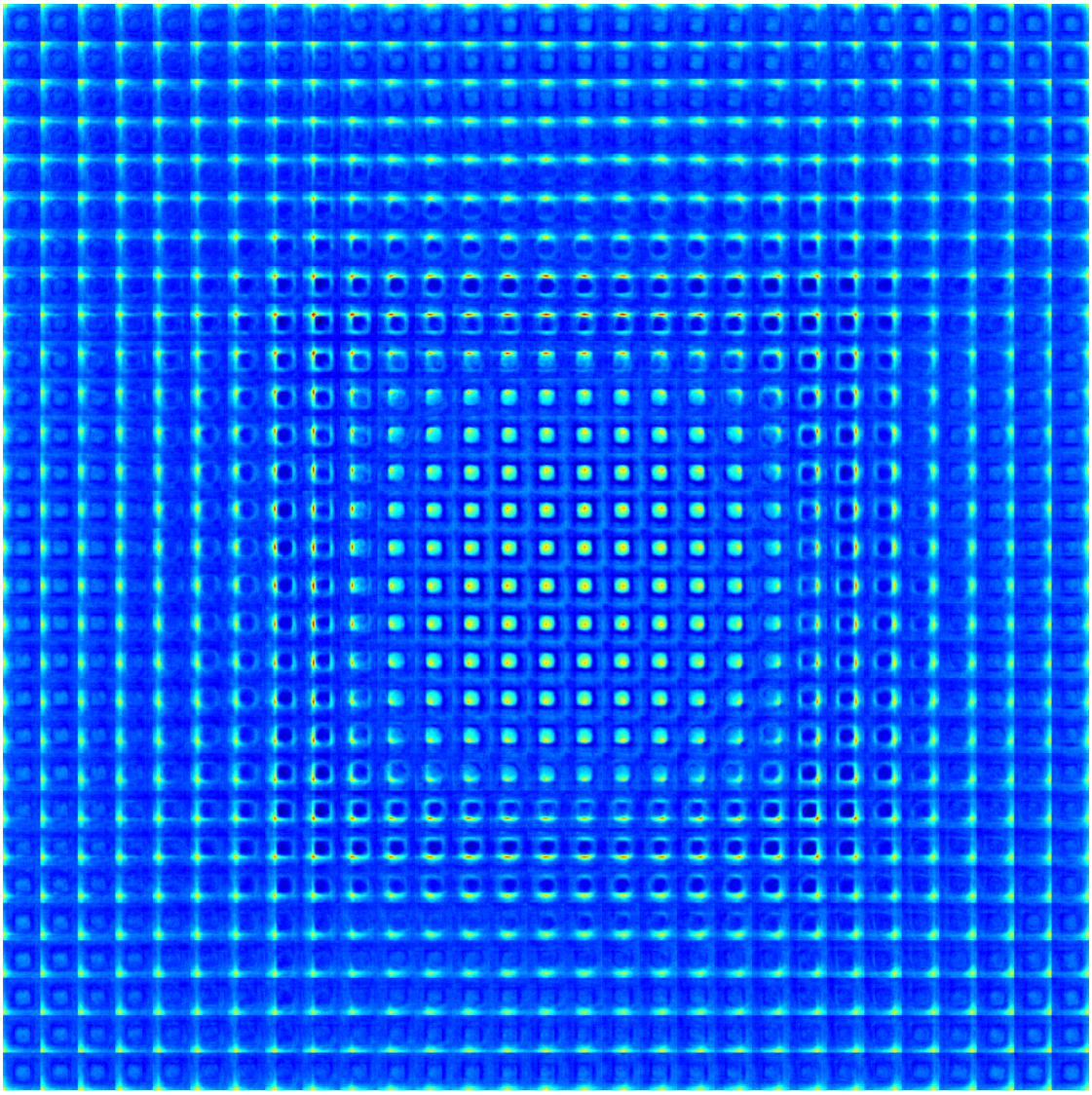


Figure 3.18: Covariance of edge response between corresponding patches using the MSER detector on image sets 1,3, and 4, which show variation in viewpoint. These have been reshaped so that each small block represents the covariance of all the pixels in one patch with respect to a particular pixel in the corresponding patch. The location of the small block specifies the pixel in the corresponding patch. For example the block at the lower right of the image shows the covariance of the all the pixels in a patch with the pixel in the lower right of the corresponding patch. The structure in the center results from the MSER region of interest operator placing regions around blobs of constant intensity. As a result the edge magnitude is small and nearly random near the center and results in the distinctive pattern shown.

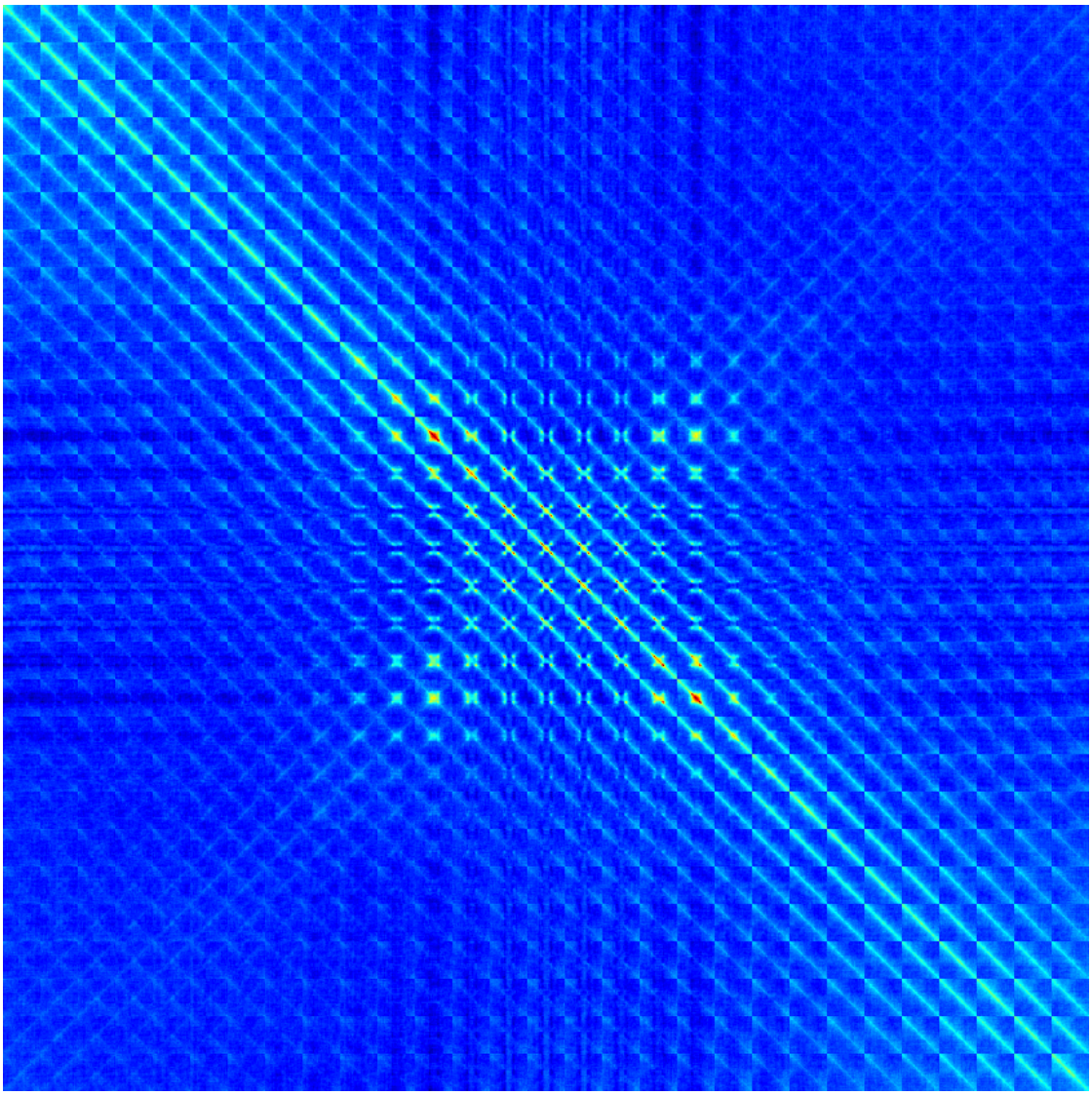


Figure 3.19: Covariance of edge response between corresponding patches using a version of the MSER detector, that returns double sized regions, on image sets 1,3, and 4, which show variation in viewpoint.

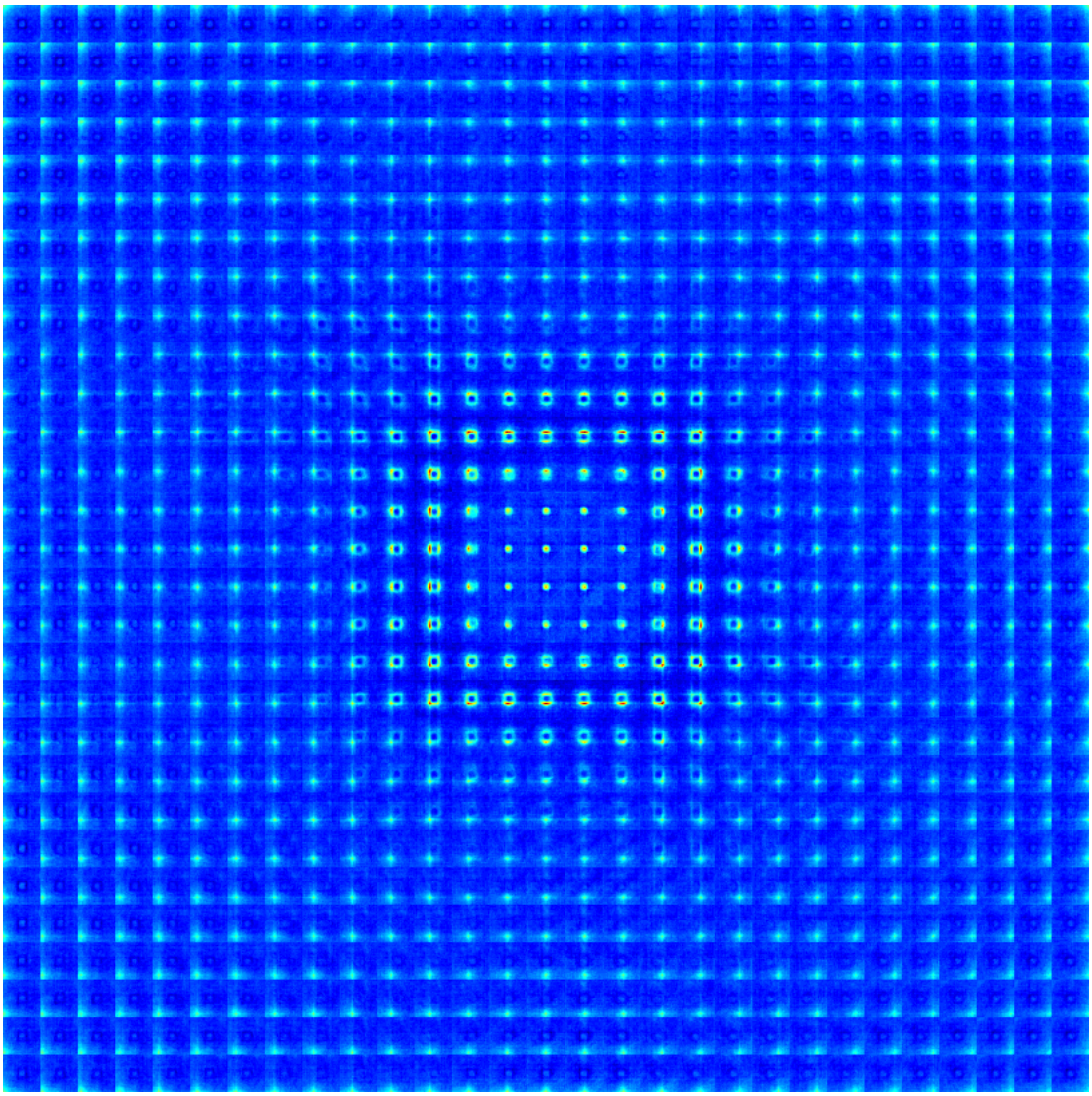


Figure 3.20: Covariance of edge response between corresponding patches using the a version of the MSER detector, that returns double sized regions, on image sets 1,3, and 4, which show variation in viewpoint. These have been reshaped so that each small block represents the covariance of all the pixels in one patch with respect to a particular pixel in the corresponding patch. The location of the small block specifies the pixel in the corresponding patch. For example the block at the lower right of the image shows the covariance of the all the pixels in a patch with the pixel in the lower right of the corresponding patch. The structure in the center results from the MSER region of interest operator placing regions around blobs of constant intensity. As a result the edge magnitude is small and nearly random near the center and results in the distinctive pattern shown.

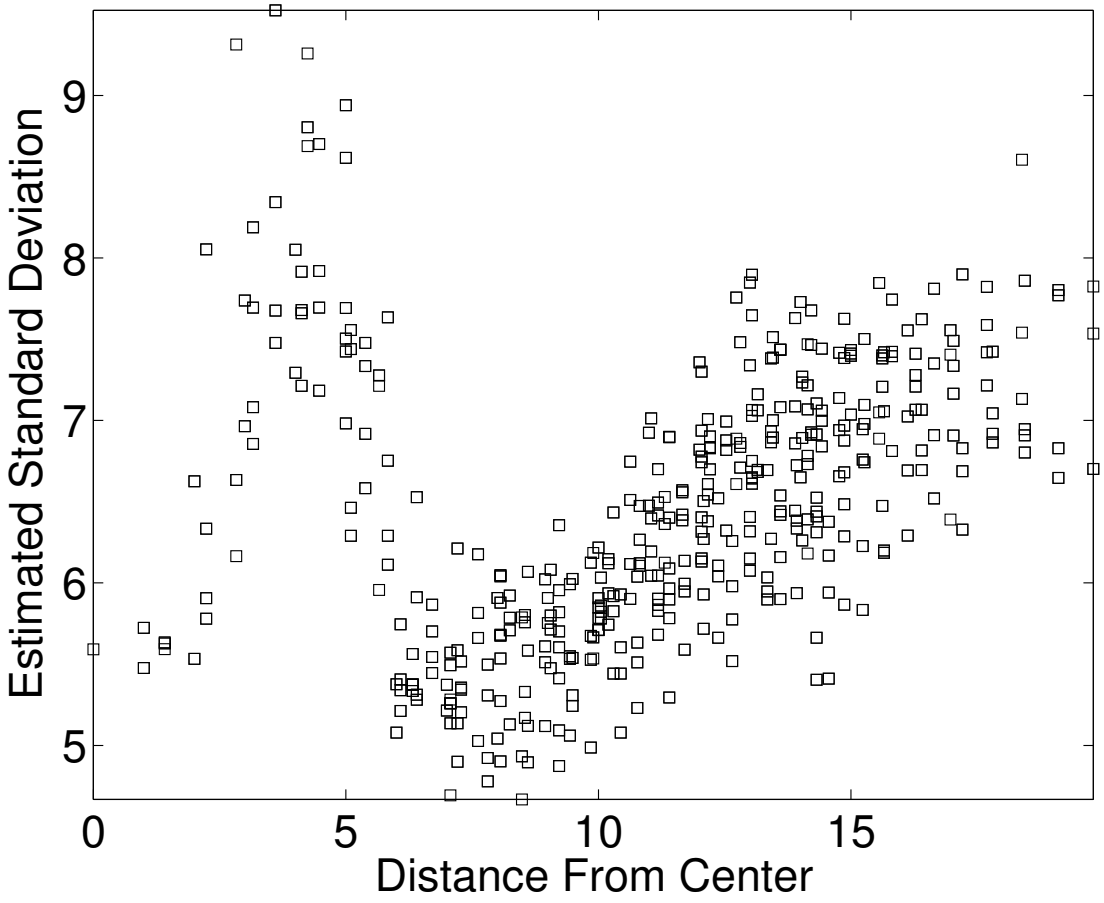


Figure 3.21: Results of fitting Gaussians to the blur patterns shown in Figure 3.20 of covariance of edge response between corresponding double patches using the a version of the MSER detector, that returns double sized regions, on image sets 1,3, and 4, which show variation in viewpoint. The estimated standard deviation is plotted against the distance from the center. The amount of blur in the covariance increases linearly but only outside of the central disc.

Chapter 4

Geometric Blur Descriptor and Image Classification

4.1 Introduction

Geometric blur offers an efficient way to average a signal over a range of transforms, but this is still too expensive to use in large scale recognition experiments. This chapter presents a technique for sub-sampling the geometric blur of a signal in order to produce a more concise descriptor useful for recognition. Experimental evidence indicates that the descriptor in fact performs well on a difficult image categorization task. The categorization task entails identifying images that contain object categories. Baseline experiments using color and texture produce results much worse that experiments using geometric blur, suggesting that the success of the geometric blur descriptor may result from a rough relationship to shape. This relationship is made more explicit by considering matchings for the purpose of alignment in Chapter 5.

4.2 Caltech 101 Dataset

The Caltech 101 [Fei-Fei *et al.*, 2004] dataset consists of images from 101 categories of objects: from accordion to kangaroo to yin-yang, available at [cal,]. Example images from 100 of the categories can be seen in figure 4.1. There are a wide variety of image categories: man-made objects, animals, indoor images and outdoor images, drawings, etc. In addition many of the images have background clutter. There are up to 800 images in a category, although many categories contain 50 or fewer images. Some categories offer more variation and clutter than others as can be seen Figure 4.2 showing the average of the images in each class. Motivated by these images we next develop a feature descriptor based on geometric blur.

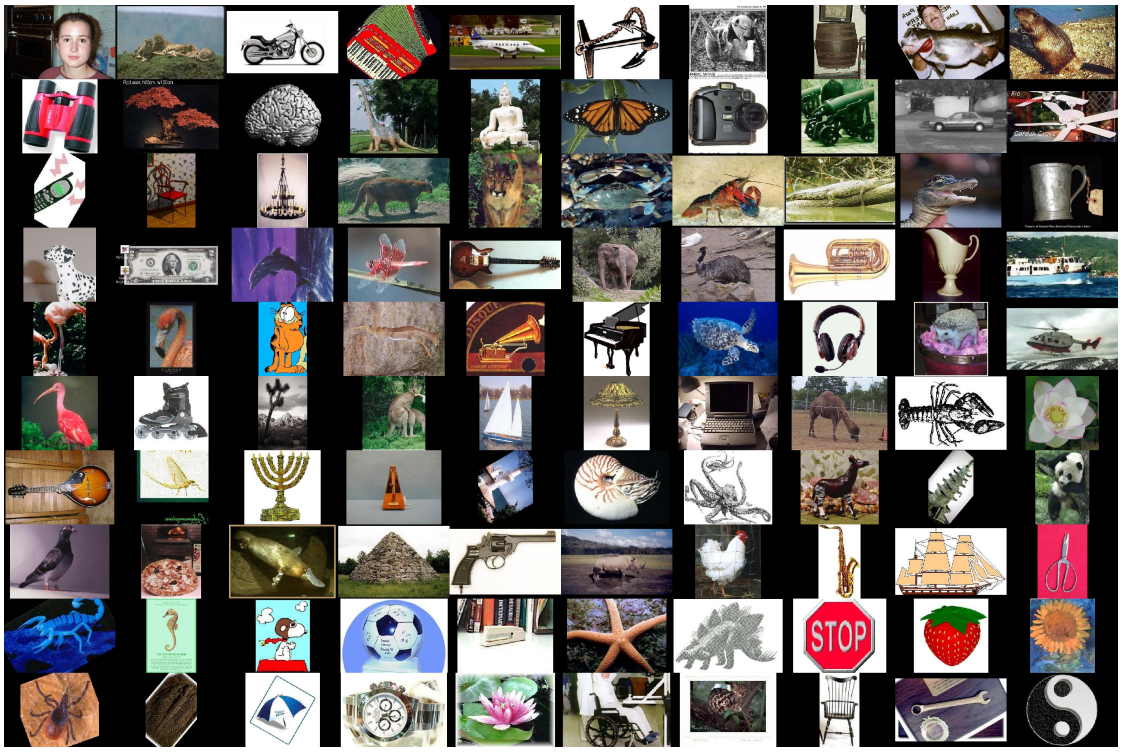


Figure 4.1: Example images from 100 of the categories in the Caltech 101 dataset.

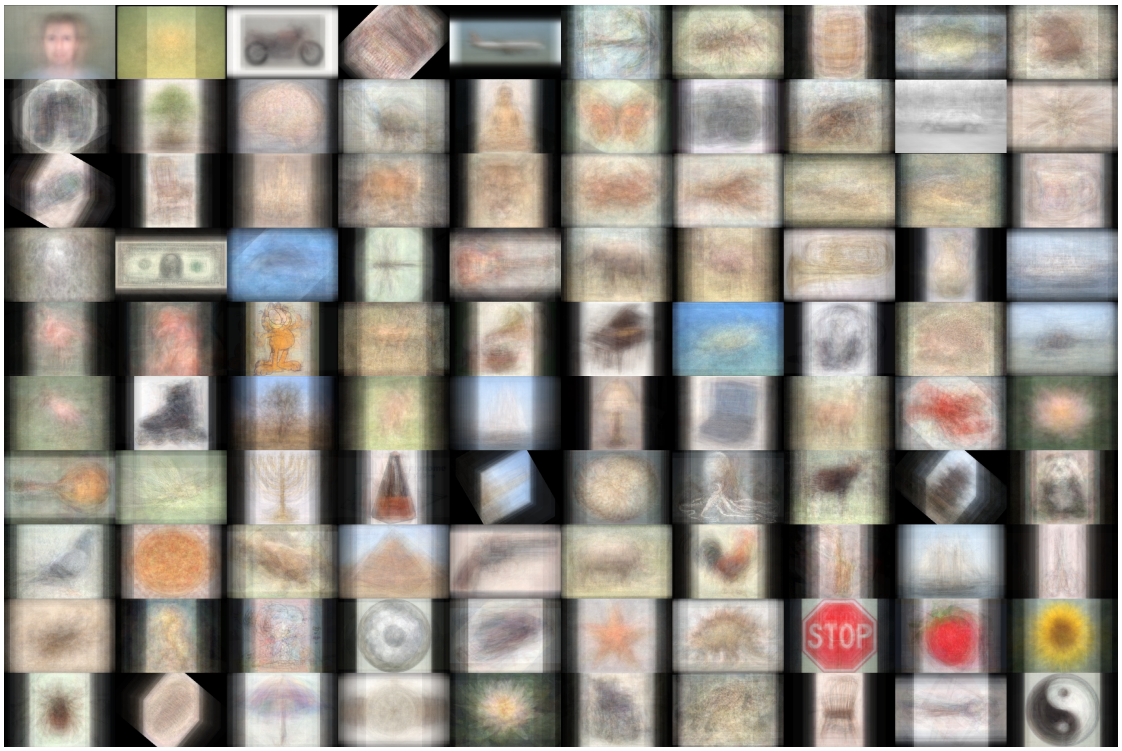


Figure 4.2: Average images for 100 of the Caltech 101 dataset categories.

4.3 Geometric Blur Descriptor

Feature descriptors and interest point / region of interest operators are the head and tail respectively of a thorny beast indeed. The results in Chapter 3 show that the choice of interest point operator effects the resulting variation that must be tolerated by a feature descriptor.

One benefit of the spatially varying blur is that geometric blur can be used for localization. Our original work [Berg and Malik, 2001] concentrates mainly on this aspect of geometric blur. This is quite different from other contemporary descriptors such as SIFT [Lowe, 1999] that rely on an interest point operator to select similar locations for potential matches. As a result a somewhat promiscuous interest point operator can be used in conjunction with geometric blur, and the localization of

the best match can be left up to the descriptor itself. We will place interest points anywhere in an image where there is a strong edge response, using sampling with repulsion to spread interest points throughout the image.

In the case of the Caltech 101 dataset we observe a relatively small amount of in-plane rotation, and as a results choose a constant region of interest. This does not preclude a multi-scale approach, it just means that scale will not be determined locally. Figure 4.3 shows a sparse sample of some interest points in an image as well as the region of interest and descriptor for one. Figure 5.1 shows the full set of interest points for an image.

Two design choices are necessary to use geometric blur: the source for sparse feature channels, and the blur kernel and amount. Once these have been made constructing a descriptor is accomplished by sub-sampling the geometric blur for each channel.

Feature Channels

Motivated by the wide range of appearance in the dataset we base the feature channels on a coarse scale edge detector. The best results are obtained using the boundary detector of [Martin *et al.*, 2004]. This boundary detector is constructed not to respond to texture, and produces relatively consistent boundary maps. In addition a simple and computationally less expensive edge detector based on elongated derivative of Gaussian filters is used for comparison [Morrone and Burr, 1988]. In both cases edge detection results are split up by orientation and oriented non-max suppression is applied producing multiple sparse channels as shown in Figure 4.3[1].

Blur Kernel

As before we use a simple blur kernel based on a Gaussian. If $G_a(x)$ is a Gaussian

[1] As an example of an alternate feature channel, our work in [Efros *et al.*, 2003] uses sparse channels derived from optical flow, followed by "geometric" blur in the temporal domain.

with standard deviation a then:

$$K_x(y) = G_{\alpha|x|+\beta}(y)$$

is our blur kernel. The kernel is normalized with respect to the L^2 norm.

Sub-sampling

The geometric blur of a signal should be sub-sampled using a pattern that matches the amount of blur introduced. In particular in the periphery fewer samples are required. For the kernel we consider above this implies a density of samples decreasing linearly with distance from the origin. The sampling pattern used in these experiments is shown in Figure 4.3.

A quick summary of possible steps for computing geometric blur descriptors for an image follows:

1. Use an edge detector to compute oriented edge channels for the image.
2. Choose interest points using random sampling with repulsion on points with high edge energy.
3. Compute multiple blurred versions of the channels as described in Section 2.2.5.
4. Around each interest point, for each channel, sample points according to the dart-board patter in Figure 4.3. These samples should be drawn from the appropriate blurred version of the channel.
5. These samples form the geometric blur descriptors.

Following the motivation in Chapter 2 the descriptors are compared using normalized correlation.

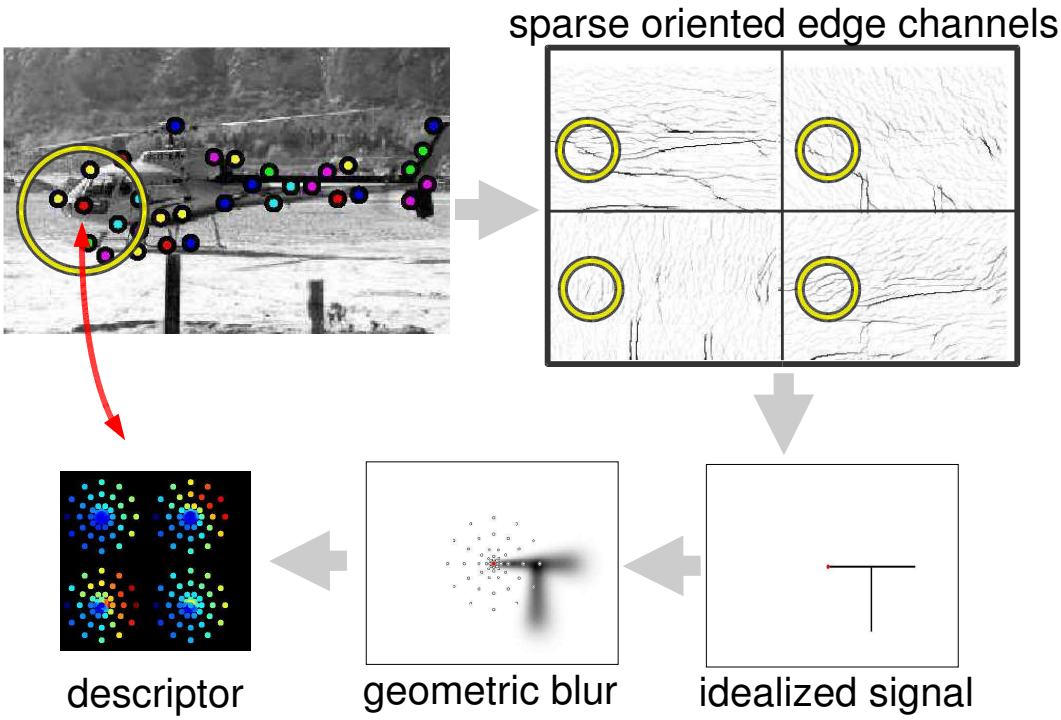


Figure 4.3: The steps to compute a geometric blur descriptor. Starting with a feature point on an image (shown in **red** in the **upper left**) and a region of interest (indicated in **yellow**). The sparse feature channels are cropped out as shown in the **upper right**. Geometric blur is applied to each channel (shown here with an idealized signal for clarity) and the signal is sub-sampled. The final descriptor is the vector of values shown as colored dots at the **lower left**.

4.4 Experiments

We first describe the experimental setup for testing image categorization on the Caltech 101 dataset and then present base-line results using color and texture features. These are followed by results using the geometric blur descriptors described above.

Basic Setup: Fifteen exemplars were chosen randomly from each of the 101 object classes and the background class, yielding a total 1530 exemplars. For each class, we select up to 50 testing images, or "probes" excluding those used as exemplars. Results for each class are weighted evenly so there is no bias toward classes with more images. To do this the percentage of queries from a class that are classified correctly is averaged over all of the classes. This is the same as the mean diagonal entry in the class confusion matrix. It turns out the standard deviation of this average confusion over choices of training and testing is quite small, on the order of 2%, as might be expected from averages over so many draws from somewhat similar distributions.

4.4.1 Experimental results

Whole Image Baseline

The first baseline experiment is motivated by the apparent structure in the average image for some of the categories in Figure 4.2. All of the images are resized to the same size and then reshaped into a long vector of pixel values. A simple nearest neighbor classifier using the L^2 norm produces an average recognition rate of 15%. Somewhat surprisingly it makes no difference whether color information is included or not.

Color Baseline

The second baseline experiment uses color histograms over the entire image and a simple nearest neighbor classifier. The color histograms are computed on the a, b values of the L, a, b color space jointly with 50 bins in each dimension for a total of

2500 bins. The histograms are smoothed and normalized before comparison using L^2. This results in a an average recognition rate of 13%.

Texture Baseline

Recently Hao Zhang [Zhang, 2005] and others have conducted similar experiments using texture resulting in recognition rates of 17%.

Geometric Blur Descriptor

We take two approaches to comparing images using geometric blur descriptors. The first is based on voting. Each feature in a query image is classified as "voting" for one category using a nearest neighbor classifier based on all the features in the training set. The image is assigned to the category with the most votes. This results in a recognition rate of 52%. The second strategy classifies images using nearest neighbor. The similarity function used is the average similarity between a feature in one image and its best match in the other. This results in a recognition rate of 40%[2].

4.5 Conclusion

We have developed a descriptor based on geometric blur that proves effective in classifying images of objects in the Caltech 101 dataset. The significantly better performance using geometric blur features over color or texture features indicate that the blur descriptor has access to some additional information, possibly related to rough local shape. Although we primarily emphasize the use of geometric blur in finding correspondences between shapes as addressed in the following chapter, it is also at the heart of the best performing image classification schemes for the Caltech

[2]It is worth noting that nearest neighbor is not necessarily the optimum strategy. Here it serves as a simple classifier and and is consistent across the various cues. Recent work on learning local classifiers to improve nearest neighbor classification [Zhang, 2005] improves the results on texture from 17% to 25%, and for the second geometric blur method from 40% to 56%. Combining geometric blur and texture improves performance to 59%. These improvements from using a more powerful classification technique do not alter the relative performance of the cues, and consistently indicate that by far the most informative cue is the similarity of geometric blur descriptors.

101 dataset [Zhang, 2005]. In addition it has been used as a local model of shapes to improve edge detection with mid-level cues, providing the best improvement out of techniques compared on a dataset of natural images [Ren *et al.*, 2005]. In Chapter 5 a matching framework is built around geometric blur.

Chapter 5

Alignment as a Discrete Matching Problem

5.1 Introduction

Our thesis is that recognizing object categories, be they fish or bicycles, is fundamentally a problem of deformable shape matching. We have developed a geometric blur descriptor that provides some estimate as to whether two regions of an image might have similar underlying shapes. In order to determine whether there is an alignment between whole objects we look for matchings between objects that map regions to similar regions and maintain the rough assembly of the regions.

These two ideas of rough shape: that local regions have similar edge structure and that the relationship between regions should be maintained, are both consistent with the idea that shape is that which is preserved under some set of transformations.

We will formulate the matching problem as an optimization, trying to satisfy the following constraints:

1. Corresponding regions on the two shapes should have similar local structure.

This will be measured by a geometric blur descriptor.

2. Minimizing geometric distortion: If i and j are points on the model corresponding to i' and j' respectively, then the vector from i to j, $\vec{r}_{ij}$ should be consistent with the vector from i' to j', $\vec{r}_{i'j'}$. If the transformation from one shape to another is a translation accompanied by pure scaling, then these vectors must be scalar multiples. If the transformation is a pure Euclidean motion, then the lengths must be preserved. Etc.

3. Smoothness of the transformation from one shape to the other. This enables us to interpolate the transformation to the entire shape, given just the knowledge of the correspondences for a subset of the sample points. We use regularized thin plate splines to characterize the transformations.

The similarity of point descriptors and the geometric distortion is encoded in a cost function defined over the space of correspondences. We purposely construct this to be an integer quadratic programming problem (cf. Maciel and Costeira [Maciel and Costeira, 2003]) and solve it using fast-approximate techniques[1]. These approximate techniques show good performance on the problem instances generated. In addition a similar framework is used on different low level features in our work with Xiaofeng Ren [Holub *et al.*, 2005] on localizing humans in still images.

5.2 Related Work

There have been several approaches to shape recognition based on spatial configurations of a small number of keypoints or landmarks. In geometric hashing [Lamdan *et al.*, 1990], these configurations are used to vote for a model without explicitly solving

[1]It is worth noting that this formulation is amenable to various probabilistic interpretations, maximum likelihood estimation for a product of Gaussian models among others.

for correspondences. Amit et al. [Amit *et al.*, 1997] train decision trees for recognition by learning discriminative spatial configurations of keypoints. Leung et al. [Leung *et al.*, 1995], Schmid and Mohr [Schmid and Mohr, 1997], and Lowe [Lowe, 2004] additionally use gray level information at the keypoints to provide greater discriminative power. Lowe's SIFT descriptor has been shown in various studies e.g. [Mikolajczyk and Schmid., 2003] to perform very well particularly at tasks where one is looking for identical point features. Recent work extends this approach to category recognition [Fergus *et al.*, 2003] [Fei-Fei *et al.*, 2003] [Fei-Fei *et al.*, 2004], and to three-dimensional objects [Rothganger *et al.*, 2003].

It should be noted that not all objects have distinguished key points (think of a circle for instance), and using key points alone sacrifices the shape information available in smooth portions of object contours. Approaches based on extracting edge points are, in our opinion, more universally applicable. Huttenlocher et al. developed methods based on the Hausdorff distance [Huttenlocher *et al.*, 1993]. A drawback for our purposes is that the method does not return correspondences. Methods based on Distance Transforms, such as [Gavrila and Philomin, 1999], are similar in spirit and behavior in practice. Work based on shape contexts is indeed aimed at first finding correspondences [Belongie *et al.*, 2001][Mori *et al.*, 2001] and is close to the spirit of this work. Another approach is the non-rigid point matching of [Chui and Rangarajan, 2003] based on thin plate splines and "softassign".

One can do without extracting either keypoints or edge points: Ullman et al propose using intermediate complexity features, a collection of image patches,[Ullman *et al.*, 2002].

For faces and cars the class specific detectors of [Viola and Jones, 2001] [Schneiderman and Kanade, 2000] [Schneiderman, 2004] have been very successful. These techniques use simple local features, roughly based on image gradients, and a cascade of classifiers for efficiency. Recent work on sharing features [Torralba *et al.*, 2004] has

extended this to multiclass problems.

A distinction of this work is that we are mainly looking for correspondences between shapes as a whole in the presence of large intraclass variation. As a result we do not expect individual local features to give correct correspondences in isolation.

5.3 Geometric Distortion Costs

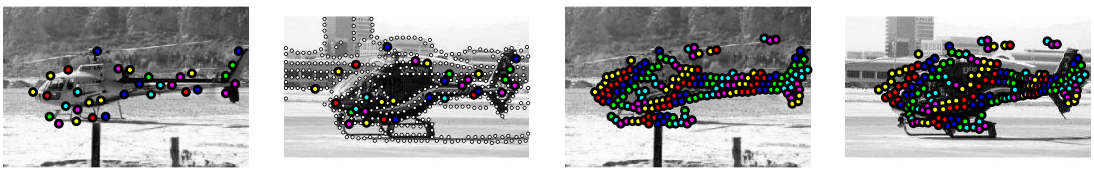


Figure 5.1: *An exemplar with a subset of feature points marked (***left***), the novel "probe" image with all feature points in white, and the feature points found to correspond with the exemplar feature points marked in corresponding colors (***left center***), the exemplar with all its feature points marked in color, coded by location in the image (***right center***), and the probe with the exemplar feature points mapped by a thin plate spline transform based on the correspondences, again colored by position in the exemplar (***far right***). See Figure 5.4 for more examples*

We consider correspondences between feature points $\{p_i\}$ in model image P and $\{q_j\}$ in image Q. A correspondence is a mapping σ indicating that p_i corresponds to $q_{\sigma(i)}$. To reduce notational clutter we will sometimes abbreviate $\sigma(i)$ as i', so σ maps p_i to $q_{i'}$.

The quality of a correspondence is measured in two ways: how similar feature points are to their corresponding feature points, and how much the spatial arrangement of the feature points is changed. We refer to the former as the match quality, and the later as the distortion of a correspondence.

We express the problem of finding a good correspondence as minimization of a cost function defined over correspondences. This cost function has a term for

the match quality and for the geometric distortion of a correspondence: $\text{cost}(\sigma) = \omega_{\text{m}} C_{\text{match}}(\sigma) + \omega_{\text{d}} C_{\text{distortion}}(\sigma)$

Where constants ω_{m} and ω_{d} weigh the two terms. The match cost for a correspondence is:

$$C_{\text{match}}(\sigma) = \sum_{i} c(i, i') \tag{5.1}$$

Where $c(i, j)$ is the cost of matching i to j in a correspondence. We use the negative of the correlation between the feature descriptors at i and j as $c(i, j)$.

We use a distortion measure computed over pairs of model points in an image. This will allow the cost minimization to be expressed as an integer quadratic programming problem.

$$C_{\text{distortion}}(\sigma) = \sum_{ij} H(i, i', j, j') \tag{5.2}$$

Where $H(i, j, k, l)$ is the distortion cost of mapping model points i and j to k to l respectively. While there are a wide variety of possible distortion measures, including the possibility of using point descriptors and other features, in addition to location, we concentrate on geometric distortion and restrict ourselves to measures based on the two offset vectors $r_{ij} = p_j - p_i$ and $s_{i'j'} = q_{j'} - q_{i'}$.

$$C_{\text{distortion}}(\sigma) = \sum_{ij} \text{distortion}(r_{ij}, s_{i'j'})$$

Our distortion cost is made up of two components:

$$C_{\text{distortion}}(\sigma) = \sum_{ij} \gamma \mathrm{d_a}(\sigma) + (1-\gamma)\mathrm{d_l}(\sigma) \tag{5.3}$$

$$d_a(\sigma) = \left(\frac{\alpha_d}{|r_{ij}|} + \beta_d\right) \left|\arcsin\left(\frac{s_{i'j'} \times r_{ij}}{|s_{i'j'}||r_{ij}|}\right)\right| \tag{5.4}$$

$$d_l(\sigma) = \frac{|s_{i'j'}| - |r_{ij}|}{(|r_{ij}| + \mu_d)} \tag{5.5}$$

where d_a penalizes the change in direction, and d_l penalizes change in length. A correspondence σ resulting from pure scale and translation will result in $d_a(\sigma) = 0$, while σ resulting from pure translation and rotation will result in $d_l(\sigma) = 0$. The constants α_d, β_d, μ_d, are all terms allowing slightly more flexibility for nearby points in order to deal with local "noise" factors such as sampling, localization, etc. They should be set relative to the scale of these local phenomena. The constant γ weighs the angle distortion term against the length distortion term.

Outliers Each point p_i, in P, is mapped to a $q_{\sigma(i)}$, in Q. This mapping automatically allows outliers in Q as it is not necessarily surjective – points q_j may not be the image any point p_i under σ. We introduce an additional point q_{null} and use $\sigma(i) = \text{null}$ to allow a point p_i to be an outlier. We limit the number of points p_i which can be assigned to q_{null}, thus allowing for outliers in both P and Q.

5.4 Correspondence Algorithm

Finding an assignment to minimize a cost function described by the terms in Equations 5.2 and 5.1 above can be written as an Integer Quadratic Programming (IQP)

problem.

$$\text{cost}(x) = \sum_{a,b} H(a,b)x_a x_b + \sum_a c(a)x_a \tag{5.6}$$

Where the binary indicator variable x has entries x_a, that if 1, indicate $\sigma(a_i) = a_j$. We then have $H(a,b) = H(a_i, a_j, b_i, b_j)$, and $c(a) = c(a_i, a_j)$ from Equations 5.2 and 5.1.

We constrain x to represent an assignment. Write x_{ij} in place of $x_{a_i a_j}$. We require $\sum_j x_{ij} = 1$ for each i. Futhermore if we allow outliers as discussed in Section 5.3, then we require $\sum_i x_{i\text{null}} \leq k$, where k is the maximum number of outliers allowed. Using outliers does not increase the cost in our problems, so this is equivalent to $\sum_i x_{i\text{null}} = k$. Each of these linear constraints are encoded in a row of A and an entry of b. Replacing H with a matrix having entries $H_{ab} = H(a,b)$ and c with a vector having entries $c_a = c(a)$. We can now write the IQP in matrix form:

$$\begin{aligned} \min \text{cost}(x) = & x'Hx + c'x \ \text{ subject to,} \\ & Ax = b, \ \ x \in \{0,1\}^n \end{aligned} \tag{5.7}$$

5.4.1 Approximation

Integer Quadratic Programming is NP-hard, however specific instances may be easy to solve. We follow a two step process that results in good solutions to our problem. We first find the minimum of a linear bounding problem, an approximation to the quadratic problem, then follow local gradient descent to find a locally minimal assignment. Although we do not necessarily find global minima of the cost function in practice the results are quite good.

We define a linear objective function over assignments that is a lower bound for

our cost function in two steps. First compute $q_a = \min \sum_b H_{ab} x_b$. Note that from here on we will omit writing the constraints $Ax = b$ and $x \in \{0, 1\}^n$ for brevity.

If x_a represents $\sigma(i) = j$ then q_a is a lower bound for the cost contributed to any assignment by using $\sigma(i) = j$. Now we have $L(x) = \sum_a (q_a + c_a) x_a$ as a lower bound for $cost(x)$ from Equation 5.7. This construction follows [Maciel and Costeira, 2003], and is a standard bound for a quadratic program. Of note is the operational similarity to geometric hashing.

The equations for q_a and L are both integer linear programming problems, but since the vertices of the constraint polytopes lie only on integer coordinates, they can be relaxed to linear programming problems without changing the optima, and solved easily. In fact due to the structure of the problems in our setup they can be solved explicitly by construction. If n is the length of x, each problem takes $O(n)$ operations with a very small constant. Computing q_a for $a = 1 \ldots n$ requires $O(n^2)$ time.

We then perform gradient descent changing up to two elements of the assignment at each step. This takes $O(n^2)$ operations per step, and usually requires a very small number of steps (we put an upper bound on the number of steps). In practice we can solve problems with $m = 50$ and $n = 2550$, 50 possible matches for each of 50 model points with outliers, in less than 5 seconds.

5.5 Correspondence results

Given a model image P of an object, and a target image Q, possibly containing an instance of a similar object we find a correspondence between the images as follows:

1. Extract sparse oriented edge maps from each image.

2. Compute features based on geometric blur descriptors at locations with high edge energy.

3. Allow each of m feature points from P to potentially match any of the k most similar points in Q based on feature similarity and or proximity.

4. Construct cost matrices H and c as in Section 5.3.

5. Approximate the resulting Binary Quadratic Optimization to obtain a correspondence. Store the cost of the correspondence as well.

6. Extend the correspondence on m points to a smooth map using a regularized thin plate spline [Powell, 1995].

See Figures 5.1 and 5.4 for a number of examples. In the leftmost column of the figures is the image, P, shown with m points marked in color. In the middle left column is the target image Q with the corresponding points found using our algorithm. A regularized thin plate spline is fit to this correspondence to map the full set of feature points on the object in P, shown in the middle right column, to the target, as shown on the far right column. Corresponding points are colored the same and points are colored based on their position (or corresponding position) in P – in P colors are assigned in uniform diagonal stripes, the distortion of these striped in the far right column of the figure gives some idea of the distortion in the correspondence.

5.6 Recognition Experiments

Our recognition framework is based on nearest neighbors.

Preprocessing: For each object class we store a number of exemplars, possibly replicated at multiple scales, and compute features for all of the exemplars. Features are only computed on the support of the objects. At this point object supports are marked by hand. The following chapter shows how to find them automatically.

Indexing: Extract features from a query image. For each feature point in an exemplar, find the best matching feature point in the query based on normalized

correlation of the geometric blur descriptors. The mean of these best correlations is the similarity of the exemplar to the query. We form a shortlist of the exemplars with highest similarity to the query image.

Correspondence: Find a correspondence from each exemplar in the shortlist to the query as described above. Pick the exemplar with the least cost.

We address two object recognition problems, multiclass recognition and face detection. In the multiple object class recognition problem, given an image of an object we must identify the class of the object and find a correspondence with an exemplar. We use the Caltech 101 object class dataset consisting of images from 101 classes of objects: from accordion to kangaroo to yin-yang, available at [cal,]. This dataset includes significant intra class variation, a wide variety of classes, and clutter. On average we achieve **48%** accuracy on object classification with quite good localization on the correctly classified objects. This compares favorably with the original paper on this dataset producing 16% [Fei-Fei *et al.*, 2004].

We also consider face detection for large faces, suitable for face recognition experiments. Here the task is to detect and localize a number of faces in an image. The face dataset we use is sampled from the very large dataset used in [Berg *et al.*, 2004] consisting of news photographs collected from yahoo.com. With only 20 exemplar faces our generic system provides a ROC curve with slightly better generalization, and slightly worse false detection rate than the quite effective specialized face detector of Mikolajczyk [Mikolajczyk, 2002] used in [Berg *et al.*, 2004].

We apply our technique to two different data sets, the Caltech set of 101 object categories (available here [cal,]) and a collection of news photographs containing faces gathered from yahoo.com (provided by the authors of [Berg *et al.*, 2004]). In the experiments that follow, we utilize the same parameters for both datasets except for those specifically mentioned.

For all images edges are extracted at four orientations and a fixed scale. For the

Caltech dataset where significant texture and clutter are present, we use the boundary detector of [Martin *et al.*, 2004] at a scale of 2% of the image diagonal. With the face dataset, a quadrature pair of even and odd symmetric Gaussian derivatives suffices. We use a scale of $\sigma = 2$ pixels and elongate the filter by a factor of 4 in the direction of the putative edge orientation.

Geometric blur features are computed at 400 points sampled randomly on the image with the blur pattern shown in Figure 4.3. We use a maximum radius of 50 pixels (40 for faces), and blur parameters $\alpha = 0.5$ and $\beta = 1$.

For correspondence we use 50 (40 for faces) points, sampled randomly on edge points, in the correspondence problem. Each point is allowed to match to any of the most similar 40 points on the query image based on feature similarity. In addition for the Caltech 101 dataset we use $\gamma = 0.9$ allowing correspondences with significant variation in scale, while for the faces dataset we handle scale variation partly by repeating exemplars at multiple scales and use $\gamma = 0.5$.

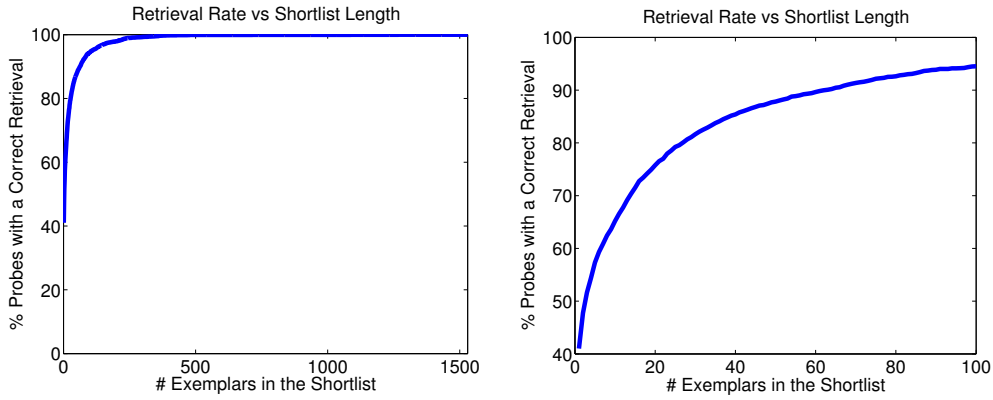


Figure 5.2: *For a probe or query image exemplars are ranked according to feature similarity. We plot the percentage of probes for which an exemplar of the correct class was found in the shortlist. Here the first exemplar is correct 41% of the time.* **Left** *Full curve.* **Right** *Curve up to shortlist length 100 for detail.*

5.7 Caltech 101 Results

Basic Setup: Fifteen exemplars were chosen randomly from each of the 101 object classes and the background class, yielding a total 1530 exemplars. For each class, we select up to 50 testing images, or "probes" excluding those used as exemplars. Results for each class are weighted evenly so there is no bias toward classes with more images.

The spatial support of the objects in exemplars is acquired from human labeling. The top entry in the shortlist is correct 41% of the time. One of the top 20 entries is correct 75% of the time. (Figure 5.2). [2]

Recognition and localization: Using each of the top ten exemplars from the shortlist we find a good correspondence in the probe image. We do this by first sampling 50 locations on the exemplar object and allowing each to be matched to its 50 best matching possibilities in the probe with up to 15% outliers. This results in a quadratic programming problem of dimension 2550. We use a distortion cost based mainly on the change in angle of edges between vertices ($\gamma = 0.9$). This allows matches with relatively different scales (Figure 5.4 line 3). The exemplar with the lowest distortion correspondence gives **48%** correct classification, at the same time providing localization. Note that this is using a simple nearest neighbor classifier and generative models. A baseline experiment comparing grayscale images using SSD and 1-nearest neighbor classification gives 16%. At press, the best results from the Caltech group are 40% using discriminative methods [**?**]. No other techniques have addressed correspondence at the level of detail presented here.

Multiscale: We compute exemplar edge responses and features at a second scale for each exemplar resulting in twice as many exemplars. This improves shortlist perfor-

[2] We note that these results are on the Caltech 101 dataset as presented in 5.7, which contains some duplicates. Using the currently available dataset [cal,] which has no duplicates the performance drops by approximately 3% across all experiments, in this case to 38% and 72% respectively. For the recognition results using correspondence performance drops from 48% with duplicates to 45% without duplicates.

mance by 1% or less, and does not change recognition performance. This illustrates the lack of scale variation in Caltech 101. The face dataset exhibits a large range of scale variation.

5.8 Face Detection Results

We apply the same technique to detecting medium to large scale faces for possible use in face recognition experiments. The face dataset is sampled from the very large dataset in [Berg *et al.*, 2004] consisting of A.P. news photographs. A set of 20 exemplar faces split between front, left, and right facing, was chosen from the database by hand, but without care. The test set was selected randomly from the remaining images on which the face detector of [Mikolajczyk, 2002] found at least one 86×86 pixels or larger face. We use the generic object recognition framework described above, but after finding the lowest cost correspondence we continue to look for others. A comparison of the ROC curves for our detector and that of [Mikolajczyk, 2002] is found in Figure 5.3. Our detector has an advantage in generalization, while producing more false positives. While not up the the level of specialized face detectors, these are remarkably good results for a face detector using 20 exemplars and a generative model for classification, without any negative training examples.

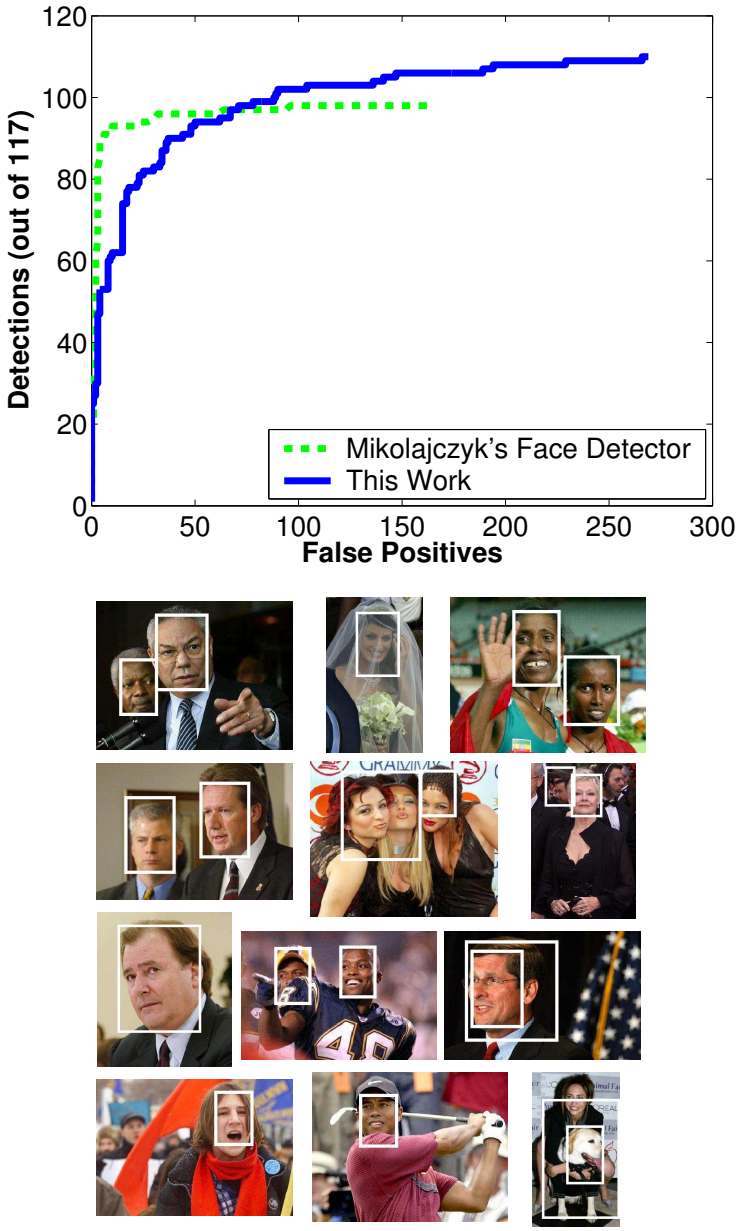


Figure 5.3: **Top** *ROC curves for our face detector using 20 exemplar images of faces (split between frontal and profile) and the detector of Mikolajczyk. Mikolajczyk's detector has proven to be effective on this dataset. simply finding sets of feature points in an image that have a good correspondence, based on distortion cost, to an exemplar. Good correspondences allow detection and localization of faces using a simple generative model, no negative examples were used.* **bottom** *Detections from our face detector marked with rectangles.*

Figure 5.4: *Each row shows an alignment found using our technique described in section 5.4. Leftmost is an exemplar with some feature points marked. Left center is a probe image with the correspondences found indicated by matching colors (all possible feature matches are shown with white dots). All of the feature points on the exemplar are shown center right, and their image using a thin plate spline warp based on the correspondence are shown in the right most image of the probe. Note the ability to deal with clutter (1,6), scale variation(3), intraclass variation all, also the whimsical shape matching (2), and the semiotic difficulty of matching a bank note to the image of a bank note painted on another object (5).*

Chapter 6

Models of Variation

6.1 Introduction

Given an engine for correspondence between objects we can build a model for the variation in an object's configuration and appearance. This process is largely unconscious to humans. When we identify someone by saying, "Jill has somewhat pointy ears." there is no thought given to the process of aligning everyone's head, building a model of pointiness for the protruding bit on the side, aligning Jill's head to the model and seeing that yes indeed, she does have somewhat pointy ears.

As a proof of concept experiment we will attempt to model the variation in images of a category of object. The result is a process for taking a number of unsegmented photographs of instances of a category of object and segmenting out the commonly occurring object.

6.2 Approach

The procedure for doing this is quite simple. An image is considered as a reference image. We find a matching from this reference image to each of other images of

that object category. Given these matchings we identify how well each part of the reference image matches the other images *after alignment*. This is a model of the variation in the appearance of that part of the image after alignment. It is somewhat subtle, but the object being modeled is the image and the part of the image that is consistently aligned is the instance of the object category in the image. It would then be possible to repeat the process using the segmented object and model its variation over instances.

6.3 Experiment

In the alignment and recognition experiments in Chapter 5, exemplar objects were hand segmented from their backgrounds. We now show how this can be automated by finding the repetitive aspects of objects in the example images. Ideally this would be computed for all images simultaneously. We show that in many cases it is sufficient to find the similar parts in pairs of images independently.

Starting with a set of example images $\{I_i\}$ from an object class find the support of the object in an image I_{i_0} as follows. For each image I_j where $j \neq i_0$:

1. Find a correspondence from I_{i_0} to I_j[1].
2. Use a regularized thin plate spline to map all of the feature points in I_{i_0} to I_j.
3. For each mapped feature from I_{i_0}, the quality of the match is the similarity to the best matching nearby feature in I_j. The median quality of match for a feature is the measure of how common that feature is in the training images.

Feature points with median quality within 90% of the best for that image are considered part of the object. Repeating the recognition experiments from Chapter 5,

[1]Here we allow 40% outliers instead of 15% as used in the recognition experiments.

the shortlist accuracy improves by 1-4% (Fig. 6.1). While the estimated support is usually not perfect, recognition performance is similar to that using hand segmented images, 48%. The process is depicted in Figure 6.1.

The learned models of support reflect a region of the image that is consistent across training images, as opposed to individual discriminative features. For instance the cheek on a face is not by itself discriminative for faces, but when considering faces transformed into alignment the cheek is usually consistent.

6.4 Discussion

The procedure presented is an example of using correspondence to build a model of variation. It contrasts with almost all current models used for object recognition which find local features or combinations of local features which are discriminative for a class. As an example for a face, the eyes and mouth would be locally discriminative. Here by looking for alignments of whole objects we can identify that the entire face region is consistent across the set of images.

Still, this is proof of concept procedure is probably not the optimum solution to automatically segmenting objects in images. In particular this is a purely generative approach relying on only positive examples, and relying on a sufficient amount of variation in the background. One problematic example is the "automobiles from the side" images. The vast majority of these images were taken of parked cars from across the street. The result is that almost the entire image aligns well when any two images are compared. This technique can be extended to a more discriminative model where not only should positive examples match well, but negative examples should match poorly.

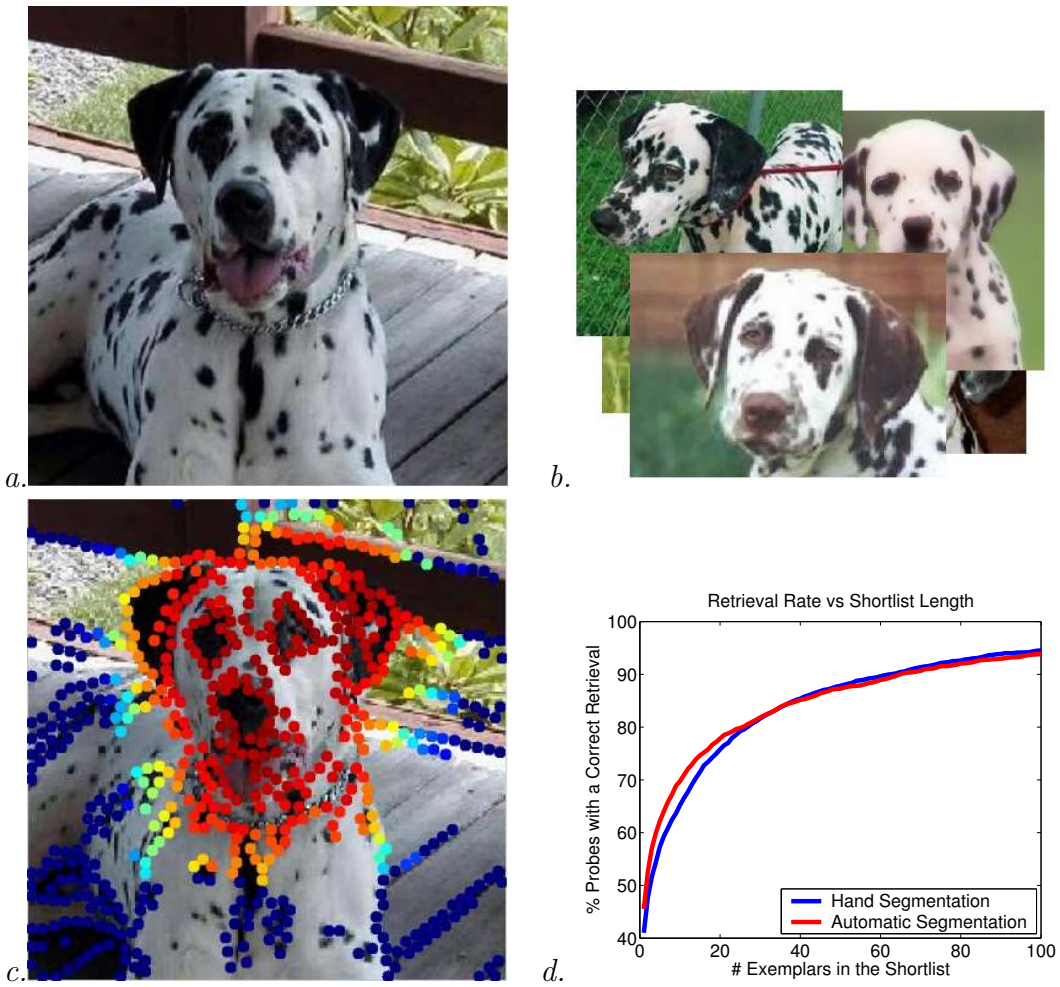


Figure 6.1: *Illustrating automatic model segmentation: One training image* (**a.**) *the remaining 14 training images* (**b.**) *colors indicate how well on average feature points match after aligning transforms to each of the other training images* (**c.**) *At lower right, the percentage of probes for which an exemplar of the correct class was found in the shortlist. The* **blue** *curve shows performance with hand segmented exemplars, the* **red** *curve shows performance with* automatically *segmented exemplars. For hand segmented exemplars the first exemplar is correct 41% of the time, for automatically segmented exemplars 45%.* (**d.**)

Bibliography

[Amit *et al.*, 1997] Y. Amit, D. Geman, and K. Wilder. Joint induction of shape features and tree classifiers. *IEEE Trans. PAMI*, 19(11):1300–1305, November 1997.

[Belongie *et al.*, 2001] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. In *ICCV*, pages I.454–461, 2001.

[Berg and Malik, 2001] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, pages 607–614, 2001.

[Berg *et al.*, 2004] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y. W. Teh, E. Learned-Miller, and D. A. Forsyth. Names and faces in the news. In *CVPR*, pages 848–854, 2004.

[Berg *et al.*, 2005] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, pages 26–33, 2005.

[cal,] Caltech 101 dataset
www.vision.caltech.edu/feifeili/101_ObjectCategories .

[Chui and Rangarajan, 2003] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. *CVIU*, 89:114–141, 2003.

[Efros *et al.*, 2003] A.A. Efros, A.C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. 9th Int. Conf. Computer Vision*, volume 2, pages 726–733, 2003.

[Fei-Fei *et al.*, 2003] L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *ICCV*, pages 1134–1141, 2003.

[Fei-Fei *et al.*, 2004] 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 *CVPR, Workshop on Generative-Model Based Vision*, 2004.

[Fergus *et al.*, 2003] R. Fergus, P. Perona, and A Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, pages 264–271, 2003.

[Fischler and Elschlager, 1973] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computers*, C-22(1):67–92, 1973.

[Gavrila and Philomin, 1999] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. 7th Int. Conf. Computer Vision*, pages 87–93, 1999.

[Grenander *et al.*, 1991] U. Grenander, Y. Chow, and D.M. Keenan. *HANDS: A Pattern Theoretic Study Of Biological Shapes*. Springer, 1991.

[Holub *et al.*, 2005] A. Holub, M. Welling, and P. Perona. Combining generative models and fisher kernels for object recognition. In *ICCV*, pages 136– 143, 2005.

[Huttenlocher *et al.*, 1993] D.P. Huttenlocher, G. Klanderman, and W. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Trans. PAMI*, 15(9):850–863, Sept. 1993.

[Lades *et al.*, 1993] M. Lades, C.C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Computers*, 42(3):300–311, March 1993.

[Lamdan *et al.*, 1990] Y. Lamdan, J.T. Schwartz, and H.J. Wolfson. Affine invariant model-based object recognition. *IEEE Trans. Robotics and Automation*, 6:578–589, 1990.

[Leung *et al.*, 1995] T.K. Leung, M.C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. 5th Int. Conf. Computer Vision*, pages 637–644, 1995.

[Lowe, 1999] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[Lowe, 2004] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[Maciel and Costeira, 2003] J. Maciel and J Costeira. A global solution to sparse correspondence problems. *PAMI*, 25(2):187–199, 2003.

[Martin *et al.*, 2004] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *PAMI*, 26(5):530–549, 2004.

[Mikolajczyk and Schmid., 2003] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, pages 257–263, 2003.

[Mikolajczyk, 2002] K. Mikolajczyk. *Detection of local features invariant to affines transformations*. PhD thesis, INPG, 2002.

[Mori *et al.*, 2001] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *CVPR*, volume 1, pages 723–730, 2001.

[Mori *et al.*, 2005] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *PAMI*, 27(11):1832–1837, 2005.

[Morrone and Burr, 1988] M. Morrone and D. Burr. Feature detection in human vision: A phase dependent energy model. *Proc. Royal Soc. of London B*, 235:221–245, 1988.

[Palmer *et al.*, 1981] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. In J. Long and A. Baddeley, editors, *Attention and Performance*, volume 9, pages 135–151. Hillsdale, NJ: Erlbaum, 1981.

[Palmer, 1975] S. Palmer. Visual perception and world knowledge: Notes on a model of sensory cognitive interaction. In D. A. Norman and D. E. Rumelhart, editors, *Explorations in cognition*, page 279307. San Francisco: W. H. Freeman, 1975.

[Piotrowski and Campbell, 1982] J. N. Piotrowski and F. W. Campbell. A demonstration of the visual importance and flexibility of spatial-frequency amplitutde and pahse. *Perception*, 11(3):337–346, 1982.

[Powell, 1995] M. J. D. Powell. A thin plate spline method for mapping curves into curves in two dimensions. In *CTAC*, Melbourne, Australia, 1995.

[Ren *et al.*, 2005] X. Ren, C. Fowlkes, and J. Malik. Mid-level cues improve boundary detection. Technical Report 05-1382, U.C. Berkeley, Computer Science Division, 2005.

[Rosch, 1973] E. Rosch. Natural categories. *Cognitive Psychology*, 4(3):328–350, 1973.

[Rothganger *et al.*, 2003] F. Rothganger, S. Lazebnik, C Schmid, and J Ponce. 3d object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *CVPR*, pages II:272–275, 2003.

[Schmid and Mohr, 1997] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Trans. PAMI*, 19(5):530–535, May 1997.

[Schneiderman and Kanade, 2000] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR*, pages 746–751, 2000.

[Schneiderman, 2004] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection. In *CVPR*, pages 29–36, 2004.

[Slater and Morison, 1985] A. Slater and V. Morison. Slate constancy and slant perception at birth. *Perception*, 12:707–718, 1985.

[Thouless, 1931] R. H. Thouless. Phenomenal regression to the real object. *British Journal of Psychology*, 21:339–359, 1931.

[Tootell *et al.*, 1982] R. B. H. Tootell, M. S. Silverman, E. Switkes, and R. L. De Valois. Deoxyglucose analysis of retinotopic organization in primate striate cortex. *Science*, 220:737–739, 1982.

[Torralba *et al.*, 2004] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, pages 762–769, 2004.

[Ullman *et al.*, 2002] S. Ullman, M. Vidal-Naquet, and E Sali. Visual features of intermediate complexity and their use in classification. *Nat. Neur.*, 13:682–687, 2002.

[Viola and Jones, 2001] P. Viola and M. Jones. Robust real-time object detection. *2nd Intl. Workshop on Statistical and Computational Theories of Vision*, 2001.

[Zhang, 2005] H. Zhang. Pers. comm., Dec. 2005.