Multiple View Feature Descriptors from Image Sequences via Kernel Principal Component Analysis

Jason Meltzer[1], Ming-Hsuan Yang[2], Rakesh Gupta[2], and Stefano Soatto[1]

[1] University of California, Los Angeles, CA 90095, USA,
[2] Honda Research Institute, Mountain View, CA 94041, USA

Abstract. We present a method for learning feature descriptors using multiple images, motivated by the problems of mobile robot navigation and localization. The technique uses the relative simplicity of small baseline tracking in image sequences to develop descriptors suitable for the more challenging task of wide baseline matching across significant viewpoint changes. The variations in the appearance of each feature are learned using kernel principal component analysis (KPCA) over the course of image sequences. An approximate version of KPCA is applied to reduce the computational complexity of the algorithms and yield a compact representation. Our experiments demonstrate robustness to wide appearance variations on non-planar surfaces, including changes in illumination, viewpoint, scale, and geometry of the scene.

1 Introduction

Many computer vision problems involve the determination and correspondence of distinctive regions of interest in images. In the area of robot navigation, a mobile platform can move through its environment while observing the world with a video camera. In order to determine its location, it must create a model that is rich enough to capture this information yet sparse enough to be stored and computed efficiently. By dealing with only sparse image statistics, called *features,* these algorithms can be made more efficient and robust to a number of environmental variations that might otherwise be confusing, such as lighting and occlusions. Usually, these features must be tracked across many images to integrate geometric information in space and time. Thus, one must be able to find correspondences among sets of features, leading to the idea of *descriptors* which provide distinctive signatures of distinct locations in space. By finding features and their associated descriptors, the correspondence problem can be addressed (or at least made simpler) by comparing feature descriptors.

In applications such as N-view stereo or recognition, it is frequently the case that a sparse set of widely separated views are presented as input. For such *wide baseline* problems, it is necessary to develop descriptors that can be derived from a single view, since no assumptions can be made about the relative viewpoints among images. In contrast, in the cases of robot navigation or real-time structure

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3021, pp. 215–227, 2004.

from motion, a video stream is available, making it possible to exploit *small baseline* correspondence by tracking feature locations between closely spaced images. This provides the opportunity to incorporate multiple views of a feature into its signature, since that information is present and easily obtained. Under these circumstances, the problem of tracking across frames is relatively simple, since the inter-frame motion is small and appearance changes are minimal. Many existing feature trackers, such as [12,35,37], can produce chains of correspondences by incrementally following small baseline changes between images in the sequence. These trackers, however, are far from ideal and often drift significantly within a handful frames of motion, thus they cannot be used for wide baseline matching. In order to reduce this effect, features must be recognized despite a variety of appearance changes. Additionally, when a sudden large change in viewpoint occurs, it is necessary to relate features that have already been observed to their counterparts in a new image, thus maintaining the consistency of the model that the features support. In this paper, we propose to integrate the techniques of short baseline tracking and wide baseline matching into a unified framework which can be applied to correspondence problems when video streams are available as input.

One may question the necessity of such a multi-view descriptor, given that many single view features work well (see the next section for an overview). In some cases, in fact, multiple views will add little to the descriptor matching process, since the variability can be well modelled by a translational or affine transformation. When these assumptions are violated, such as when there are non-planar surfaces in a scene, or complex 3D geometry, multiple views of a feature can render significant robustness to viewpoint changes. When the geometry itself is changing, such a descriptor is necessary to capture this variability. A multiple view descriptor can provide a generic viewpoint, meaning the results of matching will be less sensitive to particular viewpoints (which may be confusing special cases). Perhaps the most compelling argument in favor of the multi-view approach is that, in many applications, the data is already there. When this is the case, it makes sense to try to leverage the data available, rather than discard it after processing each frame.

1.1 Related Work

The problem of finding and representing distinctive image features for the purposes of tracking, reconstruction, and recognition is a long-standing one. Recently, a number of authors (Schmid *et al* [7,23], Lowe [21,20], Baumberg [1], Tuytelaars and Van Gool [38,39], Schaffalitzky and Zisserman [28,29]) have developed *affine invariant* descriptors of image locations. These expand upon the pioneering work of Lindeberg [19], Koenderink [15], and others who study image scale space and its properties. The general approach of these methods is to find image locations which can be reliably detected by searching for extrema in the scale space of the image [19]. Given different images of a scene taken over small or wide baselines, such descriptors can be extracted independently on each pair, then compared to find local point correspondences.

Lowe's scale invariant feature transform (SIFT) [20] considers an isotropic Gaussian scale space, searching for extrema over scale in a one-dimensional scale space. The difference-of-Gaussian function is convolved with the image, computed by taking the difference of adjacent Gaussian-smoothed images in the scale space pyramid. Once the extrema are located (by finding maxima and minima in neighborhoods of the scale space pyramid), they are filtered for stability then assigned a canonical orientation, scale and a descriptor derived from gradient orientations in a local window. The descriptor is adapted from Edelman *et al* [8], which models the outputs of "complex cells" that respond to particular image gradients within their receptive fields. In a similar manner, SIFT samples gradients around the points of interest and amalgamates them into a 4x4 array of 8-bin orientation histograms. This constitutes the 128 element SIFT descriptor. SIFT has been shown to match reliably across a wide range of scales and orientation changes, as well as limited 3D perspective variation [24].

Mikolajczyk and Schmid's affine invariant interest point detector [23] seeks to find stable areas of interest in the affine image scale space, which has three parameters of scale. It first selects initial feature locations using a multi-scale Harris corner detector [12] then applies an iterative technique to find the best location, scale, and shape transformation of the feature neighborhood. The procedure converges to yield a point location in the image, as well as a canonical transformation which can be used to match the feature despite arbitrary affine transformations of its neighborhood. Descriptors consist of normalized Gaussian derivatives computed on patches around the points of interest. These patches are transformed by the canonical mapping used in the detection phase of the algorithm, which yields scale and skew invariance. Rotational invariance is obtained using steerable filters, and affine photometric invariance is achieved by normalizing all of the derivatives by the first. A dimension 12 descriptor is the final output of the procedure, involving derivatives up to 4th order.

Tuytelaars and Van Gool [39] have developed methods which explicitly take into account a variety of image elements, such as corners, edges, and intensity. They find and characterize affinely invariant neighborhoods by exploiting properties of these elements, then match similar points using geometric and photometric constraints to prune false matches. An explicit assumption they make is that the areas of interest selected lie on approximately planar regions, though their experiments demonstrate robustness to violations of this assumption.

The aforementioned works (which represent only a small fraction of the latest literature on the topic, see also [1,7,9,5,12,14,15,20,21,22,23,26,27,28,29,34,36,38, 39,40] for some others) focus on the problem of extracting properties from local patches in a *single* image in order to match these locations in subsequent images of the same scene. In [9], Ferrari *et al* present a method for matching features across multiple unordered views, which is derived from pairwise view matches using the detector described in [39]. When a video stream is available, as often is the case when using cameras on mobile robots, more information is present than can be obtained from a single image of a scene or an arbitrary set of such images. It is therefore reasonable to seek a *multi-view descriptor*, which incorporates information from across multiple adjacent views of a scene to yield better feature correspondences.

1.2 Rationale and Overview of the Approach

We address the wide-baseline correspondence problem under the specific scenario of autonomous guidance and navigation, where high frame-rate video is available during both training ("map building") and localization, but viewing conditions can change significantly between the two. Such changes affect both the domain of the image (geometric distortion due to changes of the viewpoint and possibly deformations of the scene) and its range (changes in illumination, deviation from Lambertian reflection). If $w : \mathbb{R}^2 \to \mathbb{R}^2$ denotes a piecewise smooth function, and $I : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^+$ denotes the image, then in general two given images are related by $I_2(x) = \rho(I_1(w(x)))$ where ρ is a functional that describes range deformations, and w describes domain deformations. Such changes are due to both intrinsic properties of the scene ξ (shape, reflectance) and to *nuisance factors* ν (illumination, viewpoint), so we write formally $\rho = \rho_{\xi,\nu}(I)$ and $w = w_{\xi,\nu}(x)$. The goal of the correspondence process is to associate different images to a common cause (the same underlying scene ξ) despite the nuisances ν.

A "feature" is a statistic of the image, $\phi : I \to \mathbb{R}^k$ that is designed to facilitate the correspondence process.[1] Ideally one would want a feature statistic that is invariant with respect to all the nuisance factors: $\phi \circ \rho_{\xi,\nu}(I(w_{\xi,\nu}(x))) = f_\xi(x)$ independent of ν, for some function f and for all allowable nuisances ν. Unfortunately this is not possible in general, since there exists no single-view statistic that is invariant with respect to viewpoint or lighting conditions, even for Lambertian scenes. Nuisances that can be moded-out in the representation are called *invertible*.[2] What nuisance is invertible depends on the representation of the data. If we consider the data to be a single image, for instance, viewpoint is not an invertible nuisance. However, if multiple adjacent views of the same scene are available, as for instance in a video from a moving camera, then viewpoint can be explicitly accounted for, at least in theory. Additionally, changes in viewpoint elicit irradiance changes that are due to the interplay of reflectance and illumination, and it is therefore possible that "insensitive" (if not invariant) features can be constructed. This is our rationale for designing feature descriptors that are based *not* on single views, but on multiple adjacent views of the same scene.

Any correspondence process relies on an underlying model of the scene, whether this is stated explicitly or not: our model of the scene is a constellation of planar patches that support a radiance density which obeys a diffuse+specular reflection model. This means that for patches that are small enough one either sees the Lambertian albedo, or a reflection of the light source, and therefore the rank of an aggregation of corresponding views is limited [13]. We represent

[1] Facilitation should be quantified in terms of computational complexity, since the benefit of using features to establish correspondence is undermined by Rao-Blackwell's theorem, that guarantees that a decision based on any statistic of the data achieves a conditional risk that is no less than the decision based on the raw data.

[2] Euclidean, affine, and projective image motion are invertible nuisances, in the sense that they can be eliminated by pre-processing the image. However, viewpoint is not invertible, unless the scene has special symmetries that are known *a priori*.

the multi-view descriptor by a rank-constraint in a suitable inner product of a deformed version of the image: $\text{rank}(\mathcal{T}) = r$, where the tensor $\mathcal{T}$ is defined by

$$\mathcal{T} \doteq (\Phi(I_1(w_1(x))), \Phi(I_2(w_2(x))), \ldots, \Phi(I_n(w_n(x)))) \tag{1}$$

for a suitable function Φ that maps the image to a higher-dimensional space.[3] The modeling "responsibility" is shared by the transformation w and the map Φ: the more elaborate the one, the simpler the other. What is not modeled explicitly by w and Φ goes to increase the rank of $\mathcal{T}$. In this general modeling philosophy, our approach is therefore broadly related to [6,10,11]. In this work, we consider the following combinations:

Translational domain deformation, generic kernel: We use the simplest possible $w(x) = x + T$, a generic map Φ, and all the responsibility for modeling deformations of the domain and the range is relegated to the principal components of the tensor $\mathcal{T}$. In this case, the descriptor is invariant with respect to plane-translations, but all other deformations contribute to the rank $r \simeq n$.

Affine domain deformation, generic kernel: $w(x) = Ax + T$, and additional geometric distortion due to chages of viewpoint and photometric variability is relegated to the principal components of $\mathcal{T}$.

Viewpoint deformation: In this case, $w(x)$ depends on the 3-D structure of the scene, and is explicitly inverted by projective reconstruction and normalization. Therefore, $\mathcal{T}$ is viewpoint invariant and its principal components only models photometric variability.

2 Proposed Solution

We relegate deformations in the images not accounted for by the transformation w to the principal components of $\mathcal{T}$. Principal component analysis (PCA) operates on the premise that a low dimensional basis suffices to approximate the covariance matrix of the samples, thereby providing a compact representation. Given M observation images, PCA diagonalizes the covariance matrix $C = \frac{1}{M}\sum_{j=1}^{M} \mathbf{y}_j \mathbf{y}_j^T$ (where $\mathbf{y}_j$ can be considered a vectorized image patch, and without loss of generality we assume that $\mathbf{y}$ is pre-processed to be zero mean) by solving an eigenvalue equation [2]. The Karhunen-Loeve (KL) transform is an efficient method to compute the basis (principal components), which can be carried out using singular value decomposition (SVD) [3] . For the case where we have a stream of incoming data, the sequential Karhunen-Loeve algorithm exploits the low dimension approximation characteristic by partitioning and transforming the data into blocks of orthonormal columns to reduce computational and memory

[3] We use Φ to indicate the map from the data to what is known in the kernel-machine community as "feature space" and ϕ for the feature that we have defined in this section (i.e. an invariant image statistic). The notation should not be confusing in the end, since ϕ will comprise principal components of $\Phi(I_j)$.

requirements [18,4]. In other words, this algorithm essentially avoids the computation of a full-scale SVD at every time step by only applying the necessary computation to smaller data blocks for updating the KL basis incrementally.

2.1 Incremental Update of Kernel Principal Components

Since PCA aims to find an optimal low dimensional linear subspace that minimizes the reconstruction error, it does not perform well if the data points are generated from a nonlinear manifold. Furthermore, PCA encodes the data based on second order dependencies (pixel-wise covariance among the pixels), and ignores higher-order statistics including nonlinear relations among the pixel intensity values, such as the relationships among three or more pixels in an edge or a curve, which can capture important information for recognition.

The shortcomings of PCA can be overcome through the use of kernel functions with a method called kernel principal component analysis (KPCA). In contrast to conventional PCA which operates in the input image space, KPCA performs the same procedure as PCA in a high dimensional space, F, related to the input by the (nonlinear) map $\Phi : \mathbb{R}^N \longrightarrow F, \quad \mathbf{y} \mapsto Y$. [4] If one considers $\mathbf{y} \in \mathbb{R}^N$ to be a (vectorized) image patch, $Y \in F$ is this image patch mapped into F. The covariance matrix for M vectors in F is $C' = \frac{1}{M} \sum_{j=1}^{M} \Phi(\mathbf{y_j})\Phi(\mathbf{y_j})^T$, assuming $\sum_{k=1}^{M} \Phi(\mathbf{y_k}) = 0$ (see [31] for a method to center $\Phi(\mathbf{y})$). By diagonalizing C', a basis of kernel principal components (KPCs) is found. As demonstrated in [30], by using an appropriate kernel function $k(\mathbf{x}, \mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, one can avoid computing the inner product in the high-dimensional space F. The KPCs are implicitly represented in terms of the inputs (image patches) $\mathbf{y}$, the kernel k, and a set of linear coefficients β, as $\Psi = \sum_{i=1}^{M} \beta_i \Phi(\mathbf{y}_i)$, $\Psi \in F$.

To choose an appropriate kernel function, one can either estimate it from data or select it *a priori*. In this work, we chose the Gaussian kernel

$$k(\mathbf{w}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{w} - \mathbf{y}\|^2}{2\sigma^2}\right) \tag{2}$$

based on empirical study, though a principled yet computationally expensive method for learning the kernel from data using quadratic programming was recently demonstrated by Lanckriet et al. [17].

Unfortunately, there is no "online" version of KPCA, as exists for standard PCA [4,18]. In order to avoid computations in the high-dimensional space, all computations are performed through the kernel in terms of linear combinations of input vectors. Hence, in traditional KPCA, all of the input vectors (image patches) must be stored in order to perform classification. This is unacceptable for feature descriptors, since the storage requirement is high, the computational complexity grows with more examples, and the representation is not compact.

[4] F is typically referred to as "feature space," but to avoid confusion we will refrain from using that name.

To avoid this problem, we have employed an online (i.e. constant time and memory) version, *approximate* KPCA, which continually produces a fixed number of approximations of the input patches. These approximations and their expansion coefficients, along with the kernel function, form a compact representation of the KPCA basis in high-dimensional space, which can be used to compare an observed feature with this descriptor. The technique is based on the one of Schölkopf *et al* [33] to find approximate "pre-images" of vectors implicitly represented in high-dimensional space. Given a vector $\Psi = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{y}_i)$, we seek an approximation $\Psi^* = \sum_{i=1}^{L} \beta_i \Phi(\mathbf{z}_i)$ with $L < M$. For a particular pre-image $\mathbf{z}$, [33] demonstrates that it is sufficient to minimize the distance between Ψ and its projection onto $\Phi(\mathbf{z})$, which is equivalent to maximizing

$$\frac{\langle \Psi, \Phi(\mathbf{z}) \rangle^2}{\langle \Phi(\mathbf{z}), \Phi(\mathbf{z}) \rangle}. \tag{3}$$

By differentiating and substituting the Gaussian kernel function for inner products, the following fixed-point iteration for $\mathbf{z}$ is obtained:

$$\mathbf{z}^{n+1} = \frac{\langle \mathbf{Y}, (\alpha \cdot * \mathbf{K}^n) \rangle}{\alpha^T \mathbf{K}^n} \tag{4}$$

where $\mathbf{Y}$ is a matrix of input vectors $\mathbf{y}_i$ (image patches), α is the vector of coefficients, $\mathbf{K}^n$ is the vector of kernels $[k(\mathbf{y}_1, \mathbf{z}^n), ..., k(\mathbf{y}_N, \mathbf{z}^n)]^T$, and $\cdot *$ is the element-wise multiplication operator. In order to find a number of such approximations, $\mathbf{z}_1, ...\mathbf{z}_M$, we set $\Psi^{m+1} = \Psi^m - \beta^m \Phi(\mathbf{z}^m)$, where $\mathbf{z}^m$ is found using (4). One can solve for the optimal coefficients of the expansion, β, in each iteration to yield $\Psi^* = \sum_{i=1}^{L} \beta_i \Phi(\mathbf{z}_i)$ [32].

In order to match a newly observed image to existing descriptors, our algorithm searches the image for patches which have a small residual when projected onto the stored KPCA descriptors. That is, it finds $\mathbf{y}$, a patch from the new image, and ψ, a descriptor (KPCA basis), such that the following is minimized for a choice of ψ and $\mathbf{y}$.

$$\left\| \Phi(\mathbf{y}) - \sum_{i=1}^{N} \frac{\langle \Phi(\mathbf{y}), \psi_i \rangle}{\langle \psi_i, \psi_i \rangle} \psi_i \right\|^2 \tag{5}$$

where ψ_i is a kernel principal component and N is the number of components in the descriptor.

2.2 Feature Descriptors through Incremental Update of KPCA

Our method for extracting feature descriptors from image sequences proceeds as follows:

1. **Bootstrap with a small-baseline tracker**: Read a number of frames of the input sequence, track the features using a standard tracking method, and store the image patches of each feature. As a translation-invariant tracker, $w(x) = x + T$, we use Lukas and Kanade's [37] classic algorithm; for affine-invariant tracker, $w(x) = Ax + T$, we use the Shi and Tomasi (ST) algorithm.

2. **Construct kernel basis**: Perform KPCA using the Gaussian kernel separately on each feature's training sequence or reduced training set found in step 3.
3. **Approximate kernel basis**: Form an approximate basis for each feature by finding approximate patches which lead to the least residual estimate of the original basis in high-dimensional space. Create L such patches for each feature. In our algorithm, L is a tuning parameter. Further discussion of "pre-image" approximation in KPCA can be found in [32,33].

The above algorithm yields a set of descriptors, each corresponding to a particular feature. Given a novel image of the same scene, these descriptors can be matched to corresponding locations on the new image using (5).

3 Experiments

We performed a variety of experiments with the proposed system to test the efficacy of small and wide baseline correspondence. In the experiments, two phases were established: *training* and *matching*, which correspond to short and wide baseline correspondence. In the training phase, a video sequence was recorded of a non-planar object; this object underwent a combination of 3D rotation, scaling, and warping with respect to the camera. The Shi-Tomasi (ST) [35] tracker (or Lucas-Kanade (LK), in the case of translational tracking) was used to obtain an initial set of points, then the procedure of the previous section was used to track these locations and develop feature descriptors via approximate KPCA. Note that we do not show experiments for projective reconstruction, since we did not see any benefit to this approach using our data sets. In future experiments with more severe depth variations, we expect to see significant benefits by normalizing the projective reconstruction.

In the matching phase, a test image from outside the training sequence was used to find wide-baseline correspondences. First, initial features were selected using the ST or LK selection mechanism. The purpose of this was to find the most promising locations based on the same criteria used in the tracking phase. In the case of affine tracking, the ST tracking algorithm then performed affine warping in a neighborhood around each candidate point, seeking to minimize the discrepancy between this point and each stored in the training phase. In the translational case, no such warping was applied. The quality of a candidate match was calculated by finding the projection distance of this patch onto the basis of the descriptor using (5). Finally, candidate matches that fell below a threshold distance were selected, and the best among those was chosen as the matching location on the test image.

The results for matching are displayed in the figures using an image of the training sequence for reference, but the matching process does not use that image. Rather, it matches the descriptors derived from all of the video frames. Any image in the training sequence could be used for this visualization.

It is important to note a few aspects of our experiments. First, we are only concerned with feature *selection* or *tracking* insofar as they influence the ex-

perimental quality. Any consistent feature selector and tracker can be used to estimate the candidate points, or even no feature tracker at all (the entire image or a neighborhood could be searched, for example). For computational reasons, we used the ST selector and tracker for the experiments, which proved robust enough for our purposes.

A number of tuning parameters are present in the algorithms. For the LK and ST trackers, one must choose a threshold of point selection, which bounds the minimum of the eigenvalues of the second moment matrix of the image around the point. This was set, along with a minimum spacing between points of 10 pixels, such that about 50 points were selected on the object of interest. During tracking, a pyramid of scales is used to impose scale invariance. We chose to calculate three levels of this pyramid in all experiments, hence the pyramid included the image and downsampled versions 1/2 and 1/4 of its original size. For the KPCA algorithm, the tuning parameters are S, the size of the image patches, N, the number of kernel principal components to keep, L, the number of approximate patches to keep (to support the approximate kernel principal component basis), and σ, the bandwidth of the Gaussian kernel. We used $S = 31 \times 31$, $N = 8$ for the translational case, $N = 3$ for the affine case, $L = 4$, and $\sigma = 70$. These were selected by experiment.

While we did not optimize our experiments for speed or programming efficiency, we found the average time for tracking between adjacent frames to be approximately 10 seconds. The code was executed in Matlab on a 1.7GHz Pentium IV processor. The video frames and test images were 640x480 8-bit greyscale pixels, and about 50 feature locations were tracked every frame. The wide baseline matching had similar time requirements, taking up to twenty minutes to match 50 features, since a brute force combinatorial search was used. Optimizations in compiled code, as well as search heuristics, would increase these speeds dramatically. We have developed a C++ version of the tracking phase of this code, which runs at speeds of greater than 15Hz on the same hardware.

The figures in the following pages show a selection of many experiments. In all figures, lines link corresponding points between views. The results were pruned of matches outside the relevant objects to make the figures less cluttered. When comparing the choice of tracker, we found that affine tracking required fewer principal components than translational tracking to produce similar correspondence rates. When attempting to match scenes that are rotated or scaled with respect to the training sequence, the affine tracking scheme has a clear advantage, since these domain transformation are explicitly accounted for by the tracking mechanism. Such variability could not be represented in the translational case unless it was observed in the training sequence.

4 Concluding Remarks

We have presented a novel method for extracting feature descriptors from image sequences and matching these to new views of a scene. Rather than derive invariance completely from a model, our system learns the variability in the images

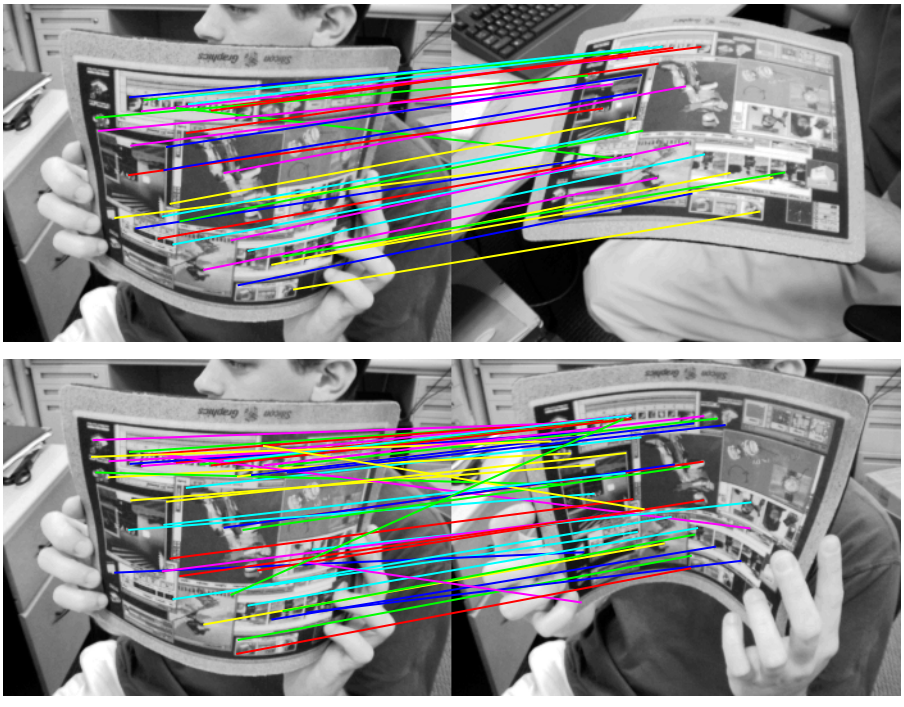

Fig. 1. Affine tracking + KPCA: A non-planar surface undergoing warping and viewpoint changes. (Top-left) The first image of the training sequence. (Top-right) The test image, outside of the training sequence. 27 feature locations were correctly matched, with 2 false positives. (Bottom-left) Image from the training sequence. (Bottom-right) The warped object. 40 locations correctly matched, 6 false positives.

directly from data. This technique is applicable in situations where such data is already available, such as robot navigation, causal structure from motion, face tracking, and object recognition.

There are a number of ways in which our system can be extended. Because a kernel technique is used, we must approximate the basis comprising the feature descriptor with virtual inputs. The best way to do this remains an open problem, and we are investigating other methods in addition to the one presented here ([16, 25], for example). When tracking and matching, we use the ST selector to provide an initial guess for feature locations. While convenient, this may not be the best choice, and we are experimenting with the use of more modern feature selectors ([20,23,28]). In cases where the observed scene is known to be rigid, robust structure from motion techniques (RANSAC or a robust Kalman Filter, for example) can be used to remove incorrect correspondences and suggest potential feature locations. Finally, the experimental code must be translated into more efficient form, allowing it to be used in near real-time on mobile platforms.

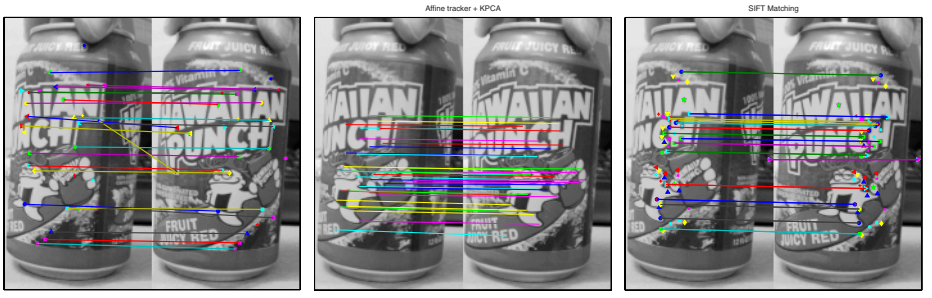

Fig. 2. Translational, Affine, and SIFT: The figures show the results of matching using our algorithms and SIFT on a rotating can. (Left) The translational Multi-View technique correctly matches a number of features near the edges of the can. There are 22 correct correspondences and 8 false positives. (Center) The affine Multi-View matches 32 locations and zero outliers, but with fewer matches along the edges. (Right) SIFT correctly matches more locations overall, but toward the center portion of the can where the transformation is approximately affine. Some lines between matches are included for clarity.

Fig. 3. Translational + KPCA: Matching using a non-planar surface undergoing warping and scale changes. During training, the object was moved away from the camera and deformed. (Left) shows the first image of the training sequence. (Right) shows the test image, which is outside of the training sequence. The shapes and colors indicate the corresponding points. In this example, 50 feature locations were correctly matched, with 10 false positives.

Acknowledgements. This work was conducted while the first author was an intern at the Honda Research Institute in 2003. Meltzer and Soatto are supported in part by the following grants: AFOSR F49620-03-1-0095, ONR N00014-03-1-0850, NSF ECS-0200511, and NSF CCR-0121778. The authors thank David Ross, David Lowe, and the Intel OpenCV project for providing code.

References

1. A. Baumberg. "Reliable feature matching across widely separated views," *Proc. CVPR,* 2000.
2. P.Belhumeur, J.Hespanha, D.Kriegman. "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *Proc. ECCV*, 1996.
3. C. Bishop. *Neural Networks for Pattern Recognition.* Oxford U. Press, 1995.
4. J. Bunch and C. Nielsen. "Updating the singular value decomposition." *Numerische Mathematik,* 31:111–129, 1978.
5. G. Carneiro and A. Jepson. "Phase-based local features." *Proc ECCV,* 2002.
6. T. Cootes, G. Wheeler, K .Walker and C. Taylor. "View-Based Active Appearance Models." *Image and Vision Computing,* Vol.20, 2002, pp. 657–664.
7. Y. Dufournaud, C. Schmid, R. Horaud. "Matching images with different resolutions," *Proc. CVPR*, 2000.
8. S. Edelman, N. Intrator, and T. Poggio. "Complex cells and object recognition," unpublished manuscript, 1997.
9. V. Ferrari, T. Tuytelaars and L. Van Gool. "Wide-baseline muliple-view correspondences." *Proc. CVPR,* Madison, USA, June 2003.
10. A. Fitzgibbon and A. Zisserman. "Joint manifold distance: a new approach to appearance based clustering." *Proc. CVPR*, 2003.
11. B. Frey and N. Jojic. "Transformed Component Analysis: Joint Estimation of Spatial Transformations and Image Components." *Proc. ICCV*, 1999.
12. C. Harris and M. Stephens. "A combined corner and edge detector." *Alvey Vision Conference,* 1988.
13. H. Jin, S. Soatto, A. Yezzi. "Multi-view stereo beyond Lambert." *CVPR,* 2003.
14. A. Johnson and M. Herbert. "Object recognition by matching oriented points." *CVPR,* 1997.
15. J. Koenderink and A. van Doorn. "Generic neighborhood operators," *IEEE Trans. Pattern Analysis and Machine Intell.,* vol. 14, pp. 597–605, June 1992.
16. J. Kwok and I. Tsang. "The Pre-Image Problem in Kernel Methods." *Proc. 20th Int. Conf. on Machine Learning,* 2003.
17. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, M. Jordan. "Learning the kernel matrix with semi-definite programming." *Proc. 19th Int. Conf. on Machine Learning,* Sydney, Australia, 2002.
18. A. Levy, M. Lindenbaum. "Sequential Karhunen-Loeve Basis Extraction and its Application to Images." *IEEE Trans. on Image Processing,* Aug. 2000.
19. T. Lindeberg. *Scale-Space Theory in Computer Vision.* Boston: Kluwer Academic Publishers, 1994.
20. D. Lowe. "Distinctive image features from scale-invariant keypoints," Preprint, submitted to *IJCV.* Version date: June 2003.
21. D. Lowe. "Object recognition from local scale-invariant features," *Proc. ICCV,* Corfu, Greece, September 1999.
22. K. Mikolajczyk and C. Schmid. "Indexing based on scale invariant interest points." *Proc. 8th ICCV,* 2001.
23. K. Mikolajczyk, C. Schmid. "An affine invariant interest point detector." *Proc. ECCV,* 2002.
24. K. Mikolajczyk and C. Schmid. "A performance evaluation of local descriptors." *Proc. CVPR,* June 2003.
25. D. Pavlov, D. Chudova, and P. Smyth. "Towards scalable support vector machines using squashing." *Proc. Int. Conf. on Knowledge Discovery in Databases,* 2000.

26. P. Pritchett and A. Zisserman. "Wide baseline stereo matching." *6th ICCV,* 1998.
27. F. Rothganger *et al.* "3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints." *Proc. CVPR,* June 2003.
28. F. Schaffalitzky and A. Zisserman. "Viewpoint invariant texture matching and wide baseline stereo," *Proc. ICCV,* Jul 2001.
29. F. Schaffalitzky and A. Zisserman. "Multi-view matching for unordered image sets, or 'How do I organize my holiday snaps?' " *7th ECCV,* Copenhagen, 2002.
30. B. Schölkopf, A. Smola. *Learning with Kernels.* Cambridge: The MIT Press, 2002.
31. B. Schölkopf, A. Smola, K. Müller. "Nonlinear component analysis as a kernel eigenvalue problem." *Neural Computation*, 10, 1299–1319, 1998.
32. B. Schölkopf et al. "Input Space vs. Feature Space in Kernel-Based Methods," *IEEE Transactions on Neural Networks.* 1999.
33. B. Schölkopf, P. Knirsch, A. Smola and C. Burges, "Fast Approximation of Support Vector Kernel Expansions, and an Interpretation of Clustering as Approximation in Feature Spaces", *DAGM Symposium Mustererkennung,* Springer Lecture Notes in Computer Science, 1998.
34. C. Schmid and R. Mohr. "Local Greyvalue Invariants for Image Retrieval." *Pattern Analysis and Machine Intelligence,* 1997.
35. J. Shi and C. Tomasi. "Good Features to Track," *IEEE CVPR,* 1994.
36. D. Tell and S. Carlsson. "Wide Baseline Point Matching Using Affine Invariants Computed from Intensity Profiles." *Proc. 6th ECCV,* 2000.
37. C. Tomasi, T. Kanade. "Detection and tracking of point features." Tech. Rept. CMU-CS-91132. Pittsburgh: Carnegie Mellon U. School of Computer Science, 1991.
38. T. Tuytelaars and L. Van Gool. "Wide Baseline Stereo based on Local, Affinely invariant Regions," *British Machine Vision Conference,* pp. 412–422, 2000.
39. T. Tuytelaars, and L. Van Gool. "Matching Widely Separated Views based on Affine Invariant Regions," to appear in *Int. J. on Computer Vision,* 2004.
40. R. Zabih and J. Woodfill. "Non-Parametric Local Transforms for Computing Visual Correspondence." *Proc. ECCV,* 1994.