Depth Based Image Registration via 3D Geometric Segmentation

Bing Han, Christopher Paulson, and Dapeng Wu
Department of Electrical and Computer Engineering
University of Florida Gainesville, FL 32611
Correspondence author: Prof. Dapeng Wu, wu@ece.ufl.edu, http://www.wu.ece.ufl.edu

Abstract

Image registration is a fundamental task in computer vision and it significantly contributes to high-level computer vision and benefits numerous practical applications. Although many image registration techniques have been proposed in the past, there is still a need for further research because many issues such as the parallax problem remain to be solved. The traditional image registration algorithms suffer from the parallax problem due to their underling assumption that the scene can be regarded approximately planar which is not satisfied when large depth variations exist in the images with high-rise objects. To address the parallax problem, we present a new strategy for 2D image registration by leveraging the depth information from 3D image reconstruction. The novel idea is to recover the depth in the image region with high-rise objects to build accurate transform function for image registration. We use a geometric segmentation algorithm to partition 3D point cloud to multiple geometric structures and at the same time, estimate the parameters of each geometric structure. Experimental results show that our proposed method is able to mitigate the parallax problem and achieve better performance than the existing image registration scheme.

Index Terms

3D reconstruction, image registration, depth estimation, parallax problem, geometric segmentation

I. Introduction

Image registration is a fundamental task in image processing and computer vision, which matches two or more images taken at different times and different viewpoints, by geometrically aligning reference and sensed images. There has been a broad range of techniques developed over the years in literature. A comprehensive survey of image registration methods was published in 1992 by Brown [1], including many classic methods still in use. Due to the rapid development of image acquisition devices, more image registration techniques emerged afterwards and were covered in another survey published in 2003 [2].

Different applications due to distinct image acquisition require different image registration techniques. In general, manners of the image acquisition can be divided into three main groups:

- *Different viewpoints (multiview analysis)*. Images of the same scene are acquired from different viewpoints. The aim is to gain a larger 2D view or a 3D representation of the scanned scene.
- *Different times*. Images of the same scene are acquired at different times, often on regular basis, and possibly under different conditions. The aim is to find and evaluate changes in the scene which appeared between the consecutive image acquisitions.
- *Different sensors*. Images of the same scene are acquired by different sensors. The aim is to integrate the information obtained from different source streams to gain more complex and detailed scene representation.

Due to the diversity of images to be registered and various types of degradations, it is impossible to design a universal method applicable to all registration tasks. Every method should take into account not only the assumed type of geometric deformation between the images but also the radiometric deformations and noise corruption, required registration accuracy and application-dependent data characteristics. Nevertheless, the majority of the registration methods consists of the following four steps: feature detection, feature matching, transform model estimation, image resampling and transformation.

A widely used feature detection method is corner detection. Kitchen and Rosenfeld [3] proposed to exploit the second-order partial derivatives of the image function for corner detection. Dreschler and Nagel [4] searched for the local extrema of the Gaussian curvature. However, corner detectors based on the second-order derivatives of the image function are sensitive to noise. Thus Forstner [5] developed a more robust, although time consuming, corner detector, which is based on the first-order derivatives only. The reputable Harris detector [6] also uses first-order derivatives for corner detection.

Feature matching includes area-based matching and feature-based matching. Classical area-based method is cross-correlation (CC) [7] exploit for matching image intensities directly. For feature-based matching, Goshtasby [8] described the registration based on the graph matching algorithm. Clustering technique, presented by Stockman et al. [9], tries to match points connected by abstract edges or line segments.

After the feature correspondence has been established the mapping function is constructed. The mapping function should transform the sensed image to overlay it over the reference image [10][11][12][13].

Finally interpolation methods such as nearest neighbor function, bilinear, and bicubic functions are applied to the output of the registered images.

The prevailing image registration methods, such as Davis and Keck's algorithm [14], [15], assume all the feature points are coplanar and build a homography transform matrix to do registration. The advantage is that they have low computational cost and can handle planar scenes conveniently; however, with the assumption that the scenes are approximately planar, they are inappropriate in the registration applications when the images have large depth variation due to the high-rise objects, known as the parallax problem. Parallax is an apparent displacement of difference of orientation of an object viewed along two different lines of sight, and is measured by the angle or semi-angle of inclination between those two lines. Nearby objects have a larger parallax than further objects when observed from different positions. Therefore, as the viewpoint moves side to side, the objects in the distance appear to move slower than the objects close to camera.

In this paper, we propose a depth based image registration algorithm by leveraging the depth information. Our method can mitigate the parallax problem caused by high-rise scenes in the images by building accurate transform function between corresponding feature points in multiple images. Given an image sequence, we first select a number of feature points and then match the features in all images. Then we estimate the depth of each feature point from feature correspondences. With the depth information, we can project the image in 3D instead of using a homography transform. Further more, fast and robust image registration algorithm can be achieved by combining the traditional image registration algorithms and depth based image registration method proposed in this paper. The idea is that we first compute the 3D structure of a sparse feature points set and then divide the 3D point cloud (obtained by sparse 3D reconstruction) into multiple approximately planar regions. For each region, we can perform a depth based image registration. Accordingly, our proposed image registration is able to mitigate the parallax problem due to the use of depth information.

The remainder of this paper is organized as follows. Section II describes the overall structure of our image registration system. Section III reviews the 3D reconstruction algorithm we used in our new method. In Section IV, we describe how to use 3D depth information for 2D image registration and propose a non-linear deterministic annealing algorithm for geometric segmentation. Section V presents the experimental results and we compare our algorithm with Davis and Keck's algorithm. We conclude this paper in Section VI.

II. System Overview

Due to the diversity of images to be registered and various types of degradations, it is impossible to design a universal method applicable to all registration tasks. Every method should take into account not only the assumed type of geometric deformation between the images but also the radiometric deformations and noise corruption, required registration accuracy and application-dependent data characteristics. Nevertheless, the majority of the registration methods consists of the following four steps: feature detection,

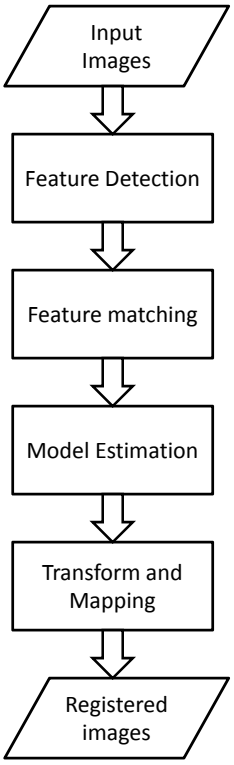

Fig. 1. Flowchart for 2D image registration.

feature matching, transform model estimation, image resampling and transformation. Although they may differ in some specific part, most image registration approaches are generally based on the same procedure. The procedure is given in Fig. 1.

Instead of matching the whole image pixel to pixel directly, the technique of feature detection and matching is widely used in image registration algorithms in order to find the relation between the two images to be registered. A feature is also known as point of interest, including edges, corners, blobs and ridges. Feature matching is to establish the feature correspondence between the features extracted from two images. With the feature correspondence, a mapping function can be constructed and transform the sensed image to the reference image.

In our new image registration system, we use a 3D model instead of 2D motion model used in existing works. Our system is slightly different from the previous one. As shown in Fig. 2, in our new system, we first apply 3D reconstruction to the input images and recover the 3D geometric structure of the scene in the images. The 3D model is more accurate compared to the 2D motion models estimated in the previous works. Then we segment the 3D point cloud into multiple regions, each of which is modeled by a plane. With the segmentation, we can estimate the 3D depth for every pixel in each region and recover the dense structure of the 3D scene. The 3D dense structure allows us to obtain pixel-by-pixel correspondence between two consecutive images. We describe the 3D reconstruction algorithm in Section III. In Section IV, we present the geometric segmentation and depth based mapping in 3D, and also propose a expanded deterministic annealing algorithm for geometric segmentation.

III. 3D RECONSTRUCTION FROM VIDEO SEQUENCES

Here, we simply review the classic eight point 3D reconstruction algorithm [16][17]. When developing a stereo vision algorithm for registration, the requirements for accuracy vary from those of standard stereo algorithms used for 3D reconstruction. For example, a multi-pixel disparity error in an area of low texture, such as a white wall, will result in significantly less intensity error in the registered image than the same

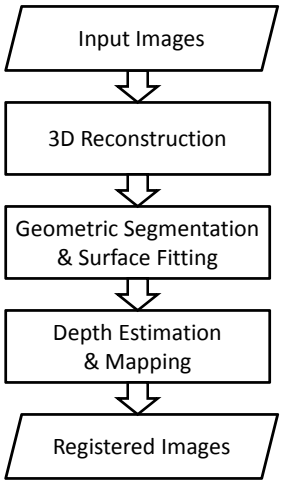

Fig. 2. Our new image registration system.

disparity error in a highly textured area. In particular, edges and straight lines in the scene need to be rendered correctly.

The 3D reconstruction algorithm is implemented using the following steps. First, geometric features are detected automatically in each individual images. Secondly, feature correspondence is established across all the images. Then the camera motion is retrieved and the camera is calibrated. Finally the Euclidean structure of the scene is recovered.

A. Feature selection

The first step in 3D reconstruction is to select candidate features in all images for tracking across different views. Ma et al. [17] use point feature in reconstruction which is measured by Harris' criterion,

$$C(\mathbf{x}) = \det(G) + k \times \text{trace}^2(G) \tag{1}$$

where $\mathbf{x} = [x, y]^T$ is a candidate feature, $C(\mathbf{x})$ is the quality of the feature, k is a pre-chosen constant parameter and G is a 2×2 matrix that depends on $\mathbf{x}$, given by

$$G = \begin{bmatrix} \sum_{W(\mathbf{x})} I_x^2 & \sum_{W(\mathbf{x})} I_x I_y \\ \sum_{W(\mathbf{x})} I_x I_y & \sum_{W(\mathbf{x})} I_y^2 \end{bmatrix} \tag{2}$$

where $W(\mathbf{x})$ is a rectangular window centered at $\mathbf{x}$ and I_x and I_y are the gradients along the x and y directions which can be obtained by convolving the image I with the derivatives of a pair of Gaussian filters. The size of the window can be decided by the designer, for example 7×7. If $C(\mathbf{x})$ exceeds a certain threshold, then the point $\mathbf{x}$ is selected as a candidate point feature.

B. Feature matching

Once the candidate point features are selected, the next step is to match them across all the images. In this subsection, we use a simple feature tracking algorithm based on a translational model.

We use the sum of squared differences (SSD) [18] as the measurement of the similarity of two point features. Then the correspondence problem becomes looking for the displacement $\mathbf{d}$ that satisfies the following optimization problem:

$$\min_{\mathbf{d}} \doteq \sum_{\mathbf{x} \in W(\mathbf{x})} [I_2(\mathbf{x} + \mathbf{d}) - I_1(\mathbf{x})]^2 \tag{3}$$

where $\mathbf{d}$ is the displacement of a point feature of coordinates $\mathbf{x}$ between two consecutive frames I_1 and I_2. Lucas and Kanade also give the closed form solution of 3

$$\mathbf{d} = -G^{-1}\mathbf{b} \tag{4}$$

where

$$\mathbf{b} \doteq \begin{bmatrix} \sum_{W(\mathbf{x})} I_x I_t \\ \sum_{W(\mathbf{x})} I_y I_t \end{bmatrix} \tag{5}$$

G is the same matrix we used to compute the quality of the candidate point feature in Eq. (1), and $I_t \doteq I_2 - I_1$.

C. Estimation of camera motion parameters

In this subsection, we recover the projective structure of the scene from the established feature correspondence. We will follow the notation used in Ma et al.'s book [17].

The method for estimating camera motion parameters [17] is based on a perspective projection model with a pinhole camera. Suppose we have a generic point $p \in \mathbb{R}^3$ with coordinates $\mathbf{X}_p = [X, Y, Z, 1]^T$ relative to a world coordinate frame. Given two image frames of one scene which is related by a motion $g = (R, T)$, the two image projection point $\mathbf{x}_1$ and $\mathbf{x}_2$ are related as follows:

$$\lambda_1 \check{\mathbf{x}}_1 = \Pi_1 \mathbf{X}_p, \quad \lambda_2 \check{\mathbf{x}}_2 = \Pi_2 \mathbf{X}_p \tag{6}$$

where $\check{\mathbf{x}} = [x, y, 1]^T$ is measured in pixels; λ_1 and λ_2 are the depths of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively; $\Pi_1 = [J, 0]$ and $\Pi_2 = [JR, JT]$ are the camera projection matrices; and J is the camera calibration matrix. In order to estimate λ_1, λ_2, Π_1 and Π_2, we need to introduce the epipolar constraint. From Eq. (6), we have

$$\check{\mathbf{x}}_2^T J^{-T} \hat{T} R J^{-1} \check{\mathbf{x}}_1 = 0 \tag{7}$$

The fundamental matrix is defined as:

$$F_m \doteq J^{-T} \hat{T} R J^{-1} \tag{8}$$

With the above model, we could estimate the fundamental matrix F_m via Algorithm 1, which is given in Ref. [17]. Then we could decompose the fundamental matrix to recover the projection matrices Π_1 and Π_2 and the 3D structure. We only give the solution here by canonical decomposition:

$$\lambda_1 \check{\mathbf{x}}_1 = \mathbf{X}_p, \lambda_2 \check{\mathbf{x}}_2 = (\widehat{T'})^T F_m \mathbf{X}_p + T' \tag{9}$$

D. Depth estimation

The Euclidean structure $\mathbf{X}_e$ is related to the projective reconstruction $\mathbf{X}_p$ by a linear transform $H \in \mathbb{R}^{4\times4}$,

$$\Pi_{ip} \sim \Pi_{ie} H^{-1}, \mathbf{X}_p \sim H \mathbf{X}_e, i = 1, 2, ..., m \tag{10}$$

where $\sim$ means equality up to a scale factor and

$$H = \begin{bmatrix} J & 0 \\ -\nu^T J & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{11}$$

With the assumption that J is constant, we could estimate the unknowns J and ν with a gradient decent optimization algorithm [17]. In order to obtain a unique solution, we also assume that the scene is generic and the camera motion is rich enough.

Fig. 3 shows the first frame and the 88th frame of the video sequence 'house'. In our experiment, we will register all the frames in the video sequence to the first frame. Fig. 4 show the selected feature points on the first frame which are used for camera pose estimation. Fig. 5 show the estimated 3D positions of the feature points and the estimated camera pose of the 1st and 88th frame.

Algorithm 1 Eight-point algorithm

Input: a set of initial feature correspondences expressed in pixel coordinates $(\mathbf{x'}_1^j, \mathbf{x'}_2^j)$ for $j = 1, 2, ..., n$:

1)Construct the matrix $\chi \in \mathbb{R}^{n\times 9}$ from the transformed correspondences $\tilde{\mathbf{x}}_1^j \doteq [\tilde{x}_1^j, \tilde{y}_1^j, 1]^T$ and $\tilde{\mathbf{x}}_2^j \doteq [\tilde{x}_2^j, \tilde{y}_2^j, 1]^T$, where the jth row of χ is given by $[\tilde{x}_1^j\tilde{x}_2^j, \tilde{x}_1^j\tilde{y}_2^j, \tilde{x}_1^j, \tilde{y}_1^j\tilde{x}_2^j, \tilde{y}_1^j\tilde{y}_2^j, \tilde{y}_1^j, \tilde{x}_2^j, \tilde{y}_2^j, 1]^T \in \mathbb{R}^9$.

2)Find the vector $F^s \in \mathbb{R}^9$ of unit length such that $||\chi F^s||$ is minimized as follows:

a)Compute the singular value decomposition (SVD) of $\chi = U\Sigma V^T$ and define F^s to be the ninth column of V.

b)Unstack the nine elements of F^s into a square 3×3 matrix $\tilde{F}$.

3)Imposing the rank-2 constraint:

a)Compute the SVD of the matrix F recovered from data to be $\tilde{F} = U_F diag\{\sigma_1, \sigma_2, \sigma_3\}V_F^T$.

b)Impose the rank-2 constraint by letting $\sigma_3 = 0$ and reset the fundamental matrix to be $F = U_F diag\{\sigma_1, \sigma_2, 0\}V_F^T$.

Output: F.

(a) The 1st frame in the 'house' video sequence

(b) The 88th frame in the 'house' video sequence

Fig. 3. Frames used for image registration

IV. Image registration with depth information

Once we obtain the 3D structure of the feature points, the motion, and calibration of the camera, we can start to register the rest of the pixels in the images with the estimated depth information. The traditional image registration algorithms, such as the algorithm proposed by Davis and Keck [14], [15], try to register the two images by computing the homography matrix H between corresponding feature points. The limit of this algorithm is that they assume all the points in the physical world are coplanar or approximately coplanar, which is not true with high-rise scenes. In order to mitigate this problem, we propose a novel algorithm which first segment the image geometrically and then perform the registration to each region with depth estimation.

A. Geometric Segmentation and Surface Fitting

Not all points in an image are suitable for matching or tracking. The feature points that we have selected are only a small portion of a whole image. Therefore, the first reconstruction is a sparse 3D reconstruction. The sparse structure is not suitable for human visualization. For this reason, a dense matching is necessary to establish a 3D geometric view. It is known that the most popular solution for dense matching is based on the epi-polar constraint. This approach uses geometric constraints to restrict correspondence search from 2D to 1D range. The main disadvantage of this approach is that the dense

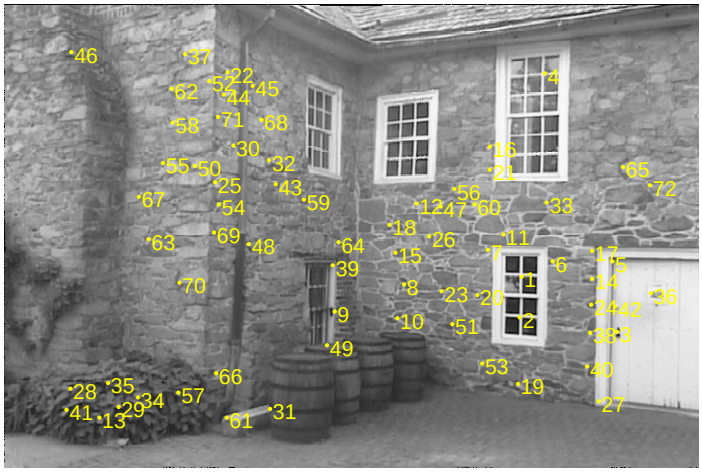

Fig. 4. Feature points selected on the 1st frame.

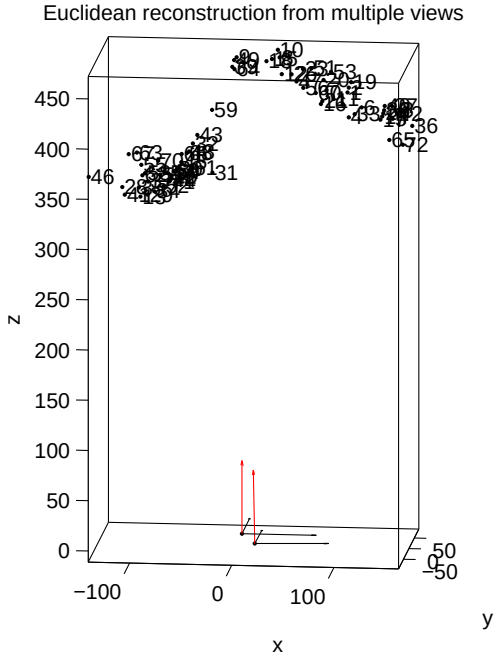

Fig. 5. Estimated camera pose of the 1st and 88th frame, and estimated 3D positions of feature points.

depth map is not smooth because of outliers. Lhuillier and Quan [19] proposed a dense matching method called quasi-dense approach. However, the non-smoothness problem still exists.

In this section, we propose an expanded deterministic annealing approach for space partitioning and surface fitting in 3D Euclidean space. Under the assumption that the 3D scene under study consists of a few geometric structures, we design a non-linear function to map the data point from geometrical space to surface model space and apply deterministic annealing in the feature space to partition the feature space into multiple regions. For each region, we use a linear plane model to fit the 3D points in the region.

We call our method as expanded deterministic annealing method. Our method has three merits: 1) the ability to avoid many poor local optima; 2) the ability to minimize the cost function even if its gradients vanish almost everywhere; 3) the ability to achieve non-linear separation of 3D points. However, there is no closed form solution to the expanded deterministic annealing problem; therefore we use a gradient descent algorithm to solve this problem. Next, we present the problem of geometric segmentation and surface fitting.

Given a set of 3D points $\{\mathbf{y}_i\}$, we would like to find multiple geometric surfaces that best fit the 3D point cloud $\{\mathbf{y}_i\}$. The problem can be formulated as below

$$\min_{\{\theta_k\}_{k=1}^K} \sum_{k=1}^{K} \sum_{\mathbf{y}_i \in \mathcal{C}_k} d(\mathbf{y}_i, g_{\theta_k}) \tag{12}$$

where K is the number of surfaces (i.e., planes here); $\{\theta_k\}_{k=1}^K$ denotes the set $\{\theta_1, \theta_2, \cdots, \theta_K\}$; $\mathcal{C}_k$ denotes the cluster of 3D points that belong to the k-th plane; $\mathbf{y}_i = [x_i, y_i, z_i]^T$ is the i-th point; $\theta_k = [a_k, b_k, c_k]^T$ is the parameter vector of the k-th plane model, where $\frac{1}{a_k}$, $\frac{1}{b_k}$, and $\frac{1}{c_k}$ are intercepts of the plane on x-axis, y-axis, and z-axis, respectively; and $d_{i,k}^2$ is the squared distance (fitting error) between $\mathbf{y}_i$ and plane model $g_{\theta_k}(\mathbf{y}) = \mathbf{y}^T\theta_k = 1$, which is defined as

$$d_{i,k}^2 = d^2(\mathbf{y}_i, g_{\theta_k}) = (\mathbf{y}_i^T\theta_k - 1)^2 \tag{13}$$

This is a joint problem of model selection and parameter estimation, i.e., we need to determine how many surfaces (or the number of clusters of 3D points) and estimate the parameters of the parametric model of each surface. This problem is particularly challenging because the more surfaces, the smaller fitting error but the higher probability of over-fitting; the fewer surfaces, the larger fitting error.

The problem in (12) can be solved by deterministic annealing (DA) [20]. The DA approach to clustering has demonstrated substantial performance improvement over traditional supervised and unsupervised learning algorithms. DA mimics the annealing process. DA works as below. First, it minimizes the cost function subject to a constraint on the degree of randomness of the solution. The constraint on Shannon entropy, is gradually shrunk as the temperature reduces, and the constraint eventually vanishes as the temperature goes to zero; hence the solution of DA converges to the solution of minimizing the original cost function. Similar to the simulated annealing [21], the cooling schedule allows DA to avoid many poor local optima. The DA approach has been adopted in a variety of research fields, such as graph-theoretic optimization and computer vision. Rao et al. [22] apply DA to solving a piecewise regression problem.

In this paper, we propose a new approach, called expanded deterministic annealing (EDA), to solve the geometric segmentation and surface fitting problem. Specifically, we first use a non-linear function to map the input 3D points to a high dimensional feature space using the local geometric structure of each 3D point. Then we apply deterministic annealing to the points in the feature space for clustering (i.e., geometric segmentation) and surface fitting. Different from Ref. [22], our EDA approach leverages local geometric structure for clustering. Next, we present our EDA approach.

The input data is a set of 3D points, $\mathbf{y}_i = [x_i, y_i, z_i]^T$ $(i = 1, \cdots, N)$. Under the assumption that L nearest 3D points of a given point $\mathbf{y}_i$ are on the same local plane, we use the least squares method to estimate this local plane model parameters, denoted by $\mathbf{L}_i = [a_i, b_i, c_i]^T$. Let

$$\mathbf{f}_i = \begin{bmatrix} \mathbf{y}_i \\ \mathbf{L}_i \end{bmatrix}$$

So we expand a 3D point $\mathbf{y}_i$ to 6D point $\mathbf{f}_i$; then we apply DA to this expanded 6D space to solve the geometric segmentation and surface fitting problem. Define

$$P_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$P_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

In our EDA algorithm, we use the following (new) distortion function

$$D(\mathbf{f}_i, g_{\theta_k}) = \beta \times D_1(P_1\mathbf{f}_i, g_{\theta_k}) + D_2(P_2\mathbf{f}_i, g_{\theta_k}), \tag{14}$$

where $D_1(\mathbf{y}_i, g_{\theta_k}) = d_{i,k}^2$, which is defined in (13); $D_2(P_2\mathbf{f}_i, g_{\theta_k})$ is defined by

$$D_2(P_2\mathbf{f}_i, g_{\theta_k}) = 1 - (P_2\mathbf{f}_i^T \cdot \theta_k)^2; \tag{15}$$

and β is a positive real number, which balances the two types of distortions D_1 and D_2. Note that D_1 quantifies the fitting error between a given 3D point $\mathbf{y}_i$ and the global plane, which $\mathbf{y}_i$ belongs to; D_2 quantifies the difference between the local plane model of $\mathbf{y}_i$ and the global plane model of $\mathbf{y}_i$; and $P_2\mathbf{f}_i^T \cdot \theta_k$ is a cosine similarity between the two plane models. One novelty of our EDA algorithm is the introduction of D_2, which can be regarded as locally averaged distortion and help mitigate the effect of outliers, i.e., an outlier 3D point only contributes distortion D_1 weighted by β. E.g., if we choose $\beta = 0.1$, then the contribution from D_1 is reduced by a factor of 10.

Denote the number of clusters by K. We apply DA to partition $\{\mathbf{y}_i\}$ into K clusters and estimate the parameters of planes, each of which corresponds to one cluster. Since K is not a given parameter, our EDA algorithm will search for the optimal value of K, as shown in Algorithm 2. For a given value K, our EDA algorithm solves the following problem.

$$\min_{\{\theta_k\}_{k=1}^K} F = D - TH \tag{16}$$

where $\theta_k = [a_k, b_k, c_k]^T$ $(k = 1, \cdots, K)$ is the surface model parameter to be estimated; D is defined in (14); and H is the entropy constraint. We define D and H as follows:

$$D = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} P(\mathbf{y}_i \in g_{\theta_k}) \times D(\mathbf{f}_i, g_{\theta_k}), \tag{17}$$

$$H = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} P(\mathbf{y}_i \in g_{\theta_k}) \times \log P(\mathbf{y}_i \in g_{\theta_k}), \tag{18}$$

where

$$P(\mathbf{y}_i \in g_{\theta_k}) = \frac{\exp(-\frac{D(\mathbf{f}_i, g_{\theta_k})}{T})}{\sum_{j=1}^{K}\exp(-\frac{D(\mathbf{f}_i, g_{\theta_j})}{T})} \tag{19}$$

We use a gradient descent algorithm to solve the problem (16) as shown in Algorithm 2. We explain Step 2 of Algorithm 2 as below. Since a plane model is specified by equation $g_\theta(\mathbf{y}) = \mathbf{y}^T\theta = 1$, given 3D points $\{\mathbf{y}_i\}_{i=1}^N$, for $K = 1$, we can estimate the parameter vector θ of the plane that fits all the 3D points, by solving the following problem

$$\hat{\theta} = \arg\min_{\theta} \sum_{i=1}^{N} (\mathbf{y}_i^T\theta - 1)^2. \tag{20}$$

Hence, $\hat{\theta}$ can be obtained by the least squares solution as below

$$\hat{\theta} = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\vec{1}_N, \tag{21}$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N]^T$ is a matrix of dimension $N \times 3$, and $\vec{1}_N = [1, 1, \cdots, 1]^T$ is a vector of dimension N. In Step 4b, $\Theta = [\theta_1^T, \theta_2^T, \cdots, \theta_K^T]^T$; $\nabla_\Theta F$ denotes the gradient of F with respect to Θ.

Algorithm 2 EDA based joint geometrical segmentation and surface fitting

1) **Input**: $\mathbf{y}_1$, $\mathbf{y}_2$, ..., $\mathbf{y}_N$;
   K_{max}: maximum number of clusters
   T_{init}: starting temperature
   T_{min}: minimum temperature
   α: cooling rate (must be < 1)
   I_{max}: maximum iteration number
   ε: threshold for F
   ϵ: threshold for plane merging
   σ^2: variance of Gaussian perturbation
2) **Initialization**
   Compute θ_1 via (21); $T = T_{init}; K = 2; \theta_2 = \theta_1; P(\mathbf{y}_i \in g_{\theta_1}) = P(\mathbf{y}_i \in g_{\theta_2}) = \frac{1}{2}, \forall i$.
3) **Perturb**
   Generate Gaussian vector δ_k of zero mean and covariance matrix $\sigma^2 I$; $\theta_k \leftarrow \theta_k + \delta_k$ $(k = 1, 2)$;
   $F_{old} = D - TH$;
   j=0;
4) **Loop until convergence**
4a) For each i and each k, compute $P(\mathbf{y}_i \in g_{\theta_k})$ via (19);
4b) Update the surface models
      $\Theta \leftarrow \Theta - \gamma \nabla_\Theta F$ (γ is obtained by Armijo rule);
   $F = D - TH$;
   $j = j + 1$;
   If ($j < I_{max}$ and $(F_{old} - F)/F_{old} > \varepsilon$)
      $F_{old} = F$; Goto Step 4a;
5) **Model size determination**
   { if ($||\theta_k - \theta_m||_2 < \epsilon$), then replace θ_k and θ_m by $(\theta_k + \theta_m)/2$ } $\forall k, m$;
   K = number of planes after merging;
6) **Cooling**
   $T = \alpha T$;
   If ($T < T_{min}$)
      perform Step 4a, 4b and Step 5 for $T = 0$
      Goto Step 9
7) **Duplication**
   Replace each plane by two planes at the same location; $K = 2K$;
8) **Goto Step 3**
9) **Output**: $\{\theta_k\}_{k=1}^K$.

B. 3D Point Correspondence

Here, we only consider two images. Suppose for the first image, we have the 3D point set $\{\mathbf{X}_e^j\}_{j=1}^n$, which could be divided into N_c clusters, denoted by $\mathcal{X}_1$, $\mathcal{X}_2$, $\cdots$, $\mathcal{X}_{N_c}$. For each cluster, we assume there are at least three non-collinear points, which usually can be satisfied. Then we model each cluster by a plane. We use $\mathcal{X}_1$ as an example. We can use the following plane model to fit the points in $\mathcal{X}_1$,

$$\mathbf{X} \cdot \mu = 1 \tag{22}$$

where $\mu = [a, b, c, 0]^T$ is the plane parameter vector and can be estimated by the least squares method, given 3D points in $\mathcal{X}_1$.

For an arbitrary image point $\mathbf{x}^i = [x^i, y^i]^T$, which is an image of a 3D point in $\mathcal{X}_1$, we could estimate its depth λ^i by solving the following equation,

$$\lambda^i \check{\mathbf{x}}^i = H_1^{-1} \Pi_1 \mathbf{X}_e^i \tag{23}$$

where $\check{\mathbf{x}}^i = [x^i, y^i, 1]^T$, H_1^{-1} and Π_1 are estimated by Eq. (9) and Eq. (11), respectively. In Eq. (23), only λ^i is unknown and with the constraint on $\mathbf{X}_e^i$ via Eq. (22), we can obtain the value of λ^i.

Then, with $\Pi_1 = [I, 0]$, we could have $\mathbf{X}_p^i = [\lambda_1^i x^i, \lambda_1^i y^i, \lambda_1^i, 1]^T$. from Eq. (6), we can get the relation between two image projection point $\mathbf{x}_1^i$ and $\mathbf{x}_2^i$ as follows:

$$\check{\mathbf{x}}_2^i = \Pi_2 \mathbf{X}_p^i. \tag{24}$$

where $\check{\mathbf{x}}_2^i = [\lambda_2^i x_2^i, \lambda_2^i y_2^i, \lambda_2^i]^T$. We could then get the position of the corresponding point $\mathbf{x}_2^i = [x_2^i, y_2^i]^T$ in the second image.

V. Experimental Results

In this section, we conduct experiments to demonstrate that our proposed approach is able to segment 3D point cloud into appropriate geometric structures and register the images more accurately. The rest of the section is organized as below. Subsection V-A shows the estimation accuracy of our proposed EDA algorithm for synthetic data with the knowledge of ground truth. Subsection V-B investigates the accuracy of our proposed image registration algorithm for real-world data.

A. EDA on Synthetic Data

In this subsection, we show the estimation accuracy of EDA algorithm for synthetic data with the knowledge of ground truth; the synthetic data here does not contain noise. We also compare EDA to both PI algorithm and API algorithm. PI algorithm is a projection based iterative geometric segmentation algorithm (PI for short) based on the same principle of the Lloyd algorithm (a.k.a., K-means) [23]. API algorithm is an adaptive projection based iterative algorithm (API for short) based on the same principle of ISODATA, which generalizes K-means by allowing K to be unspecified.

To generate synthetic data, we first determine the number of planes, denoted by K; then specify the analytic form of each of the K planes and the area of each plane; for each plane, we uniformly generate 100 3D points on the plane area. The same data set is applied to the three algorithms. In the experiment, we run each algorithm 1000 times; in different run, a different set of 3D points are (randomly) generated; then we compute the average performance of each algorithm in terms of average squared approximation error and correct identification rate.

Table I shows average squared approximation error for PI, API, and EDA algorithms, where the squared approximation error is quantified by the Euclidean norm of the estimation error for the plane model parameters. We test four different number of planes, i.e., $K = 3, 4, 5, 6$. As observed from Table I, the approximation error of EDA is negligible comparing to that of PI and API, which demonstrates that the EDA algorithm significantly outperforms both PI and API algorithms in terms of estimation accuracy for the plane models. This is because EDA is able to separate the 3D point cloud in a non-linear manner, and can avoid many poor local optima.

Table II shows correct identification rate for PI, API, and EDA algorithms. The correct identification rate is quantified by the percentage of 3D points whose plane memberships are correctly identified. Note that there are K planes and we have the ground truth of which plane a 3D point belongs to. We observe that the correct identification rates of EDA and API are much higher than that of the PI algorithm. The reason why the API algorithm outperforms the PI algorithm is that the API algorithm is not sensitive to initialization while the PI algorithm is very sensitive to initialization. Again, EDA performs the best among the three algorithms in terms of correct identification rate. This is again because EDA is able to separate the 3D point cloud in a non-linear manner, and can avoid many poor local optima. We also observe that if the number of plane is too large, like $K > 10$, the performance get worse. The reason is

TABLE I
AVERAGE SQUARED APPROXIMATION ERROR.

K	PI	API	EDA
3	3.77×10^{-1}	3.00×10^{-9}	1.17×10^{-12}
4	4.01×10^{-1}	9.81×10^{-8}	2.21×10^{-12}
5	2.43×10^{-1}	2.86×10^{-9}	3.06×10^{-12}
6	2.94×10^{-1}	8.801×10^{-9}	3.00×10^{-12}
10	5.38×10^{-1}	3.32×10^{-8}	7.81×10^{-10}
20	1.01	6.73×10^{-7}	5.28×10^{-8}

TABLE II
CORRECT IDENTIFICATION RATE.

K	PI	API	EDA
3	83%	96%	99%
4	79%	93%	99%
5	82%	94%	97%
6	78%	97%	98%
10	73%	92%	95%
20	66%	88%	91%

that as the total number of planes increases, it becomes harder to distinguish two similar planes and it becomes possible that many of the points may be classified into wrong categories. Therefore, we consider it as one of the limitations of the EDA algorithm.

B. EDA on Real World Data

In this section, we show the registration accuracy of our proposed method for real world data. In our experiment, given a set of images to be registered, we regard the first image's local coordinate system as world coordinate system. So the first image can be viewed as a reference image. Then the rest of the images are registered to the reference image. We also applied the algorithm proposed by Davis and Keck [14] to register the input images for comparison purpose.

The experiment is conducted on several video sequences. Here, we use the 'oldhouse' test sequence as an example. The data includes a sequence of 88 images captured from one camera. We first select 72 feature points in the first image and then find the corresponding feature points in the rest of the images. The depth estimates of these points are calculated by the algorithm introduced in Section III.

Fig. 3 is the 1st frame and the 88th frame in the test image sequence. Fig. 6 is the registration result using our algorithm and Fig. 7 is the output of the algorithm proposed by Davis and Keck [14]. Fig. 8 shows the difference image between the registered image and the first image using our algorithm and Fig. 9 shows the difference image from the algorithm of Davis and Keck [14]. We can see that our result can mitigate the parallax problem since the roof and wall corners are registered correctly; on the contrary, the registered image by the algorithm of Davis and Keck [14] has a lot of artifacts caused by the parallax problem. We also show some registration results using our algorithm in Fig. 10 and Fig. 11.

In order to further compare our algorithm to the algorithm proposed by Davis and Keck, we compute the root of mean squared errors (RMSE) of the registration results from both algorithms. Fig. 12 shows that the registration error of our algorithm is less than 50% than that of the algorithm proposed by Davis and Keck.

We also show some experimental results with test sequence 'break dancers'. Fig. 13 shows the original image captured by the 4th camera. Fig. 14 is the registration result using our algorithm and Fig. 15 is the output of the algorithm proposed by Davis and Keck.

The result shows that our image registration algorithm can mitigate the parallax problem because most of the scene is registered without vibration, as opposed to registration results under the algorithm of Davis

Fig. 6. Result of our algorithm, in which the 88th frame is registered to the 1st frame.

Fig. 7. The test result under the algorithm of Davis and Keck, in which the 88th frame is registered to the 1st frame.

and Keck in which the high-rise scene in the sensed images significantly moved after registration to the reference images. The reason is that the algorithm of Davis and Keck assumes all the points in the images are coplanar. While this assumption is satisfied when the distance between the camera and the interested scene is so large that the small depth variation can be neglected, it fails in the case of high-rise scene. Therefore, depth information should be used to accomplish the registration for this specific high-rise region of the images.

Finally, we would like to point out that the algorithm proposed by Davis and Keck [14] assumes a planar registration. Their scheme was designed for use with high-altitude aerial imagery where planar transformations are fairly good approximations. Furthermore, their scheme uses RANSAC to remove poor matching points during the computation. This can help to deal with some depth discontinuities that may be present in the high-altitude aerial images. In our experiments, the test images contain salient 3D scenes; these images are out of the domain for the algorithm of Davis and Keck. This is the reason why the algorithm of Davis and Keck does not perform well.

Fig. 8. The difference image between the registered 88th image (using our algorithm) and the 1st image.

Fig. 9. The difference image between the registered 88th image (using the algorithm of Davis and Keck) and the 1st image.

VI. Conclusion

In this paper, we propose a new 2D image registration method by leveraging depth information. While traditional image registration algorithms fail to register high-rise scene accurately because the points cannot be assumed to be simply planar, our image registration algorithm can mitigate the parallax problem.

Our future works include:

- Develop a robust 3D model based on the state-of-the-art depth estimate algorithm [24][25] given a video sequence. The reliability of the depth estimates is crucial to depth-based registration algorithm;

Fig. 10. The 37th frame in the 'house' video sequence.

Fig. 11. Result of our algorithm, in which the 37th frame is registered to the 1st frame.

therefore, the highly robust 3D reconstruction technique is required to implement our algorithm. Up to now, most recent depth recovery algorithms reported in the literature claim to recover consistent depth from some challenging video sequences [24][25]. We can apply or modify this state-of-the-art depth map recovery method to develop depth-based image registration algorithm.

- Combine depth-based image registration method with traditional algorithms. In other words, we can use depth information to register high-rise region while applying traditional registration algorithm for other planar region of the image. The purpose is to tradeoff between the accuracy of the registration

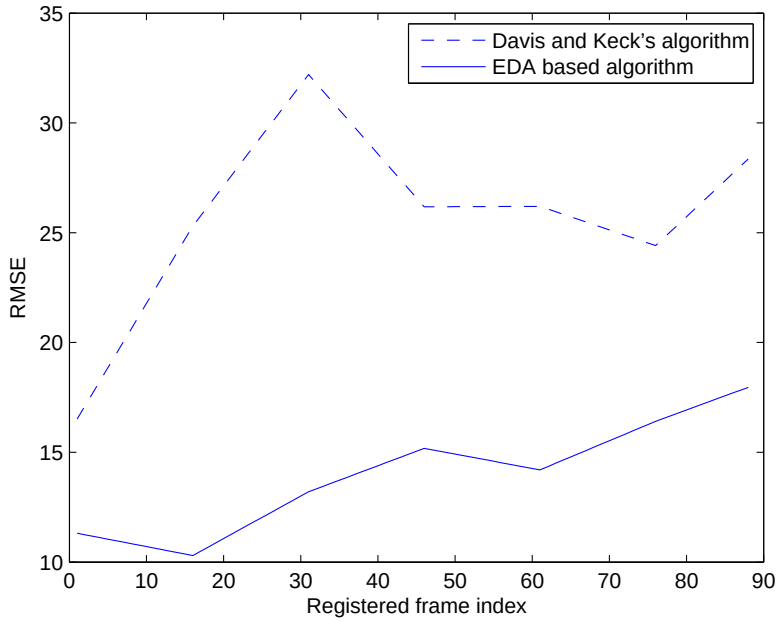

Fig. 12. Result of our algorithm, compared to that under the algorithm of Davis and Keck, in which all the 88 frames are registered to the 1st frame.

Fig. 13. Original image of sequence 'break dancers'.

and the high computational cost introduced by 3D reconstruction. The combined algorithm thus can enjoy both the high efficiency of the traditional algorithm and the high robustness of the depth-based registration method.

- We would use our depth-based image registration algorithm in practical applications to further verify the performance of our algorithm compared to the traditional ones.

Fig. 14. Result of our algorithm, in which the image captured by the 4th camera is registered to the image from the 3rd camera.

DISCLAIMERS

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

ACKNOWLEDGEMENT

This material is based on research sponsored by AFRL under agreement number FA8650-06-1-1027. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The authors would like to thank Dr. James W. Davis and Mark Keck from the Ohio State University for permission to use their registration algorithm and code. Our thanks also go to Olga Mendoza for constructive suggestions for improving the paper.

REFERENCES

[1] L. Brown, "A Survey of Image Registration Techniques," *ACm Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.
[2] B. Zitova and J. Flusser, "Image Registration Methods: A Survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, 2003.
[3] L. Kitchen and A. Rosenfeld, "Gray-Level Corner Detection," 1980.
[4] L. Dreschler and H. Nagel, *Volumetric model and 3D-trajectory of a moving car derived from monocular TV-frame sequences of a street scene*. Univ., Fachbereich Informatik, 1981.
[5] W. Forstner and E. Gulch, "A fast operator for detection and precise location of distinct points, corners and centres of circular features," in *Proceedings of Intercommission Conference on Fast Processing of Photogrammetric Data*, pp. 281–305, 1987.
[6] J. Noble, "Finding Corners," *Image and Vision Computing*, vol. 6, no. 2, pp. 121–128, 1988.
[7] W. Pratt *et al.*, "Digital Image Processing," *New York*, pp. 429–32, 1978.
[8] A. Goshtasby and G. Stockman, "Point Pattern Matching Using Convex Hull Edges," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 5, pp. 631–636, 1985.
[9] G. Stockman, S. Kopstein, and S. Benett, "Matching Images to Models for Registration and Object Detection via Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 229–241, 1982.
[10] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Manhattan-world stereo," in *IEEE Conference on Computer Vision and Pattern Recognition 2009*, pp. 1422–1429, IEEE, 2009.
[11] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Reconstructing building interiors from images," in *IEEE 12th International Conference on Computer Vision 2009*, pp. 80–87, IEEE, 2009.

Fig. 15. Result of Davis and Keck's algorithm, in which the image captured by the 4th camera is registered to the image from the 3rd camera.

[12] D. Gallup, J. Frahm, and M. Pollefeys, "Piecewise planar and non-planar stereo for urban scene reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition 2010*, pp. 1418–1425, IEEE, 2010.

[13] S. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *Proc. ICCV*, pp. 1881–1888, 2009.

[14] J. Davis and M. Keck, "OSU Registration Algorithm," *Internal Report, Ohio State University, USA*.

[15] O. Mendoza, G. Arnold, and P. Stiller, "Further exploration of the object-image metric with image registration in mind," in *Proceedings of the SPIE, Symposium on Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, vol. 6974, pp. 05–12, April 2008.

[16] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms, MA Fischler and O. Firschein, eds*, pp. 61–62, 1987.

[17] Y. Ma, S. Soatto, J. Kosecka, Y. Ma, S. Soatta, J. Kosecka, and S. Sastry, *An invitation to 3-D vision*. Springer, 2004.

[18] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, vol. 3, p. 3, 1981.

[19] M. Lhuillier and L. Quan, "A Quasi-Dense Approach to Surface Reconstruction from Uncalibrated Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 418–433, 2005.

[20] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression, and Related Optimization Problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.

[21] S. Kirkpatrick, "Optimization by Simulated Annealing: Quantitative Studies," *Journal of Statistical Physics*, vol. 34, no. 5, pp. 975–986, 1984.

[22] A. Rao, D. Miller, K. Rose, and A. Gersho, "A Deterministic Annealing Approach for Parsimonious Design of Piecewise Regression Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 2, pp. 159–173, 1999.

[23] S. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[24] G. Zhang, J. Jia, T. Wong, and H. Bao, "Recovering consistent video depth maps via bundle optimization," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2008.

[25] G. Zhang, J. Jia, T. Wong, and H. Bao, "Consistent Depth Maps Recovery from a Video Sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 974–988, 2009.