

Alignment of Images Captured Under Different Light Directions

Sema Berkiten and Szymon Rusinkiewicz,

Princeton University

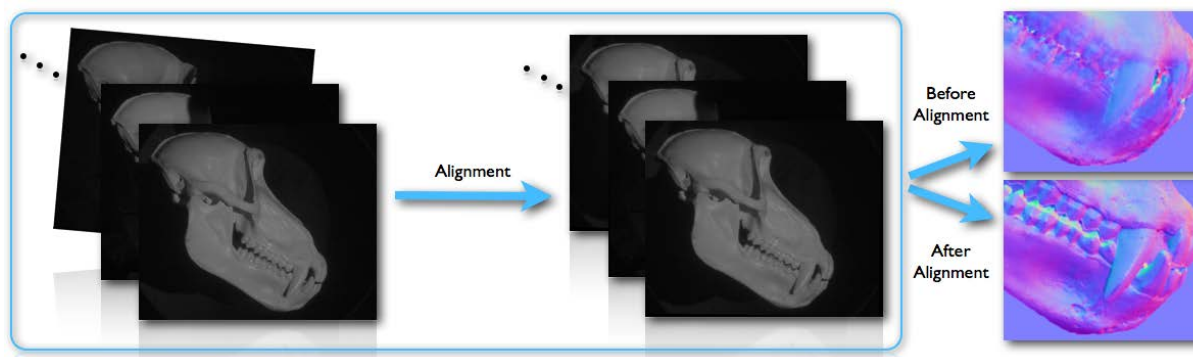


Figure 1: Misaligned images are aligned with our graph-based alignment method so that they can be used in various applications such as photometric stereo. **Left to right:** images before and after alignment, and normal maps calculated from misaligned and aligned images.

Abstract

Image alignment is one of the first steps for most computer vision and image processing algorithms. Image fusion, image mosaicing, creation of panoramas, object recognition/detection, photometric stereo and enhanced rendering are some of the examples in which image alignment is a crucial step. In this work, we focus on alignment of high-resolution images taken with a fixed camera under different light directions. Although the camera position is largely fixed, there might be some misalignment due to perturbations to the camera or to the object, or the effect of optical image stabilization, especially in long photo shoots. Based on our experiments, we observe that feature-based techniques outperform pixel-based ones for this application. We found that SIFT [Low04] and SURF [BTVG06] provided very reliable features for most cases. For feature-based approaches, one of the main problems is the elimination of outliers, and we solve this problem using the RANSAC framework. Furthermore, we propose a method to automatically detect the transformation model between images. The datasets that we focus on have around 10-100 images, of the same scene, and in order to take advantage of having many images, we explore a graph-based approach to find the strongest connectivities between images. Finally, we demonstrate that our alignment algorithm improves the results of photometric stereo by showing normal maps before and after alignment.

1. Introduction

Image alignment is the process of overlaying images of the same scene under different conditions, such as from different viewpoints, with different illumination, using different sensors, or at different times. In this work, we focus on the alignment of multiple images of the same scene under varying illumination.

In the literature, there are many works on image alignment, and most of them aim to solve a specific problem. For example, algorithms for medical image alignment mostly use intensity values of the image, while object recognition algorithms usually try to find some descriptive salient features. In short, the structure of the algorithm is shaped by the specifications of the problem itself.

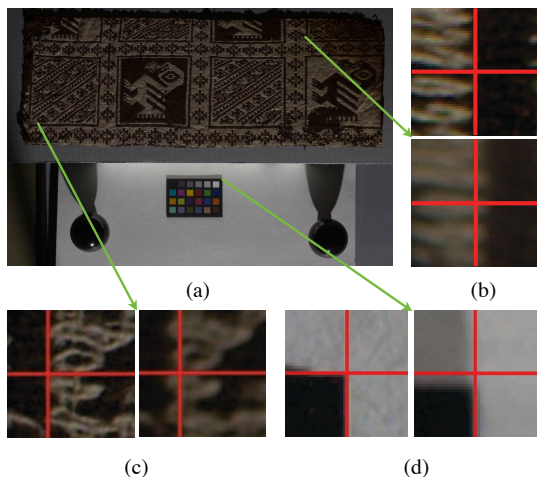


Figure 2: Effect of image stabilization (IS). *a*) A sample image from the textile dataset. *b*), *c*), and *d*) Closeup images showing the image stabilization effect, red lines are placed on the same locations to show the misalignment between frames because of IS.

In this work, we focus on aligning images of the same object under different light directions, keeping other conditions almost the same. Our main contribution is four-fold:

- Evaluation of which approaches to image alignment are most stable under different illumination,
- Progressively solving for more and more complex image transformations, to use the least general (and hence most robust) model necessary for good alignment,
- Simultaneous alignment of multiple images using spanning trees of graphs,
- Demonstration of the effect of the image alignment on photometric stereo.

The image datasets that we focus on are usually used as inputs to other applications, such as polynomial texture mapping (PTM) [MGW01], shape and detail enhancement [FAR07], photometric stereo and enhanced rendering [MWGA06] etc. For example, PTM takes several images taken from the same view point but under different light directions, and constructs the coefficients of a bi-quadratic polynomial per texel. These coefficients are used later to reconstruct the surface color under varying light directions. Although the camera and the object in the capture setup should be kept still, sometimes there might be some misalignment between images because of perturbations to the camera/object in the scene or the effect of optical image stabilization. For example, Figure 2 shows some misalignments caused by image stabilization. Therefore, image registration is necessary to make sure that input images are well aligned.

However, many techniques in the literature are either only partially invariant to illumination change or not invariant

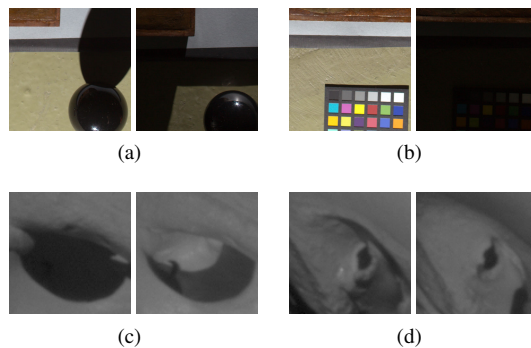


Figure 3: Closeup images from the panel dataset (top row) and the skull dataset (bottom row). Different light directions might cause different cast shadows (*a*, *b*, and *c*), moving specular highlights (*a*, and *d*), and different visible edges with different sharpness (all).

at all [Zit03]. For images taken under different light directions, the main problem is the non-linear illumination change, which is not trivial to deal with because different light directions may lead to cast shadows, moving specular highlights, local changes in brightness on the image, or even the loss of some information (e.g. if the light direction is perpendicular to a geometric edge on the object, it might not be recognizable on the image; or a texture edge might not be visible if it is in shadow), Figure 3 illustrates these problems. In early experiments, we observed that pixel-based methods such as normalized cross correlation, L2 difference of intensity values, pixel dot product etc. fail to find correct alignments for our datasets most of the time. It is because pixel-based methods rely on intensity values only, and these are very unreliable under non-linear illumination changes. That is why, in this work, we focus on feature-based methods.

In this paper, we first summarize some image registration algorithms in the literature. Then we explain our approach to solve the matching problem for images taken under different illumination. Finally, we show the results.

1.1. Related Work

Image registration (alignment) is transforming a source image to the coordinate system of the reference image. Images may be taken either at different times, from different views, under different lighting, or with different sensors. Researchers have been working on the image registration problem for decades to solve alignment problems in various fields such as medical imaging, remote sensing, image fusion, panorama, change detection, recognition etc.

Brown [Bro92] published a very broad survey on image registration techniques in 1992. More recently, Zitova et al. published another survey paper in 2003 including more recent works on image alignment [Zit03]. In 2006, a tutorial

for image alignment and stitching was published by Szeliski [Sze06].

The many algorithms for image alignment can be broadly categorized as pixel-based or feature-based. Pixel-based methods use intensity values directly: for example they may rely on normalized-cross-correlation and its variants, similarity measures using pixel dot products or L2 distance. Matching, fitting and validation steps are calculated simultaneously for a preselected transformation model [Zit03].

For template matching, Kaneko et al. proposed the selective correlation coefficient, which is very similar to cross-correlation (CC), but it extracts a correlation mask-image differently than CC. They used increment sign correlation to extract the mask, and the mask is enhanced with four-pixel majority rule [KMI02], [KSI03]. Silveira et al. proposed a method for real-time visual tracking. They modeled the illumination change and image motion by solving a second-order optimization problem minimizing the intensity difference based on illumination and image motion models [SM07]. By using only the strongest image gradients with a pyramidal refinement strategy, Eisemann and Durand align flash and no flash images [ED04].

One of the well known pixel-based algorithms in computer vision and graphics is optical flow, proposed by Lucas and Kanade [LK81]. It assumes constant flow in a local neighborhood and solves optical flow equations by least squares. Optical flow and its variants have been used to solve image alignment problems in various works [KUWS03], [KMK05], [Bar06], [BAHH92]. To find only translational misalignment, Ward proposed median threshold bitmaps in image pyramids for hand-held photographs with varying exposures [War03].

Feature-based algorithms consist of three main blocks: salient feature extraction, feature matching, and estimation of the transformation. Various types of feature detection algorithms have been proposed throughout the years such as line, contour, and region detectors. However, salient feature points are easier to deal with than lines, contours or surfaces. The Harris corner detector [HS88] has been used for many years to detect corner-like points. Recently, feature descriptors became more popular because of their distinctive and invariant natures. The Scale-Invariant Feature Transform (SIFT) was proposed by Lowe in 2004; since then, it has been used widely because of its shift and scale-invariance and its distinctive descriptors [Low04]. Furthermore, Lowe showed that SIFT shows high performance on object recognition. Later on, Brown and Lowe used SIFT on unordered panorama images for stitching, together with the RANSAC framework for outlier elimination and a probabilistic model for verification [BL07]. Tang et al. similarly used a variant of SIFT with RANSAC to align medical microscopic sequence images [TDS08]. It is shown in [WWX*10] that the same approach (SIFT and RANSAC) works well for multi-modal image registration-aligning infrared to visible images. Bay

et al. proposed a similar but faster approach to SIFT called Speeded Up Robust Features (SURF) [BTVG06]. Winder et al. introduced another configuration to compute salient feature descriptors and presented comparisons to SIFT [WB07], [WHB09].

For feature matching, the least efficient method is brute force comparison of L2 distance between feature descriptors. If the number of features is large, such as for object recognition, k-d trees or similar data structures can be used to speed up the search [BL07]. Even if a highly distinctive descriptor is used, there might be some false matches called outliers, and a randomized framework to eliminate outliers called RANSAC is often preferred because it works with up to fifty percent outliers [Fis81]. Mikolajczyk et al. published comparisons of steerable filters, PCA-SIFT, differential invariants, spin images, SIFT, complex filters, moment invariants, and cross-correlation for different types of interest regions [MS05]. Also an intensive survey on local invariant feature descriptors can be found in [TM08].

As a hybrid of feature- and pixel-based methods, normalized cross-correlation is used with a Harris-Laplacian detector in [ZHG06] to make NCC rotation and scale invariant.

1.2. Overview

The outline of the rest of paper is as follows:

- **Single-target alignment:** Our core algorithm uses salient features and the RANSAC framework to eliminate outliers.
 - **Feature detection:** We propose a normalized Harris corner detector to extract features. Also, we experimented on SIFT [Low04] and SURF [BTVG06] salient features and show that they outperform the Harris detector.
 - **Feature matching:** We use the Euclidean distance between feature locations for normalized Harris corners, Euclidean distance between feature descriptors for SIFT and SURF for feature matching. Wrong matches because of impreciseness in feature detectors are eliminated with RANSAC [Fis81].
 - **Progressive Transformation:** We propose an algorithm which automatically detects the best transformation type between two images instead of assuming one transformation type such as affine or projective.
- **Graph-based alignment:** Instead of aligning each image in the dataset to one target image independently, we show that alignment can be improved by constructing a spanning tree based on image similarities and aligning each image to its neighbor towards the root.
- **Results:** We showed the results of our alignment algorithm on several datasets, and demonstrate that the image alignment improves the results of photometric stereo.

2. Single-Target Image Alignment

Image alignment is a correspondence problem of mapping one image to another. In this section, we explain the single-target alignment algorithm in three sections: feature extraction, feature matching, and transformation models.

2.1. Feature Extraction

Although invariance to geometric deformation, and shift and scale invariance to illumination are explored well in previous work, there is no feature extraction algorithm which is invariant to non-linear illumination changes such as varying light direction. We therefore explore a number of feature detection algorithms on our datasets, with the aim of discovering which ones perform best in the presence of large-scale lighting changes.

2.1.1. Normalized Harris Corner Detector

Intuitively, we expect that many corner-like features in an object's texture can be precisely localized regardless of illumination. We therefore begin by exploring the performance of the Harris corner detector. Unfortunately, while this detector is invariant to shifts in brightness (since it is based on the gradient), it is not invariant to multiplicative changes in brightness. As a result, this detector is highly sensitive to illumination, and extracts more corner points in bright regions. We therefore explore a *normalized* Harris corner detector, based on a structure tensor that is normalized by the local image intensity:

$$C = \frac{\begin{bmatrix} \sum_W g(i,j)I_x(\mathbf{w})^2 & \sum_W g(i,j)I_x(\mathbf{w})I_y(\mathbf{w}) \\ \sum_W g(i,j)I_x(\mathbf{w})I_y(\mathbf{w}) & \sum_W g(i,j)I_y(\mathbf{w})^2 \end{bmatrix}}{\sum_W g(i,j)I(\mathbf{w})^2} \quad (1)$$

where W is the interest window around each pixel, (i, j) are pixel offsets within the window, g is a 2D Gaussian filter, \mathbf{w} is the global pixel location, and $I(\mathbf{w})$ is the image intensity at that pixel.

2.1.2. SIFT

The Scale-Invariant Feature Transform (SIFT) is a scale- and rotation-invariant local feature detector proposed by Lowe [Low04], and used for many computer vision problems since then. There are several works showing its robustness in the literature [MS05], [Pav08], [KMW11]. SIFT consists of four main steps: scale-space extrema detection, accurate key-point localization, orientation assignment, and calculation of a key point descriptor.

In the first step, candidate interest points are detected by finding extrema over scale and image space. The second step refines the interest points' locations by fitting a 3D quadratic function to the scale-space function, which is approximated by the Taylor expansion. In the third step, each feature point is assigned a dominant orientation, which is detected by finding the maximum of the local orientation histogram around

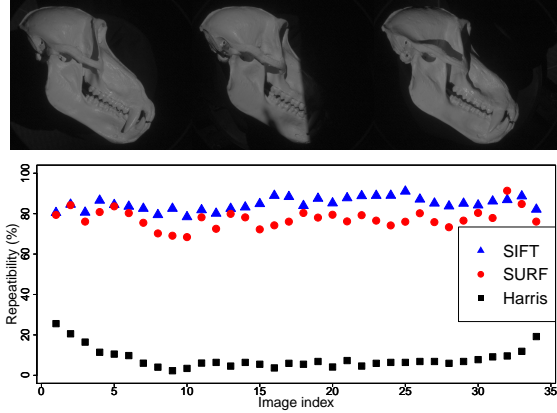


Figure 4: *Top:* Three images from the skull dataset. *Bottom:* Repeatability ratios of different feature detectors on this dataset. (First image is selected as the target image.)

the feature point. In the final step, a descriptor vector is calculated for each feature point. This descriptor is built by concatenating local orientation histograms of 4×4 sub-windows of a 16×16 window around the feature point. Further details of SIFT can be found in [Low04].

2.1.3. SURF

The Speeded-up robust feature (SURF) detector proposed by Bay et al. [BTVG06], is also a scale and rotation invariant local feature detector which is partially inspired by SIFT. Its main purposes are to be computationally less expensive and to be as distinctive as SIFT. In order to speed-up the computations, SURF uses integral images and box-filter approximations to the second derivative of Gaussian.

2.1.4. Comparison of Feature Extraction Methods

In Figure 4, the repeatability of features on a typical dataset is explored for the three different types of feature detectors above. The repeatability ratio between the target (t) and the source (s) images is formulated as follows:

$$R = \frac{\text{Number of } fp_i\text{'s}}{\text{Total number of features}} \quad (2)$$

where,

$$fp_i = \{ \text{feature pair } i \mid \sqrt{2} \geq \| (x_s(i), y_s(i)) - (x_t(i), y_t(i)) \| \}$$

It is obvious that, despite the improvements of normalization, the Harris corner detector is outperformed by SIFT and SURF on this dataset, and indeed we find that this behavior generalizes across many different types of images. As a result, we use SIFT as the feature extractor throughout the rest of the paper.

Method selected by progressive algorithm	Ground-truth error for:				
	Translation	Tr+Rot	Tr+Rot+Sc	Affine	Projective
Translation	0.13	0.25	0.33	0.38	15.7
Translation+Rotation	6.26	0.23	0.23	0.44	4.7
Translation+Rotation+Scale	17.7	2.64	0.43	0.56	4.7
Affine	14.5	2.94	1.13	0.47	2.1
Projective	18.4	2.5	1.01	0.55	0.23

Table 1: Average ground-truth alignment errors in pixels for image collections in Table 2. The progressive algorithm selects the method shown at left in each row. As seen from the ground-truth errors (to which the progressive algorithm did not have access), the progressive method generally picks the transformation type giving the lowest error.

2.2. Feature Matching

The naive way to match feature points is to calculate the Euclidean distance between feature descriptors and compare them. Another approach is to calculate the ratio of Euclidean distance to the closest neighbor and to the second closest neighbor and eliminate the ones which have high ratio, because a low ratio implies that it is a correct correspondence with high probability. For tasks which require searching a large feature database, such as object recognition, special data-structures such as k-d tree or search strategies are used. For example, [Low04] uses the Best-Bin-First (BBF) search algorithm, which returns the closest neighbor with high probability. The BBF is a variant of k-d tree, using a priority queue based on closeness, and it terminates after a specific number of neighbors are explored.

In this work, we use the naive method (exhaustive search on Euclidean distances), but we apply a constraint on the distance between the locations of feature points on the image plane, because we have assumed the deformation between images will be small which is the property of our specific problem, so corresponding points cannot be very far away from each other. In particular, we eliminate all matches if the feature points are more than 128 pixels apart.

2.3. Transformation Models

To calculate a transformation matrix for given feature mappings between two images, the transformation type has to be selected first. We consider several classes of transformations, of increasing numbers of degrees of freedom (DOF): translation only (2 DOF), translation and rotation (3 DOF), similarity (4 DOF), affine (6 DOF), and projective or homography (8 DOF).

The projective transformation model is the most generic among all, which is why it is commonly used, especially if the type of deformation is not given a priori. However, this generality comes at a price, since incorrect correspondences and even small errors in feature point localization can result in significant errors in the transformation.

2.3.1. Progressive Transformation Model

Selection of the transformation model determines the number of degrees of freedom, and thereby the constraints on the transformation matrix. When the deformation type on input images is not known in advance, the projective transformation model is commonly chosen. However, when there is only a translational difference between two images, for example, the estimated transformation will be more erroneous than it would be if a translational model were selected, because of localization error on the feature extraction step.

In order to obtain maximally accurate and robust estimates of the transformation, we propose a *progressive* model to select a deformation type automatically, similar to the model selection algorithm proposed by [Tor98]. In particular, we use the following algorithm inspired by RANSAC:

Algorithm for progressive transformation

repeat

- For each transformation model, from translational to projective:
 - Estimate the transformation matrix by fitting the least squares on feature matches in a RANSAC framework, store the number of correspondences which agree with the estimated matrix (inlier).

until the maximum number of inlier matches is smaller than the predefined threshold, τ :

Select the final transformation: the one with the maximum number of inlier correspondences.

The reason why this approach works is the presence of localization error on the feature point locations. Otherwise, if all feature points were localized perfectly, we would expect to get the same results for different transformation models. In Table 1, average single-target alignment errors for different types of transformation models are shown. We observe that, in a majority of cases, the progressive algorithm picks the transformation model giving the lowest ground-truth pixel error.

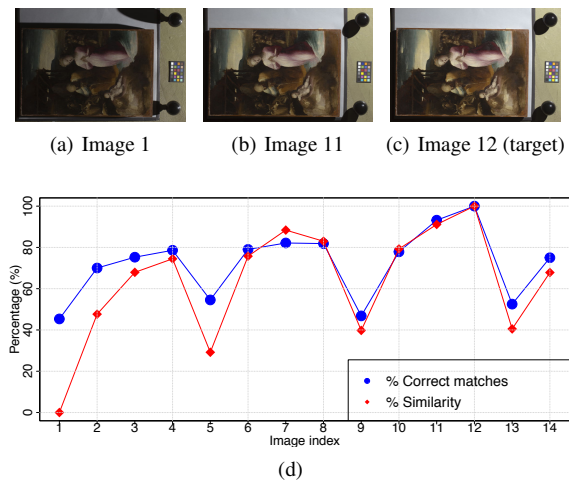


Figure 5: (a-c) Images 1, 11, and 12 in the dataset. (d) Percentage of correct feature matches, as well as image similarities (inverse of normalized and scaled L2 differences of low-resolution images) for the panel dataset. Image 12 is the target image.

3. Graph-Based Image Alignment

So far we have explored single-target image alignment, but aligning all images in the dataset to one target image does not give good results for all cases. In particular, it is inevitable to get badly aligned results when images in the dataset have a lot of geometrical variations and less texture, such as images in the skull dataset, because different light directions will result in different shadows, varying local gradients, and thereby different local image features. For example, Figure 5 shows the number of correct matches between one fixed target image (Image 12) and the remaining images in the dataset. We observe that the more similar the illumination condition, the higher the number of correct matches.

Based on experiments such as this, we conclude that it is possible to leverage the availability of multiple images by not attempting alignment to a single target. Instead, we formulate multi-image alignment as finding a *spanning tree* in a graph in which each vertex V represents an image and each edge E represents similarity. For efficiency, it is desirable to have the edge weights easily computable. Fortunately, as shown in Figure 5, simple L2 image difference (on down-sampled images) is highly correlated with the number of correct feature matches. We may therefore use low-resolution L2 similarity as a proxy for feature similarity when constructing our graph. Also, we observed that this proxy gives higher weights for image pairs with close light positions.

One way of visualizing the similarities between images quantitatively is Laplacian Eigenmaps [BN01]. This is a spectral clustering technique used to solve the dimension re-

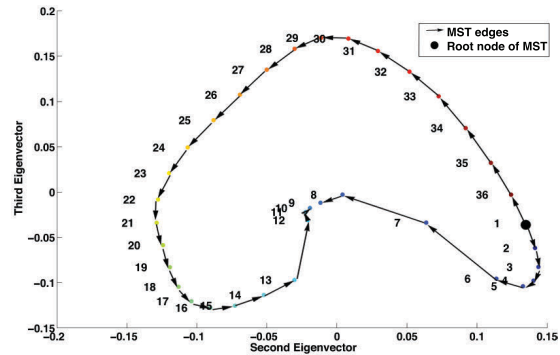


Figure 6: Each dot represents an image and located based on the second and third eigenvectors of the Laplacian Eigenmaps for the skull dataset (36 images). MST-edges are shown with black arrows, the first node is the root node of the MST.

duction problem. It takes an affinity matrix whose elements are the Euclidean distance between corresponding images. Its optimal solution is the eigenvector corresponding to the second smallest eigenvalue. Figure 6 shows the eigenvectors corresponding to the second and third smallest eigenvalues for the skull dataset of 36 images.

3.1. Minimum Spanning Tree vs Shortest Path Tree

Once we have constructed an image similarity graph G , we are left with the task of extracting a subset of edges on which to perform full image alignment. (Of course, including more than the minimum subset of edges and combining the results with least squares could reduce error, but also increases sensitivity to bad correspondences, and is not explored in this paper.) We compare two algorithms for extracting a spanning tree:

- A minimum spanning tree (MST) is the one which has the smallest total weight among all possible spanning trees of G . We use Prim’s algorithm [Pri57] to construct the MST.
- A shortest path tree (SPT) with root vertex v is the spanning tree of G containing all shortest paths from v to other vertices. Dijkstra’s algorithm [Dij59] can be used to construct an SPT from a connected graph.

Figure 7 shows alignment errors (on a logarithmic scale) for single-target alignment, SPT, and MST. We observe that graph-based methods typically outperform single-target alignment. Total alignment errors are indicated in the legends, and for the skull dataset we observe that the total single-target alignment error (130 pixels, 3.7 pixels per image) is far from acceptable. On the other hand, MST gives roughly 0.5 pixel error per image for the same dataset. For the panel dataset, alignment results for each approach are very close to each other (about 0.3 pixel error per image) because this dataset is feature-rich, texture-rich, and the object has a flat surface. On the other hand, the skull dataset is

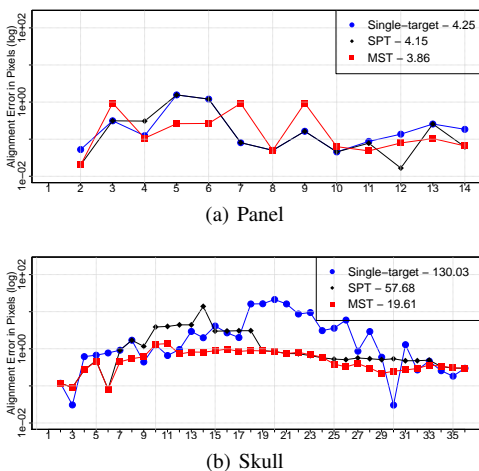


Figure 7: Alignment errors in pixels (in log scale) for each image. Single-target alignment, SPT, and MST for panel (a) and skull (b) datasets. The sum of alignment errors are indicated in the legend.

more challenging and MST outperforms both single-target and SPT in this dataset. Although it is not very clear that MST always performs better than SPT, we use MST for the tests in the Results section, because MST outperforms both single-target and SPT in the most challenging case that we have (skull dataset).

4. Results

In Table 2, the test results on different datasets for three different algorithms (single-target alignment with homographic transformation type, single-target alignment with progressive transformation, graph-based alignment with progressive transformation type) are demonstrated. Three test cases are formed: i) datasets with ground truths: images on the original datasets are perfectly aligned and images are manipulated randomly (transformation types, from translation only to homographic, and manipulation amounts are set randomly); ii) datasets with gold standard: the original datasets have misalignments and they are aligned by manually selecting correspondences to calculate the gold standard; iii) experimenting on the number of key-points on each image in the dataset. In the table, the third column is the average number of key-points on images in a dataset, the fourth column is the pixel resolutions of images in the dataset, and the subsequent columns show average and maximum alignment errors for the three algorithms. Alignment error for a given estimated matrix (E) and the ground truth matrix (G) is calculated as follows:

$$e = \frac{1}{4} \sum_{i=1,2,3,4} \|G^{-1}E p_i - p_i\| \quad (3)$$

where the p_i are the four corner points of the image. The reason for calculating the alignment error on the corner points

is that the maximum error will appear on the corners for the transformation types that we consider.

We observe that single-target alignment fails when the projective transformation model is assumed for large collections. On the other hand, the progressive model mostly gives successful results. The graph-based method using MST and progressive transformation leads to the alignment error of 0.5-1.5 pixels on average, while the single-target alignment algorithm results in alignment error of 0.5-7 pixels on average. And the graph-based algorithm works robustly on very challenging datasets such as the first and second collections in Table 2. While the maximum error is unacceptable on most of the datasets when the single-target alignment is used, the graph-based method gives acceptable results.

We also show the effect of the alignment on photometric stereo in Figure 3 by demonstrating the normal map before and after alignment. On the first three examples, it is obvious that the details cannot be recovered with photometric stereo if they are not well-aligned. And in the last row, we observed that misalignment causes embossing effect on moderately flat surfaces. Also, mean-square errors (MSE) between ground truth and each normal map are indicated under each closeup image. We observed that image alignment improves the MSE at least ten times.

5. Conclusion and Future Work

In this work, we proposed a feature-based framework to align images of the same object exposed to light from different directions. We showed that total alignment error for a dataset can be reduced by a graph-based approach rather than single-target alignment. Also, we demonstrated the importance of the image alignment by showing how much our algorithm improves the results of photometric stereo.

The obvious limitation of this work is caused by feature detectors because they are not invariant to non-linear illumination changes. Even though they can handle small illumination changes, there is no guarantee that feature locations and descriptors will be consistent for large changes. In particular, lack of texture and geometry variations results in less reliable salient features. In future work, in order to cope with unreliable image features, surface geometry (normal-maps) can be included to feature descriptors iteratively. For badly warped images, we can allow the user to add some hard constraints, such as selecting a few control points on target and source images.












Dataset	#Images	#Points	Resolution	Alignment Errors (in pixels)						Running-Time (in secs)		
				Projective		Progressive		MST-Progressive		Key calc /image	Single- target	Graph- based
				Avg.	Max	Avg.	Max	Avg.	Max			
Ground Truth: Perfectly aligned images are randomly manipulated (ranging from only translation to homographic manipulations) to create test sets.												
	36	430.7	1190x980	45.47	113	2.197	13.4	0.5957	1.87	1.7	3.9	2.2
	19	1748	2184x1456	43.19	245.3	6.676	32.87	1.351	3.081	4.1	9.6	8.2
	49	1073	1728x2592	71.04	284.4	0.5062	1.507	0.6031	1.571	1.8	7.4	10.0
	64	572.6	2184x1456	27.73	186.4	0.5868	4.11	0.9675	2.304	1.4	8.3	8.1
	36	1153	1312x864	29.25	88.99	0.6456	2.612	0.8497	2.281	1.8	9.5	7.1
	42	488.4	1024x1024	29.66	102.5	2.309	15.47	1.655	6.451	5.6	6.2	5.2
	34	986.7	2184x1456	47.74	286.5	3.46	38.84	1.698	3.732	14.1	19.0	14.7
Gold Standard: it is acquired by manually aligning images												
	47	873.3	1728x2592	105.6	406.4	0.8966	2.096	0.9756	1.73	1.6	7.5	7.9
	4	1206	2592x1728	13.61	46.68	1.148	2.143	1.277	2.658	27.53	1.15975	1.10477
Experiment on the number of key-features												
	68	4569	2799x1868	87.63	499.3	0.5423	1.333	0.6249	1.56	3.2	51.6	66.0
	68	489.3	2799x1868	104.6	484.3	1.052	5.943	0.9628	3.335	5.2	11.6	11.1

Table 2: Test results for different datasets.

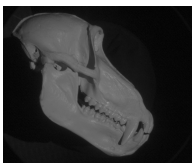
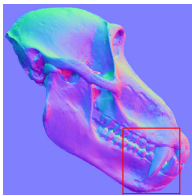
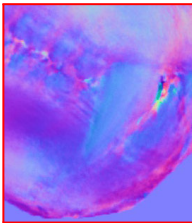
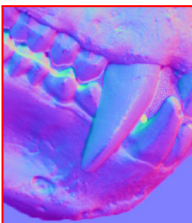
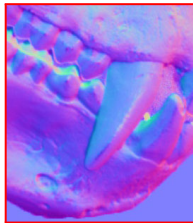
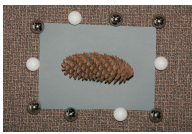
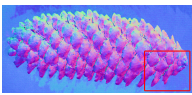
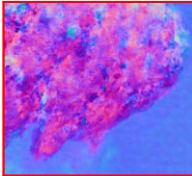
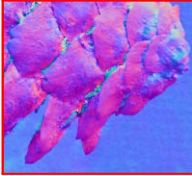
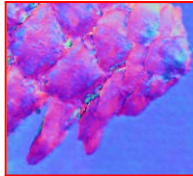







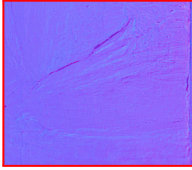

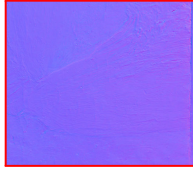
Sample Image	Normal Map for Ground Truth	Normal Maps for Misaligned dataset	Normal Maps for Ground Truth	Normal Maps after Alignment
		 MSE: 0.004		 MSE: 0.0003
		 MSE: 0.026		 MSE: 0.01
		 MSE: 0.005		 MSE: 0.0009
		 MSE: 0.001		 MSE: 0.0002

Table 3: Effect of alignment on photometric stereo. **The first column** shows example images from each dataset, the normal maps calculated on either ground truth or gold standard dataset are shown on **the second column**. **The third, fourth and the fifth columns** demonstrate close-up images of the normal maps calculated on misaligned, ground truth and aligned datasets respectively. The close-up regions are indicated with a red square on each normal map on the second column. Mean square errors (MSE) between normal maps for aligned and ground truth datasets and between normal maps for misaligned and ground truth datasets are indicated under the close-up images.

References

- [BAHH92] BERGEN J. R., ANANDAN P., HANNA T. J., HINGORANI R.: Hierarchical model-based motion estimation. Springer-Verlag, pp. 237–252. 3
- [Bar06] BARTOLI A.: Groupwise Geometric and Photometric Direct Image Registration. In *British Machine Vision Conference* (2006), pp. 157–166. 3
- [BL07] BROWN M., LOWE D. G.: Automatic panoramic image stitching using invariant features. *Int. J. Comput. Vision* 74, 1 (Aug. 2007), 59–73. 3
- [BN01] BELKIN M., NIYOGI P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14* (2001), MIT Press, pp. 585–591. 6
- [Bro92] BROWN L.: A survey of image registration techniques. *ACM computing surveys (CSUR)* (1992). 2
- [BTVG06] BAY H., TUYTELAARS T., VAN GOOL L.: Surf: Speeded up robust features. In *Computer Vision and ECCV 2006*, Leonardis A., Bischof H., Pinz A., (Eds.), vol. 3951 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2006, pp. 404–417. 1, 3, 4
- [Dij59] DIJKSTRA E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik 1*, 1 (1959), 269–271. 6
- [ED04] EISEMANN E., DURAND F.: Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.* 23, 3 (Aug. 2004), 673–678. 3
- [FAR07] FATTAL R., AGRAWALA M., RUSINKIEWICZ S.: Multiscale shape and detail enhancement from multi-light image collections. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 26, 3 (Aug. 2007). 2
- [Fis81] FISCHLER M.: Random sample consensus. *Communications of the ACM* (1981). 3
- [HS88] HARRIS C., STEPHENS M.: A combined corner and edge detector. *Alvey vision conference* (1988). 3
- [KMI02] KANEKO S., MURASE I., IGARASHI S.: Robust image registration by increment sign correlation. *Pattern Recognition* 35, 10 (2002), 2223 – 2234. 3
- [KMK05] KIM Y.-H., MARTÍNEZ A. M., KAK A. C.: Robust motion estimation under varying illumination. *Image and Vision Computing* 23, 4 (2005), 365 – 375. 3
- [KMW11] KHAN N., MCCANE B., WYVILL G.: Sift and surf performance evaluation against various image deformations on benchmark dataset. In *Digital Image Computing Techniques and Applications (DICTA), 2011 International Conference on* (dec. 2011), pp. 501 –506. 4
- [KSI03] KANEKO S., SATOH Y., IGARASHI S.: Using selective correlation coefficient for robust image registration. *Pattern Recognition* 36, 5 (2003), 1165 – 1173. 3
- [KUWS03] KANG S. B., UYTENDAELE M., WINDER S., SZELISKI R.: High dynamic range video. *ACM Trans. Graph.* 22, 3 (July 2003), 319–325. 3
- [LK81] LUCAS B. D., KANADE T.: An iterative image registration technique with an application to stereo vision. pp. 674–679. 3
- [Low04] LOWE D.: Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110. 1, 3, 4, 5
- [MGW01] MALZBENDER T., GELB D., WOLTERS H.: Polynomial texture maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 519–528. 2
- [MS05] MIKOLAJCZYK K., SCHMID C.: A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 10 (Oct. 2005), 1615–1630. 3, 4
- [MWGA06] MALZBENDER T., WILBURN B., GELB D., AMBRISCO B.: Surface enhancement using real-time photometric stereo and reflectance transformation. In *Rendering Techniques* (2006), Akenine-Müller T., Heidrich W., (Eds.), Eurographics Association, pp. 245–250. 2
- [Pav08] PAVLIDIS T.: An evaluation of the scale invariant feature transform (sift). *An Evaluation of SIFT* (2008). 4
- [Pri57] PRIM R. C.: Shortest connection networks and some generalizations. *Bell Systems Technical Journal* (1957), 1389–1401. 6
- [SM07] SILVEIRA G., MALIS E.: Real-time visual tracking under arbitrary illumination changes. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (june 2007), pp. 1 –6. 3
- [Sze06] SZELISKI R.: Image Alignment and Stitching: A Tutorial. *Foundations and Trends® in Computer Graphics and Vision* 2, 1 (2006), 1–104. 3
- [TDS08] TANG C., DONG Y., SU X.: Automatic registration based on improved sift for medical microscopic sequence images. In *Proceedings of the 2008 Second International Symposium on Intelligent Information Technology Application - Volume 01* (Washington, DC, USA, 2008), IITA '08, IEEE Computer Society, pp. 580–583. 3
- [TM08] TUYTELAARS T., MIKOLAJCZYK K.: Local invariant feature detectors - a survey. *Foundations and Trends in Computer Graphics and Vision* (2008). 3
- [Tor98] TORR P. H. S.: Geometric motion segmentation and model selection. *Phil. Trans. Royal Society of London A* 356 (1998), 1321–1340. 5
- [War03] WARD G.: Fast, robust image registration for compositing high dynamic range photographs from handheld exposures. *JOURNAL OF GRAPHICS TOOLS* 8 (2003), 17–30. 3
- [WB07] WINDER S. A. J., BROWN M.: Learning local image descriptors. In *In CVPR* (2007), pp. 1–8. 3
- [WHB09] WINDER S. A. J., HUA G., BROWN M.: Picking the best daisy. In *CVPR* (2009), pp. 178–185. 3
- [WWX*10] WANG B., WU D., XU W., LU Q., LI F., LIU S., GAO G., LAI R.: A new image registration method for infrared images and visible images. In *Image and Signal Processing (CISP), 2010 3rd International Congress on* (oct. 2010), vol. 4, pp. 1745 –1749. 3
- [ZHG06] ZHAO F., HUANG Q., GAO W.: Image matching by normalized cross-correlation. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on* (may 2006), vol. 2, p. II. 3
- [Zit03] ZITOVA B.: Image registration methods: a survey. *Image and Vision Computing* 21, 11 (Oct. 2003), 977–1000. 2, 3