Learning Two-View Stereo Matching

Jianxiong Xiao, Jingni Chen, Dit-Yan Yeung, and Long Quan

Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong
{csxjx,jnchen,dyyeung,quan}@cse.ust.hk

Abstract. We propose a graph-based semi-supervised symmetric matching framework that performs dense matching between two uncalibrated wide-baseline images by exploiting the results of sparse matching as labeled data. Our method utilizes multiple sources of information including the underlying manifold structure, matching preference, shapes of the surfaces in the scene, and global epipolar geometric constraints for occlusion handling. It can give inherent sub-pixel accuracy and can be implemented in a parallel fashion on a graphics processing unit (GPU). Since the graphs are directly learned from the input images without relying on extra training data, its performance is very stable and hence the method is applicable under general settings. Our algorithm is robust against outliers in the initial sparse matching due to our consideration of all matching costs simultaneously, and the provision of iterative restarts to reject outliers from the previous estimate. Some challenging experiments have been conducted to evaluate the robustness of our method.

1 Introduction

Stereo matching between images is a fundamental problem in computer vision. In this paper, we focus on matching two wide-baseline images taken from the same static scene. Unlike many previous methods which require that the input images be either calibrated [1] or rectified [2], we consider here a more challenging scenario in which the input contains two images only without any camera information. As a consequence, our method can be used for more general applications, such as motion estimation from structure.

1.1 Related Work

Many stereo matching algorithms have been developed. Traditional stereo matching algorithms [2] were primarily designed for view pairs with a small baseline, and cannot be extended easily when the epipolar lines are not parallel. On the other hand, existing wide-baseline methods [3] depend heavily on the epipolar geometry which has to be provided, often through off-line calibration, while other methods can only recover very sparse matching [4,5].

Although the epipolar geometry could be estimated on-line, those approaches still fail frequently for wide-baseline image pairs since the sparse matching result is fragile and the estimated fundamental matrix often fits only to some parts of the image but not

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part III, LNCS 5304, pp. 15–27, 2008.

the entire image. Region growing based methods [6,7] can achieve denser matching, but may easily get trapped in local optima. Therefore its matching quality depends heavily on the result of the initial sparse matching. Also, for image pairs with quite different pixel scales, it is very difficult to achieve reasonable results due to discrete growing.

Recent research shows that learning techniques can improve the performance of matching by taking matched pairs as training data or by learning the probabilistic image prior [8] that encodes the smoothness constraint for natural images. However, for a test image pair, the information learned from other irrelevant images is very weak in the sense that it is unrelated to the test image pair. Thus the quality of the result greatly depends on the training data.

1.2 Our Approach

In this work, we explore the dense matching of uncalibrated wide-baseline images by utilizing all the local, regional and global information simultaneously in an optimization procedure. We propose a semi-supervised approach to the matching problem requiring only two input images taken from the same static scene. Since the method does not rely on any training data, it can handle images from any scene with stable performance.

We consider two data sets, $\mathcal{X}^1$ and $\mathcal{X}^2$, corresponding to the two input images with $n^1 = r^1 \times c^1$ pixels and $n^2 = r^2 \times c^2$ pixels, respectively. For $p = 1, 2$,

$$\mathbf{X}^p = \left(\mathrm{x}_1^p, \mathrm{x}_2^p, \ldots, \mathrm{x}_{(s^p-1)\times c^p+t^p}^p, \ldots, \mathrm{x}_{n^p}^p\right)^T, \tag{1}$$

where $\mathrm{x}_{(s^p-1)\times c^p+t^p}^p$ represents the pixel located at the coordinate position (s^p, t^p) in the p-th image space, $s^p \in \{1, \cdots, r^p\}$, and $t^p \in \{1, \cdots, c^p\}$. In this paper, we define $q = 3 - p$, meaning that $q = 1$ when $p = 2$ and $q = 2$ when $p = 1$, and let $i = (s^p - 1) \times c^p + t^p$. For each pixel x_i^p, we want to find a matching point located at coordinate position (s^q, t^q) in the q-th (continuous) image space, where $s^q, t^q \in \mathcal{R}$. Hence, we can use a label vector to represent the position offset from a point in the second image to the corresponding point in the first image: $\mathrm{y}_i^p = (v_i^p, h_i^p)^T = \left((s^1, t^1) - (s^2, t^2)\right)^T \in \mathcal{R}^2$. In this way, our label vector representation takes real numbers for both elements, thus supporting sub-pixel matching. Let $\mathbf{Y}^p = (\mathrm{y}_1^p, \cdots, \mathrm{y}_{n^p}^p)^T$ be the label matrix, and $\mathrm{O}^p = (o_1^p, \cdots, o_{n^p}^p)^T$ be the corresponding visibility vector: $o_i^p \in [0, 1]$ is close to 1 if the 3D point corresponding to the data point x_i^p is visible in the other image, and otherwise close to 0 such as a point in the occluded region. This notion of visibility may also be interpreted as matching confidence.

Obviously, nearby pixels are more likely to have similar label vectors. This smoothness constraint, relying on the position of the data points, can be naturally represented by a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ where the node set $\mathcal{V}$ represents the data points and the edge set $\mathcal{E}$ represents the affinities between them. In our setting, we have two graphs $\mathcal{G}^1 = \langle \mathcal{V}^1, \mathcal{E}^1 \rangle$ and $\mathcal{G}^2 = \langle \mathcal{V}^2, \mathcal{E}^2 \rangle$ for the two images where $\mathcal{V}^1 = \{\mathrm{x}_i^1\}$ and $\mathcal{V}^2 = \{\mathrm{x}_i^2\}$. Let $\mathcal{N}(\mathrm{x}_i^p)$ be the set of data points in the neighborhood of x_i^p. The affinities can be represented by two weight matrices $\mathbf{W}^1$ and $\mathbf{W}^2$: w_{ij}^p is non-zero iff x_i^p and x_j^p are neighbors in $\mathcal{E}^p$.

In recent years, matching techniques such as SIFT [4] are powerful enough to recover some sparsely matched pairs. Now, the problem here is, given such matched pairs as labeled data $\left\langle \mathbf{X}_l^1, \mathbf{Y}_l^1 \right\rangle$, $\left\langle \mathbf{X}_l^2, \mathbf{Y}_l^2 \right\rangle$ and the affinity matrices $\mathbf{W}^1$ and $\mathbf{W}^2$, we want to infer the label matrices for the remaining unlabeled data $\left\langle \mathbf{X}_u^1, \mathbf{Y}_u^1 \right\rangle, \left\langle \mathbf{X}_u^2, \mathbf{Y}_u^2 \right\rangle$. For the sake of clarity of presentation and without loss of generality, we assume that the indices of the data points are arranged in such a way that the labeled points come before the unlabeled ones, that is $\mathbf{X}^p = \left((\mathbf{X}_l^p)^T, (\mathbf{X}_u^p)^T \right)^T$. For computation, the index of the data point can be mapped by multiplying elementary matrices for row-switching transformations.

In what follows, we formulate in Sec. 2 the matching problem under a graph-based semi-supervised label propagation framework, and solve the optimization problem via an iterative cost minimization procedure in Sec. 3. To get reliable affinity matrices for propagation, in Sec. 4 we learn $\mathbf{W}^1$ and $\mathbf{W}^2$ directly from the input images which include color and depth information. The complete procedure of our algorithm is summarized in Alg. 1. More details are given in Sec. 5. Finally, extensive experimental results are presented in Sec. 6.

2 Semi-supervised Matching Framework

Semi-supervised learning on the graph representation tries to find a label matrix $\hat{\mathbf{Y}}^p$ that is consistent with both the initial incomplete label matrix and the geometry of the data manifold induced by the graph structure. Because the incomplete labels may be noisy, the estimated label matrix $\hat{\mathbf{Y}}_l^p$ for the labeled data is allowed to differ from the given label matrix $\mathbf{Y}_l^p$. Given an estimated $\hat{\mathbf{Y}}^p$, consistency with the initial labeling can be measured by

$$C_l^p \left(\hat{\mathbf{Y}}^p, \mathrm{O}^p \right) = \sum_{\mathrm{x}_i^p \in \mathcal{X}_l^p} o_i^p \left\| \hat{\mathrm{y}}_i^p - \mathrm{y}_i^p \right\|^2 . \tag{2}$$

On the other hand, consistency with the geometry of the data in the image space, which follows from the smooth manifold assumption, motivates a penalty term of the form

$$C_s^p \left(\hat{\mathbf{Y}}^p, \mathrm{O}^p \right) = \frac{1}{2} \sum_{\mathrm{x}_i^p, \mathrm{x}_j^p \in \mathcal{X}^p} w_{ij}^p \phi \left(o_i^p, o_j^p \right) \left\| \hat{\mathrm{y}}_i^p - \hat{\mathrm{y}}_j^p \right\|^2 , \tag{3}$$

where $\phi \left(o_i^p, o_j^p \right) = \frac{1}{2} \left((o_i^p)^2 + \left(o_j^p \right)^2 \right)$. When o_i^p and o_j^p are both close to 1, the function value is also close to 1. This means we penalize rapid changes in $\hat{\mathbf{Y}}^p$ between points that are close to each other (as given by the similarity matrix $\mathbf{W}^p$), and only enforce smoothness within visible regions, i.e., o^p is large.

2.1 Local Label Preference Cost

Intuitively, two points of a matched pair in the two images should have great similarity in terms of the features since they are two observations of the same 3D point. Here, we use a similarity cost function $\rho_i^p(\mathrm{y})$ to represent the similarity cost between the pixel x_i^p in one image and the corresponding point for the label vector y in the other image space (detailed in Subsec. 5.2). On the other hand, if o_i^p is close to 0, which means that x_i^p is almost invisible and the matching has low confidence, the similarity cost should not be charged. To avoid the situation when every point tends to have zero visibility to prevent cost charging, we introduce a penalty term τ_i^p. When o_i^p is close to 0, $(1 - o_i^p)\,\tau_i^p$ will increase. Also, τ_i^p should be different for different x_i^p. Textureless regions should be allowed to have lower matching confidence, that is, small confidence penalty, and vice versa. We use a very simple difference-based confidence measure defined as follows

$$\tau_i^p = \max_{\mathrm{x}_j^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} \left\{ \left\| \mathrm{x}_i^p - \mathrm{x}_j^p \right\| \right\}. \tag{4}$$

Now, we can define the local cost as

$$C_d^p\left(\hat{\mathbf{Y}}^p, \mathrm{O}^p\right) = \sum_{\mathrm{x}_i^p \in \mathcal{X}^p} \left(o_i^p \rho_i^p \left(\hat{\mathrm{y}}_i^p\right) + \left(1 - o_i^p\right) \tau_i^p \right). \tag{5}$$

2.2 Regional Surface Shape Cost

The shapes of the 3D objects' surfaces in the scene are very important cues for matching. An intuitive approach is to use some methods based on two-view geometry to reconstruct the 3D surfaces. While this is a reasonable choice, it is unstable since the structure deduced from two-view geometry is not robust especially when the baseline is not large enough. Instead, we adopt the piecewise planar patch assumption [7]. Since two data points with high affinity relation are more likely to have similar label vectors, we assume that the label vector of a data point can be linearly approximated by the label vectors of its neighbors, as in the manifold learning method called locally linear embedding (LLE) [9], that is

$$\mathrm{y}_i^p = \sum_{\mathrm{x}_j^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} w_{ij}^p \mathrm{y}_j^p. \tag{6}$$

Hence, the reconstruction cost can be defined as

$$C_r\left(\mathbf{Y}^p\right) = \sum_{\mathrm{x}_i^p \in \mathcal{X}^p} \left\| \mathrm{y}_i^p - \sum_{x_j^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} w_{ij}^p \mathrm{y}_j^p \right\|^2 = \left\| \left(\mathbf{I} - \mathbf{W}^p\right) \mathbf{Y}^p \right\|_F^2. \tag{7}$$

Let $\mathbf{A}^p = \mathbf{W}^p + \left(\mathbf{W}^p\right)^T - \mathbf{W}^p \left(\mathbf{W}^p\right)^T$ be the adjacency matrix, $\mathbf{D}^p$ the diagonal matrix containing the row sums of the adjacency matrix $\mathbf{A}^p$, and $\mathbf{L}^p = \mathbf{D}^p - \mathbf{A}^p$ the un-normalized graph Laplacian matrix. Because of the way $\mathbf{W}^p$ is defined in Sec. 4, we have $\mathbf{D}^p \approx \mathbf{I}$. Therefore,

$$C_r\left(\mathbf{Y}^p\right) \approx \mathrm{tr}\left(\left(\mathbf{Y}^p\right)^T \mathbf{L}^p \mathbf{Y}^p \right) = \sum_{\mathrm{x}_i^p, \mathrm{x}_j^p \in \mathcal{X}^p} a_{ij}^p \left\| \mathrm{y}_i^p - \mathrm{y}_j^p \right\|^2. \tag{8}$$

This approximation induces the representation of $a_{ij}^p \left\| \mathrm{y}_i^p - \mathrm{y}_j^p \right\|^2$, which makes the integration of the cost with visibility much easier.

Now, the data points from each image lie on one 2D manifold (image space). Except for the occluded parts which cannot be matched, the two 2D manifolds are from the same 2D manifold of the visible surface of the 3D scene. LLE [10] is used to align the two 2D manifolds (image spaces) to one 2D manifold (visible surface). The labeled data (known matched pairs) are accounted for by constraining the mapped coordinates of matched points to coincide. Let $\mathcal{X}_c^p = \mathcal{X}_l^p \cup \mathcal{X}_u^p \cup \mathcal{X}_u^q$, $\hat{\mathbf{Y}}_c^p = \left(\left(\hat{\mathbf{Y}}_l^p\right)^T, \left(\hat{\mathbf{Y}}_u^p\right)^T, \left(\hat{\mathbf{Y}}_u^q\right)^T\right)^T$ and $\mathrm{O}_c^p = \left((\mathrm{O}_l^p)^T, (\mathrm{O}_u^p)^T, (\mathrm{O}_u^q)^T\right)^T$. We partition $\mathbf{A}^p$ as

$$\mathbf{A}^p = \begin{bmatrix} \mathbf{A}_{ll}^p & \mathbf{A}_{lu}^p \\ \mathbf{A}_{ul}^p & \mathbf{A}_{uu}^p \end{bmatrix}. \tag{9}$$

Alignment of the manifold can be done by combining the Laplacian matrix as in [10], which is equivalent to combining the adjacency matrix:

$$\mathbf{A}_c^p = \begin{bmatrix} \mathbf{A}_{ll}^p + \mathbf{A}_{ll}^q & \mathbf{A}_{lu}^p & \mathbf{A}_{lu}^q \\ \mathbf{A}_{ul}^p & \mathbf{A}_{uu}^p & 0 \\ \mathbf{A}_{ul}^q & 0 & \mathbf{A}_{uu}^q \end{bmatrix}. \tag{10}$$

Imposing the cost only on the visible data points, the separate LLE cost of each graph is summed up:

$$C_r^p\left(\hat{\mathbf{Y}}^1, \hat{\mathbf{Y}}^2, \mathrm{O}^1, \mathrm{O}^2\right) = \sum_{\mathrm{x}_i^p, \mathrm{x}_j^p \in \mathcal{X}_c^p} (a_c^p)_{ij}\, \phi\big((o_c^p)_i, (o_c^p)_j\big) \left\|(\hat{\mathrm{y}}_c^p)_i - (\hat{\mathrm{y}}_c^p)_j\right\|^2, \tag{11}$$

where $(a_c^p)_{ij}$ is the element of $\mathbf{A}_c^p$.

2.3 Global Epipolar Geometric Cost

In the epipolar geometry [11], the fundamental matrix $\mathbf{F}_{12} = \mathbf{F}_{21}^T$ encapsulates the intrinsic projective geometry between two views in the way that, for x_i^p at position (s^p, t^p) in one image with matching point at position (s^q, t^q) in the other image, the matching point (s^q, t^q) should lie on the line $(a_i^p, b_i^p, c_i^p) = (s^p, t^p, 1)\, \mathbf{F}_{pq}^T$. This global constraint affects every matching pair in the two images. For x_i^p, we define $d_i^p(\mathrm{y})$ to be the squared Euclidean distance in the image space of the other image between the corresponding epipolar line (a_i^p, b_i^p, c_i^p) and the matching point (s^q, t^q):

$$d_i^p(\mathrm{y}) = \frac{\left(a_i^p s^q + b_i^p t^q + c_i^p\right)^2}{\left(a_i^p\right)^2 + \left(b_i^p\right)^2}, \tag{12}$$

where $\mathrm{y} = (v, h)^T = \left(\left(s^1 - s^2\right), \left(t^1 - t^2\right)\right)^T$. The global cost is now the sum of all squared distances:

$$C_g^p\left(\hat{\mathbf{Y}}^p, O^p\right) = \sum_{\mathrm{x}_i^p \in \mathcal{X}^p} o_i^p d_i^p\left(\hat{\mathrm{y}}_i^p\right). \tag{13}$$

2.4 Symmetric Visibility Consistency Cost

Assume that x_i^p in one image is matched with x_j^q in the other image. x_j^q should also have a label vector showing its matching with x_i^p in the original image. This symmetric visibility consistency constraint motivates the following visibility cost

$$C_v^p\left(\mathrm{O}^p,\hat{\mathrm{Y}}^q\right)=\beta\sum_{\mathrm{x}_i^p\in\mathcal{X}^p}\left(o_i^p-\gamma_i^p\left(\hat{\mathrm{Y}}^q\right)\right)^2+\frac{1}{2}\sum_{\mathrm{x}_i^p,\mathrm{x}_j^p\in\mathcal{X}^p}w_{ij}^p\left(o_i^p-o_j^p\right)^2, \qquad (14)$$

where $\gamma\left(\hat{\mathrm{Y}}^q\right)$ is a function defined on the p-th image space. For each x_i^p, its value via the γ function indicates whether or not there exist one or more data points that match a point near x_i^p from the other view according to $\hat{\mathrm{Y}}^q$. The value at pixel x_i^p is close to 0 if there is no point in the other view corresponding to a point near x_i^p, and otherwise close to 1. The parameter β controls the strength of the visibility constraint. The last term enforces the smoothness of the occlusion that encourages spatial coherence and is helpful to remove some isolated pixels or small holes of the occlusion.

The γ function can be computed as a voting procedure when $\hat{\mathrm{Y}}^q$ is available in the other view. For each point x_j^q at position (s^q,t^q) in $\mathcal{X}^q$ with label $\mathrm{y}_j^q=\left(v_j^q,h_j^q\right)^T=\left(\left(s^1,t^1\right)-\left(s^2,t^2\right)\right)^T$, equivalent to be matched with a point at position (s^p,t^p), we place a 2D Gaussian function $\psi\left(s,t\right)$ on the p-th image centered at the matched position $\mathrm{c}_j=\left(s^p,t^p\right)^T$. Now, we get a Gaussian mixture model $\sum_{\mathrm{x}_j^q}\psi_{\mathrm{c}_j}\left(s,t\right)$ in the voted image space. Truncating it, we get

$$\gamma^p\left(s,t\right)=\min\Big\{1,\sum_{\mathrm{x}_j^q\in\mathcal{X}^q}\psi_{\mathrm{c}_j}\left(s,t\right)\Big\}. \qquad (15)$$

Our matching framework combines all the costs described above. We now present our iterative optimization algorithm to minimize the costs.

3 Iterative MV Optimization

It is intractable to minimize the matching and visibility costs simultaneously. Therefore, our optimization procedure iterates between two steps: 1) the M-step estimates matching given visibility, and 2) the V-step estimates visibility given matching. Before each iteration, we estimate the fundamental matrix **F** by the normalized 8-point algorithm with RANSAC followed by the gold standard algorithm that uses the Levenberg-Marquardt algorithm to minimize the geometric distance [11]. Then, we use **F** to reject the outliers from the matching result of the previous iteration and obtain a set of inliers as the initial labeled data points. The iterations stop when the cost difference between two consecutive iterations is smaller than a threshold, which means that the current matching result is already quite stable. The whole iterative optimization procedure is summarized in Alg. 1.

Algorithm 1. The complete procedure

```
1. Compute the depth and occlusion boundary images and feature vectors (Sec. 5).
2. Compute sparse matching by SIFT and the confidence penalty $\tau$, then interpolate the results
   from sparse matching with depth information to obtain an initial solution (Subsec. 5.1).
3. Learn the affinity matrices $\mathbf{W}^1$ and $\mathbf{W}^2$ (Sec. 4).
4. while (cost change between two iterations $\geq$ threshold):
   (a) Estimate the fundamental matrix $\mathbf{F}$, and reject outliers to get a subset as labeled data
       (Sec. 3),
   (b) Compute the parameters for the similarity cost function $\rho$ and epipolar cost function $d$
       (Subsec. 5.2 and 2.3),
   (c) Estimate matching given visibility (Subsec. 3.1),
   (d) Compute the $\gamma$ map (Subsec. 2.4),
   (e) Estimate visibility given matching (Subsec. 3.2).
```

3.1 M-Step: Estimation of Matching Given Visibility

Actually, the visibility term C_v imposes two kinds of constraints on the matching $\hat{\mathbf{Y}}$ given the visibility O: First, for each pixel x_i^p in the p-th image, it should not match the invisible (occluded) points in the other image. Second, for each visible pixel in the q-th image, at least one pixel in the p-th image should match its nearby points. The first restriction is a local constraint that is easy to satisfy. However, the second constraint is a global one on the matching of all points, which is implicitly enforced in the matching process. Therefore, in this step, we approximate the visibility term by considering only the local constraint [12], which means that some possible values for a label vector, corresponding to the occluded region, have higher costs than the other possible values. This variation of the cost can be incorporated into the similarity function $\rho_i^p(\mathrm{y})$ in C_d. Let $\mathbf{Y} = \left(\left(\mathbf{Y}^1\right)^T, \left(\mathbf{Y}^2\right)^T\right)^T$. Summing up all the costs and considering the two images together, our cost function is

$$C_M\left(\hat{\mathbf{Y}}\right) = \sum_{p=1,2}\left(\lambda_l C_l^p + \lambda_s C_s^p + \lambda_d C_d^p + \lambda_r C_r^p + \lambda_g C_g^p\right) + \epsilon\left\|\hat{\mathbf{Y}}\right\|^2, \tag{16}$$

where $\epsilon\left\|\hat{\mathbf{Y}}\right\|^2$ is a small regularization term to avoid reaching degenerate situations. Fixing O^1 and O^2, cost minimization is done by setting the derivative with respect to $\hat{\mathbf{Y}}$ to zero since the second derivative is a positive definite matrix.

3.2 V-Step: Estimation of Visibility Given Matching

After achieving a matching, we can recompute the γ map (Subsec. 2.4). Let $\mathrm{O} = \left(\left(\mathrm{O}^1\right)^T, \left(\mathrm{O}^2\right)^T\right)^T$. Then, summing up all the costs and considering the two images together, our cost function is

$$C_V(\mathrm{O}) = \sum_{p=1,2}\left(\lambda_l C_l^p + \lambda_s C_s^p + \lambda_d C_d^p + \lambda_r C_r^p + \lambda_g C_g^p + \lambda_v C_v^p\right) + \epsilon\left\|\mathrm{O}\right\|^2, \tag{17}$$

where $\epsilon \|\mathrm{O}\|^2$ is a small regularization term. Now, for fixed $\hat{\mathbf{Y}}^1$ and $\hat{\mathbf{Y}}^2$, cost minimization is done by setting the derivative with respect to O to zero since the second derivative is a positive definite matrix.

Since $\mathbf{W}^p$ is very sparse, the coefficient matrix of the system of linear equations is also very sparse in the above two steps. We use a Gauss-Seidel solver or a conjugate gradient method on GPU [13], which can solve in parallel a large sparse system of linear equations very efficiently. We can derive that by the way $\mathbf{W}^p$ is defined in Sec. 4 and the cost functions defined in Eq. 16 and Eq. 17, the coefficient matrix is strictly diagonally dominant and positive definite. Hence, both Gauss-Seidel and conjugate gradient converge to the solution of the linear system with theoretical guarantee.

4 Learning the Symmetric Affinity Matrix

We have presented our framework which finds a solution by solving an optimization problem. Traditionally, for $\mathbf{W}^1$ and $\mathbf{W}^2$, we can directly define the pairwise affinity between two data points by normalizing their distance. However, as pointed out by [14], there exists no reliable approach for model selection if only very few labeled points are available, since it is very difficult to determine the optimal normalization parameters. Thus we prefer using a more reliable and stable way to learn the affinity matrices.

Similar to the 3D visible surface manifold of Eq. 6 in Sec. 2.2, we make the smooth manifold and linear reconstruction assumptions for the manifold in the image space. We also assume that the label space and image space share the same local linear reconstruction weights. Then we can obtain the linear reconstruction weight matrix $\mathbf{W}^p$ by minimizing the energy function $E_{\mathbf{W}^p} = \sum_{\mathrm{x}_i^p \in \mathcal{X}^p} E_{\mathrm{x}_i^p}$, where

$$E_{\mathrm{x}_i^p} = \left\| \mathrm{x}_i^p - \sum_{\mathrm{x}_j^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} w_{ij}^p \mathrm{x}_j^p \right\|^2. \tag{18}$$

This objective function is similar to the one used in LLE [9], in which the low-dimensional coordinates are assumed to share the same linear reconstruction weights with the high-dimensional coordinates. The difference here is that we assume the sharing relation to be between the label vectors and the features [15]. Hence, the way we construct the whole graph is to first shear the whole graph into a series of overlapped linear patches and then paste them together. To avoid the undesirable contribution of negative weights, we further enforce the following constraint

$$\sum_{\mathrm{x}_j^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} w_{ij}^p = 1, \; w_{ij}^p \geq 0. \tag{19}$$

From Eq. 18, $E_{\mathrm{x}_i^p} = \sum_{\mathrm{x}_j^p, \mathrm{x}_k^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} w_{ij}^p \mathrm{G}_{jk}^i w_{ik}^p$, where $\mathrm{G}_{jk}^i = \left(\mathrm{x}_i^p - \mathrm{x}_j^p\right)^T \left(\mathrm{x}_i^p - \mathrm{x}_k^p\right)$. Obviously, the more similar is x_i^p to x_j^p, the larger will w_{ij}^p be. Also, w_{ij}^p and w_{ji}^p should be the same since they both correspond to the affinity relation between x_i^p and x_j^p. However, the above constraints do not either enforce or optimize to have this characteristic, and the hard constraint $w_{ij}^p = w_{ji}^p$ may result in violation of Eq. 19. Hence, we add a soft penalty term $\sum_{ij}\left(w_{ij}^p - w_{ji}^p\right)^2$ to the objective function. Thus the reconstruction weights of each data point can be obtained by solving the following quadratic programming (QP) problem

$$\min_{\mathrm{W}^p} \sum_{\mathrm{x}_i^p \in \mathcal{X}^p} \sum_{\mathrm{x}_j^p, \mathrm{x}_k^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} w_{ij}^p \mathrm{G}_{jk}^i w_{ik}^p + \kappa \sum_{ij}\left(w_{ij}^p - w_{ji}^p\right)^2 \tag{20}$$

$$\text{s.t. } \forall \mathrm{x}_i^p \in \mathcal{X}^p, \sum_{\mathrm{x}_j^p \in \mathcal{N}\left(\mathrm{x}_i^p\right)} w_{ij}^p = 1,\ w_{ij}^p \geq 0.$$

After all the reconstruction weights are computed, two sparse matrices can be constructed by $\mathbf{W}^p = \left[w_{ij}^p\right]$ while letting $w_{ii}^p = 0$ for all x_i^p. In our experiment, $\mathbf{W}^p$ is almost symmetric and we further update it by $\mathbf{W}^p \leftarrow \frac{1}{2}\left((\mathbf{W}^p)^T + \mathbf{W}^p\right)$. Since the soft constraint has made $\mathbf{W}^p$ similar to $(\mathbf{W}^p)^T$, this update just changes $\mathbf{W}^p$ slightly, and will not lead to unreasonable artifacts. To achieve speedup, we can first partition the graph into several connected components by the depth information and super-pixel over-segmentation on the RGB image, and break down the large QP problem into several smaller QP problems with one QP for each connected component, then solve them one by one.

5 More Details

The feature vectors are defined as RGB color. For each image, we recover the occlusion boundaries and depth ordering in the scene. The method in [16] is used to learn to identify and label occlusion boundaries using the traditional edge and region cues together with 3D surface and depth cues. Then, from just a single image, we obtain a depth estimation and the occlusion boundaries of free-standing structures in the scene. We append the depth value to the feature vector.

5.1 Label Initialization by Depth

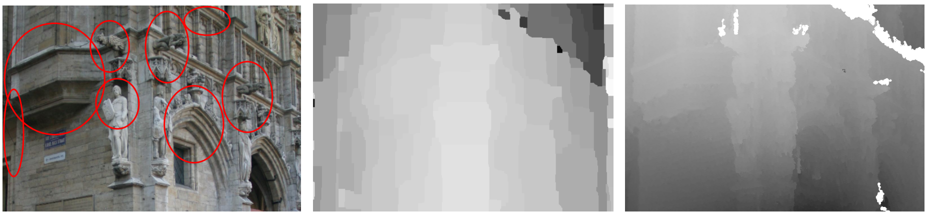
We use SIFT [4] and a nearest neighbor classifier to obtain an initial matching. For robustness, we perform one-to-one cross consistency checking, which matches points of the first image to the second image, and inversely matches points of the second image to the first image. Only the best matched pairs consistent in both directions are retained. To avoid errors on the occlusion boundary due to the similar color of background and foreground, we filter the sparse matching results and reject all pairs that are too close to the occlusion boundaries. Taking the remaining as seed points, with the depth information, region growing is used to achieve an initial dense matching [7]. Then, the remaining unmatched part is interpolated. Assuming the nearby pixels in the same partition lie on a planar surface, we estimate the homography transformation between two corresponding regions in the two images. With the estimated homography, the unknown regions are labeled and the occlusion regions are also estimated.

5.2 Computing the Similarity Cost Function

As mentioned in Sec. 2.1, the continuous-valued similarity cost function $\rho_i^p(\mathrm{y})$ represents the difference between point x_i^p and the matching point, characterizing how suitable it is for x_i^p to have label $\mathrm{y} = (v, h)^T$. Since our algorithm works with some labeled data in a semi-supervised manner by the consistent cost C_l, the local cost C_d just plays a secondary role. Hence, unlike the traditional unsupervised matching [12], our framework does not heavily rely on the similarity function $\rho_i^p(\mathrm{y})$. Therefore, for efficient computation, we just sample some values for some integer combination of h and v to compute $\rho_i^p(\mathrm{y}) = \exp(-\frac{\|\mathrm{x}_i^p - \mathrm{x}_j^q\|^2}{2\sigma^2})$. We normalize the largest sampled value to 1, and then fit $\rho_i^p(\mathrm{y})$ with a continuous and differentiable quadratic function $\rho_i^p(\mathrm{y}) = \frac{(v-v_o)^2+(h-h_o)^2}{2\sigma^2}$, where (v_o, h_o) and σ are the center and spread of the parabola for x_i^p.

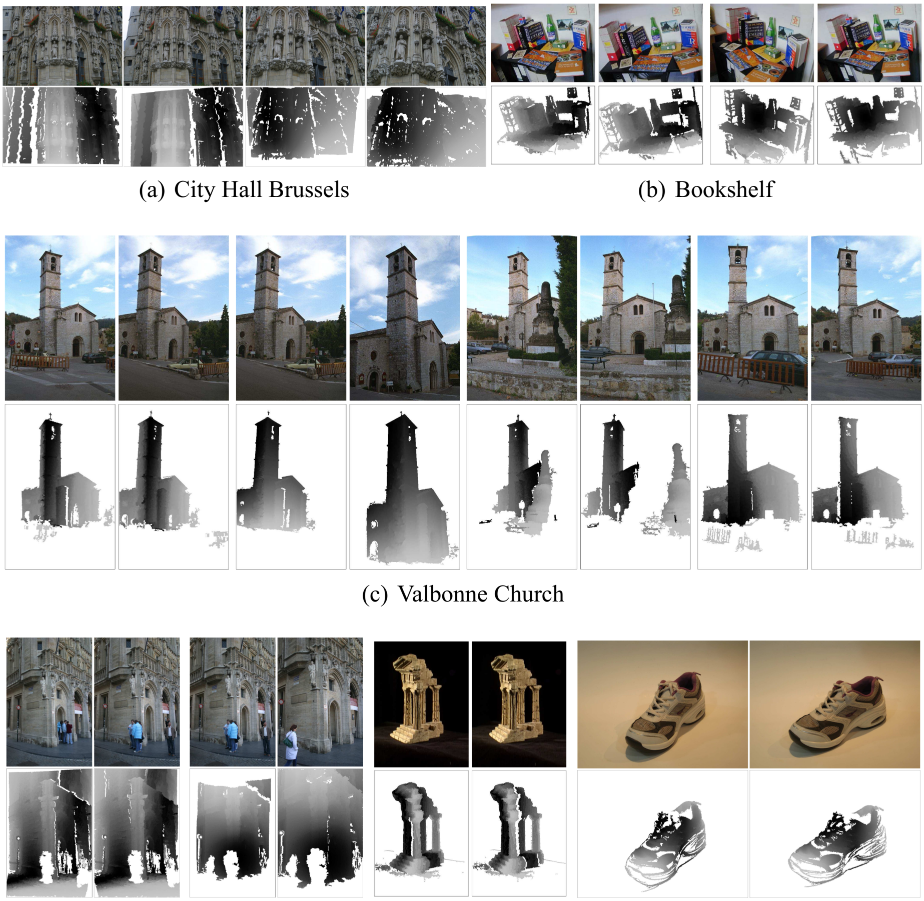
6 Experiments

In all our experiments performed on a desktop PC with Intel Core 2 Duo E6400 CPU and NVIDIA GeForce 8800 GTX GPU, the number of iterations is always less than 9 before stopping and the computation time is less than 41 seconds for each image pair, excluding the time spent on estimating the depth for a single image by [16]. We set the parameters to favor C_l and C_g in the M-step and C_v in the V-step. Since there is no ground truth in searching for good parameter values, we tune the parameters manually and then fix them for all experiments. To solve the QP problem for $\mathbf{W}^p$, we first compute a "warm start" without involving the positive constraints using the method in [9], and then run the active set algorithm on this "warm start", which converges rapidly in just a few iterations. We demonstrate our algorithm on various data set in Fig. 2, most of which have very complex shape with similar color that makes the matching problem very challenging. Compared with [17], our method can produce more detail, as shown in Fig. 1. In the figures of the matching results, the intensity value is set to be the norm of the label vector, that is $\|\mathrm{y}\|$, and only visible matching with $o \geqslant 0.5$ is shown.



(a) Input image [3] (b) Result from [17] (c) Result of our method

Fig. 1. Comparison with [17]. Attention should be paid to the fine details outlined by red circles. Also, our method can correctly detect the occluded region and does not lead to block artifacts that graph cut methods typically give. Subfig. (b) is extracted from Fig. 7 of [17].


(a) City Hall Brussels

(b) Bookshelf

(c) Valbonne Church

(d) City Hall Leuven

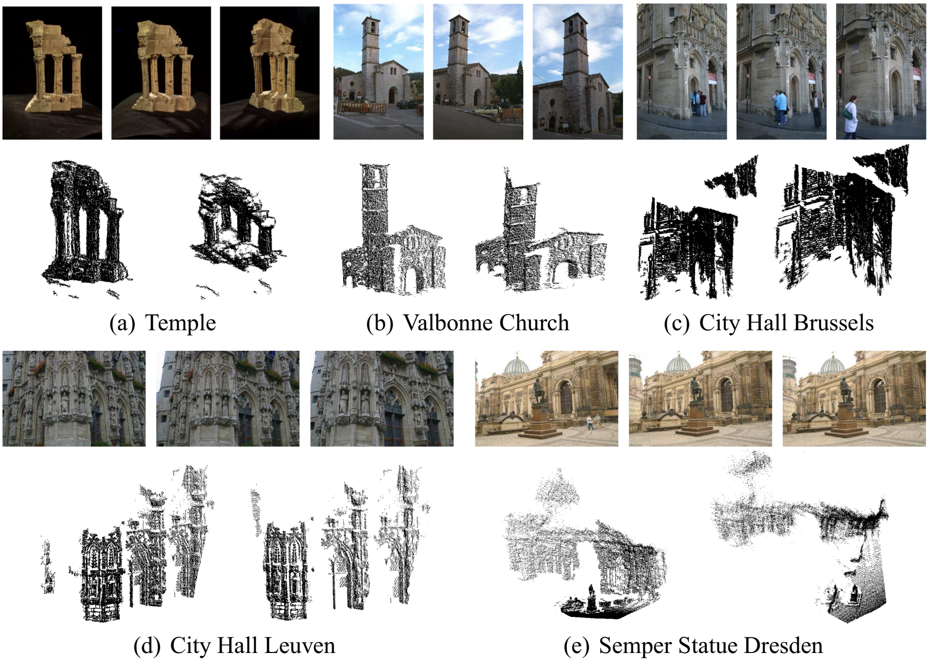
(e) Temple

(f) Shoe

Fig. 2. Example output on various datasets. In each subfigure, the first row shows the input images and the second row shows the corresponding outputs by our method.

6.1 Application to 3-View Reconstruction

In our target application, we have no information about the camera. To produce a 3D reconstruction result, we use three images to recover the motion information. Five examples are shown in Fig. 3. The proposed method is used to compute the point correspondence between the first and second images, as well as the second and third images. Taking the second image as the bridge, we can obtain the feature tracks of three views. As in [18], these feature tracks across three views are used to obtain projective reconstruction by [19], and are metric upgraded inside a RANSAC framework, followed by bundle adjustment [11]. Note that the feature tracks with too large reprojection errors are considered as outliers and are not shown in the 3D reconstruction result in Fig. 3.



(a) Temple

(b) Valbonne Church

(c) City Hall Brussels

(d) City Hall Leuven

(e) Semper Statue Dresden

Fig. 3. 3D reconstruction from three views. In each subfigure, the first row contains the three input images and the second row contains two different views of the 3D reconstruction result. Points are shown without texture color for easy visualization of the reconstruction quality.

7 Conclusion

In this work, we propose a graph-based semi-supervised symmetric matching framework to perform dense matching between two uncalibrated images. Possible future extensions include more systematic study of the parameters and extension to multi-view stereo. Moreover, we will also pursue a full GPU implementation of our algorithm since we suspect that the current running time is mostly spent on data communication between the CPU and the GPU.

Acknowledgements. This work was supported by research grant N-HKUST602/05 from the Research Grants Council (RGC) of Hong Kong and the National Natural Science Foundation of China (NSFC), and research grant 619006 and 619107 from the RGC of Hong Kong. We would like to thank the anonymous reviewers for constructive comments to help improve this work. The bookshelf data set is from J. Matas; City Hall Brussels, City Hall Leuven and Semper Statue Dresden data set are from C. Strecha; Valbonne Church data set is from the Oxford Visual Geometry Group, while the temple data set is from [1] and the shoe data set is from [20].

References

1. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, vol. 1, pp. 519–528 (2006)
2. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47(1-3), 7–42 (2002)
3. Strecha, C., Fransens, R., Gool, L.: Wide-baseline stereo from multiple views: a probabilistic account. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, vol. 1, pp. 552–559 (2004)
4. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
5. Toshev, A., Shi, J., Daniilidis, K.: Image matching via saliency region correspondences. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, pp. 1–8 (2007)
6. Lhuillier, M., Quan, L.: Match propagation for image-based modeling and rendering. IEEE Transaction on Pattern Analysis and Machine Intelligence 24(8), 1140–1146 (2002)
7. Kannala, J., Brandt, S.S.: Quasi-dense wide baseline matching using match propagation. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, pp. 1–8 (2007)
8. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, vol. 2, pp. 860–867 (2005)
9. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
10. Ham, J., Lee, D., Saul, L.K.: Learning high dimensional correspondences from low dimensional manifolds. In: Proceedings of International Conference on Machine Learning (2003)
11. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge (2004)
12. Sun, J., Li, Y., Kang, S.B., Shum, H.Y.: Symmetric stereo matching for occlusion handling. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, vol. 2, pp. 399–406 (2005)
13. Krüger, J., Westermann, R.: Linear algebra operators for GPU implementation of numerical algorithms. ACM Transactions on Graphics, 908–916 (2003)
14. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. Neural Information Processing Systems 16, 321–328 (2004)
15. Wang, F., Wang, J., Zhang, C., Shen, H.: Semi-supervised classification using linear neighborhood propagation. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, pp. 160–167 (2006)
16. Hoiem, D., Stein, A., Efros, A., Hebert, M.: Recovering occlusion boundaries from a single image. In: Proceedings of IEEE International Conference on Computer Vision (2007)
17. Tola, E., Lepetit, V., Fua, P.: A fast local descriptor for dense matching. In: Proceedings of IEEE Conference Computer Vision and Pattern Recognition, pp. 1–8 (2008)
18. Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE Transaction on Pattern Analysis and Machine Intelligence 27(3), 418–433 (2005)
19. Quan, L.: Invariant of six points and projective reconstruction from three uncalibrated images. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(1), 34–46 (1995)
20. Xiao, J., Chen, J., Yeung, D.Y., Quan, L.: Structuring visual words in 3D for arbitrary-view object localization. In: Proceedings of European Conference on Computer Vision (2008)