An Automatic Surface Fitting Method for 3D Reconstruction from 2D Video Sequence

Bing Han, Chris 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

3D reconstruction is one of the most fundamental problem in computer vision and computer graphics. 3D video reconstruction is the process of recovering the 3D geometric structure and surface from a 2D video sequence which is one of the most challenging research topics in 3D reconstruction. The challenge in 3D video reconstruction is how to align 2D image sequence pixel by pixel. Traditional stereo reconstruction methods and volumetric reconstruction methods suffer from the blank wall problem and the estimated dense depth map is not smooth for surface modeling. In this paper, We present a novel surface fitting approach for 3D dense reconstruction. We propose a non-linear deterministic annealing algorithm to decompose the 3D sparse structure to separate regions, and estimate the dense depth map by plane surface fitting. The experimental results show that the new approach can segment the 3D space geometrically and generate smoother dense depth map.

Index Terms

Geometric segmentation, surface fitting, dense matching, 3D reconstruction

I. Introduction

3D reconstruction is one of the most challenging and fundamental problem in the area of computer vision. During the recent years, a lot of approaches were developed for modeling and rendering the virtual scene from 2D videos and image sequences [1][2][3][4]. Currently, most of the systems and applications in 3D reconstruction are used for visual inspection and architecture modeling. However, there is more demand for 3D entertainment, for example, 3D movies. The change of demand results in an attention for smooth visual quality of the reconstructed scene. In this case, visual quality of the virtual scene becomes the dominant factor. While the foremost goal in previous approaches is the accuracy of the position of each point in 3D geometry.

In the last two decades, tremendous progress has been made on self-calibration and 3D surface modeling [5][6][7][8]. Most of the methods use 2D video sequences or 2D images as input and try to retrieve the depth information of the scene captured by the input video sequence. The estimated depth information helps to reconstruct the full 3D view of the scene. The existing techniques are able to well calculate the camera motion and compute a sparse depth map from the original image sequence [9][10][11][1][12]. However, fully reconstruction of a 3D scene requires the depth information of much more image pixels which requires the alignment of almost all pixels of the input images. This problem is known as dense matching problem[13][14][15].

A traditional solution to the dense matching problem is called epi-line searching. Epi-line search method uses the geometric constraints to degrade a 2D searching to a 1D range searching [16][17][18]. Although the search is constraint to 1D which seems easier to search, the blank wall problem, which is not solved in 2D feature correspondence, still exist in epi-line search. The blank wall problem is that given a texture less blank wall, it is very hard to find an accurate pixel to pixel correspondence across the input images.

Another solution to the dense matching problem is volumetric reconstruction method. Lhuillier and Quan proposed a quasi-dense approach to surface reconstruction in which they used a best first search based on combined 3D and 2D information [3][19]. Instead of using pixel-based searching and matching,

volumetric reconstruction takes the scene as a tessellation of 3D cubes, called voxels. Each voxel may be either empty or occupied by the scene structure. Various methods has been proposed to build the volumetric model which is used to generate the most consistent projections with the original images. Volumetric reconstruction could well recover the scene of the moving foreground, however, it is hard to reveal the static background structure using volumetric methods.

In this paper, we propose a novel 3D dense reconstruction method based on geometric segmentation and surface fitting. We use the existing techniques for feature correspondence, projective reconstruction and self-calibration to get the sparse points reconstruction. To address the dense matching problem, we use geometric segmentation to segment the 3D space into several separate regions, and for each region, we estimate the dense 3D depth map by surface fitting. We propose a non-linear deterministic annealing algorithm in order to partition the 3D space geometrically. With the assumption that each subspace could be modeled by a linear plane, we can retrieve the depth information for each pixel using surface fitting. The new approach is able to generate a much smoother 3D dense reconstruction comparing to the traditional methods.

This paper is organized as follows. Section II present the background and problem formulation. We present the system scheme for 3D reconstruction in Section III. Then we solve the geometric segmentation and surface fitting problem in Section IV. The experimental results are shown in Section V. Finally, Section VI concludes this paper.

II. Background and Problem Formation

In this section, we briefly review the 3D reconstruction techniques and formulate the geometric fitting problem mathematically.

A. 3D Reconstruction

3D reconstruction has been one of the most fundamental research topics in computer vision for decades. Although they may differ in some specific part, most 3D reconstruction approaches are generally based on the same pipeline [18]. The pipeline is given in Fig. 1.

The first step in 3D reconstruction from a video sequence is to group the whole video sequence into several scenes by key frames. For each scene, motion detection is needed to find moving regions from the static background. In the later part, moving foreground and static background will be treated separately and then combined together to reconstruct the scene as a whole.

The second step is sparse reconstruction. Sparse reconstruction includes several component, feature correspondence, projection reconstruction and Euclidean reconstruction. The camera motion is estimated and The Euclidean structure of the static background scene is recovered. For the moving regions, we introduce the virtual camera concept and apply the same reconstruction algorithm to recover the 3D structure. During the last two decades, tremendous progress has been made to camera self-calibration and structure computation. Sparse reconstruction starts from feature correspondence which is the most crucial part of the process. The goal of Image correspondence, also called feature correspondence, is to align different images, from a video sequence or taken separately, by finding corresponding points that describe the same point in 3D geometry [20][21]. As known to all, not all points are suitable for matching or tracking through different images, so only a few points are selected as feature points for matching [22]. So sparse reconstruction only rely on a number of distinct points which is different from the following dense reconstruction which require the correspondence of all points, if possible. Furthermore, feature points may be mismatched, known as outliers [23], which may restrict the accuracy of the reconstruction result. Given correctly matched feature points from two input images, projection reconstruction is to find the relative pose between the two views. The projective structure is mathematically expressed by fundamental matrix. Given sufficient corresponding feature points, with the assumption that the world frame is the same frame as that of the first image, we are able to compute the fundamental matrix. The projective reconstruction is determined by an arbitrary projective transformation. To solve this problem,

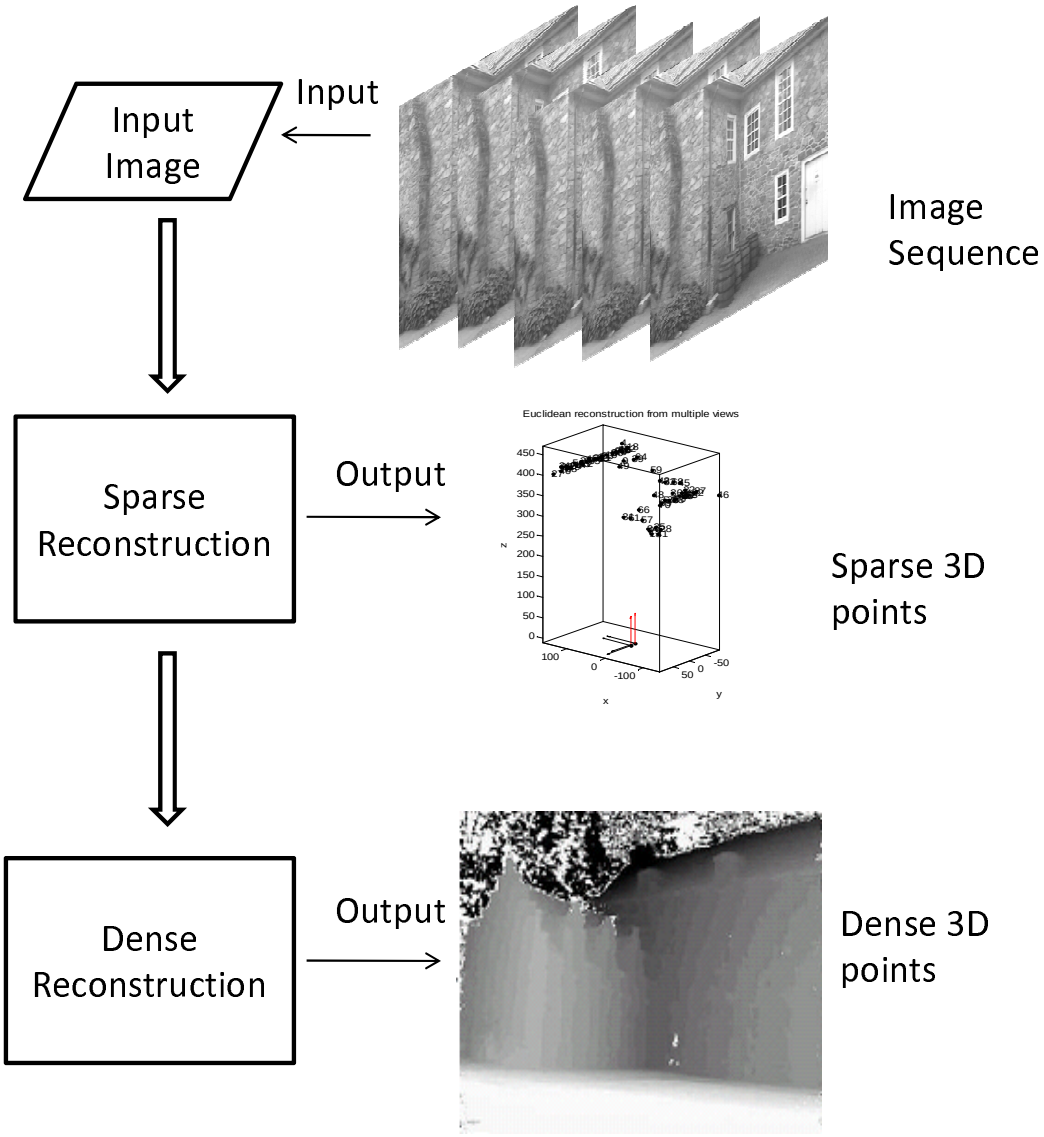

Fig. 1. The pipeline for 3D video reconstruction system.

canonical decomposition is applied to fix a particular choice of projective transformation. Therefore, the projective structure is not suitable for visualization and an update to a full-fledged Euclidean reconstruction is required to recover the metric 3D geometric structure. The update to a metric structure, determined up to an unknown scalar factor, needs the information of intrinsic parameters of the camera. Since we have no prior knowledge of the camera, this approach is called self-calibration and has received a lot of attention in recent years. The approach we present here is called absolute conic constraint, or absolute quadric constraints.

The sparse reconstruction gives a sparse structure of the desired scene; however, it could not give a satisfied visual presentation. Thus, we still need to compute the depth of a lot more points, which is known as dense reconstruction or surface reconstruction. The traditional approaches for dense reconstruction could be classified as two approaches, namely stereoscopic reconstruction and volumetric reconstruction. In this paper, we propose a novel approach to obtain the static background structure. Unlike the previous approach, we apply geometrical segmentation and surface fitting instead of dense searching and matching. Here we assume that the static background could be decomposed of several uniform regions or regular surfaces. We can then segment the whole surface into several regions based on their geometric properties. For each region, we obtain a mathematical expression by surface fitting. With the assumption that each region has

sufficient number of sparse feature points, combined with the sparse depth map, we could then compute the depth information by fitting each pixel within the estimated surface. Combining the depth map of different regions, we could finally obtain the depth map of the whole scene. The merit of this approach is that it well handles uniform regions and occlusions by mismatching issues. Also, the result is smoother than traditional stereoscopic reconstruction algorithms. The geometric fitting problem is formulated in subsection II-B and we give the solution to the problem in details in Section IV.

B. Geometric Fitting

The classic geometric fitting problem is to find a geometrical surface that best fits to a set of 3D points. Geometric fitting is commonly used in 3D model fitting and 3D visual reconstruction in computer vision.

Given a 3D point data set $\mathfrak{X} = \{\mathbf{x}_i\}, \mathbf{x}_i \in R^3, i = 1, 2, ..., n$, the geometrical fitting problem is usually stated as the optimization of a cost that measures how the geometrical surface function $\mathfrak{S} = \{\mathbf{x} : g_\theta(\mathbf{x}) = 0\}$ fits the data set $\mathfrak{X}$. The most commonly used objective function is the least squares cost,

$$D = \sum_{i=1,...N} d(\mathbf{x}_i, g_\theta)^2 \tag{1}$$

$$d(\mathbf{x}_i, g_\theta) = \min \|\mathbf{x}_i - \mathbf{x}_j\|^2, \quad \mathbf{x}_j \in \mathfrak{S} \tag{2}$$

The fitting function g_θ is learned by minimizing the design cost, D, measured over the input data set, $\mathfrak{X}$. It is well-known that for most choices of D, the cost measured during design monotonically decreases as the size of the learned fitting function g_θ is increased. With a large set of functions, it is easy to create a surface which passes through each input data point but is suspiciously complicated. The principle of Occam's razor states that the simplest model that accurately represents the data is most desirable. So we prefer to use a few basis functions which yield a smoother, simpler surface which could well approximates the original data. Generally, there are two approaches to solve the over fitting problem. One approach is to add penalty terms to the data set, like smoothness or regularization constraints. Another approach is to first build a large model and then remove some parameters by retaining only the vital model structure. Although both approaches can generate parsimonious models, the descent based learning methods all suffer from a serious limitation. The non-global optima of the cost surface may easily result in poor local minima to the descent based learning methods. Techniques adding penalty terms to the cost function further increases the complexity of the cost surface and worsen the local minimum problem.

One of the most popular clustering algorithm is Lloyd's algorithm, which starts by partitioning the input data into k initial sets. It calculates the centroid of each set via some metric. Lloyd's algorithm iteratively associates each point with the closest centroid and recalculates the centroids of the new clusters. Although widely used in real world applications, there are two serious limitations of Lloyd's algorithm. The first limitation is that the partitioning result depends on the initialization of the cluster centers, which may lead to poor local minima. The second limitation is that Lloyd's algorithm can only partition linear separable clusters. In order to avoid initialization dependence, a simple but useful solution is to use multiple restarts with different initializations to achieve a better local minima. Global k-means [24] is proposed to build the clusters deterministically, which use the original k-means algorithm as a local search step. At each step, global k-means add one more cluster based on previous partitioning result. Deterministic annealing [25] is another optimization technique to find a global minimum of a cost function. Deterministic annealing explore a larger cost surface by introducing a constraint of randomness. At each iteration, the randomness is constrained and a local optimization is performed. Finally, the imposed randomness is reduce to zero, and the algorithm optimizes over the original cost function. Kernel method [26] is used to solve the second problem by mapping the data points from input space to a higher dimensional feature space through a non-linear transformation. Then the optimization is applied in the feature space. The linear separation in the feature space turns out to be a non-linear separation in the original input space.

III. 3D Video Reconstruction

Here, we simply introduce the 3D reconstruction algorithm proposed in Ma et. al's book[7] on which our experiments are based. 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 disparity error in a highly textured area. In particular, edges and straight lines in the scene need to be rendered correctly.

A. Overview of 3D Reconstruction System

The 3D reconstruction algorithm is implemented in 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. The Euclidean structure of the scene is recovered afterward. After that, we apply the geometric segmentation algorithm described in Section IV. Finally the dense depth map is reconstructed by geometric fitting. The system scheme is given in Fig. 2.

B. Feature Selection

The first step in 3D reconstruction is to select candidate features in all images for tracking across different views. Ma et al. [7] use point feature in reconstruction which is measured by Harris' criterion,

$$C(\mathbf{x}) = \det(G) + k \times \text{trace}^2(G) \tag{3}$$

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

$$G = \begin{bmatrix} \sum_{W(\mathbf{x})} I_x^2 & \sum_{W(\mathbf{x})} I_x I_y \\ \sum_{W(\mathbf{x})} I_x I_y & \sum_{W(\mathbf{x})} I_y^2 \end{bmatrix} \tag{4}$$

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 user, for example 7×7. If $C(\mathbf{x})$ exceeds a certain threshold, then the point $\mathbf{x}$ is selected as a candidate point feature.

C. Feature Correspondence

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) 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:

$$\min_{\mathbf{d}} \sum_{\mathbf{x} \in W(\mathbf{x})} \left[I_2(\mathbf{x} + \mathbf{d}) - I_1(\mathbf{x}) \right]^2 \tag{5}$$

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 close form solution of 5

$$\mathbf{d} = -G^{-1}\mathbf{b} \tag{6}$$

where

$$\mathbf{b} \doteq \begin{bmatrix} \sum_{W_{(\mathbf{x})}} I_x I_t \\ \sum_{W_{(\mathbf{x})}} I_y I_t \end{bmatrix} \tag{7}$$

G is the same matrix we used to compute the quality of the candidate point feature in Eq. 3, and $I_t \doteq I_2 - I_1$.

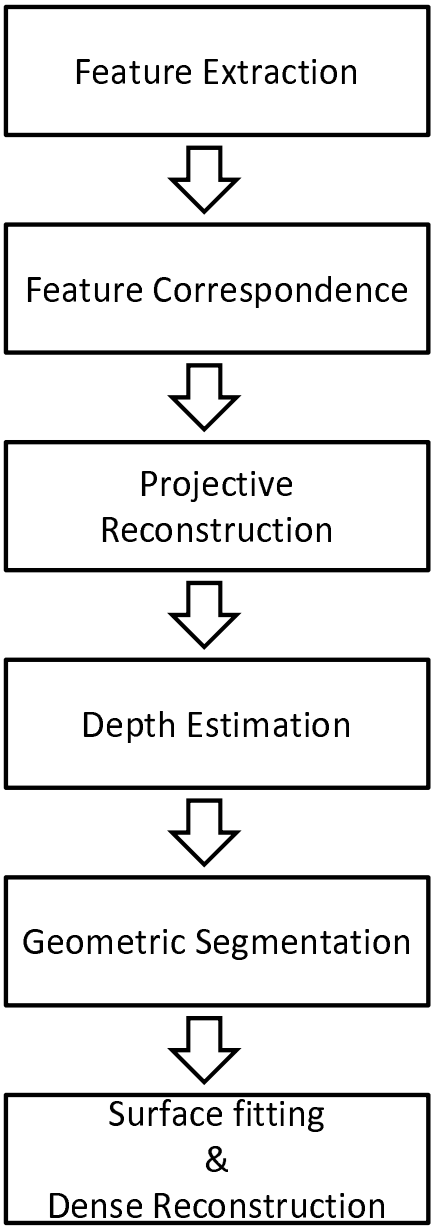

Fig. 2. The scheme for 3D video reconstruction system.

D. 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 [7]. For the detail of the proof of this algorithm, please refer to the reference.

The reconstruction algorithm is based on a perspective projection model with a pinhole camera. Suppose we have a generic point $p \in \mathbb{E}^3$ with coordinates $\mathbf{X} = [X, Y, Z, 1]^T$ relative to a world coordinate frame. Given two 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:

$$\lambda_1 \mathbf{x}_1' = \Pi_1 \mathbf{X}_p, \quad \lambda_2 \mathbf{x}_2' = \Pi_2 \mathbf{X}_p \tag{8}$$

where $\mathbf{x}' = [x, y, 1]^T$ is measured in pixels, λ_1 and λ_2 are the depth scale of $\mathbf{x}_1$ and $\mathbf{x}_2$, $\Pi_1 = [K, 0]$ and $\Pi_2 = [KR, KT]$ are the camera projection matrices and K 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 Equation (8), we have

$$\mathbf{x}'^T_2 K^{-T} \hat{T} R K^{-1} \mathbf{x}'_1 = 0 \tag{9}$$

The fundamental matrix is defined as:

$$F \doteq K^{-T} \hat{T} R K^{-1} \tag{10}$$

With the above model, we could estimate the fundamental matrix F via the Eight-point algorithm. 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:

$$\Pi_1 p = [I, 0], \Pi_2 p = [(\widehat{T'})^T F, T'], \lambda_1 \mathbf{x}'_1 = \mathbf{X}_p, \lambda_2 \mathbf{x}'_2 = (\widehat{T'})^T F \mathbf{X}_p + T' \tag{11}$$

E. Depth Estimation

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

$$\Pi_{ip} \sim \Pi_{ie} H^{-1}, \mathbf{X}_p \sim H \mathbf{X}_e, i = 1, 2, ..., m \tag{12}$$

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

$$H = \begin{bmatrix} K & 0 \\ -\nu^T K & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} \tag{13}$$

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

F. Geometric Segmentation

As we have discussed before, not all points in an image are suitable for matching or tracking. The feature points that we have selected are only a bunch of distinct points. Therefore, the previous reconstruction is a sparse scene reconstruction. The sparse structure is not suitable for human visualization. For this reason, a dense matching is necessary to establish a 3D geometric view.

In this paper, we propose a new dense matching method based on geometric segmentation. We first segment the surface of the 3D scene into several regions based on the geometric relationship. For each small homogeneous surface, we are able to model it by a plane. With the depth information of the feature points that we already get from the sparse reconstruction, we could compute the depth information for each pixel in the entire region. Since the depth information we obtained is based on a plane model, the image rendered from the 3D model is much smoother than the traditional approaches. In order to simplify the problem of surface fitting, we first segment the input image based on its geometric structure. It is different from the traditional object based image segmentation. The segmentation process is critical because proper segmentation could simplify the surface fitting. On the contrary, improper segmentation which combines too many surface areas will increase the complexity of surface modeling.

Due to the fact that the 3D data is localized to a few relatively dense clusters, 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 several regions with different sizes and shapes. For each region, we can easily find a linear plane model to fit the data. Non-linear deterministic annealing method offers three important features: 1) the ability to avoid many poor local optima; 2) the ability to minimize the cost function even its gradients vanish almost everywhere; 3) the ability to achieve non-linear separation. However, there is no close form solution for non-linear deterministic annealing problem, therefore we use a gradient descent algorithm to solve this problem. The details of this algorithm is discussed in Section IV.

G. Depth Recovery

Here, we only consider two images. Suppose for the first image, we have the 3D point set $\mathbf{X}_e^j, j = 1, 2, ..., n$ which could be divided into three clusters,$\mathbf{X}_{e1}$, $\mathbf{X}_{e2}$, $\mathbf{X}_{e3}$. For each cluster, there are at least three non-collinear points. Then we could have the plane model for this cluster. Let's take the example of $\mathbf{X}_{e1}$, suppose there are m points in the cluster and we have the plane model as follows:

$$\mathbf{A} \cdot p = 1 \tag{14}$$

where $\mathbf{A} = [\mathbf{X}_{e1}^i], i = 1, ..., m$ and $p = [a, b, c]^T$ is the plane parameter.

Given an arbitrary point $\mathbf{x}^i = [x^i, y^i]^T$ measured in pixels in the first cluster, we could estimate it's depth scale λ^i by solving the following equation.

$$\lambda^i \mathbf{x'}^i = H_1^{-1} \Pi_1 \mathbf{X}_e^i \tag{15}$$

where $\mathbf{x'}^i = [x^i, y^i, 1]^T$, H_1^{-1} and Π_1 are estimated in previous subsections. In Eq. 15, only λ^i is unknown and with the constraint on $\mathbf{X}_e^i$ with Eq. 14, we could easily get the value of λ^i.

Then, with $\Pi_1 = [I, 0]$, we could have $X_p^i = [\lambda_1^i x^i, \lambda_1^i y^i, \lambda_1^i, 1]$. from Eq. 8, we can get the relation between two image projection point $\mathbf{x}_1^i$ and $\mathbf{x}_2^i$ as follows:

$$\widehat{\mathbf{x_2^i}'} = \Pi_2 \mathbf{X}_p^i \tag{16}$$

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

IV. Geometric Segmentation based Dense Reconstruction

As we have discussed, not all points in an image are suitable for matching or tracking. The feature points that we have selected are only a bunch of distinct points. Therefore, the first reconstruction is a sparse reconstruction. The sparse structure is not suitable for human visualization. For this reason, a dense matching is necessary to establish a 3D geometric view. As known to all, 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 disadvantages of this approach are that the dense depth map is not smooth because of outliers. Lhuillier and Quan proposed another dense matching method called quasi-dense approach. They tried to combine 3D data points and 2D image information. However, the visual problem still exists.

In this paper, we propose a non-linear deterministic annealing approach for space partitioning in 3D Euclidean space. We use deterministic annealing to divide the input space into several regions with different sizes and shapes. With the partition, we can easily find a linear local surface to fit the data within each region. Deterministic annealing method offers two great features: 1) the ability to avoid many poor local optima; 2) the ability to minimize the cost function even its gradients vanish almost everywhere. Due to the fact that the data is localized to a few relatively dense clusters, we design a non-linear function to map the data point from the geometric space to surface feature space and apply deterministic annealing in the feature space instead of the geometric space. The advantage of our approach is that the estimated dense depth map is much more smooth than the traditional approaches.

Given a set of data $\mathfrak{X}$ of scattered 3D points, we would like to find the geometric surface that best fits to the scattered data. The fitting problem is usually stated as the optimization of a cost that measures how well the fitting function $g(\mathbf{x}_i)$ fits the data. The most commonly used objective function is the least squares cost. Finding a good fit is a challenging problem and may be more of an art than a science. If we use a large set of functions as the basis, we may create a surface which passes through each data point but is suspiciously complicated. Using few basis functions may yield a smoother, simpler surface which only approximates the original data. Due to the over fitting problem, we propose an new approach to optimize the objective function via space partitioning. We first partition the data set into several subsets such that

the data points $\mathbf{x}$ in each subset could be approximated by a linear surface model. In other words, we would like to use a set of plain models to approximate the date set. The objective of space partitioning is to minimize the geometric fitting error.

$$\min_{g_{\theta_k}} \sum_{k=1}^{K} \sum_{i \in C_k} d(\mathbf{x}_i, g_{\theta_k}) \tag{17}$$

where, $\mathbf{x}_i = [x_i, y_i, z_i]^T$ is the i-th point data, $\theta_k = [a_k, b_k, c_k]^T$ is the k-th linear surface model, and $d_{i,k}$ is is the fitting error between $\mathbf{x}_i$ and plane model $g_{\theta_k} = 0$ which is defined as

$$d_{i,k} = d(\mathbf{x}_i, g_{\theta_k}) = \frac{(\mathbf{x}_i^T g_{\theta_k} - 1)^2}{a_k^2 + b_k^2 + c_k^2} \tag{18}$$

A. Deterministic Annealing

The deterministic annealing (DA) approach [25] to clustering has demonstrated substantial performance improvement over traditional supervised and unsupervised learning algorithms. DA mimics the annealing process in static The advantage of deterministic annealing is its ability to avoid many poor local optima. The reason is that deterministic annealing minimizes the designed cost function subject to a constraint on the randomness of the solution. The constraint, Shannon entropy, is gradually lowered and eventually deterministic annealing optimize on the original cost function. Deterministic annealing mimics the simulated annealing [27] in statistical physics by the use of expectation. Deterministic annealing derives an effective energy function through expectation and is deterministically optimized at successively reduced temperatures. The deterministic annealing approach has been adopted in a variety of research fields, such as graph-theoretic optimization and computer vision. A. Rao et al. [28] extended the work for piecewise regression modeling. In this subsection, we will briefly review their work.

Given a data set $(\mathbf{x}, \mathbf{y})$, the regression problem is to optimize the cost that measures how well the regression function $f(\mathbf{x})$ approximates the output $\mathbf{y}$, where $\mathbf{x} \in \mathcal{R}^m$, $\mathbf{y} \in \mathcal{R}^n$, and $g : \mathcal{R}^m \rightarrow \mathcal{R}^n$. In the basic space partitioning approach, the input space is partitioned into K regions and the cost function becomes

$$\min_{g_{\theta_k}} \sum_{k=1}^{K} \sum_{i \in C_k} d(\mathbf{y}_i, f(\mathbf{x}_i, g_{\theta_k}) \tag{19}$$

where $d(\cdot, \cdot)$ is the distortion measure function. Instead of seeking the optimal hard partition directly, randomness is introduced for randomized assignment for input samples.

$$D = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} P(\mathbf{x}_i \in C_j) d(\mathbf{y}_i, f(\mathbf{x}_i, g_{\theta_k}) \tag{20}$$

In A. Rao et al.'s work, they use the nearest prototype (NP) structure as constraint and given the set of prototypes$\{\mathbf{s}_j : j = 1, 2, 3, ..., K\}$ in the input space, a Voronoi criterion is defined for NP partition

$$C = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} P(\mathbf{x}_i \in C_j) ||\mathbf{x}_i - \mathbf{s}_j|| \tag{21}$$

Although the ultimate goal is to find the hard partition, some "randomness" is desired during the assignment. Shannon entropy is introduced as a constraint of the randomness.

$$H = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{K} P(\mathbf{x}_i \in C_j) \log P(\mathbf{x}_i \in C_j) \tag{22}$$

Eventually, this constrained optimization problem could be rewritten as the minimization of the corresponding Lagrangian

$$\min_{\{\mathbf{\Lambda_j}\}\{\mathbf{s_j}\},\gamma} F = D - TH \tag{23}$$

where, γ is a nonnegative Lagrange multiplier which controls the randomness of the space partition.

B. Non-linear Deterministic Annealing

In this paper, we propose a new approach based on non-linear deterministic annealing to solve the 3D geometric fitting problem. We first use a non-linear function to map the input point data to a high dimensional feature space using the local geometric structure of the data. Then we apply deterministic annealing in the feature space to leverage the local geometric structure for clustering.

To solve the space partitioning problem, we do not use prototype to calculate the difference. The reason is that the prototype in space partitioning is generally not sufficient to represent a plane in 3D space. Instead, we estimate the linear plane model and calculate the fitting error as the Euclidean distance between the data and the plane. The traditional local optimization algorithm will likely stuck at a local optima. In order to avoid local optima, we use local geometric structure from neighboring data points and embedded the data vectors to a higher dimension as follows.

The input data is given as a 3D point, $\mathbf{x}_i = [x_i, y_i, z_i]^T$. With the assumption that nearest data points are on the same plane, we could estimate the local plane model, $\mathbf{L}_i = [a_i, b_i, c_i]^T$ of data point $\mathbf{x}_i$ and its K nearest neighbor points.

$$\mathbf{L} = \begin{bmatrix} a(\mathfrak{X}) \\ b(\mathfrak{X}) \\ c(\mathfrak{X}) \end{bmatrix} \tag{24}$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{x} \\ \mathbf{L} \end{bmatrix} \tag{25}$$

Then we revise the distortion function as follows,

$$D(\mathbf{f}_i, g_{\theta_j}) = D_1(I_1\mathbf{f}_i, g_{\theta_j}) + D_2(I_2\mathbf{f}_i, g_{\theta_j}) \tag{26}$$

$$I_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{27}$$

$$I_2 = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{28}$$

where $D_1 = d_{i,j}$ calculate the fitting error between the data point and the estimated plane, and D_2 calculate the difference between the local estimated plane model and the cluster scale estimated plane model. D_2 is defined as follows:

$$D_2(I_2\mathbf{f}_i, g_{\theta_j}) = \frac{I_2\mathbf{f}_i^T \times g_{\theta_j}}{|I_2\mathbf{f}_i| \times |g_{\theta_j}|} \tag{29}$$

After the mapping, we apply deterministic annealing algorithm to partition the data into several clusters as follows.

$$\min_{g_{\theta_j}} F = D - TH \tag{30}$$

where $g_{\theta_j} = [a_j, b_j, c_j]$ is the geometrical surface model parameter to be estimated, D is the sum of square of geometrical fitting error and H is the entropy constraint. We define D and H as follows:

$$D = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} p(\mathbf{x}_i, g_{\theta_j}) d(\mathbf{x}_i, g_{\theta_j}) = \sum_{i=1}^{N} p(\mathbf{x}_i) \sum_{j=1}^{K} p(g_{\theta_j}|\mathbf{x}_i) d(\mathbf{x}_i, g_{\theta_j}) \tag{31}$$

$$H(\mathbf{X}, g_\theta) = \sum_{i=1}^{N}\sum_{j=1}^{K} p(\mathbf{x}_i, g_{\theta_j}) \log p(\mathbf{x}_i, g_{\theta_j}) \tag{32}$$

To perform optimization we need to further analyze its terms. We can rewrite equation (32) by applying the chain rule of entropy as

$$H(\mathbf{X}, g_\theta) = H(\mathbf{X}) + H(g_\theta|\mathbf{X}) \tag{33}$$

Notice that the first term $H(\mathbf{X})$ is the entropy of the source and is therefore constant with respect to the cluster g_{θ_j} and association probabilities $p(g_{\theta_j}|\mathbf{x}_i)$. Thus we can just focus on the conditional entropy

$$H(g_\theta|\mathbf{X}) = \sum_{i=1}^{N} p(\mathbf{x}_i) \sum_{j=1}^{K} p(g_{\theta_j}|\mathbf{x}_i) \log p(g_{\theta_j}|\mathbf{x}_i) \tag{34}$$

The minimization of F with respect to association probabilities $p(g_{\theta_j}|\mathbf{x}_i)$ gives rise to the Gibbs distribution

$$p(g_{\theta_j}|\mathbf{x}_i) = \frac{\exp(-\frac{d(\mathbf{x}_i, g_{\theta_j})}{T})}{Z_x} \tag{35}$$

where the normalization is

$$Z_x = \sum_{j=1}^{K} \exp(-\frac{d(\mathbf{x}_i, g_{\theta_j})}{T}) \tag{36}$$

The corresponding minimum of F is obtained by plugging equation (35) back into equation (30)

$$F^* = \min_{p(g_{\theta_j}|\mathbf{x}_i)} F = -T\sum_{i=1}^{N} p(\mathbf{x}_i) \log Z_x \tag{37}$$

To minimize the Lagrangian with respect to the cluster model g_{θ_j}, its gradients are set to zero yielding the condition

$$\nabla_{g_{\theta_j}} F = \frac{1}{N}\sum_{i=1}^{N} p(g_{\theta_j}|\mathbf{x}_i) \nabla_{g_{\theta_j}} d(\mathbf{x}_i, g_{\theta_j}) = 0 \tag{38}$$

Since there is no close form solution for non-linear deterministic annealing problem, we use a gradient descent algorithm to solve this problem. I present our algorithm in Figure. 3.

V. Experimental Results

In this paper, I first compared three geometric segmentation algorithms, Projection based iterative geometric segmentation algorithm (PI), Adaptive projection based iterative algorithm (API), and non-linear DA based geometric segmentation algorithm(NDA), based on both synthetic data and real world data.

1) Algorithm 3 **KDA based geometrical segmentation algorithm**
2) **Set Limit**
3) K_{max}: maximum number of clusters
4) T_{init}: starting temperature
5) T_{min}: minimum temperature
6) δ: perturbation vector
7) α: cooling rate (must be < 1)
8) I_{max}: maximum iteration number
9) th: Iteration threshold
10) sth: Surface distance threshold
11) **Initialization**
12) $T = T_{init}, K = 2, \Lambda_1 = (X^T X)^{-1} X^T \vec{1}, \Lambda_2 = \Lambda_1, [p(\mathbf{\Lambda}_1|\mathbf{x}_i), p(\mathbf{\Lambda}_2|\mathbf{x}_i)] = [\frac{1}{2}, \frac{1}{2}], \forall i.$
13) **Perturb**
14) $\Lambda_j = \Lambda_j + \delta, \forall j.$
15) $L_{old} = D - TH.$
16) **Loop until convergence,** $i = 0\ \forall j$
17) For all $\mathbf{x}_i$ in the training data, compute the association probabilities

$$p(\mathbf{\Lambda}_j|\mathbf{x}_i) = \frac{\exp(-\frac{d(\mathbf{x}_i, \mathbf{\Lambda}_j)}{T})}{\sum_{j=1}^{K} \exp(-\frac{d(\mathbf{x}_i, \mathbf{\Lambda}_j)}{T})} \tag{39}$$

18) update the surface model

$$\mathbf{\Lambda}_j \longleftarrow \mathbf{\Lambda}_j + \alpha \nabla_{\mathbf{\Lambda}_j} F. \tag{40}$$

19) i = i+1;
20) if ($i > I_{max}$ or $\nabla_{\mathbf{\Lambda}_j} F < th$) End **Loop**
21) **Model Size Determination**
22) if($d(\mathbf{\Lambda}_j, \mathbf{\Lambda}_{j+1}) < sth$)
23) replace $\mathbf{\Lambda}_j, \mathbf{\Lambda}_{j+1}$ by a single plane
24) K =number of planes after merging
25) **Cooling Step**
26) $T = \alpha T.$
27) if ($T < T_{min}$)
28) perform last iteration for $T = 0$ and STOP
29) **Duplication**
30) Replace each plane by two planes at the same location, $K = 2K.$
31) **Goto Step 10**

Fig. 3. KDA based geometrical segmentation algorithm

A. NDA on Synthetic Data

The purpose of the first experiment is to compare NDA, PI, and API on synthetic data with ground truth. I generated the synthetic data using MATLAB 'randperm' function. The data is a set of 3D points on several linear planes without noise. In this experiment, I run each algorithm for 1000 times. Each time, a random data set is generated and used. We segment the same data set with different algorithms and calculate the average squared approximation error. Below is the experimental result in Table. I. K represents the number of planes in a test data set. For each plane, 100 random points are generated. The date set 1 contains 300 data in total from 3 non parallel planes. The data set 2 contains 400 data from 4 planes. The data set 3 contains 500 data from 5 planes and the data set 4 contains 600 data from 6 planes. The average squared approximation error of NDA is ignorable comparing to the errors of PI and NPI. From the experimental result, we can say that NDA algorithm outperforms both PI and API algorithms in the average squared approximation error. The reason NDA algorithm outperforms PI and API algorithms is that NDA is able to separate the space non-linearly and avoid many poor local optima.

We also measure the performance of the segmentation algorithms in percentage of correct identification of planes. We test the same data set as used in the previous experiment and compute the correct identification percentage averaging over all tests. Below is the experimental result in Table. II. We observed that correct identification rates of NDA and API are much higher than the correct identification rate of PI algorithm. The reason API algorithm outperforms PI algorithm is that API algorithm does not depends

TABLE I

THE AVERAGE SQUARED APPROXIMATION ERROR.

K	PI	API	NDA
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}

TABLE II

THE CORRECT IDENTIFICATION RATE.

K	PI	API	NDA
3	83%	96%	99%
4	79%	93%	99%
5	82%	94%	97%
6	78%	97%	98%

on random initialization while the segmentation results of PI algorithm heavily depends on initialization. Still NDA performs best among the three algorithms in correct identification rate.

B. NDA on Real World Data

In the second experiment, we test the geometric segmentation algorithm on some real world data. We use the 3D structure data set from the 'housing' image sequence. The data set includes 72 data points recovered by 3D reconstruction of 2D registered feature points. Most of the data points fall on the walls of the house in the image and we would like to estimate the surface model of the walls by geometric fitting. Fig. 4 shows the input 3D data points on the 1st frame of the 'housing' image sequence. The goal is to segment the data points into three groups and each group represent a wall in the image. Fig. 5 shows the geometric segmentation result by NDA algorithm and Fig. 6 shows the geometrical segmentation result by PI algorithm. It is pretty clear that NDA algorithm partitions the input data set into three clusters and each cluster represents a wall in the image. PI algorithm fails to find the geometric model of the walls and the data points are mixed. The experimental result on real world data shows that NDA algorithm can well segment the data sets based on their geometric relationship and the 3D surface is accurately recovered.

C. 3D Video Dense Reconstruction

In the third experiment, we integrate the NDA algorithm in the 3D video reconstruction system. The input is an image sequence and the output is a dense depth map. In our experiment, we use the 'oldhousing' image sequence. Fig. 7 shows the first frame and the 88th frame of the test image sequence 'oldhousing'. We first extract point features on all the input images. Then we apply feature correspondence algorithm to relate all the features. Fig. 4 show the selected feature points on the first frame. We then estimate the camera pose and intrinsic parameters. With the camera parameters, we are able to recover the sparse Euclidian structure of the feature points. Fig. 8 shows the estimated depth map of the selected feature points and the camera pose. After sparse reconstruction, we separate the 3D space into several regions using NDA algorithm. For each region, we use the surface fitting algorithm presented in Section III to estimate the depth information of each pixel. Combining the depth map of all regions, we can recover the 3D dense depth map of the whole frame. Fig. 9 shows the estimated dense depth map of the whole frame. Since we use surface fitting instead of searching for dense depth estimation, we do not need to worry about matching errors and outliers. The estimated dense depth map is very smooth and well represent the geometric structure of the 3D scene.

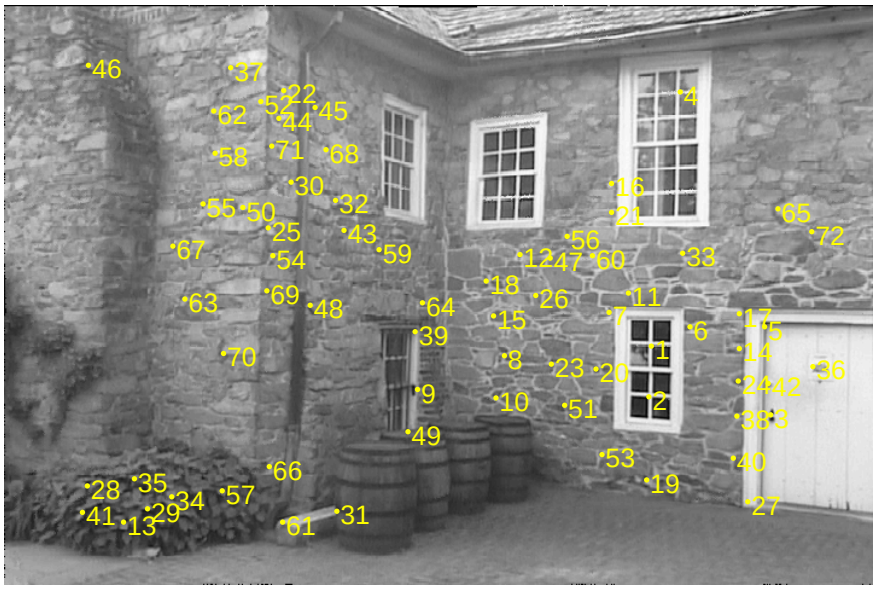

Fig. 4. The input data points on the 1st frame of 'housing' image sequence.

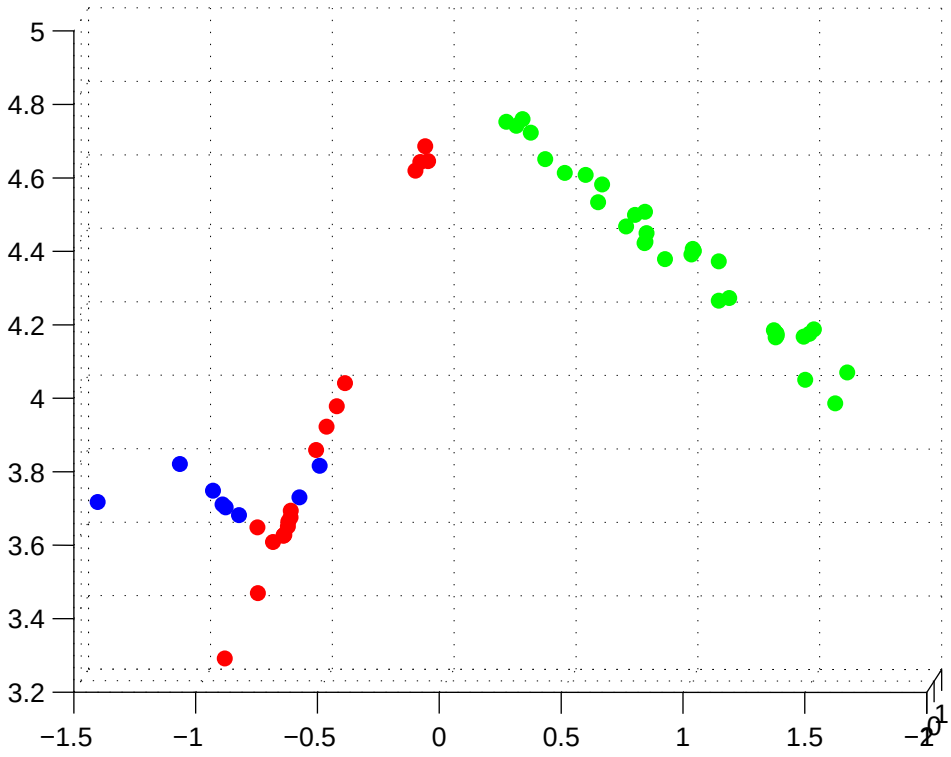

Fig. 5. The geometrical segmentation result by the NDA algorithm of 'housing' data set.

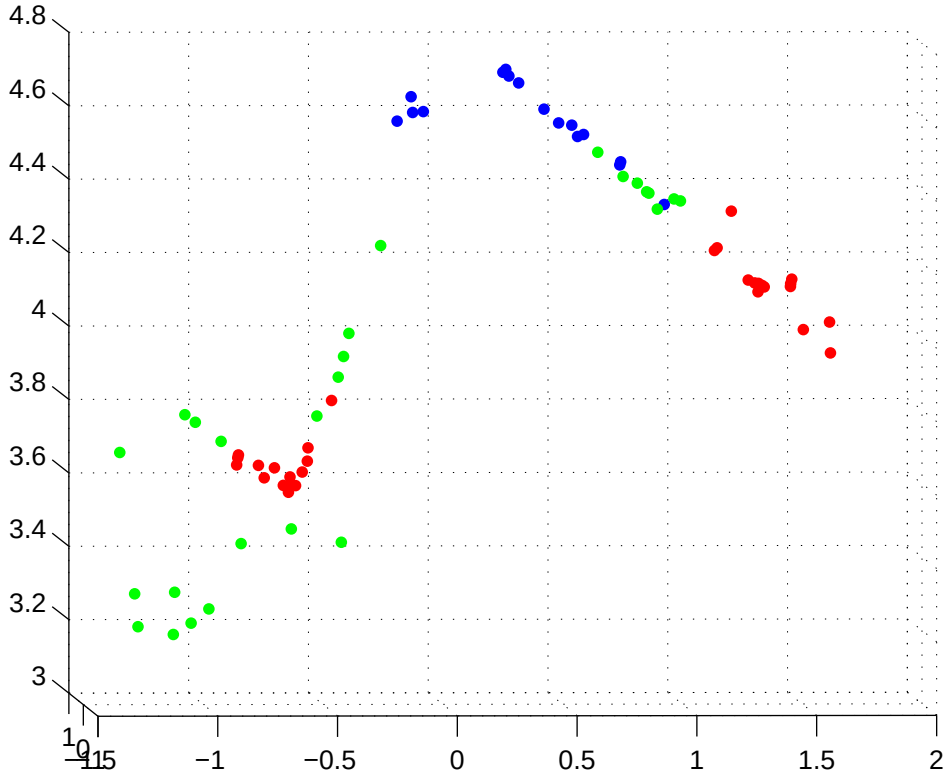

Fig. 6. The geometrical segmentation result by the PI algorithm of 'housing' data set.

VI. Conclusion

In this paper, we propose a novel solution to the dense reconstruction which is based on geometric segmentation and surface fitting. We use the existing techniques for feature correspondence, projective reconstruction and self-calibration to get the sparse points reconstruction. We propose an non-linear Deterministic Annealing algorithm to segment the 3D space into several regions based on the geometric relationship. For each region, given the intrinsic parameters from self-calibration, we can retrieve the depth information for each pixel using surface fitting. The NDA algorithm is able to separate the 3D space non-linearly and is shown to be more accurate compared to the PI and API algorithms. The new dense reconstruction approach can generate smoother dense map comparing to the traditional methods. In the future work, we will further study new surface fitting algorithm for non-linear surface models.

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.

(a) The 1st frame in the ‘oldhousing’ video sequence (b) The 88th frame in the ‘oldhousing’ video sequence

Fig. 7. Original frames used for image registration

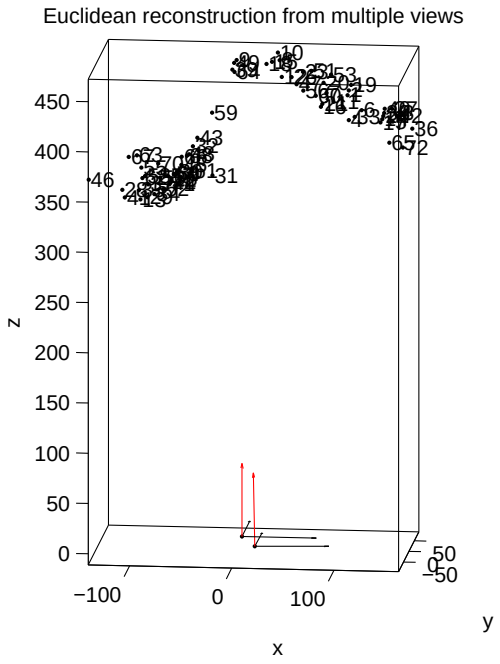

Fig. 8. The estimated sparse depth map and camera pose for the selected feature points of the 1st and 88th frames.

REFERENCES

[1] Z. Zhang, “A flexible new technique for camera calibration,” *Ieee Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[2] C. Strecha, T. Tuytelaars, and L. Van Gool, “Dense matching of multiple wide-baseline views,” in *International Conference on Computer Vision*, vol. 2, pp. 1194–1201, Citeseer, 2003.

[3] 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.

[4] H. Jin, S. Soatto, and A. Yezzi, “Multi-view stereo reconstruction of dense shape and complex appearance,” *International Journal of Computer Vision*, vol. 63, no. 3, p. 189, 2005.

[5] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press New York, NY, USA, 2003.

[6] E. Trucco and A. Verri, *Introductory techniques for 3-D computer vision*. Prentice Hall New Jersey, 1998.

[7] Y. Ma, S. Soatto, and J. Košecká, *An invitation to 3-d vision: from images to geometric models*. Springer Verlag, 2004.

[8] H. Li, B. Adams, L. Guibas, and M. Pauly, “Robust single-view geometry and motion reconstruction,” in *ACM SIGGRAPH Asia 2009 papers*, pp. 1–10, ACM, 2009.

[9] P. Beardsley, P. Torr, and A. Zisserman, “3D Model Acquisition from Extended Image Sequences,” in *Proceedings of the 4th European Conference on Computer Vision-Volume II-Volume II*, pp. 683–695, Springer-Verlag, 1996.

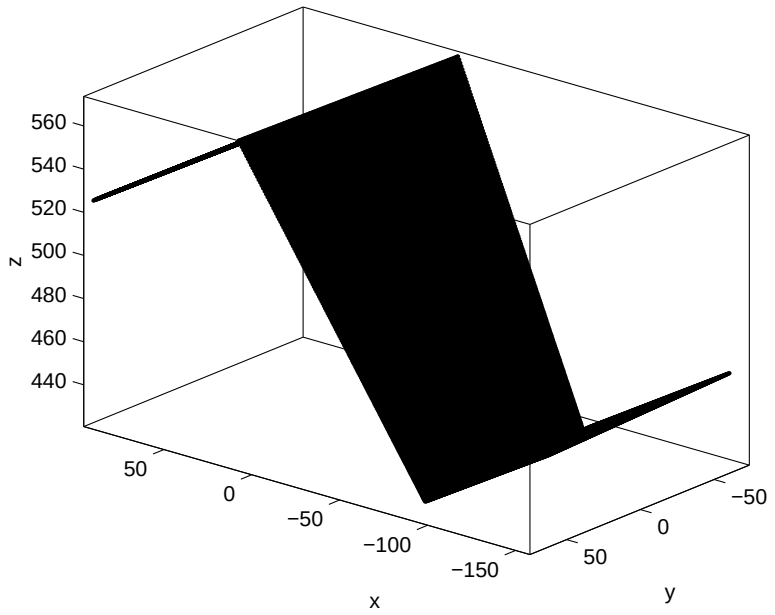

Fig. 9. The estimated dense 3D configuration.

[10] A. Fitzgibbon and A. Zisserman, "Automatic Camera Recovery for Closed or Open Image Sequences," in *Proceedings of the 5th European Conference on Computer Vision-Volume I-Volume I*, pp. 311–326, Springer-Verlag, 1998.

[11] M. Pollefeys, R. Koch, and V. Gool, "Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters," *Journal of Computer Vision*, 1998.

[12] F. Devernay and O. Faugeras, "Automatic calibration and removal of distortion from scenes of structured environments," *Investigative and Trial Image Processing*, vol. 2567, pp. 62–72, 1995.

[13] J. Yagnik and K. Ramakrishnan, "A model based factorization approach for dense 3D recovery from monocular video," in *Seventh IEEE International Symposium on Multimedia*, p. 4, 2005.

[14] V. Popescu, E. Sacks, and G. Bahmutov, "Interactive point-based modeling from dense color and sparse depth," in *Eurographics Symposium on Point-Based Graphics*, Citeseer, 2004.

[15] H. Chang, J. Moura, Y. Wu, K. Sato, and C. Ho, "Reconstruction of 3D dense cardiac motion from tagged MR sequences," in *IEEE International Symposium on Biomedical Imaging: Nano to Macro, 2004*, pp. 880–883, 2004.

[16] O. Faugeras and R. Keriven, "Complete Dense Stereovision Using Level Set Methods," in *Proceedings of the 5th European Conference on Computer Vision-Volume I-Volume I*, p. 393, Springer-Verlag, 1998.

[17] R. Koch, M. Pollefeys, and L. Gool, "Multi Viewpoint Stereo from Uncalibrated Video Sequences," in *Proceedings of the 5th European Conference on Computer Vision-Volume I-Volume I*, p. 71, Springer-Verlag, 1998.

[18] M. Pollefeys, R. Koch, M. Vergauwen, and L. Van Gool, "Automated reconstruction of 3D scenes from sequences of images," *ISPRS Journal Of Photogrammetry And Remote Sensing*, vol. 55, no. 4, pp. 251–267, 2000.

[19] M. Lhuillier and L. Quan, "Surface reconstruction by integrating 3D and 2D data of multiple views," in *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pp. 1313–1320, 2003.

[20] 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, Citeseer, 1981.

[21] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.

[22] J. Shi and C. Tomasi, "Good Features to Track," in *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pp. 593–600, 1994.

[23] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[24] A. Likas, N. Vlassis, *et al.*, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451–461, 2003.

[25] 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.

[26] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine Learning*, vol. 74, no. 1, pp. 1–22, 2009.

[27] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of Statistical Physics*, vol. 34, no. 5, pp. 975–986, 1984.

[28] 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.