Non-Rigid Dense Correspondence with Applications for Image Enhancement

Yoav HaCohen
Hebrew University

Eli Shechtman
Adobe Systems

Dan B Goldman
Adobe Systems

Dani Lischinski
Hebrew University

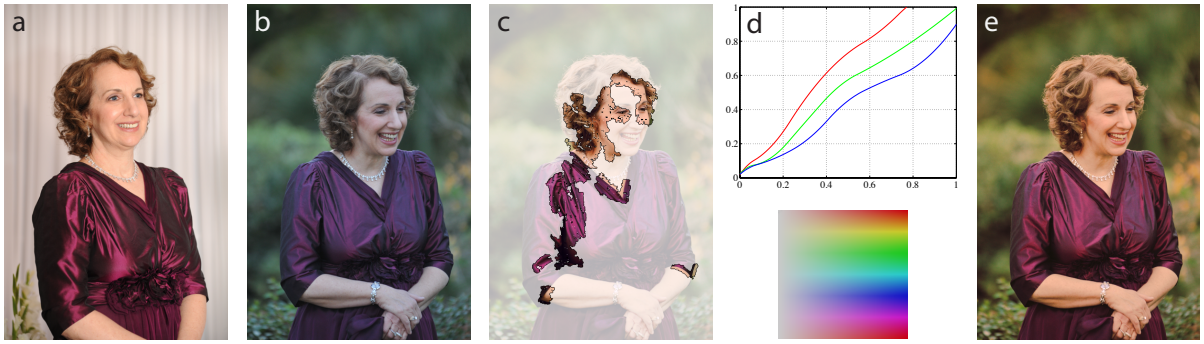


Figure 1: *Color transfer using our method. The reference image (a) was taken indoors using a flash, while the source image (b) was taken outdoors, against a completely different background, and under natural illumination. Our correspondence algorithm detects parts of the woman's face and dress as shared content (c), and fits a parametric color transfer model (d). The appearance of the woman in the result (e) matches the reference (a).*

Abstract

This paper presents a new efficient method for recovering reliable local sets of dense correspondences between two images with some shared content. Our method is designed for pairs of images depicting similar regions acquired by different cameras and lenses, under non-rigid transformations, under different lighting, and over different backgrounds. We utilize a new coarse-to-fine scheme in which nearest-neighbor field computations using Generalized PatchMatch [Barnes et al. 2010] are interleaved with fitting a global non-linear parametric color model and aggregating consistent matching regions using locally adaptive constraints. Compared to previous correspondence approaches, our method combines the best of two worlds: It is dense, like optical flow and stereo reconstruction methods, and it is also robust to geometric and photometric variations, like sparse feature matching. We demonstrate the usefulness of our method using three applications for automatic example-based photograph enhancement: adjusting the tonal characteristics of a source image to match a reference, transferring a known mask to a new image, and kernel estimation for image deblurring.

Keywords: correspondence, color transfer, PatchMatch, nearest neighbor field, deblurring

Links: DL PDF WEB

1 Introduction

Establishing correspondences between images is a long-standing problem with a multitude of applications in computer vision and graphics, ranging from classical tasks like motion analysis, tracking and stereo, through 3D reconstruction, object detection and retrieval, to image enhancement and video editing. Most existing correspondence methods are designed for one of two different scenarios. In the first scenario, the images are close to each other in time and in viewpoint, and a dense correspondence field may be established using optical flow or stereo reconstruction techniques. In the second, the difference in viewpoint may be large, but the scene consists of mostly rigid objects, where sparse feature matching methods, such as SIFT [Lowe 2004], have proven highly effective.

In this paper, we present a new method for computing a reliable dense set of correspondences between two images. In addition to the two scenarios mentioned above, our method is specifically designed to handle a third scenario, where the input images share some common content, but may differ significantly due to a variety of factors, such as non-rigid changes in the scene, changes in lighting and/or tone mapping, and different cameras and lenses. This scenario often arises in personal photo albums, which typically contain repeating subjects photographed under different conditions.

Our work is motivated by the recent proliferation of large personal digital photo collections and the tremendous increase in the number of digital photos readily available on the internet. Because of these trends, it has become increasingly possible to enhance and manipulate digital photographs by retrieving and using example or reference images with relevant content [Reinhard et al. 2001; Ancuti et al. 2008; Dale et al. 2009; Joshi et al. 2010; Snavely et al. 2006]. Many of these applications benefit from the ability to detect reliable correspondences between the input images. However, as pointed out earlier, existing correspondence methods may often find this task challenging.

For example, consider the task of color transfer from a *reference* image in Figure 1a to a *source* image 1b, which differs in illumination, background, and subject pose. Our method is able to automatically recover a set of dense correspondences between regions

that appear in both images (1c), and adjusts the source such that the result matches the tonal characteristics of the reference, as shown in Figure 1e. Figure 2 demonstrates that existing alternatives are not able to achieve a similar result on this image pair.

Our approach simultaneously recovers both a robust set of dense correspondences between sufficiently similar regions in two images (Figure 1c), and a global non-linear parametric color transformation model (Figure 1d). We extend the Generalized PatchMatch algorithm [Barnes et al. 2010] by making it robust to significant tonal differences between the images, and embed it in a new coarse-to-fine scheme, where the nearest-neighbor field computations are interleaved with color transformation model fitting. At each stage, we use aggregation of coherent regions and locally adaptive constraints to find regions within which the matches are consistent and to reject outliers before the next stage commences. Such regions are assumed to belong to content shared across the two images. In summary, our correspondence method attempts to combine the best of two worlds: It is dense, like optical flow and stereo reconstruction methods, and it is also robust to geometric and photometric variations, like sparse feature matching.

After discussing relevant previous work (Section 2) and presenting and evaluating our dense correspondence algorithm (Sections 3–4), we discuss several example-based image enhancement applications that benefit from our method (Section 5). Specifically, we first show how to improve the global color transfer implicit in our approach to produce a more locally-refined result. Next, we demonstrate the applicability of our approach for example-based image deblurring, and example-based mask transfer (foreground extraction).

2 Related Work

2.1 Correspondence

Initial correspondence methods were designed for stereo matching, optical flow and image alignment [Lucas and Kanade 1981]. These methods compute a dense correspondence field, but they are intended to operate on very similar images, typically assume brightness constancy and local motion, and tend to have errors in regions that appear in only one image.

The development of various local invariant features [Lowe 2004; Matas et al. 2002; Mikolajczyk et al. 2005] has brought about significant progress in this area. These features are robust to typical appearance variations (illumination, blur, compression), and a wide range of 3D transformations. Initial feature matching is often followed by geometric filtering steps (e.g., RANSAC using a rigid scene assumption [Lowe 2004], and geometric consistency assumptions [Cho et al. 2009]) that yield very reliable matches of 3D rigid scenes [Snavely et al. 2006]. However they are still considered less effective for matching non-rigid objects, people and scenes. In these cases both the detectors and the descriptors are less effective, and global rigid models cannot be generally applied. The Structure From Motion literature showed how sparse correspondences can be found and grouped for non-rigid objects [Zelnik-Manor and Irani 2006] (and references therein), however these methods are designed for multiple video frames with small motions. In this paper we show examples with significant non-rigidity and other appearance differences, where our dense correspondences work better than sparse feature matches, as shown in Figure 2.

More advanced methods combine sparse features with dense matching to cope with large-displacement optical flow [Brox et al. 2009], and non-rigid matching of highly different scenes [Liu et al. 2008a]. Although both demonstrated impressive dense correspondence results, they are not robust to significantly changes in scale and rotation. Our correspondence method is related to a family of methods that start with a few highly reliable feature matches, and then "densify" the correspondence around those points to obtain reliable corresponding regions [Ferrari et al. 2004; Cho et al. 2008]. However, these methods were demonstrated on a collection of rigid objects with very similar appearances. They were also applied on very coarse grids of features, and do not seem to scale well to dense pixel-to-pixel correspondences on large images. We show that our method outperforms [Cho et al. 2008] in Section 4.

[Pitié et al. 2007] SIFT Our result

Figure 2: *Failure of other methods to correctly transfer color from 1a to 1b. The SIFT column uses our parametric color transfer model, but recovered from a set of sparse SIFT correspondences.*

Our method builds upon Generalized PatchMatch (GPM) [Barnes et al. 2010], a fast randomized algorithm for finding a dense nearest neighbor field for patches that may undergo translations, rotations and scale changes. We show that GPM performs poorly on our examples, but good results can be obtained by combining it with a coarse-to-fine scheme, an iterative tonal and color correction of the input image, aggregation of consistent regions, and locally narrowing the search range of matched transformations.

2.2 Example-based enhancement

Over the years, there has been much work on the transfer of various appearance-related image attributes from one image to another. An and Pellacini [2010] provide a good survey of recent approaches. Several methods attempt to modify a *source* image by globally matching the color statistics of a *reference* image [Reinhard et al. 2001; Pitié et al. 2007]. Because the statistics of the *entire* source image are matched to those of the *entire* reference, even common content between two images may have widely varying appearance, as shown in Figure 2 (left).

Later methods have attempted to overcome this problem by using automatic co-segmentation and transferring color distributions between each pair of corresponding regions separately [Dale et al. 2009; Kagarlitsky et al. 2009]. However, such co-segmentation methods require the example and input images to be similar. Image alignment or optical flow algorithms can be used to overcome this requirement for static or small-motion cases, but none of the aforementioned works demonstrate results for more challenging cases.

The best color transfer results currently require user assistance, like that of An and Pellacini [2010] (and the references therein). In contrast, our approach provides an automatic solution for image pairs with shared content.

As large personal and online photo collections are becoming commonly available [Snavely et al. 2006], and methods for correspondence and recognition are becoming more mature and robust, new methods are being developed that utilize content-specific examples. For example, Liu *et al.* [2008b] and Dale *et al.* [2009] use global descriptors to retrieve similar examples from large online collec-

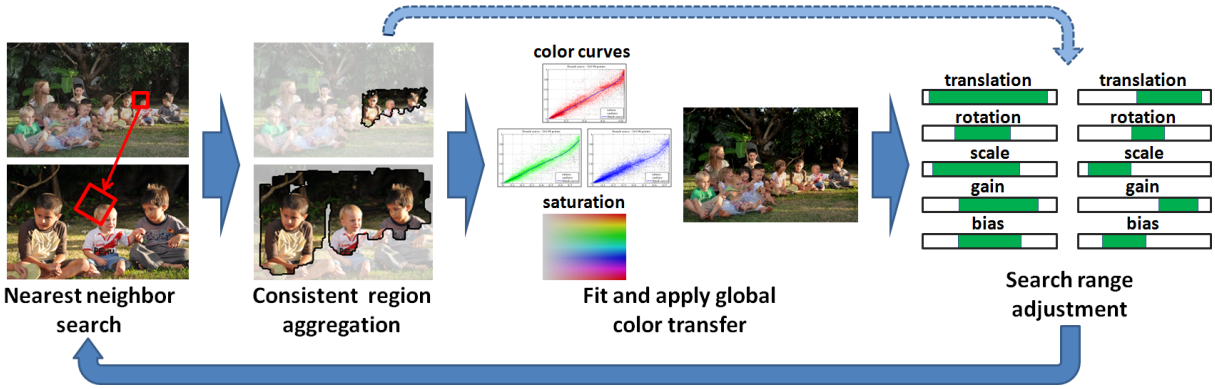


Figure 3: *The four steps of our correspondence algorithm - these are repeated several iterations at multiple scales.*

tion. Liu *et al.* [2008b] use a pixel-by-pixel approach to colorize grayscale images with edge-aware color propagation.

Other methods leverage common content in order to enhance images. Eisemann and Durand [2004] and Petschnigg *et al.* [2004] transfer appearance between flash and no-flash image pairs. Joshi *et al.* [2010] recognize recurrent faces in a personal photo collection and use these faces for deblurring and correcting lighting and color balance. Bhat *et al.* [2007] build a 3D model of the scene in order to enhance videos of static scenes using example photographs, and Eisemann *et al.* [2010] use SIFT correspondences to align static scenes for addition of high-resolution details and white-balancing.

Our method significantly extends the operating range in which this idea of using shared content to enhance images can be applied. Rather than specifically relying on the presence of a common face or a static scene, we provide a general method that robustly finds shared content, including but not limited to faces, people, static and non-static content.

3 Correspondence algorithm

Our goal is to find reliable dense correspondences between images that share some content but may differ in several scene or camera conditions. The dense region matching is non-parametric, aligning small patches transformed by simple geometric and photometric transformation to achieve robustness to such changes. We do not assume a planar or rigid scene, so matches can be found across significant changes of contents or pose. Jointly with region matching, we recover a global parametric color transformation model that combines per-channel nonlinear tone curves with a saturation matrix for cross-channel adjustments. This parametric model can extrapolate from the regions with known correspondence to other regions where the correspondence is unknown.

To recover this image correspondence model, we propose a coarse-to-fine algorithm that repeats the following four steps at each scale[1]: nearest-neighbor search, region aggregation, color transform fitting, and search range adjustment (Figure 3). First, for each patch in the source image, find its nearest neighbor in the reference image, searching over a constrained range of translations, scales, rotations, gain and bias values (Section 3.1). Second, aggregate consistent regions of matches (Section 3.2). Regions that pass a consistency test are considered to be reliable matches. Third, robustly fit a color transformation model based on these reliable consistent regions (Section 3.3). And fourth, adjust the search range for each degree of freedom of the nearest neighbor patch search in the next iteration (Section 3.4). This adjustment uses both the recovered patch correspondences and the color model to estimate plausible ranges for the color gain and bias. We repeat the coarse-to-fine procedure in two passes, in order to refine the model. See Algorithm 1 and Figure 3 for an overview of the algorithm.

[1]The factor between successive scales is $\sqrt{2}$, where the coarsest scale is chosen such that the smaller dimension is above 64 pixels.

Algorithm 1 *Non-Rigid Dense Correspondence Algorithm*

for scale = coarse **to** fine **do**
  for each patch $u \in S$ **do**
    Find a transformation $T^u = \arg\min_T \|S_u - R_{T(u)}\|_2$ (Sec. 3.1)
  end for
  Aggregate consistent matches to regions (Sec. 3.2)
    Connect adjacent patches u, v if $C(u,v) < \tau_{local}$ (eq. 1)
    Eliminate small regions
    Eliminate regions for which $C(Z) < \tau_{ratio}$ (eq. 2)
  Fit and apply a global color transformation (Sec. 3.3)
  (Optional) Estimate a blur kernel and deconvolve (Sec. 5)
  Narrow search ranges (Sec. 3.4)
end for

3.1 Nearest-neighbor search

Given a source image S and a reference image R, we compute the Nearest Neighbor Field from S to R, i.e., for each patch $u \in S$ we seek a transformation T^u such that $T^u = \arg\min_T \|S_u - R_{T(u)}\|_2$. The transformation at each patch consists of translation, rotation, uniform scale, and color bias and gain per channel. For the rest of this paper we will denote these variables as T_x, T_y, $T_{rotation}$, T_{scale}, T_{bias} and T_{gain} respectively. These transformations can locally approximate more complicated global transformations such as different pose, color curves, and more.

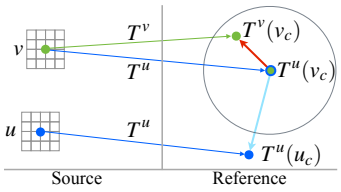
In contrast to other popular correspondence algorithms, we do not store a large feature vector in memory for each patch. Instead, we use small overlapping patches of eight by eight pixels, and a four-dimensional feature vector per pixel, which includes the three channels of *Lab* color space and the magnitude of the luminance gradient at each pixel.

Although optimizing such a high dimensional field may seem impractical, Barnes *et al.* [2010] show that the nearest neighbor field can be efficiently found in the four dimensional space of $(T_x, T_y, T_{scale}, T_{rotation})$ using their Generalized PatchMatch algorithm. We adopt this framework and extend it to support robust

color transformations and sub-pixel translation. We add eight additional dimensions for the gain and the bias of each of the four channels in our feature vector. However, unlike the geometric search dimensions, there is no need to extend the randomized search strategy over the color transformations: we obtain the optimal color bias b and gain g between a patch and its candidate match analytically in $O(1)$ using the mean μ and the variance σ^2 of the pixels in each patch by the following formula: $g(u) = \sigma(S_u)/\sigma(R_{T(u)})$, $b(u) = \mu(S_u) - g(u)\mu(R_{T(u)})$. Note that T in this case contains only the geometric part of the candidate transformation. Both the gain and the bias are clipped to lie within the current search range limits. We use Gaussian-weighted mean and variance around the center of the patch, in order to make the patch statistics rotation-invariant. They are precomputed and stored for each scale, and mipmaps are used to obtain the mean and the variance at the exact scale (since variance is not scale invariant).

3.2 Aggregating consistent regions

Although we cannot independently determine which matches are unreliable, we can aggregate matches to improve robustness by getting support from groups of matches. The likelihood that several matches agree on the transformation parameters — producing a coherent block of matches by chance — is much lower than that of any individual match to be in error [Lowe 2004; Cho et al. 2009]. We therefore apply a consistency criterion to calculate a coherence error for a group of matches together and accept sufficiently large regions if their coherence error is small.



We define adjacent patches as consistent if their nearest-neighbor field transformations from the previous stage are similar. More specifically, consider a pair of patches $u, v \in S$ with matched transformations T^u, T^v, and let v_c denote the coordinates of the center of patch v. If the two patches are matched consistently we expect the distance between $T^v(v_c)$ and $T^u(v_c)$ to be small. However, for this measure to be meaningful, the distance should be normalized (because the transformations T might involve a scale). This leads to the following definition of the consistency error between u and v:

$$C(u,v) = \frac{\|T^v(v_c) - T^u(v_c)\|_2}{\|T^u(u_c) - T^u(v_c)\|_2} \qquad (1)$$

Using this consistency error, we compute the connected components of the graph whose nodes are the patches in S, where each patch u is connected to its neighbor v if v is one of its four neighbors and if the consistency error is below a threshold: $C(u,v) < \tau_{local}$.

Thus, we obtain regions where all adjacent patch pairs are consistent, but pairs of patches further apart might not be. Our goal is to identify regions where most patch pairs are consistent. To avoid examining every pair of patches in a region Z we only consider a random subset $J(Z)$ of pairs[2]. To obtain this subset we only sample from pairs (u,v) within a certain range $\tau_{small} < \|u_c - v_c\| < \tau_{large}$. We then define the coherence error of the region, $C(Z)$, as the ratio of the inconsistent pairs to the total number of pairs in $J(Z)$:

$$C(Z) = \frac{|\{(u,v) \in J(Z) \text{ s.t. } C(u,v) > \tau_{global}\}|}{|J(Z)|}. \qquad (2)$$

[2]This sampling method assumes that the distribution of consistent pairs in a region is well behaved. In practice we sample $|J(Z)| = \sqrt{|(Z)|}$ pairs, so the entire aggregation part is done in linear time.

Finally, we accept regions whose coherency error is below a threshold τ_{ratio}. Since in small regions there is greater likelihood of the matches being coherent by chance, we exclude small regions (regions for which $|Z| < \tau_{size}$). To produce the results shown in this paper, we used $\tau_{local} = 3, \tau_{global} = 0.8, \tau_{ratio} = 0.5, \tau_{size} = 500, \tau_{small} = 8$, and $\tau_{large} = 64$.

We define the pixels in the eliminated regions as outliers, while regions that have passed the above test are considered reliable and are used to fit the global color transformation model, aligning the colors of the source image with the reference, and to adjust the search ranges for the next iterations, as described in the next section.

3.3 Global color mapping

Our global color transformation model serves two purposes: First, to iteratively improve the performance of the correspondence algorithm by narrowing the search range of the local patch transformation parameters, and second — in the context of color transfer — to produce the final result, whose tonal characteristics should match those of the reference. The color transformation is global since it is used to map all of the colors in the source, and not only those where a reliable correspondence has been established. It should be flexible enough to capture and recover various color differences, while being conservative when only a small part of the color gamut appears in the reliably matched regions. Simple adjustment of mean and variance [Reinhard et al. 2001] cannot reproduce complex variations such as saturation and nonlinear tone curve adjustments, while histogram matching and more sophisticated statistics-based methods [Pitié et al. 2007] might fail to produce a meaningful mapping for colors that do not appear in the reliably matched regions at the source image.

Hence we chose a parametric model that can be applied to predict a reasonable mapping for colors that do not appear in the input correspondences, that captures common global image adjustments and discrepancies between different imaging devices, and that can be stored in a meaningful and compact way for further manual adjustments. Our algorithm fits three monotonic curves, one per channel of the *RGB* color space, followed by a linear transform to accommodate saturation changes.

To model each of the curves we use a piecewise cubic spline with 7 breaks: two at the ends of the gamut range (*i.e.,* zero and one), and 5 uniformly distributed along the subrange on the gamut populated by reliable correspondences. Soft constraints are applied to the RGB curves outside the color range with known correspondence, so that they tend toward the identity transformation and for robustness to outliers. We constrain each of the RGB curves to pass through the points $y(-0.1) = -0.1$ and $y(1.1) = 1.1$ as well as impose hard monotonicity $y'(x) \geq 0.1$. Thus, we allow manipulations such as gamma changes to be captured by the curve, while being conservative where we do not have enough data. We use quadratic programming to solve for the curves' degrees of freedom.

To handle saturation changes, that cannot be modeled solely by independent color channel curves, we use a matrix with one degree of freedom: a uniform scale about the gray line. To compute the matrix, we project pixel colors from both images along the gray line (eliminating luminance variation) and optimize for the scale factor s that best fits the corresponding chrominances alone. The resulting matrix has the form:

$$\begin{pmatrix} s - w_r & w_g & w_b \\ w_r & s - w_g & w_b \\ w_r & w_g & s - w_b \end{pmatrix} \qquad (3)$$

Since the gray model is generally unknown, we fit this equation twice for two common models: once with uniform weights

$(w_r, w_g, w_b) = (1,1,1)/3$ and once using the YUV color space $(w_r, w_g, w_b) = (0.2989, 0.587, 0.114)$ (other gray models are similar) and choose the one that best minimizes the loss function.

3.4 Search constraints

By incorporating a separate color gain and bias for each patch, we have introduced eight additional degrees of freedom. This increases the ambiguity of each match, and thus there may be many low-cost but incorrect matches. We overcome this problem by limiting the search range of those transformations. Using the consistency criterion (Section 3.2), we detect where the Nearest-Neighbor search (Section 3.1) result is reliable, and iteratively narrow the search range for these transformations around the transformations that were found in the previous iteration.

At the initial coarse scale, the search ranges are constrained as follows: $T_x \in [0, R_w]$, $T_y \in [0, R_h]$, $T_{scale} \in [0.33, 3]$, $T_{rotation} \in [-45, 45]$, $T_{L_{bias}} \in [-30, 20]$, $T_{L_{gain}} \in [0.2, 3]$, $T_{G_{gain}} \in [0.5, 2]$, where R_w and R_h are the width and height of the reference images. We set $T_{G_{bias}} = 0$ as gradients typically change scale but not bias, and $T_{a_{gain}}, T_{b_{gain}} = 1$ as we found that a bias change is sufficient to capture chromatic changes. At each subsequent scale, we adapt the search range using the parameters of the matched transformation in the reliable regions.

Since we do not assume only one global geometric transformation, we change the search range of the geometric transformation locally, and only inside the reliable regions: For each of the reliable matches we constrain the search of the geometric parameters around its current values using a radius of 4 pixels for the translations, 10 percent for the scale, and 4 degrees for rotation.

Although a single consistent global color transformation does not always exist, we assume that if we have enough reliable correspondences, the range of their gain and the bias correspondences, combined with the global color correction, can capture the gain and bias that are required for the rest of the image. Hence, we calculate ($T_{L_{bias}}, T_{a_{bias}}, T_{b_{bias}}, T_{L_{gain}}$ and $T_{G_{gain}}$) of each of the reliable matches with respect to the color corrected image, and set the set a global search range for the color parameters to be $T_x \in \left[\min_{T^u \in Q(S)}(T_x^u), \max_{T^u \in Q(S)}(T_x^u)\right]$ where $Q(S)$ are the reliable matches and T_x is each of the color parameters. If the total area of the reliable regions is less than one percent of the source image size , we use the initial search range and no global color correction.

4 Evaluation

We extensively tested our approach on a large number of challenging pairs of images with shared content. One coarse-to-fine sweep of our basic algorithm (Alg. 1) on a 640×480 pixel image takes between 4 and 9 seconds on a 2.3GHz Intel Core i7 (2820qm) MacBook Pro (using our MATLAB/C++ implementation). The exact time depends on the interpolation method used for scaling patches from a mipmap data structure, and the exact number of GPM iterations we use. For many image pairs, a single sweep of the algorithm suffices. Furthermore, since the algorithm operates in a coarse-to-fine fashion and updates the global color transfer parameters after each iteration, we found that in most cases we obtain a very good estimate of the transfer model already at the second coarsest scale, produced after only 0.9 seconds. Thus, in the context of a color transfer application, a user would see almost immediate feedback of the estimated result. In more challenging cases we found that a second sweep with updated color may improve the correspondences, though the improvement of the global color transform is minor.

4.1 Correspondence evaluation

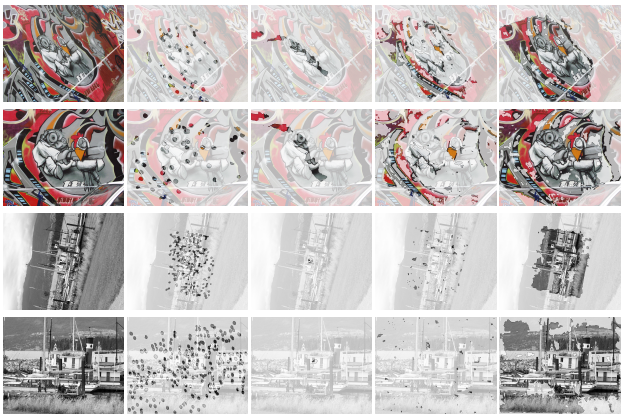
We compared our method with existing state-of-the-art dense correspondence methods: SIFT-Flow [Liu et al. 2008a] and Generalized PatchMatch (GPM), as well as with sparse SIFT correspondence [Lowe 2004]. To make a balanced comparison to GPM we used our extended implementation with 20 iterations at only the finest scale, without aggregation or narrowing the search regions (i.e., as described by Barnes *et al.* [2010]), but with the same four channels used by our method and with color bias and gain constrained to our initial search ranges. For sparse SIFT correspondence, we used circular regions with a radius of 15 pixels around the descriptor centers in the source image, and their matched circular regions in the target image, to obtain a dense correspondence in those regions. For SIFT-Flow we used the default parameter values suggested by the authors using their code.

(a) Inputs (b) SIFT (c) GPM (d) Our

Figure 4: *Qualitative comparison of matches on real-world scenes: (b) sparse SIFT features, (c) Generalized PatchMatch, and (d) our method. The unmatched regions are faded. For GPM we used our consistency criterion to eliminate outliers. For SIFT we used circular regions with a radius of 15 pixels around the descriptor centers in the source image (first and third rows), and their matched circular regions in the target image (second and forth rows), to obtain a dense correspondence in those regions. Note that SIFT incorrectly matched regions that appear in only one of the images, and that GPM has much fewer matched pixels than our method.*

Figure 4 shows a visual comparison between our method, sparse SIFT, and GPM on two pairs of real-world scenes. SIFT feature matches are typically very sparse and contain many errors that cannot be filtered easily in presence of non-rigidly moving people. For GPM we show here only large consistent regions as detected by our aggregation method. Our method typically captures much larger and more reliable matches than the other methods.

In addition to these qualitative visual comparisons, we evaluated the accuracy of our correspondence quantitatively. Since ground truth data for real scenes of this kind is scarce, we turned to a standard data set by [Mikolajczyk et al. 2005] that has been used for evaluating sparse feature based correspondence algorithms. It contains significant planar geometric transformations as well differences in sharpness, exposure and JPEG compression. Since it is known that SIFT descriptors (thus also SIFT-Flow) are robust to the latter appearance changes, we focused our comparison on large geometric deformations. Therefore we picked all subsets of pairs that have geometric deformations (the two subsets named "zoom+rotation" and the two named "viewpoint"). To accommodate these extreme

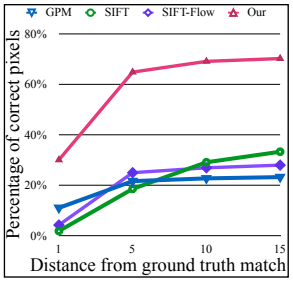


(a) Inputs (b) SIFT (c) SIFT-Flow (d) GPM (e) Our

Figure 5: *Correspondence evaluation: two examples from the dataset of Mikolajczyk et al. [2005] comparing matches recovered using sparse SIFT features (b), Generalized PatchMatch (c), SIFT-Flow (d), and our results (e). We highlight only regions of matches that fall within a radius of 15 pixels from the ground-truth. See more details in text.*

geometric deformations we extended the initial scale and rotation ranges in our method to $T_{scale} \in [0.2, 5]$, $T_{rotation} \in [-190, 190]$.

Common metrics for evaluating sparse features on this dataset do not apply to dense methods. Therefore, we adopted the metric used by [Liu et al. 2008a]. For each pixel in the first image, its match is considered correct if it falls within distance r from the ground truth location in the second image. We plotted the percent of correct matches relative to total number of input pixel with matches, as a function of r (average over all pairs in the dataset) in the graph on the right. For sparse SIFT matches we counted the pixels outside the circular descriptor regions as "incorrect". Figure 5 shows a comparison to the other methods on one "viewpoint" pair (top) and one "zoom+rotation" (bottom). In this figure we highlight correct matches that correspond to $r = 15$.



Our method outperforms the other methods on these subsets, as may be seen in the inset graph. Although GPM can handle differences in scale and rotation, it performs poorly on these pairs as it finds local independent nearest-neighbor matches on a single scale and does not try to explicitly capture consistent corresponding regions or account for any other global considerations. SIFT-Flow can handle some geometric deformations but not extreme scale and rotation differences. Note also that sparse features (like SIFT) can handle these kind of geometric deformations, but they are sparse and thus do not score high using the metric we use. Although planar transformations with little appearance changes do not reveal the full potential of our method, this dataset demonstrates the advantage of our dense correspondence algorithm in dealing with realistic transformations.

We also compared our method with the Co-recognition approach of Cho *et al.* [2008], who demonstrated impressive reliable region correspondence results, outperforming previous similar methods in many cases. We used their challenging dataset and their measure for hit-ratio h_r and background ratio b_r[3]. Note that their measure quantifies *coverage* of the common regions as opposed to pixel-wise correspondence accuracy as reported in the previous experiment. Our method performs well on this dataset with $[h_r, b_r] = [65.7\%, 4.9\%]$ on average. Note that our method is tuned for high reliability of the reported consistent regions, thus the low b_r (false-alarm rate). In order to compare the two methods for the same value of $b_r = 22.4\%$ as their method produces, we used simple dilation of 12 pixels on our results and got a slightly better h_r value of 86.9% on average (compared to their 85.5%). An example from this dataset is shown in Fig. 6. Our method often captures more accurate object boundaries, and is also likely to be more accurate inside the objects, as we compute correspondences at the *original* image resolution rather than on a very coarse grid in their method. Moreover, their dataset consists only of rigid objects and scenes with almost no differences in appearance, whereas our method works in a much wider operating range.

(a) Inputs (b) Co-recognition (c) Our

Figure 6: *Comparison to Co-recognition [Cho et al. 2008] using an example from their dataset.*

4.2 Global color transfer evaluation

Given a reference and source image pair with shared content, one may use the global parametric color transformation model we recover to automatically adjust the color and exposure of the source image to match the reference. A number of representative results are shown in Figure 7. In contrast, the statistical color transfer method of Pitié *et al.* [2007] produces poor results when the two images have different color statistics in the non-overlapping regions.

We also compared our color transfer results with color transfer based on sparse SIFT correspondences [Lowe 2004]: To perform color transfer with SIFT we fit the same global color model as described in Section 3.3, but rather than fitting it to our reliable correspondences we use circular regions centered at the matching SIFT features, where the size of each region is determined by the scale of the corresponding SIFT feature. Figure 2 shows a challenging example pair where SIFT matching fails to produce an acceptable color transfer result. The reason is that, as previously noted, SIFT tends to miss many smooth non-rigid regions, and has a high false positive rate that overwhelms our parametric color transfer model. The reader is referred to the project webpage for additional comparisons with SIFT-based color transfer.

4.3 Limitations

Our experiments did reveal that our approach has a few limitations. One limitation is that it has difficulty finding reliable correspondences in very large smooth regions, such as clear sky. Due to the

[3] $h_r = \frac{|GroundTruth \cap Result|}{|GroundTruth|}$, $b_r = \frac{|Result| - |Result \cap GroundTruth|}{|Result|}$

(a) Reference (b) Source (c) Pitié *et al.* (d) Our

Figure 7: *Automatic color transfer: comparison to the state-of-the-art method of Pitié et al. [2007].*

use of fixed-size patches, if an object appears over a different background in the two input images, the algorithm cannot extend the matching regions all the way to the object's boundaries. A third limitation is that our single global color model cannot handle cases where there are actually two or more very different color models: we find correspondences for only one. This situation might arise under strong lighting changes, or due to local user edits. Also, our global color model only handles one kind of cross-channel transformation: saturation/desaturation. Although this is the most common such transformation, it will not handle less frequent cases like hue rotations, or manipulations used to emulate chemical film development processes. A possible solution might be running the algorithm several times with different but narrower initial search ranges and aggregate the reliable matches from each pass.

5 Applications

Local color transfer

Our global parametric color transfer model produces satisfactory results in many cases. However, there are also cases where global exposure and color corrections do not produce a satisfactory result. This can occur if the lighting and shadows in the scene are too different (middle row in Figure 8), if the reference image has undergone local editing by the user after it was captured, or if there is a global transformation but it is not one of the common transformations that our model is designed to recover. In such cases, a global color transfer followed by a further local adjustment using our correspondences can yield a more satisfactory result.

We perform the local adjustment as follows: We first locally adjust the colors inside the reliable correspondence regions, and then propagate the change from the boundaries of these regions to the rest of the image using Poisson blending [Pérez et al. 2003].

The local adjustment inside the well-matched regions is done using *locally adaptive histogram matching*, a variant on the *locally adaptive histogram equalization* algorithm [Pizer et al. 1987]: the original algorithm subdivides the image to blocks, calculates a transfer function (originally histogram equalization) and smoothly interpolates the result. Since we already have a set of pixel-to-pixel correspondences between the source and the reference, we can replace the histogram equalization with histogram matching to locally match the color of each block centered at a matched pixel with the corresponding block in the reference image.

The added value of the resulting local adjustment is demonstrated in Figure 8. For example, it succeeds in improving the flesh tones in the top and bottom rows, and assigns a more accurate darker shade of green to the vegetation in the middle row. The corresponding regions on the source image are shown in column (c).

Deblurring

Deblurring by example has been demonstrated by Joshi *et al.* [2010] using recurrent faces, and by Ancuti *et al.* [2008] using static backgrounds aligned with SIFT correspondences. [Yuan et al. 2007] deblurred blurry images using noisy (aligned) examples. When a sharp example is given and there is no further data about the capturing devices, the first step is to estimate the blur kernel using an *accurate* pixel (or better, sub-pixel) alignment between the blurred pixels and the corresponding sharp ones. However we found that when the blur kernel is large, it is hard to obtain an accurate enough correspondence. Therefore we *interleave* the kernel estimation and deconvolution steps in the inner loop of our correspondence algorithm (Step 10 in Algorithm 1). The effective kernel at the coarsest resolution is usually small and can be effectively computed from the correspondences at that scale, which are further improved after each deconvolution. The kernel is then upsampled when moving to the next scale, to produce a sharper deconvolved initial source image. This process continues till the finest scale to obtain our final kernel and deconvolved image.

We modified the kernel estimation method of Cho and Lee [2009] to use a "validity" mask in addition to the "sharp" and "blurry" images as inputs. The estimation is done only for pixels inside the validity mask. In our case, the blurry input is the source image after matching its colors to the reference, and we synthesize a sharp image by assigning colors in the consistent regions of the source image using the corresponding reference locations. We call this image the "reconstructed" source and the validity mask marks the consistent pixels in it. The kernel estimation process is then followed by sparse deconvolution [Levin et al. 2007].

Accurate and dense alignment is crucial for the success of the deblurring process. Therefore, this application is an interesting test case for the quality of our correspondence method. We compare our results with the state-of-the-art blind deconvolution methods of Cho and Lee [2009] and of Levin *et al.* [2011]. To isolate the influence of the estimated kernel, we applied the same deconvolution method of [Levin et al. 2007] with same regularization weight (10^{-4}) using the estimated kernel by each method. Two examples are shown in Figure 9, where our method managed to deblur challenging images that violate the general assumptions of the blind methods.

Mask transfer

Many image editing tasks require first selecting a local region on an image, by creating either a hard mask or a soft matte, and then using this mask to locally edit (or to cut out or copy) that region. Creating masks by hand is tedious, and various interactive techniques have been devised to simplify this task (e.g., [Rother et al. 2004]). Cosegmentation [Rother et al. 2006] methods try to automatically segment the same object from two different images, assuming similarity between the object histograms and dissimilarity between the histograms of the backgrounds. Here, we assume we are given a mask that has already been created for an object in one of the images, and we wish to transfer this mask to the same object in another image. This is similar to video segmentation by propagation [Bai et al. 2009], in which a segmentation is provided for one frame, and the task is to propagate it to the rest of the video. Our problem is in some aspects more challenging, because of the large differences between images that we must handle, relative to the small frame-to-frame variations typical in video.

(a) Reference (b) Source (c) Matches (d) Our global (e) Our local (f) Zoom-in (global) (g) Zoom-in (local)

Figure 8: *Global vs. local color transfer: (a) Reference; (b) Source; (c) Matching regions in the source reconstructed from the reference; (d) Global color transfer result; (e) Local refinement of the global transfer result;*

(a) Sharp example (b) Blurry (c) [Cho and Lee 2009] (d) [Levin et al. 2011] (e) Our

Figure 9: *Comparison with blind-deconvolution methods: A sharp reference image (a) and a blurry image (b) are given. We applied the deconvolution of Levin et al. [2007] with the same parameter values and each of the kernels estimated by the blind-deconvolution methods of Cho and Lee [2009] (c), and of Levin et al. [2011] (d), and the kernel we estimated using the dense correspondence to the sharp image.*

(a) Reference (b) Input mask (c) Source (d) Trimap (e) Output mask

Figure 10: *Mask transfer: A binary mask is given (b) on the reference image (a). The goal is to transfer this mask to the source image (c). The mask and its complement are transferred to (c) using our dense correspondences, resulting in a trimap (d). Low confidence matches result in the unknown (gray) regions. A final mask (e) is computed using GrabCut with the trimap as its input.*

Our method is very simple: Given a mask for the reference image (Fig. 10(b)), first find all the consistent corresponding regions between the two images, then mark the pixels in the input image (Fig. 10(c)) that correspond to the masked pixels in the reference as "known object" pixels. Do the same for the complement of the masked pixels and mark them as "known background". The rest of the pixels are marked as "unknown". These regions are shown as white, black and gray respectively in Fig. 10(d). We slightly erode the "known" regions to avoid crosstalk, and use them as foreground/background initialization for segmentation using GrabCut [Rother et al. 2004]. Two results are shown in Fig. 10(e).

6 Summary

We have demonstrated a new correspondence method that combines dense local matching with robustness to outliers. This combination makes it possible to identify correspondences even in non-rigid objects with significant variance in their appearance characteristics, including dramatically different pose, lighting, viewpoint and sharpness. We showed that our method outperforms previous methods, which find this task challenging.

We have shown that our method is widely applicable for color transfer in real-world images, as well as additional transfer challenges such as deblurring and mask transfer. We believe this method may also prove useful for a variety of computer graphics and vision applications that currently rely on previous correspondence methods.

Acknowledgements: We thank Sunghyun Cho, Seungyong Lee and Anat Levin for the helpful discussions about the deblurring application. Special thanks go to Sunghyun for helping us adapt his code. This work was supported in part by the Israel Science Foundation founded by the Israel Academy of Sciences and Humanities.

References

AN, X., AND PELLACINI, F. 2010. User-controllable color transfer. *Computer Graphics Forum 29*, 2, 263–271.

ANCUTI, C., ANCUTI, C. O., AND BEKAERT, P. 2008. Deblurring by matching. *Computer Graphics Forum 28*, 2, 619–628.

BAI, X., WANG, J., SIMONS, D., AND SAPIRO, G. 2009. Video SnapCut: robust video object cutout using localized classifiers. *ACM Trans. Graph. 28*, 3 (July), 70:1–70:11.

BARNES, C., SHECHTMAN, E., GOLDMAN, D. B., AND FINKELSTEIN, A. 2010. The generalized PatchMatch correspondence algorithm. In *Proc. ECCV*, vol. 3, 29–43.

BHAT, P., ZITNICK, C. L., SNAVELY, N., AGARWALA, A., AGRAWALA, M., CURLESS, B., COHEN, M., AND KANG, S. B. 2007. Using photographs to enhance videos of a static scene. In *Rendering Techniques 2007*, Eurographics, 327–338.

BROX, T., BREGLER, C., AND MALIK, J. 2009. Large displacement optical flow. In *Proc. CVPR 2009*, IEEE, 41–48.

CHO, S., AND LEE, S. 2009. Fast motion deblurring. *ACM Trans. Graph. 28*, 5 (December), 145:1–145:8.

CHO, M., SHIN, Y. M., AND LEE, K. M. 2008. Co-recognition of image pairs by data-driven monte carlo image exploration. In *Proc. ECCV 2008*, vol. 4, 144–157.

CHO, M., LEE, J., AND LEE, K. 2009. Feature correspondence and deformable object matching via agglomerative correspondence clustering. In *Proc. ICCV*, 1280–1287.

DALE, K., JOHNSON, M. K., SUNKAVALLI, K., MATUSIK, W., AND PFISTER, H. 2009. Image restoration using online photo collections. In *Proc. ICCV*, IEEE.

EISEMANN, E., AND DURAND, F. 2004. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph. 23* (August), 673–678.

EISEMANN, M., EISEMANN, E., SEIDEL, H.-P., AND MAGNOR, M. 2010. Photo zoom: High resolution from unordered image collections. In *Proc. Graphics Interface*, 71–78.

FERRARI, V., TUYTELAARS, T., AND GOOL, L. J. V. 2004. Simultaneous object recognition and segmentation by image exploration. In *Proc. ECCV*, vol. 1, 40–54.

JOSHI, N., MATUSIK, W., ADELSON, E. H., AND KRIEGMAN, D. J. 2010. Personal photo enhancement using example images. *ACM Trans. Graph. 29*, 2 (April), 12:1–12:15.

KAGARLITSKY, S., MOSES, Y., AND HEL OR, Y. 2009. Piecewise-consistent color mappings of images acquired under various conditions. In *Proc. ICCV*, 2311–2318.

LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph. 26*, 3 (July).

LEVIN, A., WEISS, Y., DURAND, F., AND FREEMAN., W. T. 2011. Efficient marginal likelihood optimization in blind deconvolution. In *Proc. CVPR*, IEEE.

LIU, C., YUEN, J., TORRALBA, A., SIVIC, J., AND FREEMAN, W. T. 2008. SIFT flow: Dense correspondence across different scenes. In *Proc. ECCV*, vol. 3, 28–42.

LIU, X., WAN, L., QU, Y., WONG, T.-T., LIN, S., LEUNG, C.-S., AND HENG, P.-A. 2008. Intrinsic colorization. *ACM Trans. Graph. 27*, 5 (December), 152:1–152:9.

LOWE, D. G. 2004. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision 60*, 2, 91–110.

LUCAS, B. D., AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *Proc. DARPA Image Understanding Workshop*, 121–130.

MATAS, J., CHUM, O., URBA, M., AND PAJDLA, T. 2002. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC*, 384–396.

MIKOLAJCZYK, K., TUYTELAARS, T., SCHMID, C., ZISSERMAN, A., MATAS, J., SCHAFFALITZKY, F., KADIR, T., AND GOOL, L. V. 2005. A comparison of affine region detectors. *Int. J. Comput. Vision 65* (November), 43–72.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph. 22*, 3 (July), 313–318.

PETSCHNIGG, G., SZELISKI, R., AGRAWALA, M., COHEN, M., HOPPE, H., AND TOYAMA, K. 2004. Digital photography with flash and no-flash image pairs. *ACM Trans. Graph. 23*, 3 (August), 664–672.

PITIÉ, F., KOKARAM, A. C., AND DAHYOT, R. 2007. Automated colour grading using colour distribution transfer. *Comput. Vis. Image Underst. 107* (July), 123–137.

PIZER, S. M., AMBURN, E. P., AUSTIN, J. D., CROMARTIE, R., GESELOWITZ, A., GREER, T., ROMENY, B. T. H., AND ZIMMERMAN, J. B. 1987. Adaptive histogram equalization and its variations. *Comput. Vision Graph. Image Process. 39* (September), 355–368.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Comput. Graph. Appl.* (September 2001).

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3 (August), 309–314.

ROTHER, C., MINKA, T. P., BLAKE, A., AND KOLMOGOROV, V. 2006. Cosegmentation of image pairs by histogram matching – incorporating a global constraint into MRFs. In *Proc. CVPR 2006*, vol. 1, 993–1000.

SNAVELY, N., SEITZ, S. M., AND SZELISKI, R. 2006. Photo tourism: exploring photo collections in 3D. *ACM Trans. Graph. 25* (July), 835–846.

YUAN, L., SUN, J., QUAN, L., AND SHUM, H.-Y. 2007. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph. 26*, 3 (July).

ZELNIK-MANOR, L., AND IRANI, M. 2006. On single-sequence and multi-sequence factorizations. *Int. J. Comput. Vision 67* (May), 313–326.