Open access • Book Chapter • DOI:10.1007/978-3-540-88690-7_3

SIFT Flow: Dense Correspondence across Different Scenes — Source link

Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic ...+1 more authors

Institutions: Massachusetts Institute of Technology, École Normale Supérieure

Published on: 12 Oct 2008 - European Conference on Computer Vision

Topics: Motion field, Optical flow, Image registration, Scale-invariant feature transform and Matching (graph theory)

SIFT Flow: Dense Correspondence across Different Scenes

Ce Liu[1] Jenny Yuen[1] Antonio Torralba[1] Josef Sivic[2]
William T. Freeman[1,3]

[1]Massachusetts Institute of Technology [2]INRIA/Ecole Normale Supérieure*
[3]Adobe Systems
{celiu, jenny, torralba, billf}@csail.mit.edu josef@di.ens.fr

Abstract. While image registration has been studied in different areas of computer vision, aligning images depicting different scenes remains a challenging problem, closer to recognition than to image matching. Analogous to optical flow, where an image is aligned to its temporally adjacent frame, we propose *SIFT flow*, a method to align an image to its neighbors in a large image collection consisting of a variety of scenes. For a query image, histogram intersection on a bag-of-visual-words representation is used to find the set of nearest neighbors in the database. The SIFT flow algorithm then consists of matching densely sampled SIFT features between the two images, while preserving spatial discontinuities. The use of SIFT features allows robust matching across different scene/object appearances and the discontinuity-preserving spatial model allows matching of objects located at different parts of the scene. Experiments show that the proposed approach is able to robustly align complicated scenes with large spatial distortions. We collect a large database of videos and apply the SIFT flow algorithm to two applications: (i) motion field prediction from a single static image and (ii) motion synthesis via transfer of moving objects.

1 Introduction

Image alignment and registration is a central topic in computer vision. For example, aligning different views of the same scene has been studied for the purpose of image stitching [2] and stereo matching [3]. The considered transformations are relatively simple (*e.g.* parametric motion for image stitching and 1D disparity for stereo) and images to register are typically assumed to have the same pixel value after applying the geometric transformation.

The image alignment problem becomes more complicated for dynamic scenes in video sequences, as is the case of optical flow estimation [4–6], shown in Fig. 1(1). The correspondence problem between two adjacent frames in the video

* WILLOW project-team, Laboratoire d'Informatique de l'Ecole Normale Superieure, CNRS/ENS/INRIA UMR 8548

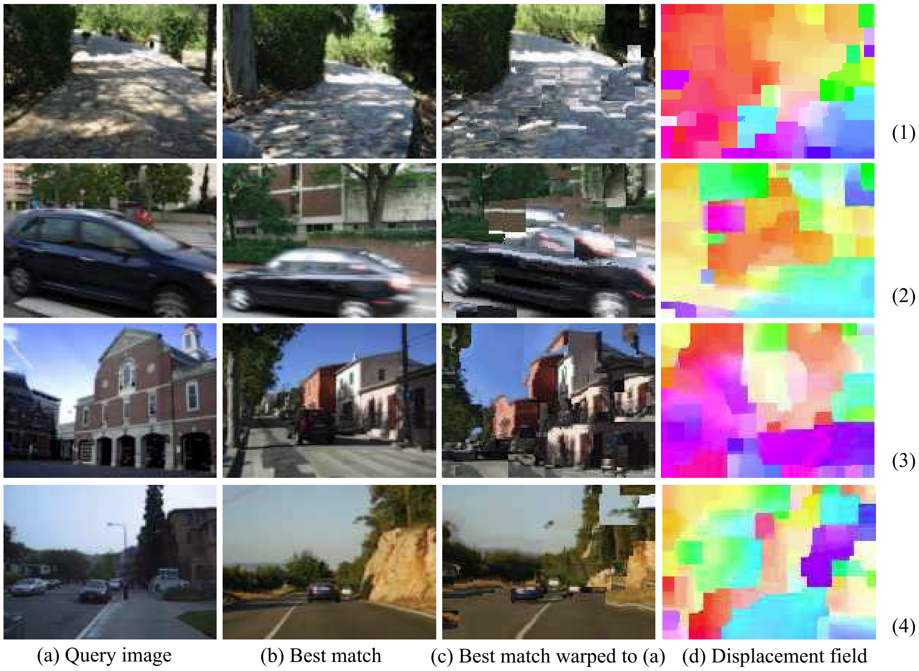


Fig. 1. Scene alignment using SIFT flow. (a) and (b) show images of similar scenes. (b) was obtained by matching (a) to a large image collection. (c) shows image (b) warped to align with (a) using the estimated dense correspondence field. (d) Visualization of pixel displacements using the color-coding scheme of [1]. Note the variation in scene appearance between (a) and (b). The visual resemblance of (a) and (c) demonstrates the quality of the scene alignment.

is often formulated as an estimation of a 2D flow field. The extra degree of freedom (from 1D in stereo to 2D in optical flow) introduces an additional level of complexity. Typical assumptions in optical flow algorithms include brigtness constancy and piecewise smoothness of the pixel displacement field.

Image alignment becomes even more difficult in the object recognition scenario, where the goal is to align different instances of the same object category, as illustrated in Fig. 1(2). Sophisticated object representations [7–10] have been developed to cope with the variations in objects' shape and appearance. However, the methods still typically require objects to be salient and large, visually very similar and with limited background clutter.

In this work, we are interested in a seemingly impossible task of aligning images depicting different instances of the same *scene category*. The two images to match may contain different object instances captured from different viewpoints, placed at different spatial locations, or imaged at different scales. In addition, some objects present in one image might be missing in the other image. Due to these issues the scene alignment problem is extremely challenging, as illustrated in Fig. 1(3) and 1(4).

Inspired by the recent progress in large image database methods [11–13], and the traditional optical flow estimation for temporally adjacent (and thus visually similar) frames, we create a large database so that for each query image we can retrieve a set of visually similar scenes. Next, we introduce a new alignment algorithm, dubbed *SIFT flow*, to align the query image to each image in the retrieved set. In the SIFT flow, a SIFT descriptor [14] is extracted at each pixel to characterize local image structures and encode contextual information. A discrete, discontinuity preserving, optical flow algorithm is used to match the SIFT descriptors between two images. The use of SIFT features allows robust matching across different scene/object appearances and the discontinuity-preserving spatial model allows matching of objects located at different parts of the scene. As illustrated in Fig. 1(3) and Fig. 1(4), the proposed alignment algorithm is able to estimate dense correspondence between images of complex scenes.

We apply SIFT flow to two original applications, which both rely on finding and aligning images of similar scenes in a large collection of images or videos. The first application is motion prediction from a single static image, where a motion field is hallucinated for an input image using a large database of videos. The second application is motion transfer, where we animate a still image using object motions transferred from a similar moving scene.

The rest of the paper is organized as follows: section 2 starts with introducing the concept of SIFT flow and describing the collected video database. Subsection 2.1 then describes the image representation for finding initial candidate sets of similar scenes. Subsection 2.2 details the SIFT flow alignment algorithm and subsection 2.3 shows some image alignment results. Applications of scene alignment to motion prediction and motion transfer are given in section 3.

2 Scene alignment using SIFT flow

We are interested in finding dense correspondences between the query image and its nearest neighbours found in a large database of images. Ideally, if the database is large enough to contain almost every possible image in the world, the nearest neighbours will be visually similar to the query image. This motivates the following analogy with optical flow, where correspondence is sought between temporally adjacent (and thus visually similar) video frames:

$$\text{Dense sampling in time : optical flow}$$
$$\text{Dense sampling in the space of all images : scene alignment using SIFT flow}$$

In other words, as optical flow assumes dense sampling of the time domain to enable tracking, SIFT flow assumes dense sampling in (some portion of) the space of natural images to enable scene alignment. In order to make this analogy possible we collect a large database consisting of 102,206 frames from 731 videos. Analogous to the time domain, we define the "temporal frames" to a query image as the N nearest neighbors in this database. The SIFT flow is then established between the query image and the N nearest neighbors. These two steps will be discussed in the next two subsections.

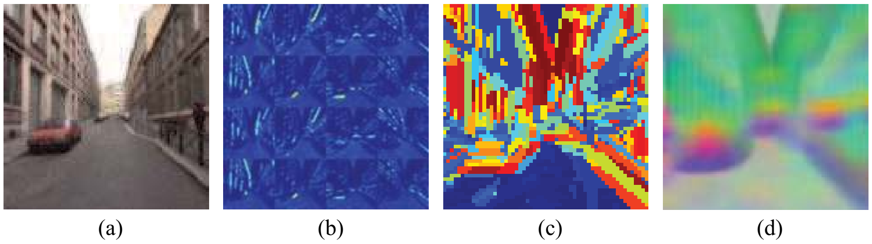


Fig. 2. Visualization of SIFT descriptors. We compute the SIFT descriptors on a regular dense grid. For each pixel in an image (a), the descriptor is a 128-D vector. The first 16 components are shown in (b) in a 4×4 image grid, where each component is the output of a signed oriented filter. The SIFT descriptors are quantized into visual words in (c). In order to improve the clarity of the visualization by mapping similar cluster centers to similar colors, cluster centers have been sorted according to the first principal component of the SIFT descriptor obtained from a large sample of our dataset. An alternative visualization of the continuous values of the SIFT descriptor is shown in (d). This visualization is obtained by mapping the first three principal components of each descriptor into the principal components of the RGB color space (*i.e.* the first component is mapped into R+G+B, the second is mapped into R-G and the third into R/2 + G/2-B). We will use (d) as our visualization of SIFT descriptors for the rest of the paper. Notice that visually similar image regions have similar colors.

2.1 Scene matching with histogram intersection

We use a fast indexing technique in order to gather candidate frames that will be further aligned using the SIFT flow algorithm to match the query image.

As a fast search, we use spatial histogram matching of quantized SIFT [14, 15]. First, we build a dictionary of 500 visual words [16] by running K-means on 5000 SIFT descriptors randomly selected out of all the video frames in our dataset. Then, the visual words are binned using a two level spatial pyramid [15, 17]. Fig. 2 shows visualizations of the high dimensional SIFT descriptors.

The similarity between two images is measured by histogram intersection. For each input image, we select the top 20 nearest neighbors. Matching is performed on all the frames from all the videos in our dataset. We then apply SIFT flow between the input image and the top 20 candidate neighbors and re-rank the neighbors based on the alignment score (described below). The frame with the best alignment score is chosen from each video.

This approach is well-matched to the similarity obtained by SIFT flow (described below) as it uses the same basic features (SIFT descriptors) and spatial information is loosely represented (by means of the spatial histograms).

2.2 The SIFT flow algorithm

As shown in Fig. 1, images of distinct scenes can be drastically different in both RGB values and their gradients. In addition, the magnitude of pixel displacements between potentially corresponding objects or scene parts can be much larger than typical magnitudes of motion fields for temporal sequences. As a result, the brightness constancy and coarse-level zero flow assumptions common in classical optical flow [4–6] are no longer valid. To address these issues, we modify the standard optical flow assumptions in the following way. First, we assume SIFT descriptors [14] extracted at each pixel location (instead of raw pixel values) are constant with respect to the pixel displacement field. As SIFT descriptors characterize view-invariant and brightness-independent image structures, matching SIFT descriptors allows establishing meaningful correspondences across images with significantly different image content. Second, we allow a pixel in one image to match any other pixel in the other image. In other words, the pixel displacement can be as large as the image itself. Note, however, that we still want to encourage smoothness (or spatial coherence) of the pixel displacement field by encouraging close-by pixels to have similar displacements.

We formulate the correspondence search as a discrete optimization problem on the image lattice [18, 19] with the following cost function

$$E(\mathbf{w}) = \sum_{\mathbf{p}} \big\| s_1(\mathbf{p}) - s_2(\mathbf{p}+\mathbf{w}) \big\|_1 + \frac{1}{\sigma^2} \sum_{\mathbf{p}} \Big(u^2(\mathbf{p}) + v^2(\mathbf{p}) \Big) + \sum_{(\mathbf{p},\mathbf{q}) \in \varepsilon} \min\Big(\alpha |u(\mathbf{p}) - u(\mathbf{q})|, d \Big) + \min\Big(\alpha |v(\mathbf{p}) - v(\mathbf{q})|, d \Big), \qquad (1)$$

where $\mathbf{w}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$ is the displacement vector at pixel location $\mathbf{p} = (x, y)$, $s_i(\mathbf{p})$ is the SIFT descriptor extracted at location $\mathbf{p}$ in image i and ε is the spatial neighborhood of a pixel (here a 4-neighbourhood structure is used). Parameters $\sigma = 300$, $\alpha = 0.5$ and $d = 2$ are fixed in our experiments. The optimization is performed using the tree-reweighted message passing algorithm (TRW-S) [20]. In the above objective function, L1 norm is employed in the first term to account for outliers in SIFT matching and a thresholded L1 norm is used in the third, regularization term to model discontinuities of the pixel displacement field. In contrast to the rotation-invariant robust flow regularizer used in [21], the regularization term in our model is decoupled and rotation dependent so that the computation is feasible for large displacements. Unlike [19] where a quadratic regularizer is used, the thresholded L1 regularizer in our model can preserve discontinuities. As the regularizer is decoupled for u and v, the complexity of the message passing algorithm can be reduced from $O(L^3)$ to $O(L^2)$ using the distance transform [22], where L is the size of the search window. This is a significant speedup since L is large (we allow a pixel in the query image to match to a 80×80 neighborhood). We also use the bipartite message passing scheme and a multi-grid as proposed in [22]. The message passing converges in 60 iterations for a 145×105 image, which is about 50 seconds on a quad-core Intel Xeon 2.83GHz machine with 16GB memory using a C++ implementation.

2.3 Scene alignment results

We conducted several experiments to test the SIFT flow algorithm on our video database. One frame from each of the 731 videos was selected as the query image and histogram intersection matching (section 2.1) was used to find its 20

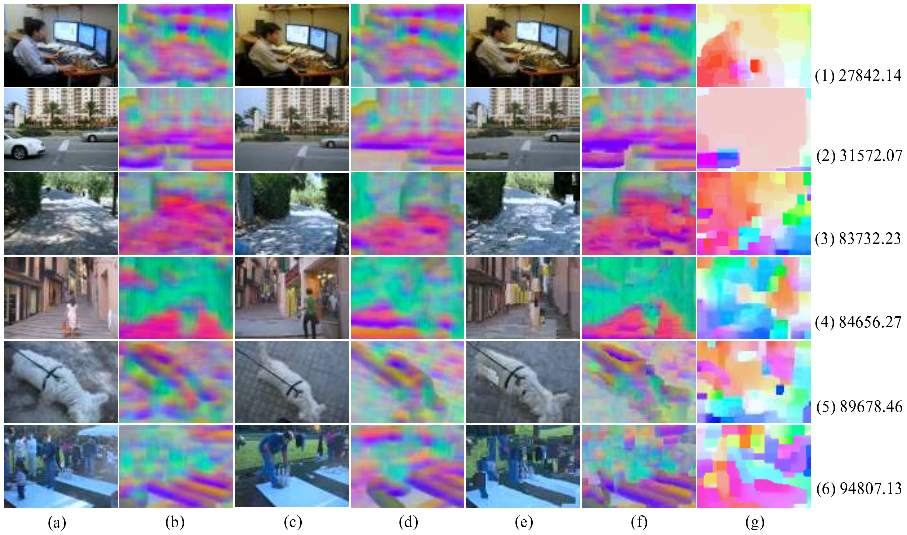


Fig. 3. SIFT flow for image pairs depicting the same scene/object. (a) shows the query image and (b) its densely extracted SIFT descriptors. (c) and (d) show the best (lowest energy) match from the database and its SIFT descriptors, respectively. (e) shows (c) warped onto (a). (f) shows the warped SIFT image (d). (g) shows the estimated displacement field with the minimum alignment energy shown to the right.

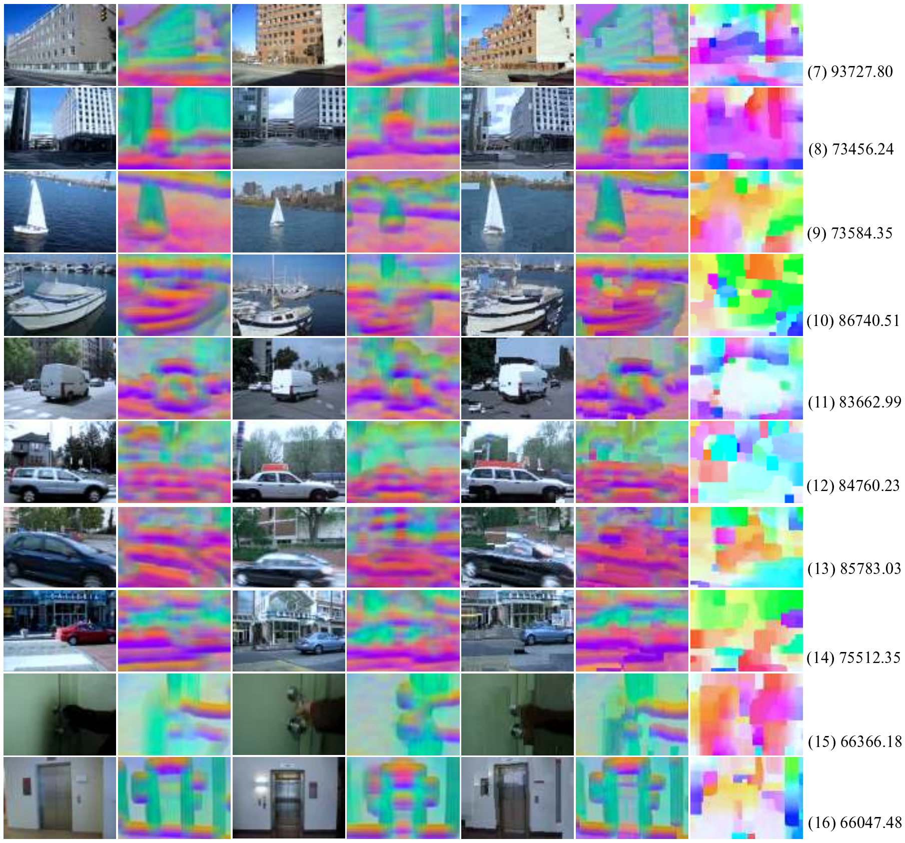


Fig. 4. SIFT flow computed for image pairs depicting the same scene/object category where the visual correspondence is obvious.

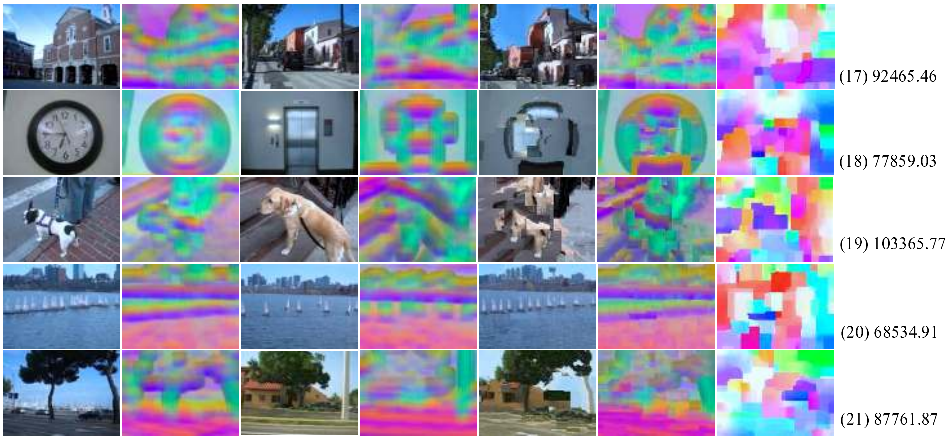


Fig. 5. SIFT flow for challenging examples where the correspondence is not obvious.

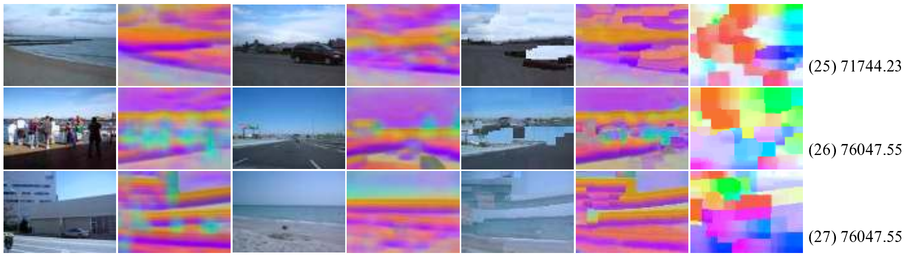


Fig. 6. Some failure examples with incorrect correspondences.

nearest neighbors, excluding all other frames from the query video. The scene alignment algorithm (section 2.2) was then used to estimate the dense correspondence (represented as a pixel displacement field) between the query image and each of its neighbors. The best matches are the ones with the minimum energy defined by (1). Alignment examples are shown in Figures 3–5. The original query image and its extracted SIFT descriptors are shown in columns (a) and (b). The minimum energy match (out of the 20 nearest neighbors) and its extracted SIFT descriptors are shown in columns (c) and (d). To investigate the quality of the pixel displacement field, we use the computed displacements to warp the best match onto the query image. The warped image and warped SIFT descriptor image are shown in columns (e) and (f). The visual similarity between (a) and (e), and (b) and (f) demonstrates the quality of the matching. Finally, the displacement field is visualized using color-coding adapted from [1] in column (g) with the minimum alignment energy shown to the right. Fig. 3 shows examples of matches between frames coming from the same video sequence. The almost perfect matching in (1) and (2) demonstrates that SIFT flow reduces to classical optical flow when the two images are temporally adjacent frames in a video sequence. In (3)–(5), the query and the best match are more distant within the video sequence, but the alignment algorithm can still match them reasonably

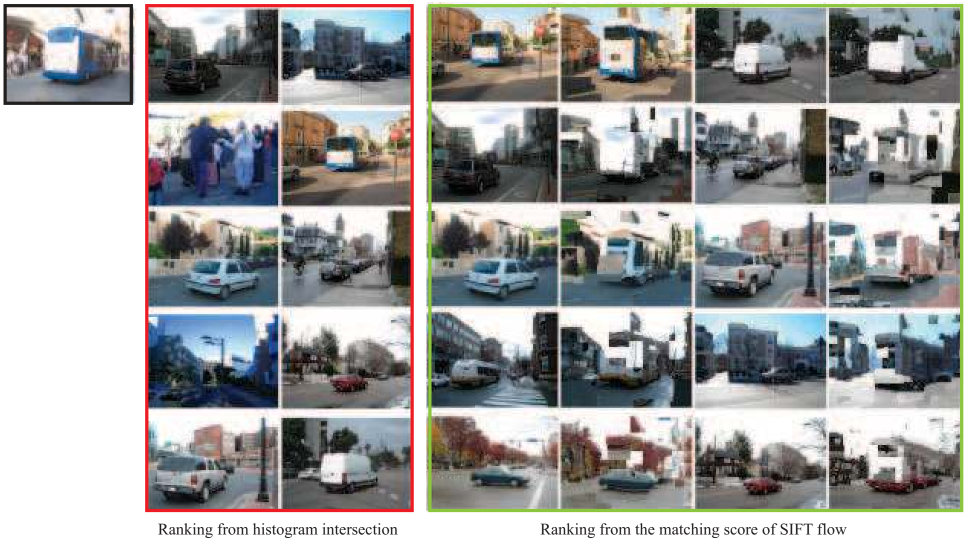

Ranking from histogram intersection
Ranking from the matching score of SIFT flow

Fig. 7. Alignment typically improves ranking of the nearest neighbors. Images enclosed by the red rectangle are the top 10 nearest neighbors found by histogram intersection, displayed in a scan-line order (left to right, top to bottom). Images enclosed by the green rectangle are the top 10 nearest neighbors ranked by the minimum energy obtained by the alignment algorithm. The warped nearest neighbor image is displayed to the right of the original image. Note how the returned images are re-ranked according to the size of the depicted vehicle by matching the size of the bus in the query.

well. Fig. 4 shows more challenging examples, where the two frames come from different videos while containing the same type of objects. The alignment algorithm attempts to match the query image by transforming the candidate image. Note the significant changes in viewpoint between the query and the match in examples (8), (9), (11), (13), (14) and (16). Note also that some discontinuities in the flow field are caused by errors in SIFT matching. The square shaped discontinuities are a consequence of the decoupled regularizer on the horizontal and vertical components of the pixel displacement vector. Fig. 5 shows alignment results for examples with no obvious visual correspondence. Despite the lack of direct visual correspondence, the scene alignment algorithm attempts to rebuild the house (17), change the shape of the door into a circle (18) or reshuffle boats (20). Some failure cases are shown in Fig. 6. Typically, these are caused by the lack of visually similar images in the video database. Note that, typically, alignment improves ranking of the K-nearest neighbors. This is illustrated in Fig. 7.

3 Applications

In this section we demonstrate two applications of the proposed scene matching algorithm: (1) motion field prediction from a single image using motion priors, and (2) motion synthesis via transfer of moving objects common in similar scenes.

3.1 Predicting motion field from a single image

The goal is, given a single static image, to predict what motions are plausible in the image. This is similar to the recognition problem, but instead of assigning labels to each pixel, we want to assign possible motions.

We built a scene retrieval infrastructure to query still images over a database of videos containing common moving objects. The database consists of sequences depicting common events, such as cars driving through a street and kids playing in a park. Each individual frame was stored as a vector of word-quantized SIFT features, as described in section 2.1. In addition, we store the temporal motion field between every two consecutive frames of each video.

We compare two approaches for predicting the motion field for the query still image. The first approach consists of directly transferring the motion of the closest video frame matched in the database. Using the SIFT-based histogram matching (section 2.1), we can retrieve very similar video frames that are roughly spatially aligned. For common events such as cars moving forward on a street, the motion prediction can be quite accurate given enough samples in the database. The second approach refines the coarse motion prediction described above using the dense correspondences obtained by the alignment algorithm (section 2.2). In particular, we compute the SIFT flow from the retrieved video frame to the query image and use the computed correspondence to warp the temporally estimated motion of the retrieved video frame. Figure 8 shows examples of predicted motion fields directly transferred from the top 5 database matches and the warped motion fields. Note that in simple cases the direct transfer is already quite accurate and the warping results only in minor refinements.

While there are many improbable flow fields (*e.g.* a car moving upwards), each image can have multiple plausible motions : a car or a boat can move forward, in reverse, turn, or remain static. In any scene the camera motion can generate motion field over the entire frame and objects can be moving at different velocities. Figure 9 shows an example of 5 motion fields predicted using our video database. Note that all the motions fields are different, but plausible.

3.2 Quantitative evaluation

Due to the inherent ambiguity of multiple plausible motions for each still image, we design the following procedure for quantitative evaluation. For each test video, we randomly select a test frame and obtain a *result* set of top n inferred motion fields using our motion prediction method. Separately, we collect an *evaluation* set containing the temporally estimated motion (from video) for the test frame (the closest to a ground truth we have) and 11 random motion fields taken from other scenes in our database, acting as distractors. We take each of the n inferred motion fields from the result set and compute their similarity (defined below) to the set of evaluation fields. The rank of the ground truth motion with respect to the random distractor motions is an indicator of how close the predicted motion is to the true motion estimated from the video sequence. Because there are many possible motions that are still realistic, we do this comparison with each of the

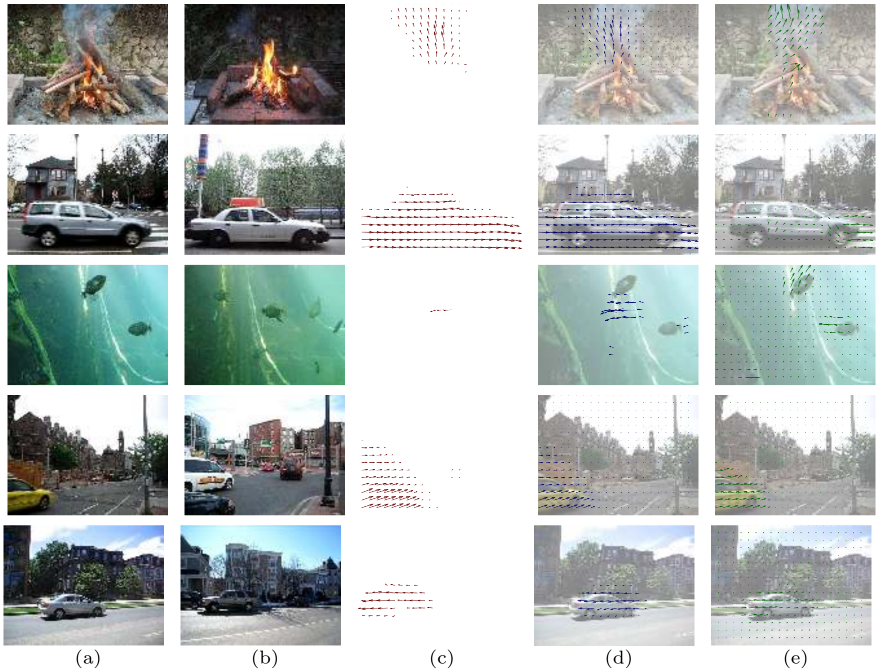


Fig. 8. Motion from a single image. The (a) original image, (b) matched frame from the video data set, (c) motion of (b), (d) warped and transferred motion field from (b), and (e) ground truth for (a). Note that the predicted motion in (d) is inferred from a single input still image, i.e. no motion signal is available to the algorithm. The predicted motion is based on the motion present in other videos with image content similar to the query image.

top n motion fields within the result set and keep the highest ranking achieved. Finally, we repeat this evaluation ten times with a different randomly selected test frame for each test video and report the median of the rank score across the different trials.

For this evaluation, we represent each motion field as a regular two dimensional motion grid filled with 1s where there is motion and 0 otherwise. The similarity between two motion fields is defined then as

$$S(\mathbf{M},\mathbf{N}) \stackrel{\text{def}}{=} \sum_{(x,y)\in G} \Big(\mathbf{M}(x,y) = \mathbf{N}(x,y)\Big) \tag{2}$$

where $\mathbf{M}$ and $\mathbf{N}$ are two rectangular motion grids of the same size, and (x, y) is a coordinate pair within the spatial domain G of grids $\mathbf{M}$ and $\mathbf{N}$.

Figure 10a shows the normalized histogram of these rankings across 720 predicted motion fields from our video data set. Figure 10b shows the same evaluation on a subset of the data that includes 400 videos with mostly streets and cars. Notice how, for more than half of the scenes, the inferred motion field is ranked 1st suggesting a close match to the temporally-estimated ground truth. Most other test examples are ranked within the top 5. Focusing on roads and

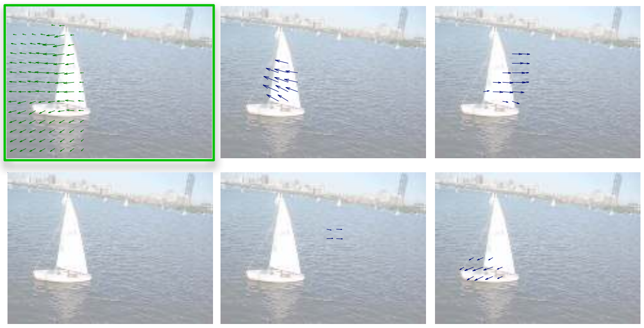
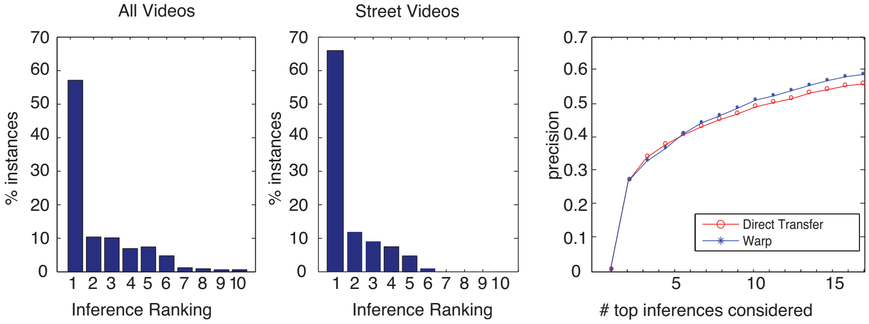


Fig. 9. Multiple motion field candidates. A still query image with its temporally estimated motion field (in the green frame) and multiple motion fields predicted by motion transfer from a large video database.



(a) (b) (c)

Fig. 10. Evaluation of motion prediction. (a) and (b) show normalized histograms of prediction rankings (result set size of 15). (c) shows the ranking precision as a function of the result set size.

cars gives even better results with 66% of test trials ranked 1st and even more test examples ranked within the top 5. Figure 10c shows the precision of the inferred motion (the percentage of test examples with rank 1) as a function of the size of the result set, comparing (i) direct motion field transfer (red circles) and (ii) warped motion field transfer using SIFT flow (blue stars).

While histograms of ranks show that the majority of the inferred motions were ranked 1st, there is still a significant number of instances with lower rank. Figure 11 shows a false negative example, where the inferred motion field was not ranked top despite the reasonable output. Notice how the top ranked distractor fields are quite similar to our prediction showing that, in some cases, where our prediction is not ranked 1st, we still produce realistic motion.

3.3 Motion synthesis via object transfer

We described above how to predict the direction and velocity of objects in a still image. Having a prior on what scenes look like over time also allows us to infer

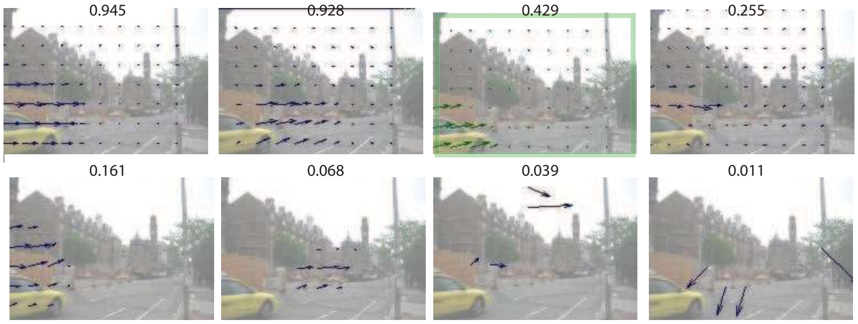


Fig. 11. Motion instances where the predicted motion was not ranked closest to the ground truth. A set of random motion fields (blue) together with the predicted motion field (green, ranked 3rd). The number above each image represents the fraction of the pixels that were correctly matched by comparing the motion against the ground truth. In this case, some random motion fields appear closer to the ground truth than our prediction (green). However, our prediction also represents a plausible motion for this scene.

what objects (that might not be part of the still image) can possibly appear. For example, a car moving forward can appear in a street scene with an empty road, or a fish can start swimming in a fish tank scene.

Based on this idea, we propose a method for synthesizing motions from a still image. The goal is to transfer moving objects from similar video scenes. In particular, given a still image q that is not part of any video in our database D, we identify and transfer moving objects from videos in D into q as follows:

1. Query D using the SIFT-based scene matching algorithm to retrieve the set of closest video frame matches $F = \{f_i | f_i$ is the ith frame from a video in $D\}$ given the query image q.
2. For each frame $f_i \in F$, we can synthesize a video sequence based on the still image q. The kth frame of the synthesized video is generated as follows:
   (a) Densely sample the motion from frame f_{i+k} to f_{i+k+1}
   (b) Construct frame q_k by transferring non-moving pixels from q and moving pixels from f_{i+k}.
   (c) Apply poisson editing [23] to blend the foreground (pixels from f_{i+k}) into the background composed of pixels from q.

Figure 12 shows examples of synthesized motions for three different scenes. Notice the variety of region sizes transferred and the seamless integration of objects into the new scenes.

Some of the biggest challenges in creating realistic composites lie in estimating the correct size and orientation of the objects to introduce in the scene [24]. Our framework inherently takes care of these constraints by retrieving sequences that are visually similar to the query image. This enables the creation of realistic motion sequences from still images with a simple transfer of moving objects.

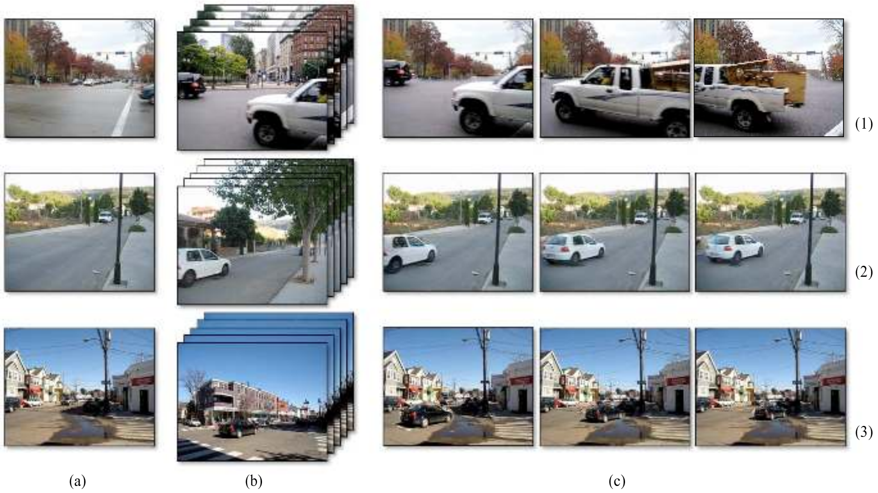


Fig. 12. Motion synthesis via object transfer. Query image (a), the top video match (b), and representative frames from the synthesized sequence (c) obtained by transferring moving objects from the video to the still query image.

4 Conclusion

We have introduced the concept of SIFT flow and demonstrated its utility for aligning images of complex scenes. The proposed approach achieves good matching and alignment results despite significant differences in appearance and spatial layout of matched images.

The goal of scene alignment is to find dense correspondence between similar structures (similar textures, similar objects) across different scenes. We believe that scene alignment techniques will be useful for various applications in both computer vision and computer graphics. We have illustrated the use of scene alignment in two original applications: (1) motion estimation from a single image and (2) video synthesis via the transfer of moving objects.

5 Acknowledgements

Funding for this work was provided by NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, NSF Career award IIS 0747120, NSF contract IIS-0413232, a National Defense Science and Engineering Graduate Fellowship, and gifts from Microsoft and Google.

References

1. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: Proc. ICCV. (2007)

2. Szeliski, R.: Image alignment and stiching: A tutorial. Foundations and Trends in Computer Graphics and Computer Vision **2**(1) (2006)
3. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Intl. J. of Computer Vision **47**(1) (2002) 7–42
4. Horn, B.K.P., Schunck, B.G.: Determing optical flow. Artificial Intelligence **17** (1981) 185–203
5. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the International Joint Conference on Artificial Intelligence. (1981) 674–679
6. Bruhn, A., Weickert, J., Schnörr, C.: Lucas/Kanade meets Horn/Schunk: combining local and global optic flow methods. Intl. J. of Computer Vision **61**(3) (2005) 211–231
7. Belongie, S., Malik, J., Puzicha, J.: Shape context: A new descriptor for shape matching and object recognition. In: NIPS. (2000)
8. Berg, A., Berg, T., Malik, J.: Shape matching and object recognition using low distortion correspondence. In: Proc. CVPR. (2005)
9. Felzenszwalb, P., Huttenlocher, D.: Pictorial structures for object recognition. Intl. J. of Computer Vision **61**(1) (2005)
10. Winn, J., Jojic, N.: Locus: Learning object classes with unsupervised segmentation. In: Proc. ICCV. (2005) 756–763
11. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Intl. J. of Computer Vision **77**(1-3) (2008) 157–173
12. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Transactions on Graphics (SIGGRAPH 2007) **26**(3) (2007)
13. Russell, B.C., Torralba, A., Liu, C., Fergus, R., Freeman, W.T.: Object recognition by scene alignment. NIPS (2007)
14. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proc. ICCV, Kerkyra, Greece (1999) 1150–1157
15. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. CVPR. Volume II. (2006) 2169–2178
16. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In: Proc. ICCV. (2003)
17. Grauman, K., Darrell, T.: Pyramid match kernels: Discriminative classification with sets of image features. In: Proc. ICCV. (2005)
18. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Pattern Analysis and Machine Intelligence **23**(11) (2001) 1222–1239
19. Shekhovtsov, A., Kovtun, I., Hlavac, V.: Efficient MRF deformation model for non-rigid image matching. In: Proc. CVPR. (2007)
20. Wainwright, M., Jaakkola, T., Willsky, A.: Exact MAP estimates by (hyper)tree agreement. In: NIPS. (2003)
21. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: Proc. ECCV. (2004) 25–36
22. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient belief propagation for early vision. Intl. J. of Computer Vision **70**(1) (2006) 41–54
23. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Trans. Graph. **22**(3) (2003) 313–318
24. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. ACM Transactions on Graphics (SIGGRAPH 2007) **26**(3) (August 2007)