Geometric Blur for Template Matching

Alexander C. Berg Jitendra Malik

Computer Science Division
University of California, Berkeley

Abstract

We address the problem of finding point correspondences in images by way of an approach to template matching that is robust under affine distortions. This is achieved by applying "geometric blur" to both the template and the image, resulting in a fall-off in similarity that is close to linear in the norm of the distortion between the template and the image. Results in wide baseline stereo correspondence, face detection, and feature correspondence are included.

1. Introduction

Many problems in computer vision involve as a key subroutine solving a version of the correspondence problem. In the case of stereo and long range motion, the goal is to find corresponding points that are projections, in the different views, of the same point in 3D space. Many approaches to object recognition also require solving the correspondence problem, in this case of a fiducial point on a 3D model or a 2D view, to the "same" point in an input. Typical approaches to solving the correspondence problem are based on some form of template matching, comparing image windows centered at the two potentially corresponding points. It is widely recognized that in spite of the considerable effort that has been devoted to this problem, we do not yet have a satisfactory solution.

In this paper we discuss our formulation of a template matching technique with the explicit goal of finding point correspondences in the presence of geometric distortion. Intuitively geometric distortion is the change in relative position of features whose local appearance is unchanged. As a concrete example, features could be intensity edges, and the geometric distortion could be due to change in pose.

A key component of our technique is that the image is first decomposed into channels of feature responses before comparison. This allows uncertainty about the position of the features to be separated from uncertainty about the appearance of the features. For the examples in this paper we will deal with edges and other features related to oriented edge energy. This component of the technique is itself generally useful in its own right.

1.1 Related Work

We address work related to making template matching robust to distortion, as well as work on matching/recognition using features.

Matching image patches is traditionally done by comparison using SSD or some variant such as normalized cross correlation. It is well understood that the effectiveness of such measures will degrade with significant viewpoint or illumination changes. This leaves two alternatives for the problem of finding the sub-window V of an image that best matches a template window W:

1. Simultaneously estimate the distortion T^*, and subwindow of the target image V^* so that $(V^*, T^*) = \text{argmin}_{(V,T)} d(V \circ T, W)$, where T is a geometric transform, usually bounded to lie in some reasonable range, and $d(\cdot, \cdot)$ is a distance-like function.

2. To "blur" the template window, performing matching in a coarse to fine way. This approach has traditionally been used in stereo and motion settings. [2]

Solution 1 above is computationally very expensive, as a result solution 2 is commonly used. One particular version of the latter approach that has been tried in object recognition settings is to train a classifier with a large number of distorted views, and rely on the generalization capabilities of the classifier to blur in an optimal manner.

Matching gray-level image windows is not the only option, though over time it has emerged as the preferred approach for finding matching points across multiple views. In the context of object recognition, where one has to allow for illumination variation in addition to viewpoint variation, there are several approaches using edge detection as a first step. Examples are template matching based on distance transforms or chamfer distance [7] and shape contexts[8]. In this work we expand the notion of features beyond edges, and explicitly formulate a notion of geometric blur for robust template matching.

1.2 Our Approach

In this paper we will try to build a generic image matching engine equally applicable to a range of tasks:

1. Long range motion, finding corresponding points in images of an object from significantly different view points, either as a result of camera or object motion. See Figure 10 for an example of very wide baseline stereo.

2. Object detection. Here the objective is to determine if an object similar to a training object is present in an image, and extract its position. See Figure 9 for an example of detecting faces.

3. Finding corresponding points on different objects. Here the objective is to find corresponding points on different objects, where unlike wide baseline stereo correspondence we are concentrating on change due to variation in the object itself. See Figure 8 for an example of finding features on faces.

In all three of these settings, matching has to proceed by considering image windows around putative corresponding points and computing some measure of similarity between these windows [1]. The challenge is in making the windows "discriminative" and the matching "robust". If windows are small, then they are not discriminative–not enough context is captured. If windows are large, then the total change in the windows is large from different camera views. In addition to the distortion due to varying perspective, there could be variation between different examples of a category, as in the case of object recognition. These problems have long been recognized in the context of stereo matching [6].

We will consider the following model for geometric distortion in images, where the observed image J is a function of the original image under some distortion T with some noise N, all over image coordinates x.

$$J(x) = I \circ T(x) + N(x)$$

In order to match windows robust to geometric distortions, the standard strategy in computer vision is to adopt a pyramid-like coarse-to-fine approach [2]. At a coarse scale of the pyramid, the image is a blurred version (typically by convolution with a Gaussian in the image intensity domain, possibly subsampled) of the image at a fine scale. This introduces positional blur uniformly at a given level – all the pixels in a window centered at a feature point have been made positionally uncertain by an amount related to the σ of the Gaussian. This is not quite the "right" thing to do if we are interested in finding point correspondences. If we let the putative corresponding points be at the center of the windows, then there is zero positional uncertainty with the central pixel of the window, and increasing positional uncertainty associated with more peripheral features. By introducing a uniform positional blur, one is simultaneously introducing *more* positional uncertainty than necessary for the central region of the window and perhaps *less* positional uncertainty than appropriate for the peripheral regions of the window. The main point of this paper is to develop a notion of "geometric blur" which takes this into account in the right way. Gaussian blur is "image blur", the right way to simplify image intensity structure and de-noise an image [9] (under certain assumptions) but is not designed with the criterion of making matching points robust under geometric distortions.

In section 2, we explain the motivation behind geometric blur in a simple 1-D setting where affine transforms reduce to dilation followed by shifts. In section 3, we develop the idea in the context of matching 2D windows and demonstrate the superiority of our geometric blur technique over Gaussian blur for 2D window matching under distortion. In section 4, results on real images for a variety of matching applications are shown.

2. Geometric Blur in 1D

A simple example using 1D signals motivates a spatially varying blur. The template $I(x)$ is a 1D signal consisting of the sum of three delta functions, as shown in Figure 1. The problem is to compare the template signal to a spatially dilated (or spatially scaled) version of the template $J(x) = I(x/a)$. Clearly the correlation at zero offset between the two is zero for almost all dilations. The typical solution is to apply a uniform Gaussian blur to the signals resulting in a gradual decrease in correlation at zero offset as dilation increases. In the figure we use a box filter for simplicity, but Gaussians give qualitatively similar results. Here the problem with uniform blurring becomes evident. Because the dilation is much more apparent farther from the origin, the signal is simultaneously "over blurred" near the origin, and "under blurred" away from the origin. As a result the correlation will have a varying sensitivity to dilation for signals of different scales. We propose the alternative, *geometric blur*, also shown in the Figure 1. Here the amount of blur is proportional to the distance from the origin. As shown, the change in correlation at zero offset is now linear in the amount of dilation, and deals with the components of the signal close to the origin consistently with those far from the origin.

In the rest of this section we will define geometric blur and make the linearity argument precise in 1D, including discussion of how the geometric blur can be calculated, and how to deal with translation.

[1] Additional constraints may be available from geometry (e.g. epipolar geometry, 3d object models) and should be exploited when possible.

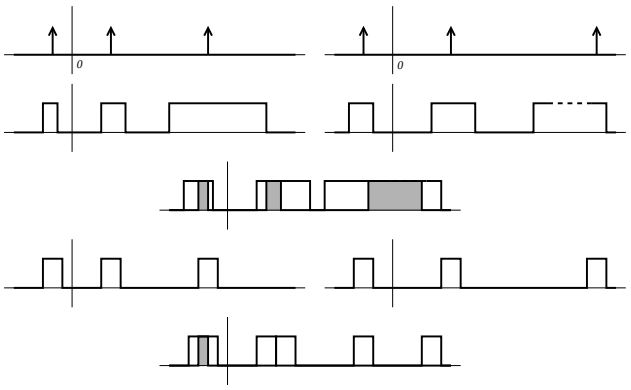

Figure 1: Top row: The right is a dilated version of the signal on the left. Second row: the signals after geometric blur. Third row: The proportional overlap of the signals after geometric blur is 1 minus a term proportional to the dilation, unlike uniform blur (second to bottom row) where the proportional overlap is dependent on the signal and the dilation in a highly non-linear manner.

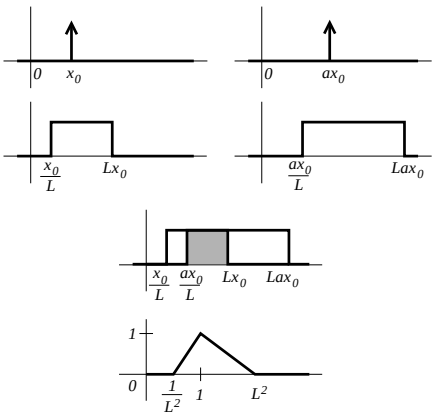

Figure 2: Top Row: A signal of a single impulse and a dilated version of the same signal, Second Row: The geometric blur of each signal, Third Row: Overlay of the two signals after geometric blur. The shaded region contributes to the correlation at zero offset. Fourth Row: Relative overlap as a function of dilation.

2.1. Definition of Geometric Blur

The geometric blur $G_I(x)$ of a signal $I(x)$ over coordinate x is defined to be:

$$G_I(x) = \int_{T \in \mathcal{T}} I(T(x))dT$$

Where the geometric distortion T is contained in some set $\mathcal{T}$ of bounded transforms. We will usually restrict $\mathcal{T}$ to be a subset of linear transformations and write Tx in place of $T(x)$. The implications of this definition are covered below and at the end of the section we discuss computation. The measure for the integral is discussed later.

2.2. Linearity in 1D

As a concrete example, consider the signal $I(x) = \delta_{x_0}(x)$ for some $x_0 > 0$ in 1D, as shown in Figure 2. Let $J(x) = \delta_{ax_0}(x)$ be a geometric distortion of I, where the distortion is a scale by a factor of $a > 0$. Now consider the geometric blur of these two signals for scaling transforms $t \in [\frac{1}{L}, L]$, for some $L > 1$. This gives us the geometric blur G_I for I as

$$\begin{aligned} G_I(x) &= \int_{t \in [\frac{1}{L}, L]} I(tx)dt \\ &= \chi_{[\frac{x_0}{L}, Lx_0]}(x) \end{aligned}$$

and similarly the geometric blur for J is

$$G_J(x) = \chi_{[\frac{ax_0}{L}, Lax_0]}(x)$$

We now define the first of two normalizations for the comparison of two geometric blurs:

$$\tilde{C}(G_I, G_J) = \frac{1}{||G_I||_1} \int_x G_I(x) G_J(x) dx$$

Using the example above we have:

$$\tilde{C}(G_I, G_J) = \begin{cases} \max(1 - \frac{a-1}{L^2-1}, 0) & 1 \leq a \\ \max(1 + \frac{a-1}{1-\frac{1}{L^2}}, 0) & 0 \leq a \leq 1 \end{cases} \quad (1)$$

The important fact here is that $\tilde{C}(G_I, G_J)$ is independent of x_0. Furthermore $\tilde{C}(G_I, G_J)$ falls off linearly in the change of the dilation a, as shown in Figure 2.

We can extend the above example to signals of the form $I(x) = \sum_i \delta_{x_i}(x)$ if the x_i are sufficiently separated so that the geometric blurs $\int_t \delta_{x_i}(tx)$ are disjoint. As the signal becomes dense, or the range of t becomes large, the blurs overlap, and linearity no longer holds. However until the signal becomes very dense the behavior is still somewhat linear.

At this point we define the second normalization for comparisons of geometric blur:

$$\hat{C}(G_I, G_J) = \frac{1}{||G_I||_2 ||G_J||_2} \int_x G_I(x) G_J(x) dx$$

This normalization gives better results for signals that are somewhat dense (see Figure 3). Unlike $\tilde{C}$, $\hat{C}$ does not fall off linearly in the change of dilation, but it is still independent of x_0 in the example above, and in practice has fall-off close to linear with respect to the change in dilation.

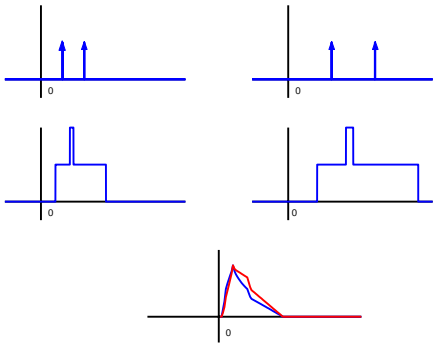

Figure 3: Top row: A signal and a dilated version of the signal. Second Row: Geometric blur of both signals. Bottom: Red line is comparison over varying dilation using $\tilde{C}$, Blue line (lower) is comparison using $\hat{C}$. The peak is at no dilation and the falloff of both is close to linear in the change in dilation.

2.3. Computation of Geometric Blur

We show that the potentially troubling computation of the geometric blur as an integral over an infinite family of distorted versions of the signal can be rewritten as a convolution with a spatially varying kernel. The resulting formulation allows straightforward approximation.

We first rewrite the integral in the definition of geometric blur

$$G_I(x) = \int_T I(T(x))dT$$

as

$$G_I(x) = \int_y I(y)\mu(\{T : T(x) == y\})dy$$

and finally as

$$G_I(x) = \int_y I(x-y)K_x(y)dy$$

Where $K_x(y)$ is a spatially varying kernel. Notice that we have replaced the integral over T lying in a family of geometric distortions with an integral over y lying in the coordinate space of the signal. The complexity has been transfered to $K_x(y)$, which depends on x, y, and the measure we use for T.

The advantage of rewriting the computation of G_I in this manner is that we can approximate the infinite family $K_x(y)$ by a discrete set of functions $\{K_{x_i}(y)\}$ corresponding to a set of $\{x_i\}$. We define $n(x) = \text{argmin}_i|x - x_i|$, and we have $G_I(x) = \int_y I(x-y)K_{x_{n(x)}}(y)dy$.

We can precompute $F_{I,K}(x,i) = \int_y I(x-y)K_{x_i}(y)dy$ and then "select" the values contributing to G_I, by simply setting: $G_I(x) = F_{I,K}(x, n(x))$. When finding the best match for a template in an image, this means that each level of blur can be compared separately using convolutions.

Concretely, consider the example from the previous subsection.

$$\begin{aligned} G_I(x) &= \int_t I(tx)dx \\ &= \int_{t\in\left[\frac{1}{L},L\right]} \delta_{x_0}(tx)dx \\ &= \int_y \delta_{x_0}(x-y)\chi_{\left[\frac{(1-L)x}{L},(L-1)x\right]}(y)dy \end{aligned}$$

Here $K_x(y)$ is $\chi_{\left[\frac{(1-L)x}{L},(L-1)x\right]}(y)$. In this case computing $F_{I,K}(x,x_i)$ simply amounts to computing increasingly blurred versions of $I(x)$, and finding the geometric blur, $G_I(x) = F(x, n(\alpha x))$ simply selects how blurry the signal at location x should be. A large value for α will result in more blur, and a small value for α will result in less blur. For $\alpha = 0$, we obtain $G_I(x) = F(x,0) = I(x)$.

3. Geometric Blur in 2D

Geometric blur as defined above for 1D extends directly to 2D. For a signal $I(x)$ the geometric blur of the signal is $G_I(x) = \int_T I(T(x))dT$ where x now varies over two dimensions, the image coordinates, and T varies over some bounded range of transforms. We will usually consider linear transforms, and will write Tx in what follows.

Since the image coordinate x varies in 2 dimensions, the family of kernels $K_x(y)$ for the geometric blur could be two dimensional. In order to simplify computation we purposely choose a range of transforms $\mathcal{T}$, or equivalently a measure on the space of transforms that results in $K_x(y)$ depending only on $|x|$. This means that the kernel functions are rotationally symmetric. By biasing the measure on T appropriately the resulting kernel functions can be taken to be simple Gaussians. For symmetric kernels we can adjust the blur just as we did in our 1D example by adjusting a parameter α where $G_I(x) = F_{I,K}(x, n(\alpha x))$. See figure 4 for an example of Geometric Blur.

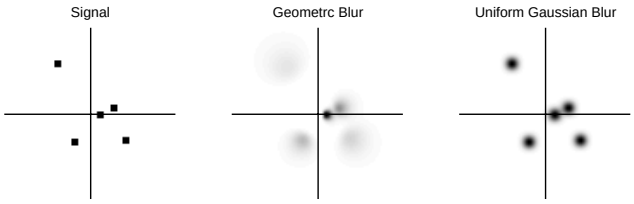

Figure 4: Geometric Blur with a spatially varying Gaussian kernel blurs more farther from the origin.

3.1 Results on Synthetic Data

We present results on synthetic data that demonstrates the behavior and capability of geometric blur.

In order to see that geometric blur helps in discrimination 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 being turned on with probability 5%. Figures 5 and 6 show the mis-classification 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.

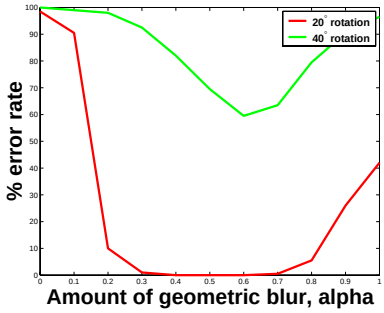

Figure 5: Identifying 200 random test images after rotation, using various amounts (α) of geometric blur.

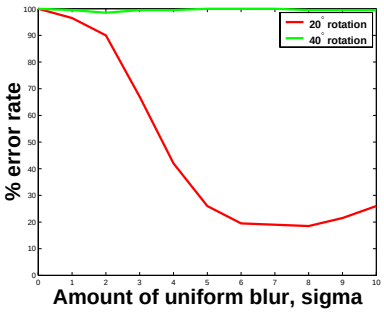

Figure 6: Identifying 200 random test images after rotation, using various amounts (σ) of uniform Gaussian blur.

3.2. Pre-processing to obtain Sparse Signals

Geometric blur is most effective when applied to sparse signals. Images when considered as 2D brightness signals are not sparse. However much work indicates that oriented edge filter responses from images are sparse [4] [5]. Also the formulation and theoretical results about geometric blur so far have assumed a non-negative signal. To meet the sparseness and non-negative requirements when considering real images, we break images up into a number of channels. Each channel is a half-wave rectified oriented edge response. In particular if $E(x)$ is a filter then two channels would be $C_1(x) = [I(x)E(x)]\,\chi_{[I(x)E(x)>0]}$ and $C_2(x) = -[I(x)E(x)]\,\chi_{[I(x)E(x)<0]}$. We also use a contrast normalization on the channels [3]. In particular if $C = [C_1(x) \ldots C_n(x)]$ is a vector of channel values at x, then the normalized version would be $\frac{1}{|C|_2+\epsilon}$ where we use an $\epsilon = 0.3$ for filters with response between +1 and -1. Figure 7 shows an image and a set of 12 channels resulting from 6 oriented edge filters.

One useful consequence of treating the positive and negative components of oriented edge responses separately is that information about zero crossings is not lost under blurring. Instead of blurring the signal response around a zero crossing to zero, the positive and negative responses are both blurred over the area, retaining the information that there was a zero crossing, but allowing uncertainty as to its position.

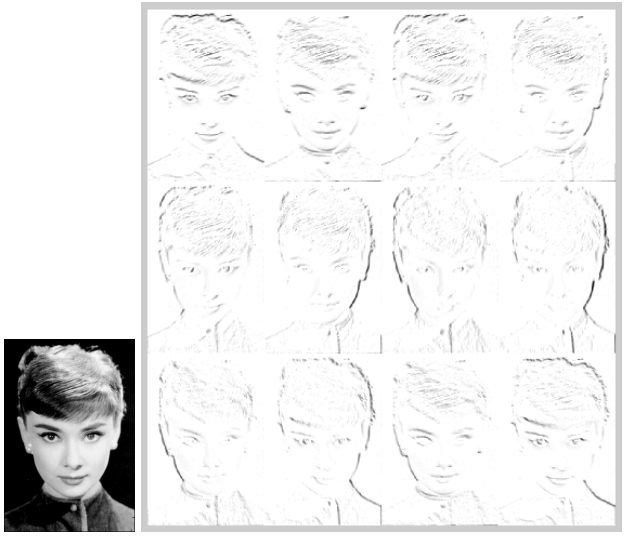

Figure 7: The twelve half-wave rectified channels contrast normalized from the response of 6 oriented edge filters on the image. White indicates zero, and black indicates a positive value. Note that the filter responses are sparse, making the individual channels appropriate for geometric blur

4. Results on Real Images

We now present results on real images in three tasks: automatic correspondence for wide base-line stereo, correspondence to a labeled template, and object detection.

4.1. Wide Base-line Stereo

Using the images shown in Figure 10, in one image an interest operator based on corners detects the 21 labeled interest points and we attempt to find a matching point in the other image. For geometric blur the images are split into channels as described above, and a template is taken around each of the feature points. Matching is done by adding together normalized cross correlation using geometric blur from each of the channels. We compare this with normalized cross correlation on the pixel intensities. Only 15 of the interest points in the original image have obvious correspondences in the other image, so we show only the 15 matches with the best matching score for each technique. The best 15 matches for Geometric blur are correct, using windows of about 1/3 the size of the entire image to provide context. Normalized cross correlation fares much worse using similarly large windows, and the poor localization for the three close to correct correspondences show the standard problem with large windows that geometric blur avoids. With smaller windows normalized cross correlation matches four points correctly, but the other 11 top matches are wrong. Notice that a number of these are corners from the bridge tower matched to incorrect corners of the bridge tower, this points to the lack of context from small windows. Using SSD instead of normalized cross correlation results in similarly bad matches. Also, although the results for geometric blur use a number of filtered channels instead of the raw gray scale value, using the channels does not improve the results of SSD or normalized cross correlation on this example. Finally note that even though the differences between the two images are clearly not an affine distortion (consider the region around one of the feature points near the span of the bridge), geometric blur performs very well, and makes the proper trade-off between local information at fine scale, and more global information at a coarse scale.

4.2. Feature Correspondence

Another application is to find features given a set of masked templates. In particular detailed faces are masked by hand, and various fiducial points on the faces are selected. For each feature point geometric blur was used to find the best match in target image. The support of the region used for comparing the geometric blur of the template and target image was fixed by the mask in the template. Results on a somewhat difficult instance are shown in Figure 8.

4.3. Object Detection

As a special case of the feature correspondence we take a single feature on the face and use this to perform face detection using geometric blur. Here a set of face templates is selected by hand, along with a mask for the extent of the face, and an origin (the nose) for distortion. Figures 9 show the results on images from the Schneiderman Kanade [1] test set for two example templates. In each case detections are noted where the correlation after geometric blur is above some threshold. A single threshold is used for all the images.

Figure 8: Left: Feature points selected by hand on a masked off image of a face. For each selected point in the left image, a template is selected around that point, and the best match for that template is found in the right image using geometric blur. There is no consistency required between points. The template for each point contains the entire face, but is blurred differently depending on which feature the template represents.

5. Conclusion

We address the problem of using templates to find point correspondences. Given this goal uniform Gaussian blur is not correct. Blur should be small near the corresponding points, and larger away from them. We define geometric blur in terms of a set of geometric distortions on an image. If we choose to model distortion with affine transformations, then the amount of blur varies linearly with distance from corresponding points.

We apply geometric blur to synthetic and real images for recognition and correspondence tasks. For real images we first break the image up into channels based on half-wave rectified responses from oriented edge filters. Then the geometric blur is applied to these feature channels separately. One difference from Chamfer distance methods that also use features and blur, is that geometric blur has a concept of point correspondence, and allows spatially varying amounts of distortion, unlike Chamfer distance methods. Another difference is that we use soft features instead of binary edges, allowing robustness to choice of threshold.

Figure 9: Face detection results. The two leftmost images are masked templates. The white circles in the other images represent the best matches using geometric blur. The templates were compared with each of the images at a range of scales (differing by a factor of 1.4). The white circles represent the matches above threshold at any scale. A single threshold per template is appropriate for detecting faces across multiple images using these templates, indicating good discriminative ability. Either template individually can produce the detection results shown for all four images, indicating good generalization.

Geometric blur seems useful in a variety of settings and we are pursuing applications to feature matching, object recognition, and wide base-line stereo.

References

[1] H. Schneiderman and T. Kanade, "A Statistical Model for 3D Object Detection Applied to Faces and Cars," *IEEE Conference on Computer Vision and Pattern Recognition*, June, 2000.

[2] J. R. Bergen and P. Anandan and K. J. Hanna and R. Hingorani, "Hierarchical Model-Based Motion Estimation", *ECCV*, pp237-52. 1992

[3] D. Heeger, "Normalization of cell responses in cat striate cortex," *Neurosci.*, 9, 181-197. 1992.

[4] D. Field, "Relations between the statistics of natural images and the response properties of cortical cells", *Journal of The Optical Society of America A.*, Vol. 4, No. 12, pp. 2379-2394, (1987)

[5] J. Huang and D. Mumford, "Statistics of Natural Images and Models", *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 541-547, 1999.

[6] Kanade, T. and Okutomi, M., "A Stereo Matching Algorithm With An Adaptive Window: Theory And Experiment", *PAMI* 16, 9. pp. 920-932. September 1994

[7] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," PAMI, vol. 10, pp. 849-865, Nov. 1988.

[8] S. Belongie, J Puzicha, J. Malik, "Shape Matching", *ICCV* pp. 454-461. 2001

[9] A. P. Witkin, "Scale-space filtering", IJCAI pp. 1019-1022. 1983.

(a) (b)

(c) (d)

Figure 10: (a) 21 points found by the interest operator in image 1. 15 of them have obvious analogues in image 2. (b) Best 15 correspondences from target image using normalized cross correlation and large windows. (3 correct 12 wrong) (note poor localization on the few "correct" points) (c) Best 15 correspondences in image 2 using geometric blur, all 15 are correct. The large support (81×81) of the window provides context, and the geometric blur makes the correct tradeoff between fine local detail, and rough global context. Note that each feature is detected independently, there is no constraint on the geometric relationship of the detected feature points. This differentiates Geometric Blur from multi-scale optical flow techniques where smoothness or regularization is used to provide consistent results. (d) Best 15 correspondences from target image using normalized cross correlation on small (best over all window sizes) windows (4 correct, 11 wrong) (note better localization, but many of the errors are bridge corners matched to incorrect bridge corners because of lack of context resulting from small windows.)