Aligning Images in the Wild

Wen-Yan Lin$^*_1$ Linlin Liu$^*_3$ Yasuyuki Matsushita$_2$ Kok-Lim Low$_3$ Siying Liu$_1$
Institute of Infocomm Research$_1$ Microsoft Research Asia$_2$
National University of Singapore$_3$
{wdlin,sliu}@i2r.a-star.edu.sg$_1$, {liulin,lowkl}@comp.nus.edu.sg$_3$, yasumat@microsoft.com$_2$
(* denotes joint first author.)

Abstract

Aligning image pairs with significant appearance change is a long standing computer vision challenge. Much of this problem stems from the local patch descriptors' instability to appearance variation. In this paper we suggest this instability is due less to descriptor corruption and more the difficulty in utilizing local information to canonically define the orientation (scale and rotation) at which a patch's descriptor should be computed. We address this issue by jointly estimating correspondence and relative patch orientation, within a hierarchical algorithm that utilizes a smoothly varying parameterization of geometric transformations. By collectively estimating the correspondence and orientation of all the features, we can align and orient features that cannot be stably matched with only local information. At the price of smoothing over motion discontinuities (due to independent motion or parallax), this approach can align image pairs that display significant inter-image appearance variations.

1. Introduction

Obtaining point-to-point correspondence across image pairs is a fundamental problem for various vision tasks, such as structure-from-motion, image super-resolution and high-dynamic-range imaging. However, even for images of the same scene, correspondence computation becomes challenging when they exhibit large inter-image appearance variations. Unfortunately, a myriad of factors affect the final appearance of an image. These include (1) illumination, that varies shading and atmospheric absorption caused by rain/ haze, (2) view point change, (3) camera settings, such as camera response function and aperture as well as intrinsics, (4) occlusion and background changes in the temporal interval between photographs, and (5) post processing, when images are enhanced using photo editing software. These factors make the problem of computing correspondence very difficult. In particular, aligning images taken under uncontrolled conditions is even harder.

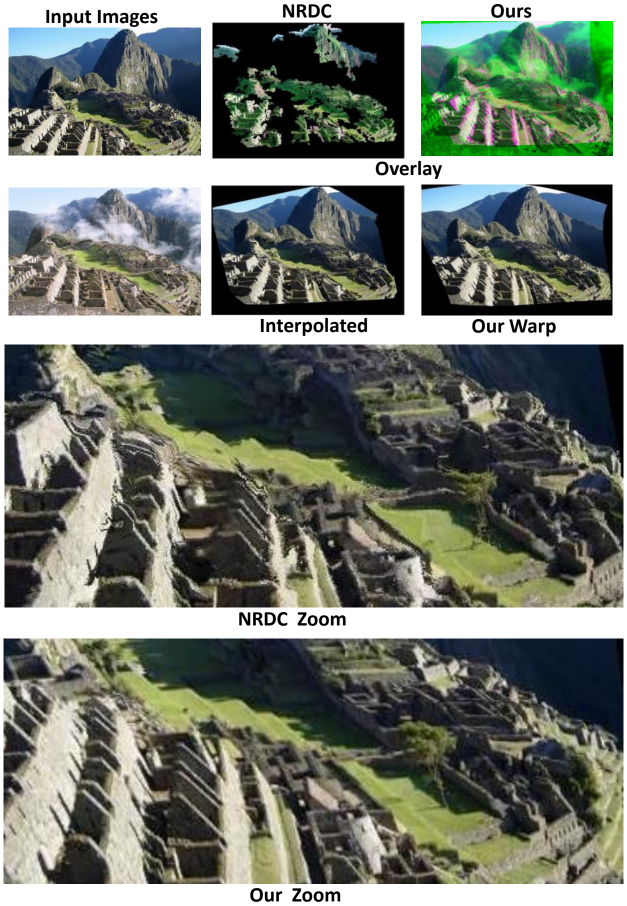

Figure 1. Two images of Machu Picchu taken under different imaging conditions. It illustrates the difficulty in of obtaining full frame warping by interpolating even fairly dense correspondence of NRDC [9]. Alignment quality is displayed by overlaying the warp's green channel with the underlying image.

Most previous attempts at aligning images taken under different conditions focus on modeling appearance variation, with Andrews *et al*. [3], Weijer *et al*. [23] and Ha-

Cohen *et al*. [9] utilizing color transforms or photometric-invariant image representations. However, modeling every eventuality is difficult. When image sections violate the model, correspondences become undefined, or large errors are incurred if a matching result is forced. Hence, problems like good quality, full frame Internet image alignment, are yet unsolved.

This paper discusses how varying imaging conditions affect descriptor-based matching and proposes a purely geometric solution. By not relying on appearance models to validate matches or remove incorrect correspondence, we generate dense, full frame warps for image pairs with large appearance variations. This is illustrated in Figure 1.

Algorithms like SIFT [13], SURF [5] and A-SIFT [16] agglomerate a patch's gradient information into a local descriptor. As noted by Lowe [13], descriptors combine invariance to affine lighting changes with robustness to non-linear lighting variation. This can be considered a weak form of photometric invariance. Descriptor-based nearest-neighbor matching is remarkably resistant to minor variations in imaging conditions. However, the number and accuracy of matches decrease sharply with increased appearance variation. Even for more sophisticated group-wise descriptor alignment techniques such as those by Shum *et al*. [20], Liu *et al*. [12], Lin *et al*. [11], attaining good quality alignment under varying imaging conditions remains a challenge. We suggest that much of this performance decline is not due to irretrievable descriptor corruption and can be alleviated by a carefully designed matching algorithm.

The key of our solution lies in the descriptor orientation. As a region quantity, a descriptor value depends on the orientation at which it encodes a patch. Hence, descriptor comparisons are only meaningful if the descriptors from both images are computed at the correct relative orientation. Usually, local image information is used to canonically orient descriptors to ensure comparability [13]. However, at increased appearance variation, local information may be insufficient to canonically orient image descriptors.

This problem's effects are subtle, as the result of descriptor mis-orientation resembles that of descriptor corruption. However, for images taken under similar conditions, works like Fan *et al*. [8] have reported significant improvements in matching performance through better orientation handling. Further more, by assuming image pairs are pre-oriented (feature orientation is not allowed to change), SIFT flow [12] allows meaningful alignment of different objects. These results lead us to posit that as imaging conditions change, descriptors remain discriminative, long after we lose the ability to canonically orient them with local information. If true, descriptors can still be utilized for matching by coupling correspondence and relative feature orientation estimation within a group-wise matching framework.

To achieve this coupling, we formulate the non-linear relationship between feature orientation and descriptors into a more manipulatable linear orientation choice. We jointly compute feature correspondence and orientation using a hierarchical series of models that progressively relax group-wise constraints. At the price of smoothing over motion discontinuities, we can use sparsely distributed feature descriptors to interpolate dense warps across image pairs taken at very different imaging conditions. This is surprising given our purely geometric approach and indicates the potential for better results by fusing both lighting and geometric cues.

Our contributions are summarized as follows

- We posit that as imaging conditions change, descriptors remain discriminative, despite loss of ability to canonically orient them with local information;
- We demonstrate this using a smoothly varying orientation and correspondence estimation algorithm to estimate full frame warps between two images of a scene taken under different imaging conditions;
- We demonstrate camera pose recovery and high-dynamic-range imaging as applications of our method.

1.1. Related Works

There is a considerable body of prior work on simultaneous region growing and orientation including works by HaCohen *et al*. [9], Barnes *et al*. [4] Vedaldi *et al*. [24] and Cheng *et al*. [7]. However, many require specific color modeling to handle illumination changes well. While more flexible, growing algorithms do not work well when local information is unstable and under appearance variation, their grown correspondence can be erroneous. Although correspondence errors can be rejected with a photometric-based thresholding such as that employed by HaCohen *et al*. [9], this results in sparse correspondences. Interpolation on such correspondence is very vulnerable to correspondence errors. In contrast, by evolving all the features together, we can interpolate across weak textures and our dense warping results can be used "as is" without further thresholding.

Our algorithm is more similar in spirit to descriptor based SIFT flow [12]. While we cannot achieve warping across images of different objects, our orientation varying descriptors permit finer alignment and better handling of orientation changes.

Our work is also an attempt at establishing relational structure between images taken under uncontrolled conditions. This motivation is shared by many works utilizing Internet images, such as "Building Rome in a Day" [2] that creates 3-D city models from community photo collections, Shrivastava *et al*.'s [19] cross-domain matching and Kemelmacher-Shlizerman *et al*.'s [10] face reconstruction from Internet images. However, these techniques are not directly applicable to our problem as they achieve stability by leveraging on dataset size, while our work takes a single image pair as input.

Compared to sophisticated color modeling techniques such as Color Eigenflows [15] and CSIFT [1], we utilize the weaker invariance of generic SIFT features [13]. Instead, we handle differing imaging conditions by focusing on geometric rather than photometric constraints.

Our formulation builds on point set registration works such as thin plate spline [6], motion coherence [25], coherent point drift [17] and smoothly varying affine [11]. Of these, we are most closely related to Lin *et al*.'s [11] work. We also utilize motion coherence [25] to parameterize our variables as a smooth variation from approximate global models. However, unlike [11] that depends on locally oriented SIFT descriptors, our joint estimation of feature orientation and correspondence handles appearance variations much better as demonstrated in Figure 2.

2. Our Approach

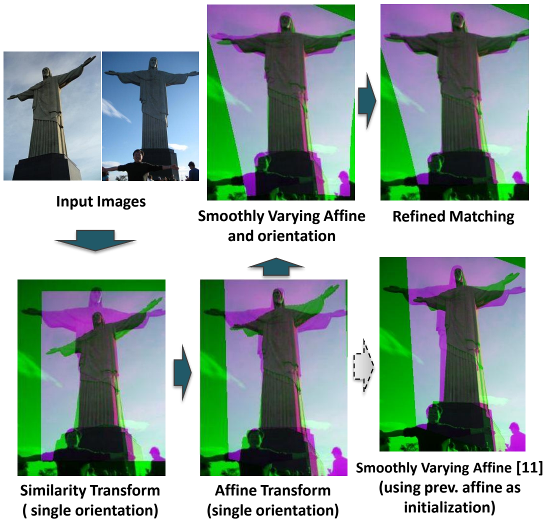
Input Images
Smoothly Varying Affine and orientation
Refined Matching
Similarity Transform (single orientation)
Affine Transform (single orientation)
Smoothly Varying Affine [11] (using prev. affine as initialization)

Figure 2. Illustration of our hierarchical approach, results using a naive smoothly varying affine [11] are shown at bottom right.

We use a smoothly varying field to warp and orient features from one image to the other. Scaling or rotating an image patch causes a permutation (with some re-sampling and anti-aliasing) of image pixels. This results in a complex, non-linear relationship between feature orientation and descriptors. To side-step the problem of optimizing that relationship, we implement a linear orientation choice mechanism. On one image, multiple differently oriented feature descriptors are computed at each spatial location. Features from the other image then choose the orientation best corresponding to their own. This "choosing" mechanism does not increase the number of variables but comes at the cost of a one-off descriptor comparison across many different orientations. Apart from feature orientation, we employ the same "choosing" mechanism to enable finer registration through feature re-localization.

2.1. Formulation

We formulate the correspondence problem between *base* and *target* images as a registration of two evolving feature sets. The base feature set is $\boldsymbol{B} = \{\mathbf{b}_i\}$ and the target feature set $\boldsymbol{T} = \{\boldsymbol{T}_j | \boldsymbol{T}_j = \{\mathbf{t}_{j\theta}\}\}$. Here, the nested target feature set is used to accommodate orientation choice.The initial feature values are computed directly from the image. These are denoted as $\boldsymbol{B}_0 = \{\mathbf{b}_{0i}\}$ and $\boldsymbol{T}_0 = \{\boldsymbol{T}_{0j}\}$, with subscript 0 representing initial values.

2.1.1 Feature Definitions

A base feature vector $\mathbf{b}_i \in \boldsymbol{B}$ takes the form

$$\mathbf{b}_i = [\begin{array}{ccc} \mathbf{b}_i^c & \mathbf{b}_i^r & \mathbf{b}_i^d \end{array}],$$

with sub-vectors $\mathbf{b}_i^c$, $\mathbf{b}_i^r$, and $\mathbf{b}_i^d$ denoting the feature's spatial coordinate, orientation, and descriptor, respectively.

Target feature definition is more complex. $\boldsymbol{T}_j = \{\mathbf{t}_{j\theta}\}$ represents a set of target features, sharing the same image coordinates but differing orientations, with $\theta \in \{1, \dots, \Theta\}$ indexing feature orientations. This gives each $\mathbf{b}_i$ feature a choice of orientation. $\mathbf{t}_{j\theta}$'s definition is similar to $\mathbf{b}_i$

$$\mathbf{t}_{j\theta} = [\begin{array}{ccc} \mathbf{t}_{j\theta}^c & \mathbf{t}_{j\theta}^r & \mathbf{t}_{j\theta}^d \end{array}],$$

with sub-vectors $\mathbf{t}_{j\theta}^c, \mathbf{t}_{j\theta}^r, \mathbf{t}_{j\theta}^d$ denoting the feature's spatial coordinate, orientation and descriptor.

Practically, our initial feature sets $\{\mathbf{b}_{0i}\}$ and $\{\boldsymbol{T}_{0j}\}$ arise from regions of interest detected using Lowe's [13] SIFT algorithm, with $i \in \{1, \dots, M\}$ and $j \in \{1, \dots, N\}$ indexing regions of interest in the base and target image, respectively. $\mathbf{b}_{0i}^c$ and $\mathbf{t}_{0j\theta}^c$ are the image coordinates of the regions of interest, given in the homogeneous form $[x \;\; y \;\; 1]^T$. Unlike in a traditional feature detection, all feature descriptors are computed at fixed orientations and scales. For the base image, all features are computed at patch radius p_0 and rotation direction r_0. For the target image, feature $\mathbf{t}_{0j\theta}$ is assigned a patch of radius p_θ and rotation angle r_θ relative to r_0. The initial target feature orientation is parameterized with respect to the base orientation. Thus,

$$\mathbf{t}_{0j\theta}^r = \begin{bmatrix} p_0/p_\theta & \cos(r_\theta) & \sin(r_\theta) \end{bmatrix}^T.$$

As only relative feature orientation is important, p_0 and r_0 are not actually assigned to $\mathbf{b}_{0i}^r$ which is a dummy variable. The SIFT features are computed at the appropriate image coordinates and orientations to obtain the descriptors $\mathbf{b}_{0i}^d$ and $\mathbf{t}_{0j\theta}^d$.

We evolve the spatial and orientation components of feature sets $\boldsymbol{B}$ and $\boldsymbol{T}$ using a hierarchy of transformations from initial $\boldsymbol{B}_0$ and $\boldsymbol{T}_0$ values. These transformations are progressively relaxed to permit point-wise evolution and finer image alignment. Feature descriptors serve as guides to the evolution and remain invariant throughout, *i.e.*, $\mathbf{b}_i^d = \mathbf{b}_{0i}^d$, and $\mathbf{t}_i^d = \mathbf{t}_{0i}^d$.

For notational simplicity, we also define Gaussian and compound Gaussian functions:

$$\begin{cases} g(\mathbf{z}, \sigma) = \exp^{-\left(\|\mathbf{z}\|^2/2\sigma^2\right)} \\ \phi_{ij\theta}(\mathbf{b}_i, \mathbf{t}_{j\theta}) = g(\mathbf{t}_{j\theta}^c - \mathbf{b}_i^c, \sigma_c) g(\mathbf{t}_{j\theta}^r - \mathbf{b}_i^r, \sigma_r) g(\mathbf{t}_{0j\theta}^d - \mathbf{b}_{0i}^d, \sigma_d) \end{cases}$$

Here $\phi_{ij\theta}(\cdot)$ is defined with respect to $\mathbf{t}_{0j\theta}^d$ and $\mathbf{b}_{0i}^d$ rather than $\mathbf{t}_{j\theta}^d$ and $\mathbf{b}_i^d$ because of the descriptor invariance $\mathbf{b}_i^d = \mathbf{b}_{0i}^d$ and $\mathbf{t}_{j\theta}^d = \mathbf{t}_{0j\theta}^d$.

2.1.2 Cost Function

To quantify alignment accuracy, we treat the base feature vectors $\mathbf{b}_i$ as generative Gaussian centroids. Thus,

$$P(\mathbf{t}_{j\theta}|\boldsymbol{B}) = \sum_{i=1}^{M} \left(\phi_{ij\theta}(\mathbf{b}_i, \mathbf{t}_{j\theta}) + \kappa\right),$$

where κ is a positive number included to handle features appearing in only one image. For interest region $\boldsymbol{T}_j$, its Θ orientation candidates are treated as mutually exclusive choices. Hence,

$$P(\boldsymbol{T}_j|\boldsymbol{B}) = \sum_{\theta=1}^{\Theta} \sum_{i=1}^{M} \left(\phi_{ij\theta}(\mathbf{b}_i, \mathbf{t}_{j\theta}) + \kappa\right). \tag{1}$$

We seek to minimize the overall negative log likelihood

$$Q(\alpha) = -\sum_{j=1}^{N} \log\left(P(\boldsymbol{T}_j|\boldsymbol{B})\right), \tag{2}$$

where α represents the parameters of the transformations applied to the initial base and target features $\mathbf{b}_{0i}$ and $\mathbf{t}_{0j\theta}$.

2.2. Hierarchy of Models

For robustness to local minima and efficiency, we do not minimize alignment cost directly. Rather, we parameterize the evolution of sets $\boldsymbol{B}$ and $\boldsymbol{T}$ using a hierarchical series of models. The simpler, more robust models serve as approximate initializations for subsequent more refined models. The process is illustrated in Figure 2.

2.2.1 Similarity Transformation

The first level of our hierarchy is the similarity transformation:

$$\begin{aligned} \mathbf{t}_{j\theta}^c &= \mathbf{S}\mathbf{t}_{0j\theta}^c, & \mathbf{b}_i^c &= \mathbf{R}\mathbf{b}_{0i}^c, \\ \mathbf{t}_{j\theta}^r &= \mathbf{t}_{0j\theta}^r, & \mathbf{b}_i^r &= [s \;\; u \;\; v]^T, \end{aligned} \tag{3}$$

where $\mathbf{S} = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and $\mathbf{R} = \begin{bmatrix} u & -v & t_1 \\ v & u & t_2 \\ 0 & 0 & 1 \end{bmatrix}$. We solve for scale, rotation and translation parameters, $\alpha = \{s, u, v, t_1, t_2\}$, by minimizing Eqn (2). The similarity transformation tightly couples feature orientation and spatial coordinates, providing a good initialization for the next level.

2.2.2 Affine Transformation

For the second level, spatial coordinates are parameterized with a global affine transformation:

$$\mathbf{t}_{j\theta} = \mathbf{t}_{0j\theta}, \quad \mathbf{b}_i^c = \mathbf{A}\mathbf{b}_{0i}^c, \quad \mathbf{b}_i^r = [s \;\; u \;\; v]^T. \tag{4}$$

We solve for parameters $\alpha = \{\mathbf{A}, s, u, v\}$ by minimizing Eqn (2). While orientation and spatial parameters are no longer directly coupled, there remains an indirect coupling effect due to their concatenation into one feature vector.

2.2.3 Smoothly Varying Orientation and Affine

Next we employ a smoothly varying orientation and affine parameterization that allows $\mathbf{b}_i$ coordinates to vary individually while robustly handling noise and missing entries:

$$\begin{aligned} \mathbf{t}_{j\theta} = \mathbf{t}_{0j\theta}, \quad \mathbf{b}_i^c &= (\mathbf{A} + \Delta\mathbf{A}_i)\, \mathbf{b}_{0i}^c, \\ \mathbf{b}_i^r &= \begin{bmatrix}\Delta s_i & \Delta u_i & \Delta v_i\end{bmatrix}^T \end{aligned} \tag{5}$$

with the smoothness being applied to the point-wise varying Δ_i coordinates. The smoothness is computed with respect to the global variables rather than to the original or initialized positions. This allows for larger motion and less vulnerability to initialization errors.

To enforce smoothness, we use the energy function

$$Q(\alpha) = -\sum_{j=1}^{N} \log\left(P(\boldsymbol{T}_j|\boldsymbol{B})\right) + \lambda\Psi(\boldsymbol{B}), \tag{6}$$

where $\alpha = \{\mathbf{A}, \Delta\mathbf{A}_i, \Delta s_i, \Delta u_i, \Delta v_i\}$ are the variable parameters. Note that while $\mathbf{A}$ and $\Delta\mathbf{A}_i$ are 3×3 matrices, only the entries of the first two rows are variables.

Smoothness function $\Psi(\cdot)$ takes the motion coherence form employed in [11, 17]. This method ensures the warping field carrying the discrete points is smooth, permitting dense warping of high dimensional sparse points.

Let $v(\mathbf{z}_{2\times 1})$ represent a 2-D smoothly varying field of some parameter. The coherence term penalizes discontinuities using

$$\int_{\mathbb{R}^2} \frac{|v'(\omega)|^2}{g'(\omega)} d\omega, \tag{7}$$

where $v'(\omega)$ is the Fourier transform of the velocity field, while $g'(\omega)$ is the Fourier transform of a Gaussian with spatial standard deviation γ.

2.2.4 Feature Re-localization

The above steps provide only approximate alignment as the features may not be consistently localized across images. For finer registration, we replace the orientation choice with a localization choice. Target set $\boldsymbol{T}$ is replaced with a new set $\widetilde{\boldsymbol{T}} = \{\widetilde{\boldsymbol{T}}_j | \widetilde{\boldsymbol{T}}_j = \{\tilde{\mathbf{t}}_{jk}\}, j \in \{1, \ldots, M\}\}$. Each $\widetilde{\boldsymbol{T}}_j$ consists of K_i localization candidates. These are high gradient regions ("Harris corners" without non-maxima suppression or thresholding) within a 30-pixel neighborhood of each evolved $\mathbf{b}_i^c$. Similar to Eqn (6), we allow base features to choose more accurate localization. Feature vectors in $\widetilde{\boldsymbol{T}}$ share the same format as $\boldsymbol{T}$, except that all orientation values are set to zero since we no longer choose orientation.

The transformation relating $\boldsymbol{B}$ and $\boldsymbol{B}_0$ is the same as that used in Eqn (5). The spatial parameters $\alpha = \{\mathbf{A}, \Delta\mathbf{A}_i\}$ are solved for by minimizing

$$Q(\alpha) = -\sum_{j=1}^{M} \log \sum_{k=1}^{K_i} \sum_{i=1}^{M} \left(\phi_{ijk}(\mathbf{b}_i, \tilde{\mathbf{t}}_{jk}) + \kappa\right) + \lambda\Psi(\boldsymbol{B}), \quad (8)$$

whose derivation is similar to Eqn (2). Note that by zeroing all orientation values, orientation variables do not affect the minimization and can be ignored.

2.3. Minimization Technique

We minimize the cost defined in Eqns (2), (6), and (8) using Expectation Maximization's iterative refinement procedure. At the m-th iteration, the alignment $\boldsymbol{B}^{\{m\}}$ and $\boldsymbol{T}^{\{m\}}$ can be used to define a set of equations which are linear in terms of the $\alpha^{\{m+1\}}$ parameters of the new alignment $\boldsymbol{B}^{\{m+1\}}$, $\boldsymbol{T}^{\{m+1\}}$. Solving these equations provides an improved alignment which is in turn used to compute a new set of equations. The process is repeated until convergence. Here, we simply state the linear update equations, with detailed derivation in the appendix. In each case, when solving for the $\{m+1\}$ parameters, the m-th registration values are treated as constants.

Notations. To simplify the description of the minimization, we define the followings:

$$\begin{cases} \overline{\phi_{ij\theta}}(\boldsymbol{B}, \boldsymbol{T}_j) = \frac{\phi_{ij\theta}(\mathbf{b}_i, \mathbf{t}_{j\theta})}{\sum_l \sum_h \phi_{hjl}(\mathbf{b}_h, \mathbf{t}_{jl}) + \kappa}, \\ \mathbf{D}_{ij\theta} = \overline{\phi_{ij\theta}}(\boldsymbol{B}^m, \boldsymbol{T}_j^m) \begin{bmatrix} \mathbf{b}_{0i}^c & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{b}_{0i}^c & \mathbf{0}_{3\times1} \end{bmatrix}, \\ \mathbf{F}_{ij\theta} = \overline{\phi_{ij\theta}}(\boldsymbol{B}^m, \boldsymbol{T}_j^m) \begin{bmatrix} -\mathbf{t}_{0j\theta(1)} & \mathbf{b}_{0i(1)} & -\mathbf{b}_{0i(2)} \\ -\mathbf{t}_{0j\theta(2)} & \mathbf{b}_{0i(2)} & \mathbf{b}_{0i(1)} \end{bmatrix}^T, \\ \mathbf{G}_{(i,j)} = g\left(\mathbf{b}_{0i}^c - \mathbf{b}_{0j}^c, \gamma\right), \end{cases}$$

where $\mathbf{D}_{ij\theta}$ and $\mathbf{F}_{ij\theta}$ are 6 × 3 and 3 × 2 data matrices, $\mathbf{G}$ is $M \times M$ affinity matrix and γ of Eqn (7) controls degree of smoothness.

The feature coordinate vectors are

$$\mathbf{c}_{ij\theta} = (\mathbf{b}_i^c)^{\{m+1\}} - (\mathbf{t}_{j\theta}^c)^{\{m+1\}},$$
$$\mathbf{r}_{ij\theta} = (\mathbf{b}_i^r)^{\{m+1\}} - (\mathbf{t}_{j\theta}^r)^{\{m+1\}}.$$

Operator $\sum_{i,j,\theta}$ represents a triple sum of indexes i, j and θ.

Similarity Transformation. Eqn (3) is updated by solving the 5 linear equations

$$\sum_{i,j,\theta} \left(\sigma_r^2 \mathbf{F}_{ij\theta}(\mathbf{c}_{ij\theta}) + \sigma_c^2 \overline{\phi_{ij\theta}}(\boldsymbol{B}, \boldsymbol{T}_j)\mathbf{r}_{ij\theta}\right) = \mathbf{0}_{3\times1},$$
$$\sum_{i,j,\theta} \overline{\phi_{ij\theta}}(\boldsymbol{B}, \boldsymbol{T}_j)\mathbf{c}_{ij\theta(1:2)} = \mathbf{0}_{2\times1}, \quad (9)$$

for the parameters $\{s, u, v, t_1, t_2\}^{\{m+1\}}$.

Affine Transformation. Eqn (4) is updated by solving the 9 linear equations

$$\sum_{i,j,\theta} \mathbf{D}_{ij\theta}(\mathbf{c}_{ij\theta}) = \mathbf{0}_{6\times1},$$
$$\sum_{i,j,\theta} \left(\overline{\phi_{ij\theta}}(\boldsymbol{B}, \boldsymbol{T}_j)\mathbf{r}_{ij\theta}\right) = \mathbf{0}_{3\times1}, \quad (10)$$

for the parameters $\{\mathbf{A}, s, u, v\}^{\{m+1\}}$.

Smoothly Varying Orientation and Affine. Parameterization of Eqn (5) is updated by solving $6 + 9 \times M$ linear equations

$$\sum_{i,j,\theta} \mathbf{D}_{ij\theta}(\mathbf{c}_{ij\theta}) = \mathbf{0}_{6\times1},$$
$$\mathbf{G}\mathbf{V} + 2\lambda\sigma_c^2 \Delta\mathbf{A} = \mathbf{0}_{M\times6},$$
$$\mathbf{G}\mathbf{U} + 2\lambda\sigma_r^2 \Delta\mathbf{R} = \mathbf{0}_{M\times3}, \quad (11)$$

for the parameters $\{\mathbf{A}, \Delta\mathbf{A}_i, \Delta s_i, \Delta u_i, \Delta v_i\}^{\{m+1\}}$. $\mathbf{V}$ and $\mathbf{U}$ are matrices whose respective i-th rows are

$$\mathbf{V}_{(i,:)} = \sum_{j\theta} \left(\mathbf{D}_{ij\theta}\mathbf{c}_{ij\theta}\right)^T, \quad \mathbf{U}_{(i,:)} = \sum_{j\theta} \overline{\phi_{ij\theta}}(\boldsymbol{B}, \boldsymbol{T}_j)\left(\mathbf{r}_{ij\theta}\right)^T$$

and the i-th row of matrices $\Delta\mathbf{A}$ and $\Delta\mathbf{R}$ are

$$\Delta\mathbf{A}_{(i,:)} = \left[\Delta\mathbf{A}_{i(1,1:3)}^{\{m+1\}}, \Delta\mathbf{A}_{i(2,1:3)}^{\{m+1\}}\right], \quad \Delta\mathbf{R}_{(i,:)} = \left((\mathbf{b}_i^r)^{\{m+1\}}\right)^T.$$

Feature Re-localization. Eqn (8) uses the same smoothly varying affine spatial parameterization as Eqn (11) but replaces orientation choice with a spatial correspondence choice. Its $6 + 6 \times M$ linear update equations are

$$\sum_{i,j,k} \widetilde{\mathbf{D}}_{ijk}(\tilde{\mathbf{c}}_{ijk}) = \mathbf{0}_{6\times1},$$
$$\mathbf{G}\widetilde{\mathbf{V}} + 2\lambda\sigma_c^2 \Delta\mathbf{A} = \mathbf{0}_{M\times6}, \quad (12)$$

with $\tilde{}$ denoting matrix variants where orientation choice terms $\mathbf{t}_{j\theta}, \theta, \Theta$ and N are replaced with the spatial choice terms $\tilde{\mathbf{t}}_{jk}, k, K_i$ and M defined in Eqn (8).

Spatial warping of a 2 × 1 base image coordinate $\mathbf{z}$ is defined by the sum of the global affine $\mathbf{A}$ and its smoothly varying affine offset $a(\mathbf{z})$. $a(\mathbf{z})$ is a continuous function defined as

$$\mathbf{W}_{M\times6} = [\mathbf{w}_1, ..., \mathbf{w}_M]^T = \mathbf{G}^+ \Delta\mathbf{A},$$
$$a(\mathbf{z}) = \sum_{i=1}^{M} \mathbf{w}_i g(\mathbf{z} - \mathbf{b}_{0i(1:2)}, \gamma), \quad (13)$$

where $\mathbf{G}^+$ is $\mathbf{G}$'s pseudo-inverse, and $\mathbf{w}_i$ is a 6 × 1 vector.

2.4. Implementation

System implementation details are as follows. SIFT descriptors are computed using VL-SIFT that allows specification of the desired feature orientation. After feature creation, image coordinates of features are normalized to zero mean, unit variance. Image alignment is computed using Algorithm 1. For the similarity transformation, its initial scale value s is set to 1, while the remaining $\{u, v, t_1, t_2\}$ parameters are initialized to the orientation neutral 0. The remaining transformations are initialized from the previous estimate. Throughout the algorithm, σ_d of the SIFT features is held constant, while σ_c and σ_r of the evolving coordinates are annealed smaller to force alignment. Parameters for the various algorithm stages are listed below. Unless otherwise stated, these parameters are used throughout the paper and the algorithm is relatively insensitive to minor variations in parameter choice.

	initial σ_c^2	initial σ_r^2	σ_d^2	λ	γ
Similarity	0.1	0.01	0.04	–	–
Affine	0.01	0.0004	0.04	–	–
Smooth. Var.	0.01	0.0004	0.04	20	4
Re-loc.	0.01	–	0.04	20	4

```
Input: Base image, Target image
for each parameterization do
    while $\sigma_c$ above threshold do
        while no convergence do
            switch{parameterization}
            case Similarity:
            update alignment with Eqn (9)
            case Affine:
            update alignment using Eqn (10)
            case Smoothly Varying Orientation and
            Affine:
            update alignment using Eqn (11)
            case Re-localization:
            update alignment using Eqn (12)
        end
        Anneal $\sigma_c = \epsilon\sigma_c$, $\sigma_r = \epsilon\sigma_r$,where $\epsilon = 0.97$.
    end
end
Output: Aligned images
```

Algorithm 1: Overall algorithm

3. Results

We align image pairs with relative scale and rotation ranges of $[0.5, 2]$ and $[-45°, 45°]$. While the algorithm can handle larger orientation changes, there is a trade off between stability to appearance variations and the permitted orientation range. Over these ranges, a mixed Matlab and C implementation of our algorithm can handle most 360×480 images within15 minutes on an Intel Core i7 computer.

In Sections 3.1 and 3.2, we evaluate our alignment results, while Sections 3.3 and 3.4 demonstrate Camera Pose Recovery and High Dynamic Range imaging as two of the applications of our alignment method.

3.1. Evaluation

To evaluate warping quality over illumination change, we capture a set of images at different locations. At each location, images are taken at a number of fixed illumination conditions. Warping is computed between images taken at different illumination conditions. The warp is transfered to image pairs taken under similar illumination conditions, which serves as evaluation ground-truth.

We compute the root-mean-square error (rmse) between images. We also compute the percentage of warped pixels whose target surroundings do not contain any similar pixels (% outliers). Surroundings are defined as a 4-pixel radius while the similar pixel threshold is set at 10 gray levels. The latter metric is similar in spirit to the “earth movers distance” and seeks to filter away illumination variation noise while avoiding over penalization of small discrepancies.

We evaluate two scenes, with the second containing significant depth discontinuity. While our algorithm's error is clearly higher on the second scene that violates its smoothness assumption, it remains stable. For benchmarking purposes, we show the results for SIFT flow [12] and Large Displacement optical flow [21].

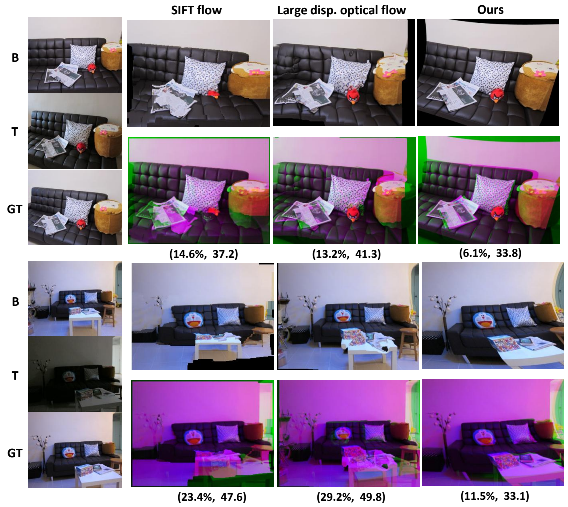

Figure 3. Left column: Base image, target image and reference ground truth. Right columns: Warps and overlays for SIFT flow [12], Large Displacement optical flow [21] and our algorithm. Errors are given below the images in the form (% outliers, rmse).

3.2. Alignment

We evaluate our alignments using standard image dataset from Mikolajczyk *et al.* [14] and Internet images. Alignments are visualized by replacing the warp's green channel with that of the target image. We also show the results of NRDC [9] and SIFT flow [12]. Our alignments are finer than SIFT flow's and we handle orientation change better. Our warps are denser than NRDC's correspondence, especially for less textured scenes. Dense warping is much more difficult than sparse matching, as it requires maintenance of a reasonable error rate while trying to maintain a perfect recall. Some examples of Internet image alignment are shown in Figure 4, with results for Mikolajczyk *et al.*'s [14] dataset in the supplementary material.[1]

For Internet images, our algorithm works well between scale ranges of $\times 3$ and out of plane rotation of $\pm 20^\circ$ degrees. Most color variations can be handled as long as images are taken during the day, however, the illumination variations caused by night photography are beyond our algorithms capability.

Note that these results are obtained without explicit color modeling, thus validating our hypothesis that proper geometric handling of SIFT descriptors is sufficient to compensate for large variations in image appearance.

3.3. Camera Pose

Our dense alignment also facilitates "tracking" a point across all images of a set. This is advantageous to camera pose recovery. By propagating correspondence across all images and using Hartley normalized image coordinates to reduce size variation caused by differing focal lengths or image resolutions, we can utilize the very stable factorization [22] algorithm for pose recovery from orthographic image streams. This recovers of camera pose up to a forward translational ambiguity. Many Internet images can be modeled as orthographic and reconstruction of a user chosen image set is shown in Figure 5. Works such as "Building Rome in a Day" [2] cannot reconstruct the scene from such small image sets because of the paucity of SIFT correspondences (approximately zero for this set).

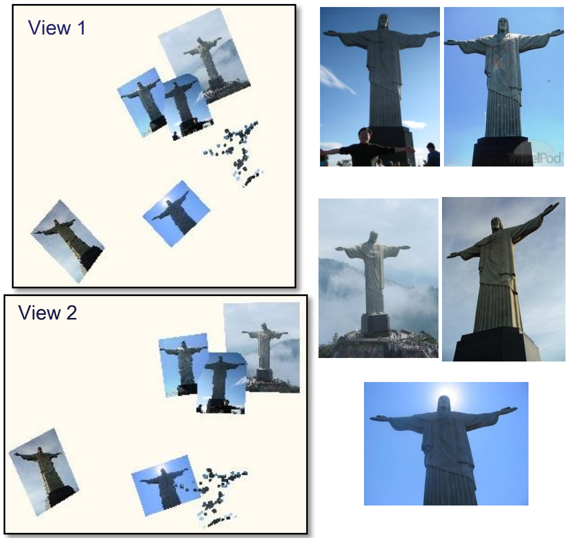

Figure 5. Left: Two different 3-D views of an orthographic reconstruction. Images represent camera positions arranged spherically around a reconstructed point cloud. Right: Input images.

3.4. High Dynamic Range Imaging

Another application is High-Dynamic-Range (HDR) Imaging, a set of techniques to increase the dynamic range between the lightest and darkest areas of an image. Typically, this is achieved by merging multiple standard-dynamic-range (SDR) images. Tone-mapping is used to display the HDR image on lower dynamic range devices [18].

One difficulty of HDR imaging is the alignment of multiple SDR images of different exposures. However, even under-exposed images contain SIFT feature our algorithm can utilize to align images. This enables HDR capture with a hand-held camera.[2] While better results can be obtained by first applying exposure correction techniques, our algorithm directly aligns the un-processed images. Results are shown in Figure 6.

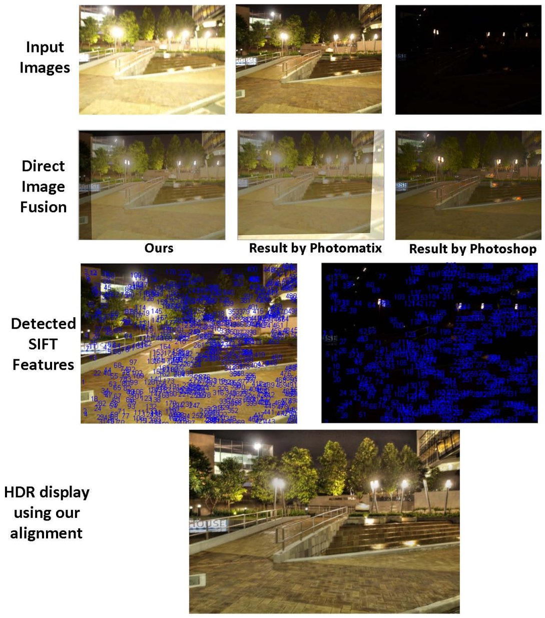

Figure 6. Alignment of SDR images using ours, Photomatix and Photoshop. Alignment quality is visible in direct fusion without de-ghosting. Other alignment software incur significant ghosting. Observe the surprising number of SIFT features on the under-exposed image. This allows us to align it with the other images despite significant appearance changes. Final result, an artistically chosen tone-mapped image after using our algorithm's alignment.

4. Discussions

Our descriptor-based algorithm can align images over large appearance variations with only geometric constraints. While there are limits to the tolerable variations (we are restricted to day images), our approach reduces the reliance on photometric models and lays the foundation for better

[1] Our algorithm's emphasis on descriptor distances does not always correspond to human perception. In particular, when matching faces, our algorithm's low emphasis on the un-textured eyes can result in an overall correct warping outline but mis-aligned facial fiducial points.

[2] This is different from matching day and night images which our algorithm does not perform well on.

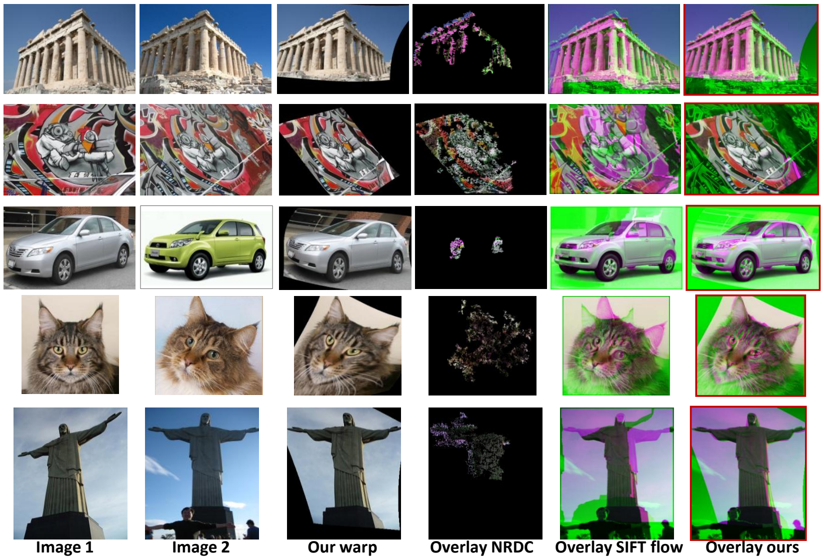

Figure 4. Left to right: Input images, our warped results and overlays for NRDC [9], SIFT flow [12] and our algorithm. Observe that we handle orientation change better than SIFT flow, while NRDC's correspondence for less textured scenes is quite sparse.

results by employing both techniques. More practically, this opens opportunities for harnessing rich, Internet image content. An example is computation of relative Internet image poses demonstrated in this paper. Our approaches drawback is smoothing over discontinuous motion. This causes errors when there is significant parallax.

Acknowledgment: This work is partly supported by Singapore MOE Academic Research Fund (Project Number: T1 251RES1104).

References

[1] A. E. Abdel-Hakim and A. A. Farag. Csift: A sift descriptor with color invariant characteristics. *CVPR*, 2006. 3

[2] S. Agarwal, Y. Furukawa, N. Snavely, B. Curless, S. M. Seitz, and R. Szeliski. Building rome in a day. *ICCV*, 2009. 2, 7

[3] R. J. Andrews and B. C. Lovell. Color optical flow. *Workshop on Digital Image Computing*, 2003. 1

[4] C. Barnes, E. Shechtman, and A. F. Dan B Goldman. The generalized patchmatch correspondence algorithm. *ECCV*, 2010. 2

[5] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *ECCV*, 2006. 2

[6] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *PAMI*, 1989. 3

[7] H. Cheng, Z. Liu, N. Zheng, and J. Yang. A deformable local image descriptor. *CVPR*, 2008. 2

[8] B. Fan, F. Wu, and Z. Hu. Aggregating gradient distributions into intensity orders. *CVPR*, 2011. 2

[9] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski. Nrdc: Non-rigid dense correspondence with applications for image enhancement. *SIGGRAPH*, 2011. 1, 2, 6, 8, 12

[10] I. Kemelmacher-Shlizerman and S. M. Seitz. Face reconstruction in the wild. *ICCV*, 2011. 2

[11] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L. F. Cheong. Smoothly varying affine stitching. *CVPR*, 2011. 2, 3, 4, 9

[12] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: Dense correspondence across different scenes. *ECCV*, 2008. 2, 6, 8, 12

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2, 3

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 2006. 6, 7, 12

[15] E. G. Miller and K. Tieu. Color eigenflows: Statistical modeling of joint color changes. *ICCV*, 2001. 3

[16] J. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2009. 2

[17] A. Myronenko, X. Song, and M. Carreira-Perpinan. Non-rigid point set registration: Coherent point drift. *NIPS*, 2007. 3, 4, 9

[18] E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda. Photographic tone reproduction for digital images. *Siggraph*, 2002. 7

[19] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *Siggraph Asia*, 2011. 2

[20] H.-Y. Shum and R. Szeliski. Construction of panoramic image mosaics with global and local alignment. *IJCV*, 1999. 2

[21] T.Brox and J.Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *PAMI*, 2010. 6

[22] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography a factorization method. *IJCV*, 1992. 7

[23] J. van de Weijer and T. Gevers. Robust optical flow from photometric invariants. *ICIP*, 2004. 1

[24] A. Vedaldi and S. Soatto. Local features, all grown up. *CVPR*, 2006. 2

[25] A. L. Yuille and N. M. Grywacz. The motion coherence theory. *ICCV*, 1988. 3