Distribution Fields with Adaptive Kernels for Large Displacement Image Alignment

Benjamin Mears
bmears@cs.umass.edu
Laura Sevilla Lara
bmears@cs.umass.edu
Erik Learned-Miller
bmears@cs.umass.edu

School of Computer Science
UMass Amherst
Amherst, MA

Abstract

While region-based image alignment algorithms that use gradient descent can achieve sub-pixel accuracy when they converge, their convergence depends on the smoothness of the image intensity values. Image smoothness is often enforced through the use of multi-scale approaches in which images are smoothed and downsampled. Yet, these approaches typically use fixed smoothing parameters which may be appropriate for some images but not for others. Even for a particular image, the optimal smoothing parameters may depend on the magnitude of the transformation. When the transformation is large, the image should be smoothed more than when the transformation is small. Further, with gradient-based approaches, the optimal smoothing parameters may change with each iteration as the algorithm proceeds towards convergence.

We address convergence issues related to the choice of smoothing parameters by deriving a Gauss-Newton gradient descent algorithm based on distribution fields (DFs) and proposing a method to dynamically select smoothing parameters at each iteration. DF and DF-like representations have previously been used in the context of tracking. In this work we incorporate DFs into a full affine model for region-based alignment and simultaneously search over parameterized sets of geometric and photometric transforms. We use a probabilistic interpretation of DFs to select smoothing parameters at each step in the optimization and show that this results in improved convergence rates.

1 Introduction

Region-based image alignment consists of finding the transformation that maps a region in one image onto the corresponding region in the second and is a fundamental low level task in many computer vision applications. Improved alignment may increase the performance of a wide range of algorithms, including those for optical flow, stereo vision, tracking, and medical image registration. Each of these areas has specific constraints and challenges, yet alignment is a basic building block for all of them. Studying alignment in isolation from a specific application area thus lays the groundwork for more specialized algorithms.

Current region-based image alignment algorithms that use gradient descent often achieve sub-pixel accuracy when they converge. Yet, their convergence depends on the smoothness

of the image intensity values. If the image regions being aligned are highly textured, current gradient-based algorithms often converge only when the ground truth transformation is small. Multiscale approaches may be used to handle larger transformations, but these multiscale approaches can also be plagued by low convergence rates since information is lost when the image regions are blurred to create image pyramids. In this work, we address the small range of convergence of gradient-based methods by deriving a Gauss-Newton algorithm that uses distribution fields (DFs), an alternative to image pyramids, and adapting this method to dynamically choose smoothing parameters. Further, we extend the algorithm to handle photometric distortions by simultaneously performing a search over parameterized sets of geometric and photometric transforms. Compared to existing algorithms, we achieve significantly higher rates of convergence on images with and without photometric distortions.

1.1 Problem Formulation and Prior Work

Following the notation of Baker and Matthews [1], let $T(\mathbf{x})$ be an image containing a fixed region for which we want to find the corresponding region in $I(\mathbf{x})$, where $\mathbf{x} = (x,y)^{\mathrm{T}}$ is a column vector. We refer to $T(\mathbf{x})$ as the template image and $I(\mathbf{x})$ as the input image. Let $W(\mathbf{x};\mathbf{p})$ be the parameterized set of warps, where $\mathbf{p}$ are the parameters. The goal of region-based alignment is to find the $\hat{\mathbf{p}}$ that minimizes some distance measure between $T(\mathbf{x})$ and $I(\mathbf{W}(\mathbf{x};\hat{\mathbf{p}}))$.

One of the early image alignment algorithms was the gradient-based Lucas-Kanade (LK) algorithm [15]. Baker and Matthews [1] give an overview of the original LK algorithm. In the LK method, the feature space consists of intensity values and the similarity measure is the L2 distance. Baker and Matthews outline variations of this algorithm as well. For example, rather than computing additive updates to $\mathbf{p}$, compositional updates can be computed [22]. With compositional updates, the current warp is updated as

$$\mathbf{W}(\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{p}) \circ \mathbf{W}(\Delta\mathbf{p}). \tag{1}$$

Further, rather than computing an incremental warp for the input image, the warp can be computed for the template and then the inverse of the warp can be applied to the current estimate of $\mathbf{W}(\mathbf{p})$ [22]. These are known as inverse algorithms and the main advantage is that much of the computation at each iteration can be precomputed since the image for which the incremental warp is computed remains the same at every iteration. A major assumption of the LK algorithm is that the image regions are relatively smooth. The success of this algorithm thus depends on the degree to which this smoothness assumption is valid. A common approach to enforce the image smoothness assumption is to adapt the use of image pyramids [6] to the alignment problem [4].

Many variations of the LK method have subsequently been made since the introduction of the algorithm in 1981 [2, 3, 5, 8, 11, 12]. In particular, Evangelidis and Psarakis [8] derived an algorithm that uses a distance measure called enhanced correlation coefficient (ECC) that is invariant to photometric distortions in brightness and contrast.

Our approach is similar to that proposed by Hager *et al.* [12] in which the measure that is optimized is the sum of squared differences (SSD) between the component-wise square root of two kernel-weighted histograms. In our approach, we optimize the SSD between histogram-based image descriptors, called distribution field (DFs) [21], computed for the two image regions. While Hager *et al.* increase the robustness of their algorithm by employing multiple weighting kernels, the robustness of our algorithm derives from the structure of

the high dimensional histogram-based image descriptor that we employ and our principled approach of selecting kernel parameters.

Our work is most similar to that of Schreiber [20]. Like our work, Schreiber generalizes the Lucas-Kanade algorithm to optimize an SSD measure of a DF-like representation. Yet, rather than blurring in the spatial dimension Schreiber uses a fixed spatial binning scheme. Further, Schreiber uses a fixed kernel size for the feature kernel. Our work differs from that of Schreiber by incorporating dynamically chosen kernels for both the feature and spatial dimensions and also including an additional search over bias and gain parameters.

In addition to region-based alignment algorithms, another class of alignment algorithms is based on matching image features from the template and input images. An overview of feature-based alignment is given by Szeliski [23]. Because feature-based methods are based on matching keypoints rather than on gradient descent, they tend to have a much larger range of convergence and so are better suited for tasks such as the creation of panoramic mosaics and object recognition across disparate views. Yet feature-based alignment requires that keypoints be found in both regions being aligned and that a suitable number of correspondences be found between these keypoints. The SIFT Flow algorithm is a recent work that has addressed this issue by computing a SIFT feature [14] at every pixel location and finding dense correspondences between two images [13]. SIFT Flow works best when the image regions are approximately the same size and so as formulated is not directly applicable to the region-based alignment problem in which a transformed version of a region in one image needs to be found within a second image. Also, the algorithm does not enforce spatial continuity of matched points, which although has advantages in some applications such as scene matching, is undesireable in other applications such as region-based alignment. We do, however, provide comparisons to a SIFT Flow-like algorithm in our experiments.

2 Distribution Fields

In this section, we review distribution fields (DFs) [21]. DFs have previously been used in the context of tracking and have been shown to have a large basin of attraction for coordinate descent over the translation parameters [21]. Further, there has been much work on representations similar to DFs [7, 9, 17, 18, 20, 24]. The basic idea of a DF is to represent a region in an image as a normalized histogram, i.e., a probability distribution, over feature values at each pixel. In this work we use grayscale intensity values as the feature values although other features could be used instead (e.g. edge intensities, RGB values for color images, etc.). The simplest DF consists of probability distributions over binned intensity values where each probability distribution is degenerate and is given by

$$D(I,\mathbf{x},f) = \begin{cases} 1 & \text{if } I(\mathbf{x}) \in \text{bin } f \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where $D(I,\mathbf{x},f)$ is the value of the DF for image I at position $\mathbf{x}$ and bin f.

Using Eq. (2) alone to represent an image provides few additional benefits over using the image itself. Indeed, when the number of bins equals the number of intensity values, the representation contains the same information as the image. Additional benefits can be gained though if the DF is "smoothed" to "spread" the information in the image. In particular, the DF can be convolved with a three-dimensional Gaussian filter with a standard deviation of σ_{xy} in the spatial directions and σ_f in the feature space dimension. By convolving with a Gaussian filter, some degree of uncertainty is allowed in both the location and

value of an image pixel. Like image blurring, convolving a DF with a Gaussian filter spreads information about intensity values to neighboring pixels, but does so with a smaller loss of information. For example, consider an image consisting of adjacent black and white pixels. Blurring this image would result in gray pixels and thus information about the original bimodal distribution of pixels would be lost. In contrast, blurring a DF representation would result in probability mass being present at both high and low pixel values for the probability distribution at each pixel location.

3 Alignment Using Distribution Fields

The proposed algorithm is based on the inverse and forward compositional LK algorithms and is similar to the algorithm derived by Schreiber [20]. The L2-distance over intensity values used in the LK algorithms is replaced with an L2-distance over histogram bins in a DF. Thus, the goal is to minimize

$$\sum_{\mathbf{x}\in R}\sum_{f}[D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f)-D_\sigma(T,\mathbf{x},f)]^2 \tag{3}$$

with respect to $\mathbf{p}$, where $I(\mathbf{W}(\mathbf{p}))$ indicates image I transformed with warp parameters $\mathbf{p}$ and D_σ indicates a distribution field blurred with parameters $\sigma=\{\sigma_{xy},\sigma_f\}$.

In the forward compositional algorithm, compositional updates to the warp, $\mathbf{W}(\mathbf{p})$, applied to the input image are computed by iteratively minimizing

$$\sum_{\mathbf{x}\in R}\sum_{f}[D_\sigma(I(\mathbf{W}(\mathbf{p}\circ\Delta\mathbf{p})),\mathbf{x},f)-D_\sigma(T,\mathbf{x},f)]^2 \tag{4}$$

and the warp is updated using Eq. (1). We can apply a first order Taylor expansion to approximate Eq. (4) as

$$\sum_{\mathbf{x}\in R}\sum_{f}\left[D_\sigma(I(\mathbf{W}(\mathbf{p}\circ\mathbf{1})),\mathbf{x},f)\ +\nabla D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f)\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\Delta\mathbf{p}-D_\sigma(T,\mathbf{x},f)\right]^2, \tag{5}$$

where $\nabla D_\sigma(I,\mathbf{x},f)=\left(\frac{\partial D_\sigma(I,\mathbf{x},f)}{\partial x},\frac{\partial D_\sigma(I,\mathbf{x},f)}{\partial y}\right)$ and $\mathbf{1}$ denotes the identity transformation. Taking the derivative of Eq. (5) with respect to $\Delta\mathbf{p}$, simplifying $\mathbf{W}(\mathbf{p}\circ\mathbf{1})$ to $\mathbf{W}(\mathbf{p})$, setting equal to zero, and solving for $\Delta\mathbf{p}$ gives

$$\Delta\mathbf{p}=\left(\sum_{\mathbf{x}\in R}\sum_{f}\left[\nabla D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f)\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^{\mathrm{T}}\left[\nabla D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f)\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]\right)^{-1}*$$
$$\left(\sum_{\mathbf{x}\in R}\sum_{f}\left[\nabla D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f)\frac{\partial\mathbf{W}}{\partial\mathbf{p}}\right]^{\mathrm{T}}\left[D_\sigma(T,\mathbf{x},f)-D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f)\right]\right). \tag{6}$$

In the forward compositional algorithm, at each iteration the template is considered fixed and the update is computed for the input image. The inverse compositional algorithm reverses the roles of the template and input image. At each iteration the input image is considered fixed and the compositional update is computed for the template. Thus, rather than iteratively minimizing Eq. (4), the inverse compositional algorithm iteratively minimizes $\sum_{\mathbf{x}\in R}\sum_f[D_\sigma(T(\mathbf{W}(\Delta\mathbf{p})),\mathbf{x},f)-D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f)]^2$ with respect to $\Delta\mathbf{p}$ and updates

the warp as $\mathbf{W}(\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{p}) \circ \mathbf{W}(\Delta\mathbf{p})^{-1}$. The update at each iteration can be derived as

$$\Delta\mathbf{p} = \left(\sum_{\mathbf{x}\in R} \sum_{f} \left[\nabla D_\sigma(T,\mathbf{x},f) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[\nabla D_\sigma(T,\mathbf{x},f) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right] \right)^{-1} \left(\sum_{\mathbf{x}\in R} \sum_{f} \left[\nabla D_\sigma(T,\mathbf{x},f) \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[D_\sigma(I(\mathbf{W}(\mathbf{p})),\mathbf{x},f) - D_\sigma(T,\mathbf{x},f) \right] \right). \quad (7)$$

3.1 Combining Forward and Inverse Compositional Algorithms

As Brooks and Arbel [5] note, the update steps computed using the forward and inverse compositional algorithms may differ and combining the results of the algorithms can result in better convergence properties, an approach first proposed by Malis [16]. Since matrix operations are composed multiplicatively rather than additively, rather than perform a component-wise average of the matrices, it is more appropriate to average the updates' logarithms [10]. To do so, the matrix logarithm (logm) of the two transforms is taken, a component-wise average is computed, and the result is re-exponentiated using the matrix exponential (expm). Thus, we compute the update as $\Delta\mathbf{p} = \mathrm{expm}\left((\mathrm{logm}(\Delta\mathbf{p}_F) + \mathrm{logm}((\Delta\mathbf{p}_I)^{-1}))/2 \right)$ at each iteration, where $\Delta\mathbf{p}_F$ is the update computed in the forward compositional algorithm and $\Delta\mathbf{p}_I$ is the update computed in the inverse compositional algorithm.

3.2 Dynamic Selection of Kernel Parameters

There are various trade-offs that need to be considered when choosing the σ_{xy} and σ_f values used for smoothing the DFs for alignment. For instance, a larger σ_{xy} may allow for a larger basin of attraction but may result in a less precise final alignment. And if σ_{xy} is chosen to be too large or too small, it can cause the algorithm to diverge.

Rather than choose fixed σ values, the values can be chosen automatically based on the current location in the search space. One idea could be to choose the σ values that minimize the current SSD between the two DFs but for this measure the optimal σ value for blurring the DFs is infinity. With an infinite σ, both DFs devolve into a uniform distribution and the SSD is zero. Rather than use an SSD metric to choose the best σ values, the best values can be chosen using the probabilistic view of DFs by maximizing the log likelihood of the current warped input image under the DF of the template image. Treating the DF of one image as an independent pixel model, this is defined to be the sum of the log probabilities of each pixel of the current warped input image under the corresponding probability distribution of the template image's DF. Our method is similar to the approach used by Narayana *et al.* for choosing the pixel-wise kernel variances in their background subtraction algorithm [17, 18]. In their approach, they use joint domain-range based kernel estimates for the background and foreground models. These can be viewed as separate DFs for the background and foreground. For the background model, at each pixel location they dynamically select the kernel parameters that maximizes the likelihood. Our approach differs from that of Narayana *et al.* in that we maximize the log probability of the entire warped input image under the DF of the template and select global rather than pixel-wise variances.

In our approach, the log likelihood of the current warped input image, $I' = W(\mathbf{p})$, under the DF of the template, smoothed using the parameters σ_{xy} and σ_f, is given by

$$l(\sigma = \{\sigma_{xy}, \sigma_f\} | T, I', R) = \sum_{\mathbf{x}\in R} \log\left(D_\sigma(T,\mathbf{x},\mathrm{bin}(I'(\mathbf{x}))) \right), \quad (8)$$

where bin is the binning function that takes an intensity value and maps it to the appropriate histogram bin. Since truncated Gaussian kernels are used for efficiency to smooth the DFs, it is possible that some entries of a DF are zero. To deal with the problem of zero probabilities (in which a single outlier can cause the likelihood to be zero), we replace $\log\left(D_\sigma(T,\mathbf{x},\text{bin}(I'(\mathbf{x})))\right)$ in Eq. (8) with $\log\left(\max\left(.0001, D_\sigma(T,\mathbf{x},\text{bin}(I'(\mathbf{x})))\right)\right)$. At each iteration in our method, an exhaustive search is performed over a finite set of σ_{xy} and σ_f values and the σ_{xy} and σ_f values that maximize Eq. (8) are used to convolve the two DFs. Since the template image remains fixed, the DFs for the template can be precomputed.

3.3 Extension to Handle Bias and Gain

As formulated thus far, the DF-based alignment algorithm can accommodate noise in the intensity values of the template and/or input image by including nonzero values in the set of choices for σ_f. Yet, simply allowing for non-zero σ_f's is not enough to handle large scale global changes in intensity such as bias (an additive offset) and gain (a multiplicative offset). To handle such global changes, we use an approach similar to the simultaneous inverse compositional (SIC) algorithm described by Baker and Matthews [2]. With this approach it is assumed that a warped version of $T(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{x})$ appears in $I(\mathbf{x})$ where the A_i are a set of basis images. In our work, we choose A_1 to be T in the inverse direction and $I(W(\mathbf{p}))$ in the forward direction and A_2 to be an image consisting of all ones. The choice of A_1 and A_2 allows for the modeling of bias and gain. While other basis images can be added to deal with more complex intensity changes, in practice the use of A_1 and A_2 together with nonzero σ_f values allows our algorithm to handle photometric distortions other than bias and gain.

To add the parameters λ_1 and λ_2 to the gradient descent, we explicitly compute the discrete gradient for the two parameters at every histogram bin in the DF. To simplify the notation in the following, let $\mathbf{q} = [\mathbf{p} \quad \lambda]$, $A_i(\mathbf{p}) = A_i(W(\mathbf{p}))$, and $I(\mathbf{q}) = I(W(\mathbf{p})) + \sum_i \lambda_i A_i(\mathbf{p})$. In the forward direction the gradients are computed by adding a small ε_{λ_i} to the parameter and evaluating $\partial D_\sigma(I(\mathbf{q}),\mathbf{x},f)/\partial\lambda_i = \left[D_\sigma(I(\mathbf{q}) + \varepsilon_{\lambda_i} A_i(\mathbf{p}),\mathbf{x},f) - D_\sigma(I(\mathbf{q}),\mathbf{x},f)\right]/\varepsilon_{\lambda_i}$. Similarly, in the inverse direction the discrete gradient is calculated as $\partial D_\sigma(T,\mathbf{x},f)/\partial\lambda_i = \left[D_\sigma(T + \varepsilon_{\lambda_i} A_i,\mathbf{x},f) - D_\sigma(T,\mathbf{x},f)\varepsilon_{\lambda_i}\right]$. In our experiments we let $\varepsilon_{\lambda_1} = .01$ and $\varepsilon_{\lambda_2} = 1$.

We then use these discrete gradients to augment the term $\nabla D_\sigma(T,\mathbf{x},f)\,(\partial\mathbf{W}/\partial\mathbf{p})$ in Eq. (7) and the term $\nabla D_\sigma(I(\mathbf{p}),\mathbf{x},f)\,(\partial\mathbf{W}/\partial\mathbf{p})$ in Eq. (6). These corresponding terms in the LK method are termed the steepest descent images by Baker and Matthews [1]. Thus, the steepest descent images become

$$\left[\nabla D_\sigma(T,\mathbf{x},f)\frac{\partial\mathbf{W}}{\partial\mathbf{p}}, \frac{\partial D_\sigma(T,\mathbf{x},f)}{\partial\lambda_1}, \frac{\partial D_\sigma(T,\mathbf{x},f)}{\partial\lambda_2}\right] \tag{9}$$

in the inverse direction and

$$\left[\nabla D_\sigma(I(\mathbf{q}),\mathbf{x},f)\frac{\partial\mathbf{W}}{\partial\mathbf{p}}, \frac{\partial D_\sigma(I(\mathbf{q}),\mathbf{x},f)}{\partial\lambda_1}, \frac{\partial D_\sigma(I(\mathbf{q}),\mathbf{x},f)}{\partial\lambda_2}\right] \tag{10}$$

in the forward direction. The $\Delta\lambda_{i,F}$ and $\Delta\lambda_{i,I}$ can then be computed together with the $\Delta\mathbf{p}_F$ and $\Delta\mathbf{p}_I$ where recall that the F subscript indicates the value computed in the forward algorithm and the I subscript indicates the value computed in the inverse algorithm. The λ_i's are then updated as $\lambda_i^{new} = \lambda_i^{old} + (\Delta\lambda_{i,F} - \Delta\lambda_{i,I})/2$. In the experiments described in the following extension, we refer to this additional search over photometric parameters as the "SIC extension." We also found that on average the best results were obtained by first normalizing

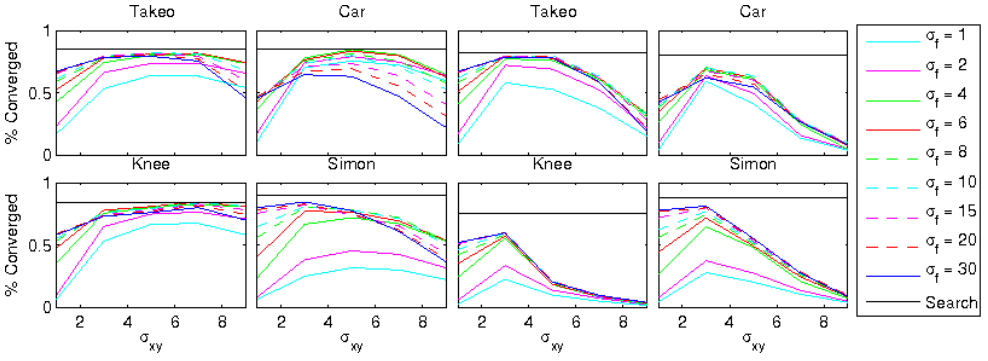

Figure 1: Results of experiments testing the use of fixed kernel parameters versus automatic selection. The first two columns show convergence rates for image pairs that differ by a geometric, but not a photometric, transformation and the second two columns show convergence rates for image pairs that differ by both a geometric and photometric transformation. "Search" refers to the automatic selection of kernel parameters described in Section 3.2. For all images the automatic selection method produced convergence rates better than or equal to any of the fixed parameters

the image regions to have a range of pixel values of 0 to 255. Note that image normalization without the SIC extension produced inferior convergence rates.

4 Experimental Setup and Results

We adopt an experimental setup similar to that of Baker and Matthews [1] and Evangelidis and Psarakis [8]. To construct an affine transformation, three canonical points are selected from the region of interest (the top left, top right, and bottom middle). Gaussian noise is then added to these points and the corresponding affine warp computed. The average magnitude of a set of transformations can be controlled by changing the standard deviation of the Gaussian noise used to generate the transformations. We use the same image regions as Baker and Matthews. Similar results were seen on other image regions.

To compare the algorithms' robustness to photometric distortions, we test our algorithm on image regions with and without photometric distortions. To generate a photometric distortion, we follow the procedure used by Evangelidis and Psarakis [8] and apply a transform of the form

$$I(\mathbf{x}) \leftarrow (I(\mathbf{x}) + 20)^{0.9} \tag{11}$$

to the input image. Further, Gaussian noise with standard deviation of 8 is applied to the pixel values of both images. To measure whether an alignment converged, the mean squared error of the canonical points in T and $I(\mathbf{W}(\mathbf{p}))$ is computed. For experiments with no photometric noise, the threshold for convergence is set to one pixel while for experiments with photometric noise the threshold is set to 1.5 pixels. Since we are mainly concerned with the convergence of the algorithm, we allow all the algorithms to run either to convergence or to a large maximum number of iterations (50) at each pyramid level. The convergence rate is averaged over 500 transformations for each of the standard deviation values of the Gaussian noise used to generate the transformations.

We first ran a set of experiments to test the dynamic selection of kernel parameters described in Section 3.2. In this experiment, we used Gaussian noise with standard deviation of 15 to generate the affine transformations. A photometric distortion specified by Eq. (11) was applied to the input images and Gaussian noise with standard deviation of 8 was added to both the template and input images. Alignments were performed over the affine and photometric parameters and the image regions were pre-normalized. We compared using fixed σ_{xy} and σ_f parameters versus choosing σ_{xy} and σ_f automatically at each step from the sets $\{1,3,5,7,9\}$ and $\{1,2,4,6,8,10,15,20,30\}$, respectively. We used the extensive set of fixed parameters as a surrogate for the method of Schreiber since his use of spatial binning can be viewed as applying a box kernel with a fixed kernel size. Also, 64 bins were used for each histogram in the DFs and the DFs were subsampled by a factor of 2 in the spatial direction which allowed for a significant speedup without significantly affecting convergence.

Figure 1 shows the results of these experiments. For all images the automatic selection method (termed "Search" in the legend) produced convergence rates better than or equal to any of the fixed parameters. The comparison to the best fixed parameters for each image is an optimistic standard since the best parameters can only be chosen by running experiments on each image with known ground-truth transformations while our approach selects parameters on-the-fly. Figure 2 shows an example transformation and the kernel parameters selected. Note that the kernel parameters tend to decrease as the alignment reaches convergence.

We next compared our method to previous approaches for region-based image alignment. For a fair comparison to our approach which pre-normalizes images, images were also pre-normalized for the comparison methods which in general greatly improved their performance on images with photometric noise and had a small effect on images without photometric noise. We compared to the multiscale LK forward additive method, a LK method in which the updates from the inverse and forward compositional algorithms were combined using the method described in Section 3.1, and a SIC version of LK. For the ECC algorithm, we used the code provided by Evangelidis and Psarakis[1] [8] although for consistency we adapted the code to use the same hierarchical approach as the LK methods rather than use the hierarchical approach implemented by the authors. Also, to compare to a SIFT Flow-like algorithm, we implemented a version of our algorithm in which a SIFT vector rather than a probability distribution over intensity values is located at each pixel location. We call this representation a SIFT DF. We found that the performance of this method was improved if each channel of the SIFT DFs were blurred spatially (i.e. each component was blurred separately across the image), with the kernel parameters chosen at each iteration to minimize the SSD between the unblured SIFT DF of the warped image and the blurred SIFT DF of the template. We did not blur this representation in the feature space dimension since a SIFT vector incorporates multiple histograms and so it does not make sense to blur across the entire vector. Further, SIFT already weights features during the construction of the feature vector.

For both the ECC and LK methods, three pyramid levels are used. For the DF-based method, the σ_{xy} and σ_f values are automatically chosen at each iteration from the sets $\{1,3,5,7,9\}$ and $\{1,2,4,6,8,10,15,20,30\}$, respectively. For the DF SIFT method, the σ_{xy} parameter is automatically chosen at each iteration from the set $\{1,3,5,7,9\}$. The set of values for σ_{xy} was chosen to be evenly spaced integers while the set for σ_f was also chosen to be integers but with larger spacings for larger values. Similar to the first experiment, the DF is subsampled by a factor of two spatially and 64 bins are used.

The first two columns of Figure 3 show the results for alignments between the input and

[1] http://xanthippi.ceid.upatras.gr/people/evangelidis/ecc/

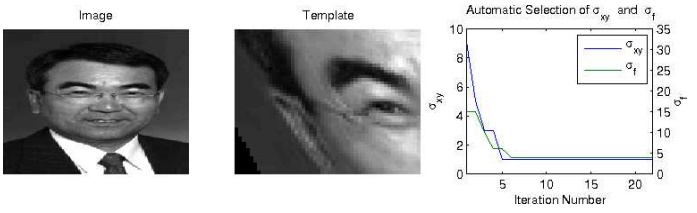

Figure 2: Selection of kernel parameters for an example transformation. The left image shows the image in which a warped version of the template shown in the middle is trying to be found. The graph on the right shows the kernel parameters selected by our method at each step during the alignment.

template images when no photometric distortion or Gaussian noise is applied to the intensity values of either image. As can be seen in the figure, the DF-based method performs similarly to, or outperforms, the other top-performing methods on all four images. The outperformance is particularly significant on the "Car" image. The "Car" image has a relatively large amount of texture compared to the other images and so relatively more information is lost when it is blurred, which is why the image pyramid-based approaches perform poorly.

The last two columns of Figure 3 show the results for alignments when a photometric distortion specified by Eq. (11) is applied to the input image and Gaussian noise with standard deviation of 8 is added to both the template and input images. The DF-based method outperforms all of the other methods.

5 Conclusion and Future Work

We have shown that by deriving an iterative Gauss-Newton algorithm that uses an error measure based on DFs and that dynamically chooses the kernel parameters at each iteration, we can achieve higher rates of convergence than existing algorithms. And these improvements are seen both with and without photometric distortions.

While our main focus was on achieving high convergence rates rather than efficiency, we believe significant speedups in our algorithm can be achieved. Two bottlenecks in our algorithm are the convolution of the DF and the high dimensionality of DFs. The latter was partially addressed by subsampling the DF and can be further addressed by experimenting with the number of bins used to construct the DF. The former can be addressed by using speedups similar to those proposed by Paris and Durand [19] for the bilateral filter. To efficiently compute an approximation to the bilateral filter, Paris and Durand construct an image representation analogous to a DF. They note that since the convolution is a low pass filter, it can be computed at a coarser resolution without introducing significant errors. A similar approach can be used with our method. Further, since our approach subsamples the DF, the result of the coarse convolution would only have to be upsampled for a fraction of the points in the DF, thus reducing the time and space complexity of our algorithm.

6 Acknowledgments

This material is based upon work supported by the National Science Foundation under NSF DGE-0907995.

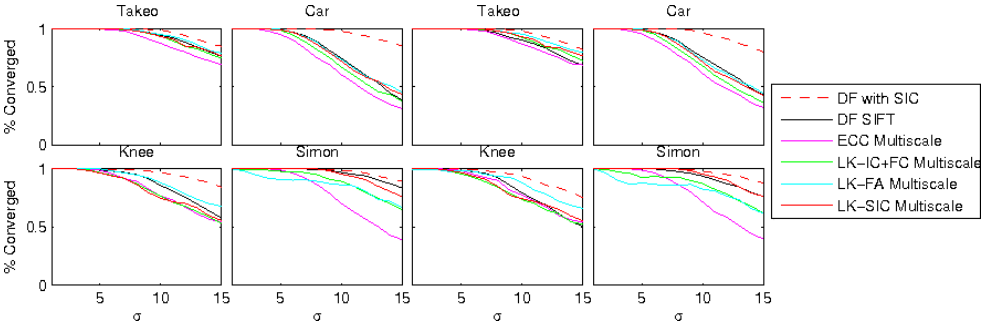

Figure 3: The first two columns show convergence rates for image pairs that differ by a geometric, but not a photometric, transformation. The last two columns show convergence rates for image pairs that differ by both a geometric and photometric transformation, and that have Gaussian noise with standard deviation of 8 added to their pixel values.

References

[1] S. Baker and I. Matthews. Lucas-Kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.

[2] S. Baker, R. Gross, and I. Matthews. Lucas-Kanade 20 years on: A unifying framework: Part 3. Technical Report CMU-RI-TR-03-35, CMU Robotics Institute, December 2003.

[3] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. Lucas-Kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-01, CMU Robotics Institute, March 2003.

[4] J. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *ECCV*, 1992.

[5] R. Brooks and T. Arbel. Generalizing inverse compositional and ESM image alignment. *IJCV*, 2010.

[6] P. Burt and E. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.

[7] A. Elgammal, R. Duraiswami, and L. Davis. Probabilistic tracking in joint feature-spatial spaces. In *CVPR*, 2003.

[8] G.D. Evangelidis and E.Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *PAMI*, 2008.

[9] M. Felsberg. Adaptive filtering using channel representations. In *Mathematical Methods for Signal and Image Analysis and Representation*, pages 31–48. 2012.

[10] P.T. Fletcher, C. Lu, and S. Joshi. Statistics of shape via principal component analysis on Lie groups. In *CVPR*, 2003.

[11] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 1998.

[12] G.D. Hager, M. Dewan, and C.V. Stewart. Multiple kernel tracking with SSD. In *CVPR*, 2004.

[13] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W.T. Freeman. SIFT Flow: Dense correspondence across different scenes. In *ECCV*. 2008.

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

[15] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[16] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *ICRA*, 2004.

[17] M. Narayana, A. Hanson, and E. Learned-Miller. Background modeling using adaptive pixelwise kernel variances in a hybrid feature space. In *CVPR*, 2012.

[18] M. Narayana, A. Hanson, and E. Learned-Miller. Improvements in joint domain-range modeling for background subtraction. In *BMVC*, 2012.

[19] S. Paris and F. Durand. A fast approximation of the bilateral filter using a signal processing approach. In *ECCV*, 2006.

[20] D. Schreiber. Generalizing the Lucas–Kanade algorithm for histogram-based tracking. *Pattern Recognition Letters*, 29(7):852–861, 2008.

[21] L. Sevilla-Lara and E. Learned-Miller. Distribution fields for tracking. In *CVPR*, 2012.

[22] H.Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *IJCV*, 48(2):151–152, 2002.

[23] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, 2006.

[24] C. Yang, R. Duraiswami, and L. Davis. Efficient mean-shift tracking via a new similarity measure. In *CVPR*, 2005.