Vol.2013-CVIM-187 No.43
2013/5/30

3D Shape Reconstruction by Dynamic Sensing with A Range Sensor

Xiangqi Huang[1,a)] Bo Zheng[1] Takeshi Masuda[2] Atsuhiko Banno[2] Katsushi Ikeuchi[1]

Abstract: Building 3D shape models of interesting targets and getting their localization are very important fundamental tasks in the fields like digital culture heritage, robotics, mixed reality, medical application and so on. Most commonly used range sensors cost a relatively long time of data acquisition. This causes data distortion and cannot be ignored especially in the case of continuously gathering data on a moving platform. Existing common solutions to this problem are either taking "stop-scan-go" strategy to avoid distortion or correcting the sensor motion using secondary sensors like GPS, inertial sensor, camera or even another range sensor. Some other researches apply linearization, discretization and other specified constraints to the problem to achieve compromised results.
This thesis focuses on efficient and accurate 3D shape reconstruction under a moving sensing system only with a single range sensor. Different from previous solutions, the proposed system works in an efficiently continuous manner. We don't have to stop the platform to obtain a stationary scan. Sensor can move under a reasonable motion mode and simultaneously scan the target or environment. Data distortion caused by continuous movement will be rectified. This continuous manner is much more efficient and attractive in practical applications.
Our method consists of two major components: 1) sensor motion estimation - recovering motion of laser projector which leads to data distortion and relies on 2) inter-frame correspondences - extracting local feature points which are robust to data distortion. In experiments, we demonstrate that the algorithm achieves substantially better performance for i) data rectification, ii) robust feature extraction.

1. Introduction

Building accurate 3D models of interesting targets is a very important fundamental task in the fields like using range finder in digital culture heritage, robotics, mixed reality, medical application and so on. [1–4]

When investigating 3D reconstruction, it is necessary to describe the kind of data being considered. Most commonly used sensors for 3D modeling are camera and laser range sensor. Compared with camera, range sensor has the following advantages: 1. It can easily get direct and highly accurate 3D data. 2. Range data is stable, whereas the 3D modeling techniques based on 2D cameras are much more sensitive to illumination, shadows, scale and pose. However, range sensor costs a relatively long time of data acquisition, e.g., Vivid 9i takes 30 seconds for one 3D scan. This causes data distortion and cannot be ignored especially in the case of continuously gathering data on a moving platform with relatively fast speed.

The way of continuously obtaining range data is obviously more efficient and practical in actual applications. This paper challenges the problem of 3D shape reconstruction using a moving range sensor.

[1] Institute of Industrial Science, the University of Tokyo, Japan
[2] The National Institute of Advanced Industrial Science and Technology, Japan
[a)] huang@cvl.iis.u-tokyo.ac.jp

2. Related Work

A typical issue coming from robotics field is to build a 3D map of complex non-flat terrain. Although range sensors can afford accurate 3D point clouds of the environment, relatively slow data acquisition rate compared with fast vehicle speed leads to distorted data and difficulties for alignment. A common solution to this problem is taking "stop-scan-go" strategy [5, 6] to avoid distortion.

In practical, gathering data continuously with moving sensors is obviously a more efficient way. There are also a lot of work trying to solve the data distortion problem for moving range sensor. Considering from the information source, they can be categorized into two classes. The first group tries to solve the problem by utilizing external information from a secondary sensor [7, 8] or prior knowledge of the environment or targets. For example, to digitize large-scale culture heritage, [9] has used laser range sensor mounted on flying balloon to cover the whole site. The uncontrollable movement of balloon causes distortions on the data. Utilizing the overlapped areas with another range sensor fixed on the ground, Banno and Ikeuchi adopt polynomial fitting to

approximate the sensor motion parameters, then align and rectify distorted data. Harrison and Newman [10] have made effort to utilize the vertical plane feature in man-made environment to refer low amplitude roll, pitch and yaw movements of vehicle moving on uneven ground. The other group only use internal information from the sensor itself. Bosse and Zlot [11] proposed a 3D scan-matching method varied from the ICP algorithm, in which sensor trajectory is discretized and constraint are linearized to deal with continuously collected data from a moving vehicle. Structured light range sensors with rolling shutter, like Microsoft Kinect, have the similar data distortion problem when it is used on a moving platform. In [12], Ringaby and Forssen proposed a scan rectification method by estimating 3D camera trajectory.

Another related line of work is rectification of rolling shutter video. This problem has been studied, and solved to some extent [13, 14]. What is different here is that in range sensor systems we have access to depth values in all pixels, and these allow us to robustly solve for the full 3D sensor trajectory, instead of resorting to affine motion [13], or rotation only models [14].

Our data rectification algorithm proposed in this paper, has utilized the correspondence constraints between 3D shapes. A lot works have been done on this problem of finding correspondence between 3D shapes. Classical rigid alignment includes methods based on sampling and verifying candidate transformation, or by applying the iterated closest point algorithm or its variations [15, 16]. More recent works focus on shapes with large variations. There are some successful works about non-rigid registration of surfaces [17–20], which usually consider the whole shape as a single feature and can not handle with large missing of data. Several recent works also deal with matching approximately isometric shapes [21, 22].

In the case of deformation coming from relative motion between range sensor and objects, it has its own properties. It usually includes large amount of missing data, due to occlusion or change of scene or view points. The shape might be deformed significantly between scans and the deformation doesn't keep the geodesic distance invariant. Thus we consider local shape feature is better choice for this problem.

3. Shape Rectification for Intra-frame Motion

Since raw range data only have sensor-oriented coordinates, data from different view points are described in different coordinate system. In the continuously data gathering mode, each point has a individual coordinate system. To correctly align those points, we need to apply unique transformation to each point to make them described in the unified coordinate system. When there are some overlapped areas between two data sets, we apply transformations to make them coincide.

Those transformations can be derived from the sensor motion. We build a proper model to describe the sensor motion. Utilizing the correspondence of feature points in overlapped areas, we can construct a cost function to evaluate the motion model. Sensor motion is estimated by minimizing the cost function. Finally the distorted data can be rectified based on the transformation derived from estimated sensor motion.

3.1 Intra-frame Motion Model

According to Chasles' Theorem [23], every rigid body motion can be realized by a rotation about an axis combined with a translation parallel to that axis. It can be represented using exponential of twist ξ, as shown in following equation:

$$G = e^{\hat{\xi}} = I + \hat{\xi} + \frac{(\hat{\xi})^2}{2!} + \frac{(\hat{\xi})^3}{3!} + \cdots \tag{1}$$

$$\text{where } \xi = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \hat{\xi} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2}$$

where the rotated angel θ is the norm of ω. v describes the translation parallel to the axis ω.

We use twist coordinate to describe rigid body kinematics for its two main advantages. The first is that they allow a global description of rigid body motion which does not suffer from singularities due to the use of local coordinates. Such singularities are inevitable when one chooses to represent rotation via Euler angles, for example. The second advantage is that screw theory provides a very geometric description of rigid motion which greatly simplifies the analysis of mechanisms. [23]

For convenience without loss of generality, we have made an assumption that the motion of sensor is smooth, which is appropriate in most practical applications. This means that the sensor motion can be represented by a smooth function whose continuity is higher than C^0. Such a smooth function can be represented by a sum of smooth basis functions, and the most simple case is the polynomial representation.

While the sensor moves, the transformation changes in accordance with time t, and the twist coordinate is represented by functions of time t as $\xi(t)$

$$\xi(t) = \xi_0 + \xi_1 \cdot t + \xi_2 \cdot t^2 + \xi_3 \cdot t^3 + \cdots \tag{3}$$

which we call *polynomial motion*. Without loss of generality, we can assume that, initially at $t = 0$, the sensor and object coordinate systems are identical, and hence $\xi_0 = 0$. We denote the coefficients of polynomial motion of N-th order by $\xi(t)_N$, where $\xi_1, \xi_2, \cdots, \xi_N$ are 6×1 twist vectors. Here we assume that all parameters are constant in the whole sensor motion sequence of a time duration T_c.

Therefore the trajectory of range sensor in the object frame $q(t)$ can be represented as:

$$q(t) = G(t) \cdot q(0) = e^{\hat{\xi(t)}} \tag{4}$$

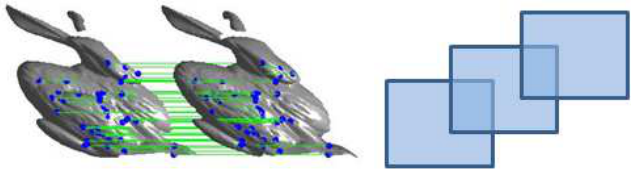

Fig. 1 Overlapped areas between sequent range images.

Given the motion model of sensor $G(t)$, a point of the object measured at time t with local range sensor coordinate p_r, can be transformed to the object frame:

$$q_o = G(t) \cdot p_r \tag{5}$$

3.2 Motion Estimation from Correspondences

As shown in Fig. 1,when we use a 3-D continuously scanning range sensor, there are overlap regions in sequent range images, through which we assume that the sensor motion is represented by a unique sensor motion model. If a point with its object coordinate p_o is measured in two range images indexed by 1 and 2, the sensor coordinates are related to the world coordinates by:

$$p_o = G(t_1) \cdot p_{r1}$$
$$p_o = G(t_2) \cdot p_{r2}$$

where p_{r1} and p_{r2} are the sensor coordinates and t_1 and t_2 are the measured time of point p_o in two range images successively. By eliminating the object coordinate p_o, the constraint equation describing the correspondence of feature points can be written as:

$$G(t_1) \cdot p_{r1} - G(t_2) \cdot p_{r2} = 0 \tag{6}$$

Each pair of corresponding points can give a constraint equation. Theoretically, when the number of equations is more than the number of unknown variables in $G(t)$, we can have a solution of sensor motion. In the actual case, we usually have redundant equations. To find an optimal solution from those constraints, we construct a cost function.

Given K points $p_1, \cdots, p_K$ with measurements $(p_{r1,i}, t_{1,i}, p_{r2,i}, t_{2,i}), 1 < i < K$, when the motion parameter of twist ξ is represented by N-th order polynomial, the optional solution is given by minimizing the cost function defined by the sum of squared errors:

$$E(\xi) = \sum_{i=1}^{K} \| G(t_{1,i}) \cdot p_{r1,i} - G(t_{2,i}) \cdot p_{r2,i} \|^2 \tag{7}$$

with $G(t) = e^{\hat{\xi(t)}}$,where $\xi(t) = \xi_1 \cdot t + \xi_2 \cdot t^2 + \cdots + \xi_N \cdot t^N$.

In this cost function, there are $6N$ unknowns. Since each pair of feature points can give 3 constraint equations, theoretically, as long as $3K \geq 6N$, there would be a solution for the problem.

We minimize the cost function to find the optimal solution. Here we adopted Levenberg-Marquardt algorithm to solve it.

3.3 Rectification from Motion

Based on the estimated sensor trajectory $G(t)$ from Eq. 7, transformations from local sensor coordinates to world coordinates can be precisely deduced for each individual measurement point.

4. Feature Correspondence for Distortion by Sensor Motion

In our proposed sensor motion estimation algorithm, point correspondences have been assumed as prior conditions. Actually, due to the nature of distorted data from moving range sensor, it's not a easy job to obtain reliable corresponding relation. Deformation coming from relative motion between range sensor and objects, has its own properties. It usually includes large amount of missing data, due to occlusion or change of scene or view points. The shape might be deformed significantly between scans and the deformation doesn't keep the geodesic distance invariant.

To solve this problem, we propose a novel algorithm of extraction of 3D shape correspondences. The basic idea is utilizing Morse theory to extract topological information from critical points of a function. We design two types of Morse function for different conditions. We then find the maximal stable energy basins from the extracted topological tree by introducing energy landscape and disconnected graph here. To describe the features, we apply an affine invariant regularization to the extracted regions. We then attach a multiple-scale description to each critical point since there are multiple extracted regions supporting the point. To apply this algorithm to our problem that 3D shape reconstruction from data distorted by sensor motion, we assume that local data deformation satisfies affine transformations. Details will be described in latter sections.

4.1 Feature Detection

4.1.1 Morse Theory

In this sub-section, we give a brief review of basic definitions and concepts of Morse theory we will use here.

Definition of critical point: given a manifold $\mathcal{M}$, suppose $f : \mathcal{M} \to R$ is a smooth function. Then $x_0 \in \mathcal{M}$ is critical point of f if and only if $df|_{x_0} \in T^{\star}_{x_0}\mathcal{M}$. The value of $f(x_0)$ is called *critical value*.

A critical point is called *nondegenerate* if its Hessian is nondegenerate.

Definition of Morse function: a smooth function f is called a Morse function if its all critical points are nongenerate.

Morse Lemma: If p_0 is nondegenerate critical point of index λ of a smooth function $f : \mathcal{M} \to R$, then there exist local coordinates $(x_i)_{1 \leq i \leq m}$ near p_0 such that $x^i(p_0) = 0$, $\forall i$, and in these coordinates we have the equality

$$f = f(p_0) - \sum_{i=1}^{\lambda} (x^i)^2 + \sum_{j=\lambda+1}^{m} (x^j)^2 \tag{8}$$

where m is $dim(\mathcal{M})$.

It can be deduced from Morse Lemma that for a Morse function $f : \mathcal{M} \longrightarrow R$ with a critical point x_0, one can perturb f by composing with an isotopy of the manifold $\mathcal{M}$, that the new function has the same critical point x_0.

We assume that the deformation caused by sensor motion satisfies the perturbation mentioned above. In other words, we assume that critical points of Morse function on 3D mesh are kept after the distortion by sensor motion. Based on this assumption, we design a algorithm slicing the 3D mesh by critical values to extract feature regions which are invariant to the deformation.

In the case of 3D data scanned by moving range sensor, the coordinates of objects always change with the movement of sensor. Thus we need to choose appropriate Morse functions which are not affected by the definition of reference frame.

4.1.2 Implicit Polynomial Morse Function

One choice is to utilize implicit polynomial (IP) which fits the object surfaces.

Given a 3D data set, the 3D IP of degree n is defined as:

$$f_n(\mathbf{x}) = \sum_{0 \le i,j,k; i+j+k \le n} a_{ijk} x^i y^j z^k \tag{9}$$

where $\mathbf{x} = (x, y, z)$ is one point in the data set.

An IP fitting of 3D surface is to find an IP whose zero set $\{\mathbf{x} | f_n(\mathbf{x} = 0)\}$ can represent the given 3D data set best.

Ideally, it means:

$$f_n(\mathbf{x}) = \sum_{0 \le i,j,k; i+j+k \le n} a_{ijk} x^i y^j z^k = 0 \tag{10}$$

Technically, it can be considered as the IP which minimizes the square error:

$$\sum (f_n(\mathbf{x}))^2 \tag{11}$$

The degree n can be adaptively determined by adopting a fast adaptive fitting method developed by Zheng [24]. It achieves $O(Nk)$ complexity where N is the number of data points and k is the number of polynomial coefficients. Denote the found IP as $\tilde{f}_n(\cdot)$:

$$\sum (\tilde{f}_n(\mathbf{x}))^2 = \min_{f_n(\cdot)} \sum (f_n(\mathbf{x}))^2 \tag{12}$$

According to [24], the value of IP fitting won't be affected by the change of coordinate system. Another advantage is that IP is invariant to affine transformation. As shown in Fig. 2, left top figure is the original Stanford bunny model and the left bottom one is the model colored by the value of its IP function $\tilde{f}_n(\mathbf{x})$. The right top figure is the one distorted by an affine transformation and the right bottom is the one colored by its IP value. We can see that the corresponding parts on the two models share the same color distributions.

IP is a global fitting method where all points are involved in the calculation. It works well on the data sets which contain the same parts of objects. But in the case of large change of the observed targets, like view point change or occlusion, it will be affected a lot by the data change.

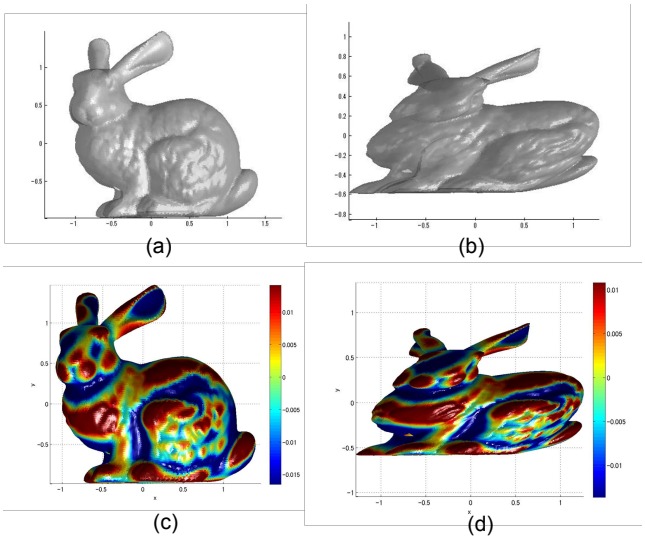

Fig. 2 Stanford bunny colored by IP values. (a) Original model. (b) The one distorted by an affine transformation. (c) Original model colored by its IP values. (d) Distorted model colored by its IP values.

4.1.3 Shape Difference of Beta stable Laplacian

To overcome the problem of IP in the case of data change, we design another Morse function based on Laplacian smoothing.

Laplacian smoothing is one of the common methods of mesh smoothing. The basic idea is moving the vertices of mesh incrementally in the direction of the Laplacian. One of the advantages of selecting Laplacian smoothing is that it uses relative positions of neighborhood vertices, which don't change with the reference frame. Another advantage is that the computational complexity of Laplacian smoothing is linear in time and space which is more practical on large meshes.

Different from the conventional Laplacian smoothing, we use the difference of the vertex and the one after smoothing:

$$I_k = \| L_{k+1}(x) - L_k(x) \| \cdot sign(n \cdot (L_{k+1}(x) - L_k(x))) \tag{13}$$

where k is the iteration times of Laplacian, n is the surface normal at point x.

Geometrically, this function describes the local shape of objects. It emphasize the parts where the shape changes fast. The more iterations of Laplacian done, the flatter the function would be, and more small changes on the shape are filtered. To determine an appropriate number of k, inspired by the concept of feature's stability in [25], we define a Beta-stable Laplacian function.

Let N_k be the norm of I_k:

$$N_k = \| I_k \| = \sum \| L_{k+1}(x_i) - L_k(x_i) \| \tag{14}$$

The variation speed δ_k of the Morse function at k is:

$$\delta_k = N_{k+1} - N_k \tag{15}$$

When δ_k is far away from zero, we say that the Morse function is not stable. A small change of k will lead to a significant change of N_k. From the point of view of geometric shape, it means a

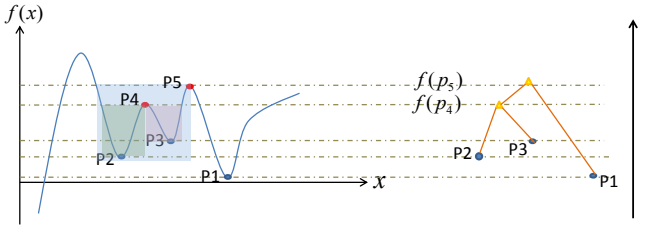

Fig. 4 Construct a disconnected graph from Morse function based on critical points.

small hump on the surface also has large Morse function value, which is undesired. In contrast, when $\delta_k \approx 0$, we say that I_k is stable.

I_k is β-stable if k is the smallest integer for which $\delta_\epsilon = 0$ for all $\epsilon \in [k, k+\beta]$.

Fig. 3 (a) shows the value of Laplacian function changing with the iteration times k. Fig. 3 (b) shows the variation speed δ_k versus k. It can be seen that the function gets stable when $k \geq 15$.

4.1.4 Maximal Stable Regions

According to Morse theory, the topology of a smooth manifold is very closely related to the critical points of the Morse function defined on the manifold. Given a smooth manifold M, sublevel set M^t is defined as:

$$M^t = \{x \in M; f(x) \leq t\} \tag{16}$$

The changes in the topology of M^t is an indicator of the presence of a critical point.

Our main interest in using this property is to slice the the manifold based on critical points.

3D mesh can be considered as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the vertices and $\mathcal{E}$ is referring to the undirected edges connecting the vertices.

First we define a connected component C in 3D mesh as a subgraph of $\mathcal{G}$, in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in $\mathcal{G}$.

Assume the boundary of the given Morse function $f(x)$ is $[a, b]$. When the value t gradually changes from a to b, the connectivities between local minimas in the sublevel set M^t change at critical values. As shown in Fig. 4, P_2 and P_3 is not connected until $t = f(p_4)$ and P_1 is not connected to P_2 and P_3 until $t = f(p_5)$.

We thus can determine the set of critical values $\{f^P\}$ for key point P by checking the change of connectivities of local minimas in M^t, like $f(P_4)$ and $f(P_5)$ in Fig. 4.

We use local minimas as key points. The support regions of local minima P, denoted as SR_P, is defined as the connected components containing P in the set of M^{f^P}. As shown in Fig. 4, the support regions of P_2 are the green and blue parts, and the support regions of P_3 are the pink and blue parts.

The change of connectivities between local minimas can be visualized using a disconnectivity graph [26]. This idea is from the field of physics used for visualizing potential energy. As shown

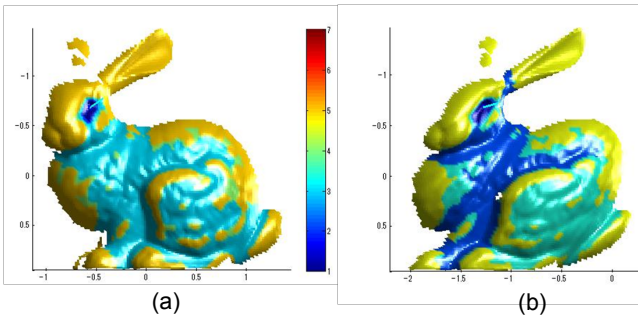

Fig. 5 Example of multi-scale feature. The key point is the one on the eye denoted by a blue arrow. (a) Support regions extracted from data scanned by static sensor. (b) Support regions extracted from data scanned by moving sensor. Different colors stand for the different scales.

in Fig. 4, all local minimas are leaves nodes represented by blue circles. The critical values f^P where connectivities change are branch nodes represented by yellow triangles. Actually the critical values f^P of key point P are all branch nodes on the path from leaf P to the root.

We adopt the depth first search algorithm to calculate the connected components in M^t. The detection of local minima can also be done at the same step. We only need to find the point with the minimal value in the newly appeared connected components while t changes.

4.1.5 Multi-scale Detection

The detail of how to extract the feature on 3D mesh is described in Algorithm 1. Geometrically the extracted regions look like basins with the bottom at key points. So we name it energy basin extraction algorithm. We only mentioned the support regions around local minimas when we describe the algorithm. Similarly, it only need to take the minus value of Morse function f to get the support regions around local maximas.

Fig. 5 shows an example of the multi-scale feature. The different colors stand for the support regions in different scales. Although they are not all the same in the two data sets because of the change of data, still they share the same region in the small scales. Thus we can find their similarity in different scales.

4.2 Feature Description

By far, we have extracted the multi-scale energy regions cut by critical points. The next step is to find an appropriate way to describe them.

Here we assume that the local deformations between the corresponding regions in range data from moving sensor approximately satisfy affine transformation, which is appropriate in normal moving vehicle. We thus apply an affine normalization to the extracted regions. Since there are multiple regions in different scales for each key point, we construct a multi-scale descriptor made by a set of a basic descriptor. Here we adopt the spin image to be the basic descriptor.

4.2.1 Affine Normalization

We use the shape normalization method based on the inertia

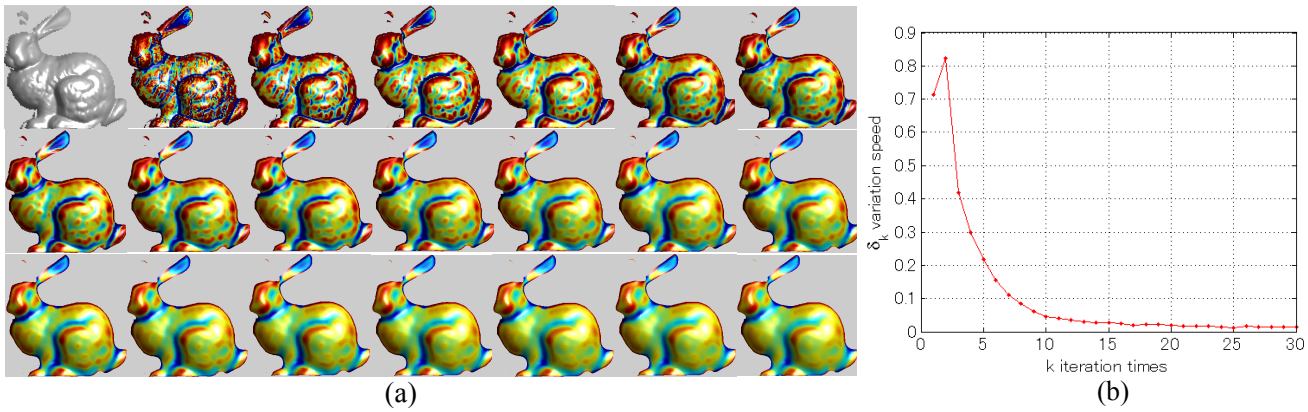

Fig. 3 (a) Difference of Laplacian on Stanford bunny model. (b) Shows the speed δ_k versus the iteration times k of Laplacian.

Algorithm 1: Energy Basin Extraction

Data: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$; $f(\mathcal{V}) \in [a, b]$
Result: a set of key points $\{P_i\}$ with corresponding multi-scale support regions SR_{P_i}

```
for t = a → b do
    Get the sublevel set G^t;
    Check connectivity of G^t and give component label C to V;
    Update the component label C_t^i for each key point P_i;
    Update the list of key points {P_i};
    if Connectivities between {P_i} change then
        Record t as a parent node for connected P_i;
        Update the disconnected graph (DG);
    end
end
for i = 1 → |P| do
    Get the path {t_j^i} from P_i to the root of DG;
    for j = 1 → |t^i| do
        Get support region SR_{P_i}^j: the connected component containing
        P_i in sublevel G^{t_j^i};
    end
end
```

matrix normalization presented by Cohignac [27].

The algorithm described in [27] is for 2D image data. Here we analogously expand it to 3D case.

Denote $\mathbb{I}_{\mathcal{F}}$ as the indicator function of a solid shape $\mathcal{F}$, assuming that $\mathcal{F}$ is previously translated so that its barycenter is at the origin of the 3D space. The moment of order (p, q, k) (p, q and k are natural integers) of $\mathcal{F}$ is defined by:

$$\mu_{p,q,k}(\mathcal{F}) = \int_{\mathbb{R}^3} x^p y^q z^k \mathbb{I}_{\mathcal{F}}(x, y, z) dx dy dz \tag{17}$$

The discrete form for 3D mesh $\mathcal{M}$ can be rewritten as:

$$\mu_{p,q,k}(\mathcal{M}) = \sum_{\mathbb{R}^3} x^p y^q z^k \tag{18}$$

where $(x, y, z) \in \mathcal{M}$

Let $\mathcal{S}_{\mathcal{F}}$ be the following 3×3 positive-definite, symmetric matrix:

$$\mathcal{S}_{\mathcal{F}} = \frac{1}{\mu_{0,0,0}} \begin{bmatrix} \mu_{2,0,0} & \mu_{1,1,0} & \mu_{1,0,1} \\ \mu_{1,1,0} & \mu_{0,2,0} & \mu_{0,1,1} \\ \mu_{1,0,1} & \mu_{0,1,1} & \mu_{0,0,2} \end{bmatrix} \tag{19}$$

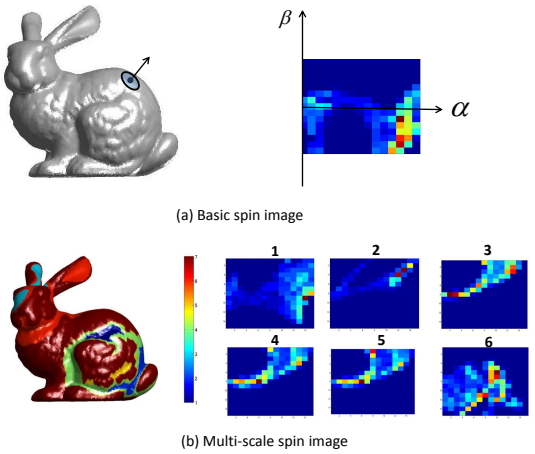

Fig. 6 (a) Basic spin image. (b) Multi-scale spin image.

where $\mu_{i,j,k} = \mu_{i,j,k}(\mathcal{M})$. According to the uniqueness of Cholesky factorization, the decomposition of $\mathcal{S}_{\mathcal{F}}$: $\mathcal{S}_{\mathcal{F}} = B_{\mathcal{F}} B_{\mathcal{F}}^T$ may be unique, where $B_{\mathcal{F}}$ is a lower-triangular real matrix with positive diagonal entries.

Let $\mathcal{A}$ be a non-singular 3×3 matrix. The normalized shape associated to $\mathcal{F}$ is the shape $\mathcal{F}' = B_{\mathcal{F}}^{-1}(\mathcal{F})$. It can be proved that the normalized shape $\mathcal{F}'$ is invariant to affine transformation $\mathcal{A}$, up to a rotation $\mathcal{Q}$. (See details in [27])

4.2.2 Description of Normalized Regions

Since our normalized shapes have rotation change varying with the coordinate system, invariance to rotation should be a desired property of the feature descriptor. Spin image is chosen as the basic descriptor for its rotation invariance and fast computation. We propose an variation algorithm of spin image.

First we simply review the original spin image algorithm proposed by Andrew E. Johnson [28]. A spin image is created for an oriented vertex on 3D mesh surface. First a partial, object-centered coordinate system is defined with respect to an oriented point: the radial coordinate α, defined as the distance to the surface normal at the point, and the elevation coordinate β, defined as the signed distance to the tangent plane at the point. A 2-D accumulator indexed by α and β is incremented in the surface

mesh which is within the support region of this vertex, as shown in Fig. 6 (a).

To use spin image in our algorithm, for each support region SR_P^j belonging to the key point P, a spin image is calculated, as shown in Fig. 6 (b). Since the size of each support region SR_P^j is different, the coordinates range of α and β of the spin image is set as the maximal distance along each coordinate axis from the key point to the furthest point in the region. The resolution of bins in the accumulator is set the same for all regions.

Given the spin image SI_j for each support region SR_P^j, We construct our descriptor of the key point P as:

$$D(P) = \{SI_1, SI_2, \cdots, SI_N\} \tag{20}$$

where $N = |SR_P|$ is the cardinality of the set $\{SR_P\}$.

The similarity of two key points P and Q is defined as the minimal pairwise distance of spin images between their supporters:

$$d(P,Q) = \min_{SI_i \in \{D(P)\}, SI_j \in \{D(P)\}} \| SI_i - SI_j \| \tag{21}$$

This descriptor compares features in different scales and is more robust to occlusions and the change of view points.

4.3 Matching Criteria and Evaluation

Consider two 3D meshes $\mathcal{M}$, $\mathcal{N}$ to be matched. Let $K^{\mathcal{M}}$ and $K^{\mathcal{N}}$ be the detected key points set of $\mathcal{M}$ and $\mathcal{N}$ respectively. As described in Eq. 21, let $d(p,q)$ be the similarity of two key points p and q.

We say key point $p \in K^{\mathcal{M}}$ is matched to q_1:

$$q_1 = \arg\min_{q \in K^{\mathcal{N}}} d(p,q) \tag{22}$$

$$\text{if } \frac{d(p,q_2)}{d(p,q_1)} > 1.5, \text{ where } q_2 = \arg\min_{q \in K^{\mathcal{N}} \setminus q_1} d(p,q)$$

It means if the best match is at least 50% better than the second best match for p, then p has a matched key point.

To evaluate the performance of the proposed algorithm, we use *accuracy* and *repeatability* [29]:

$$\text{Accuracy} = \frac{\text{correct matches found in } \mathcal{M} \text{ and } \mathcal{N}}{\text{total matches found in } \mathcal{M} \text{ and } \mathcal{N}} \tag{23}$$

$$\text{Absolute Repeatability} = \text{repeatable key points} \tag{24}$$

$$\text{Relative Repeatability} = \frac{\text{repeatable key points}}{\min\{|K^{\mathcal{M}}|, |K^{\mathcal{N}}|\}} \tag{25}$$

A key point $p \in K^{\mathcal{M}}$ is said to be *repeatable* if the distance from its nearest neighbor, a key point $q \in K^{\mathcal{N}}$, after transformed according to the ground truth transformation matrix G_T, is less than a threshold ϵ:

$$\| p - G_T q \| \leq \epsilon \tag{26}$$

5. Experiments

5.1 Evaluation on Rectification

To verify and evaluate our proposed algorithm, we test it with simulated data which can afford ground truth. To show the robustness under different conditions, like the complexity of sensor motion and the shape of objects, we build a data set which contains the regular motion modes and typical targets in applications such as digitalization of culture heritages, autonomous vehicle.

Table 1 Virtual sensor specification

Vertical range	30°
Horizontal range	90°
Angular resolution	1/16°
fps	0.5
Scanning order	Raster

5.1.1 Setting up

The characteristic of virtual range sensor is set up as Table 1. We generate the range data by simulating the process that a straight line hits on the object surface in raster order with a given time sequence and given angular resolution. Thus we can obtain measured data, scanning time and corresponding ground truth.

We don't consider the feature detection procedure here and just make an assumption that the correspondence of feature points in different frames is known and established by methods like manual operation or feature matching. Another assumption is that all feature points are taken from static rigid objects. In our experiments, we generate feature points in a manual way, randomly choosing points from the overlapped areas of two successive frames.

5.1.2 Evaluation Methods

To evaluate the rectification performance, we compute the point distances between the rectified points $\{\mathbf{R}\}_1^N$ to ground truth $\{\mathbf{S}\}_1^N$. Here we define the absolute rectification error as:

$$e_i = \| \mathbf{R}_i - \mathbf{S}_i \| \tag{27}$$

The absolute error is affected by point resolution, which is related to sensor specifications and distances between sensors and observed objects. In order to evaluate the algorithm despite of those factors, we define the normalized rectification error as:

$$E_i = \frac{e_i}{d_p} \tag{28}$$

where $d_p = \frac{1}{K}\sum_{i=1}^{K} d_i$ is the mean point-wise distance of the object, as shown in Fig.7.

5.1.3 Tests on Common Motion Modes

To evaluate our algorithm in different types of sensor motions, we test it under 5 general motion modes as shown in Fig.8. As shown in the first column of Fig.9, measurements are distorted by sensor motion. The second column of Fig.9 shows our rectified results. Table 2 shows the parameters of simulated motion modes in the data set.

5.1.4 Tests on Different Target Sets

We verify the proposed algorithm with different targets, as shown in Fig.10. As Fig.10 shows, as long as enough corresponding point are sampled, the proposed algorithm works on different targets.

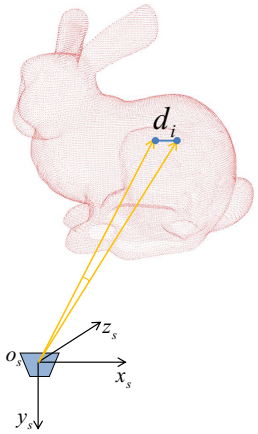

Fig. 7 Point-wise distance.

Table 2 Regular motion modes

Motion description
1.Constant velocity
2.Constant acceleration
3.Pure rotation
4.Combination of 2 and 3
5.Uniform circular motion

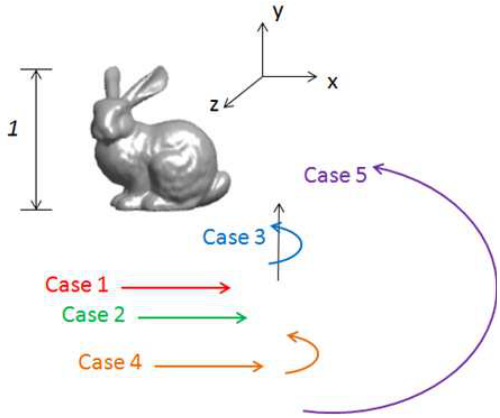

Fig. 8 Simulated motion modes

5.1.5 Robustness to Correspondence Noise

To verify the robustness of the proposed algorithm, we add noise to the correspondence constraints. The unit of noise is mesh resolution. As shown in Fig.12, the rectification error has a linear relation with correspondence noise.

5.1.6 Discussion on Optimization

In the experiment, we notice that for a given polynomial motion $G(t)_N$, the initial guess of the parameters in the Levenberg-Marquardt optimization algorithm has large effect on the result and iteration time. Empirically, we found that use the estimation result of low order polynomial motion $G(t)_{i-1}$ as the initial guess of $G(t)_i$ can give better result than directly estimating $G(t)_i$. In most cases we can achieve an acceptable solution by starting this iterative procedure from $G(t)_1$ with the initial guess $\xi_1 = 0$.

5.2 Evaluation on Feature Correspondence

To verify and evaluate the proposed 3D corresponding extraction algorithm, we use models from the Stanford 3D Scanning Repository [30] and AIM@Shape Shape Repository [31]. First test is extracting correspondence between original model and the one distorted by affine transformations. The second test is between the static data and the synthesized data gathered by virtual moving sensor.

Fig. 13 and Fig. 14 are examples of extracted feature regions

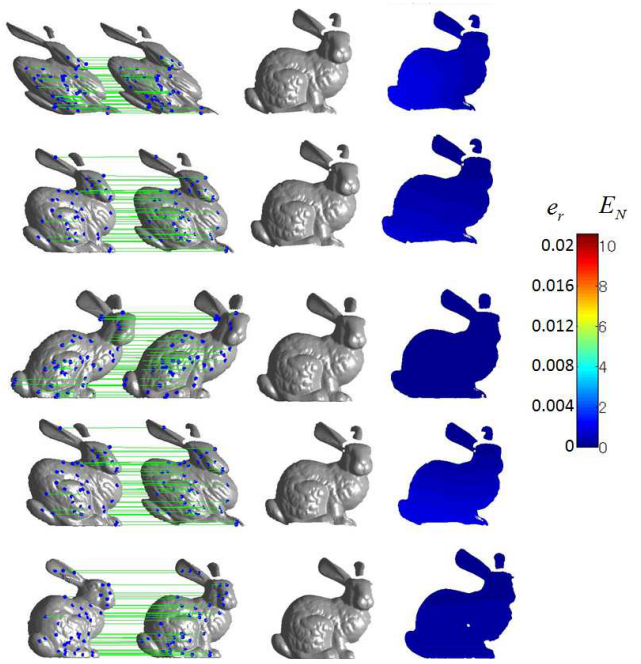

Fig. 9 Simulation experiment on Stanford bunny model under 5 common motion modes. The first column: distorted data and correspondence between consecutive range image. The second column: rectified data using our method. The third column: rectification error.

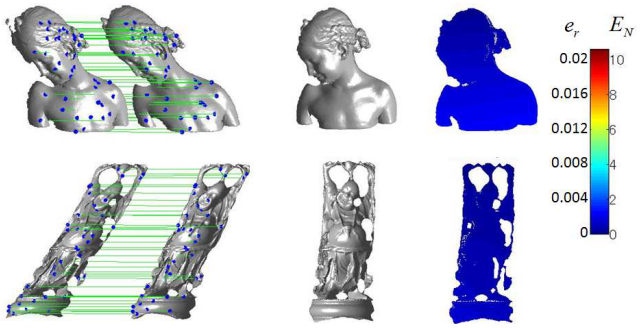

Fig. 10 Simulation experiment with other objects under constant velocity motion. The first column: distorted data and correspondence between consecutive range image. The second column: rectified data using our method. The third column: rectification error.

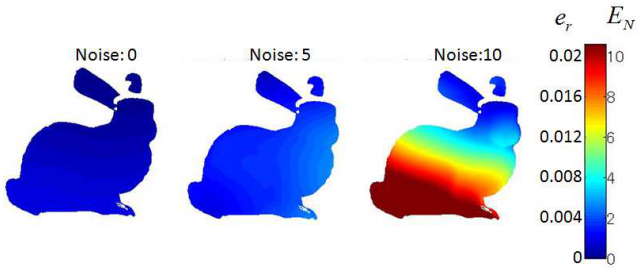

Fig. 11 Rectification error with correspondence noise

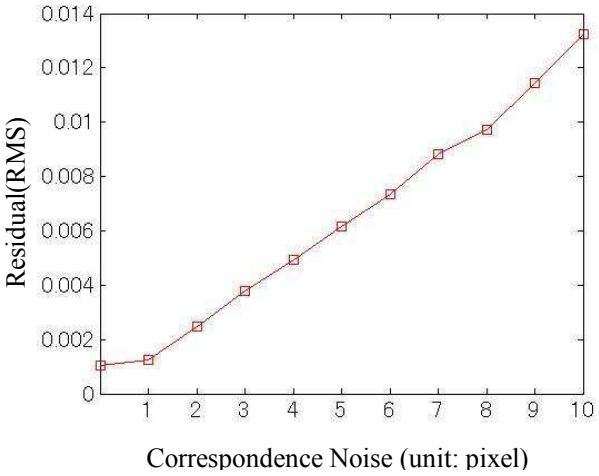

Fig. 12 Rectification error with respect to correspondence noise.

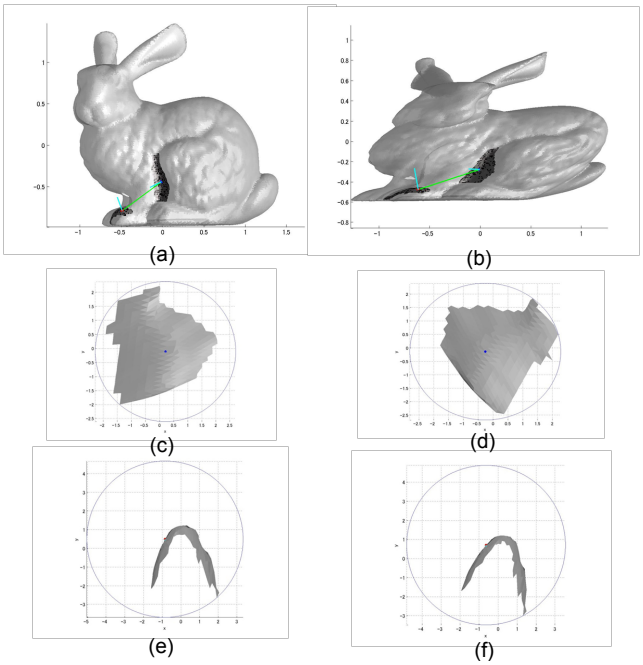

Fig. 13 Example of extracted feature regions and their affine normalizations. (a) Two key points with their support regions extracted from the original data. (b) Corresponding key points with their support regions extracted from the data distorted by affine transformations. (c) and (e) are the normalized regions from the original data. (d) and (f) are the corresponding normalized region from distorted data.

and their affine normalizations. To show the performance of affine normalization, only a single-scale corresponding region is shown for one key point. As shown in the sub figures (a) and (b), the exact corresponding region is extracted from the distorted data. After the affine normalization, the normalized regions have the same shapes, up to a rotation.

5.2.1 Evaluation on Affine Distortion

To verify the performance of proposed algorithm to affine distortion, a series set of affine transformations are applied to original object models.

As shown in Fig. 15 and Fig. 16, we apply different affine transformation to the bunny model and compare the extracted feature

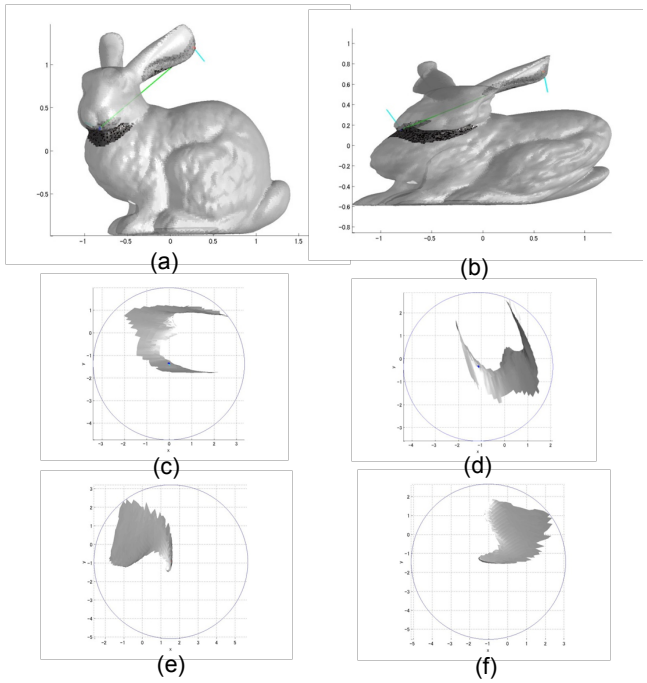

Fig. 14 Another example of extracted feature regions and their normalizations. (a) Two key points with their support regions extracted from the original data. (b) Corresponding key points with their support regions extracted from the data distorted by affine transformations. (c) and (e) are the normalized regions from the original data. (d) and (f) are the corresponding normalized region from distorted data.

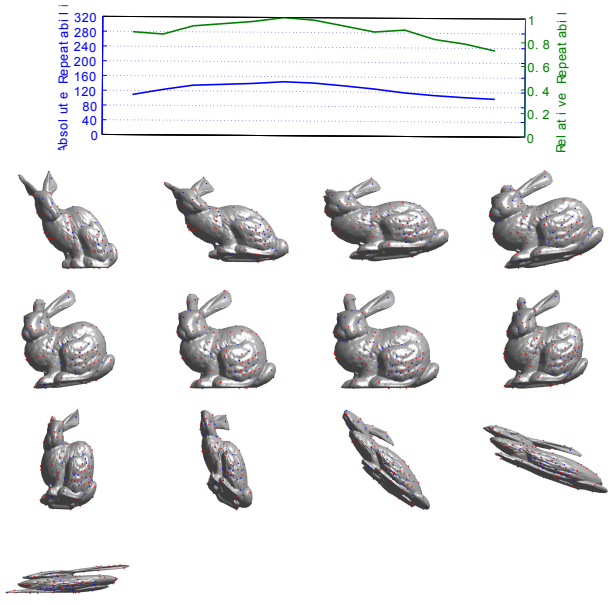

Fig. 15 Repeatability curve of Stanford bunny model applied a set of affine distortion, using IP Morse function.

points with the original model. The results show that although beta-stable Laplacian Morse function can detect feature points twice than IP Morse function, IP is more stable on different distortions.

We apply the same set of affine transformations to other models. The repeatability curves using IP Morse function and beta-stable Morse function are shown in Fig. 17 and Fig. 18, respec-

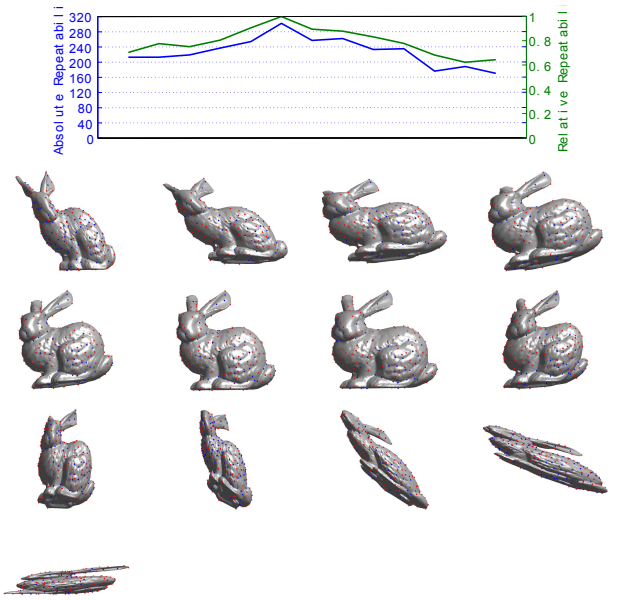

Fig. 16 Repeatability curve of Stanford bunny model applied a set of affine distortion, using beta-stable Laplacian Morse function.

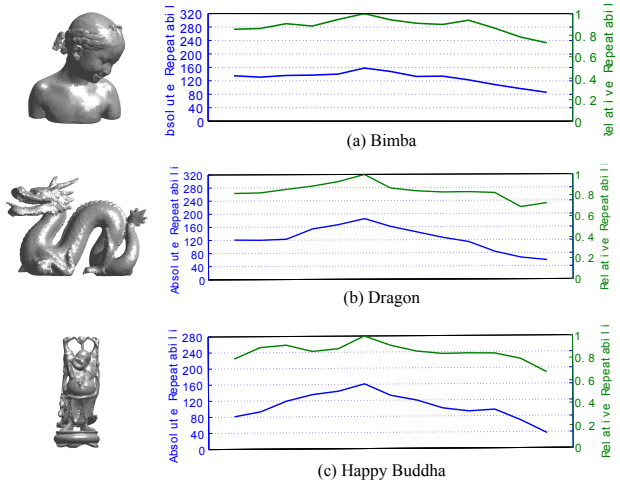

Fig. 17 Repeatability curve of other models applied the same set of affine distortion, using IP Morse function.

tively. The results show that IP does have a more stable performance than Laplacian in the affine distortion. Fig. 19 (a) and (b) show the matching accuracy of using IP and Laplacian, respectively. Fig. 20 and Fig. 20 show the matching result of using IP and Laplacian on different models with the same affine transformation, respectively.

5.2.2 Evaluation on Synthetic Data from Moving Range Sensor

Fig. 22 and Fig. 23 are the matching results between the synthetic static data and distorted data from moving sensor with different velocities, using beta-stable Laplacian Morse function.

6. Applications

6.1 Application in 3D Reconstruction of Indoor Environment

The problem of fast 3D modeling of indoor environment at-

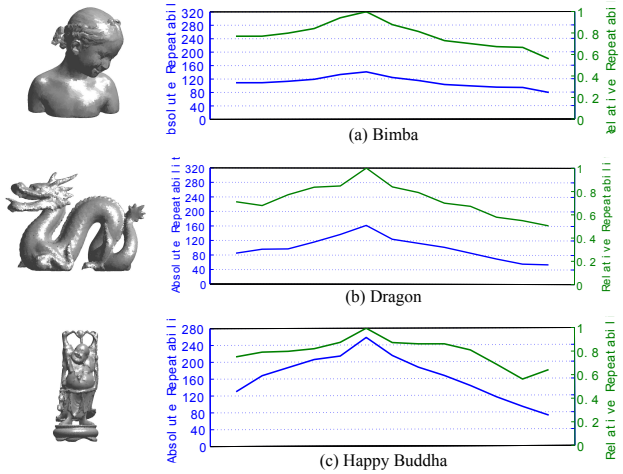

Fig. 18 Repeatability curve of other models applied the same set of affine distortion, using beta-stable Laplacian Morse function.

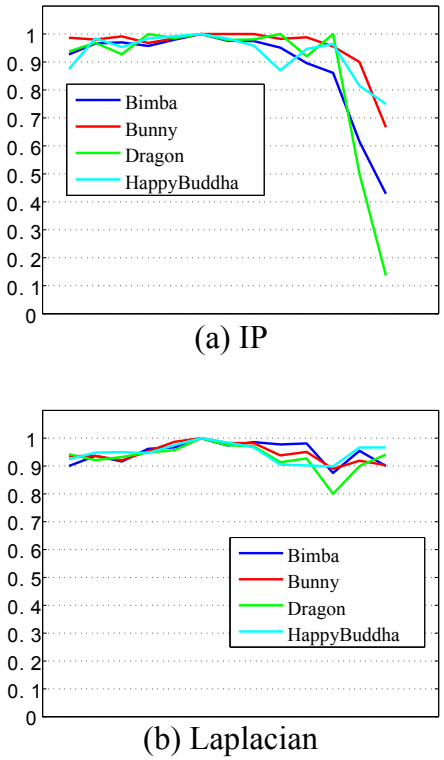

Fig. 19 (a) Match accuracy using IP Morse function. (b) Match accuracy using beta-stable Laplacian Morse function.

tacks a lot of attention because of its wide applications to intelligent building, since 3D models can provide richer information about the environment. The state-of-the-art technologies of 3D modeling usually involve devices like camera, laser scanner, GPS with mobile platform. Laser scanner based systems have better accuracy than camera-based systems. Since high accurate localization device like GPS cannot receive satellite signals in indoor environment, information fusion with other devices like camera, IMU and encoder is a common solution.

Here we apply our proposed algorithm to build 3D model of an office environment, as shown in Fig.24. The purpose of this experiment is to verify that our algorithm can collect and self-

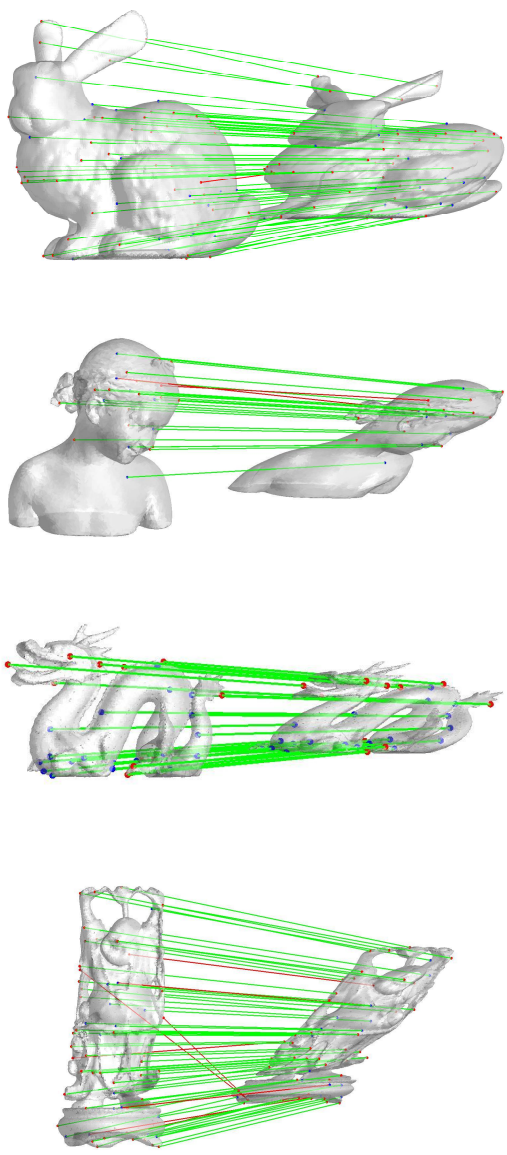

Fig. 20 One group of matching result of the ones using IP Morse function.

Table 3 Specification of IHI 3D laser range sensor

Model Number	IHI
Light source	JIS C 6802 Class1
Vertical field of view	60°
Horizontal field of view	90°
Measurable distance	0.1[m] ~ 200[m]
vertical angular resolution	0.01°
horizontal angular resolution	0.15°
Frame rate	0.3 ~ 10Hz

calibrate data with moving platform using a single 3D laser range scanner. Such system has lower cost and works more efficiently.

6.1.1 Setting up

As shown in Fig.25, the laser range scanner is mounted on a cart. In order to cover more upper space, we set the scanning central axis a small upward angle with the horizontal plane. Details of sensor specification is in Table 3.

In the experiment, we mannually move the platform in the of-

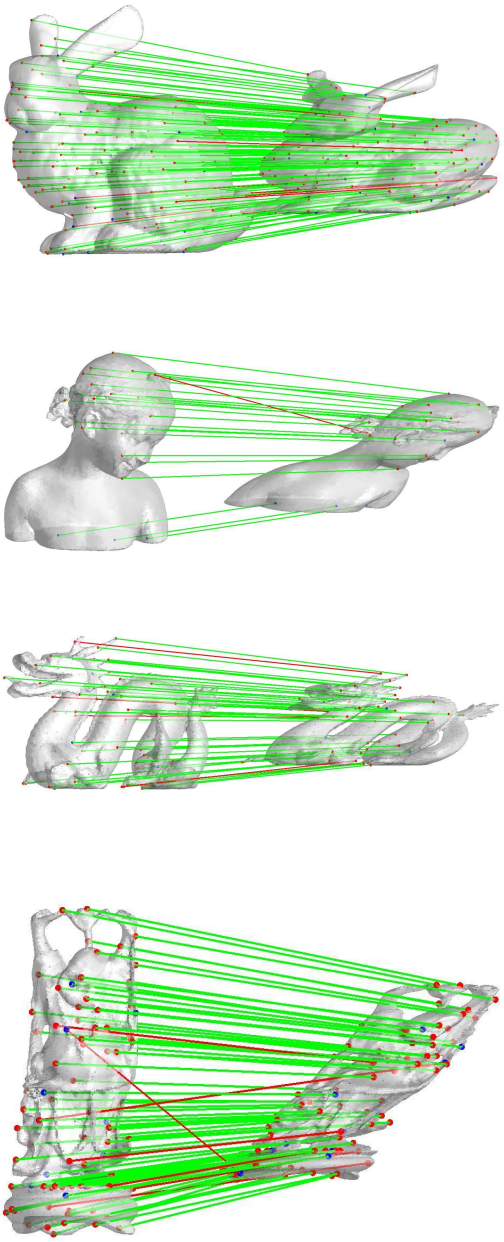

Fig. 21 One group of matching result of the ones using beta-stable Laplacian Morse function.

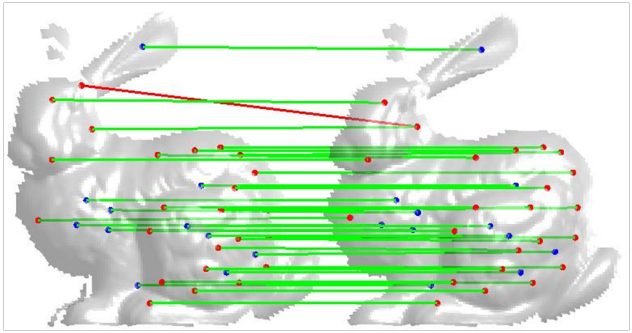

Fig. 22 Correspondence between synthetic static data and constant velocity (C_1) data using our proposed algorithm.

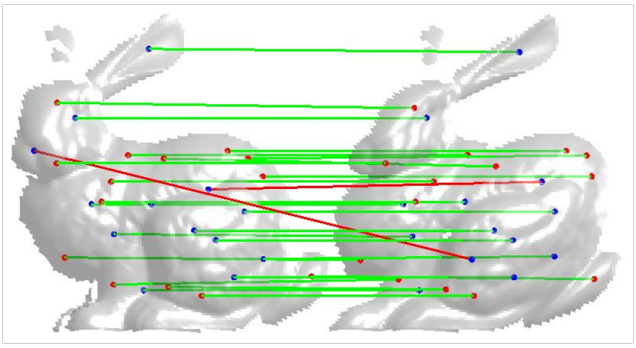

Fig. 23 Correspondence between synthetic static data and constant velocity (C_2) data using our proposed algorithm.

Fig. 24 Scene of the indoor experiment.

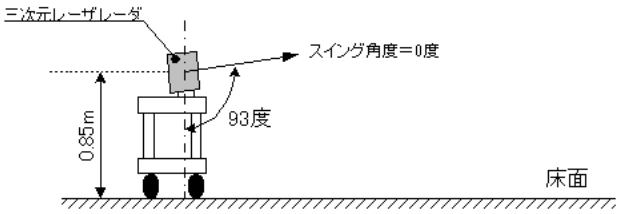

Fig. 25 Setting up of range sensor in the indoor experiment.

fice to scan it in one round. Examples of data distortion caused by sensor movement are shown in Fig.26.

6.1.2 Results and Discussion

In order to evaluate our algorithm without the effect of performance of corresponding extraction step, we manually pick a set of corresponding points, as the green lines shown in Fig.27. Fig.28 shows the comparison between distorted data and rectified data by our algorithm.

One way to evaluate the data rectification is to check how well the point clouds can be aligned with ground truth, which is the point clouds $\{\mathbf{S}\}_1^N$ obtained by static range sensor. To align two point clouds, we adopt iterative closest point (ICP) method [15] to find the best alignment transformation, yeilding aligned point clouds $\{\mathbf{X}\}_1^M$ for raw distorted data, and $\{\mathbf{Y}\}_1^K$ for rectified data.

Now we can compute closest point distances for all points in $\{\mathbf{X}\}_1^M$ to $\{\mathbf{S}\}_1^N$:

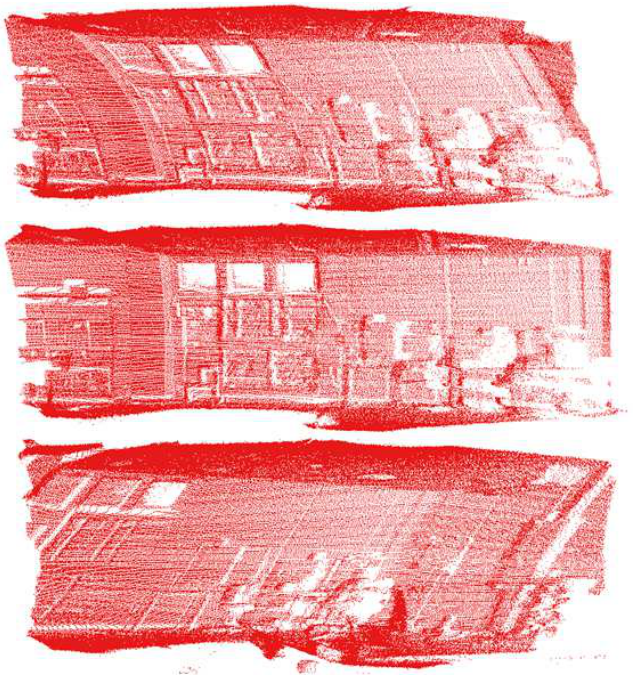

Fig. 26 Distorted data of the indoor scene.

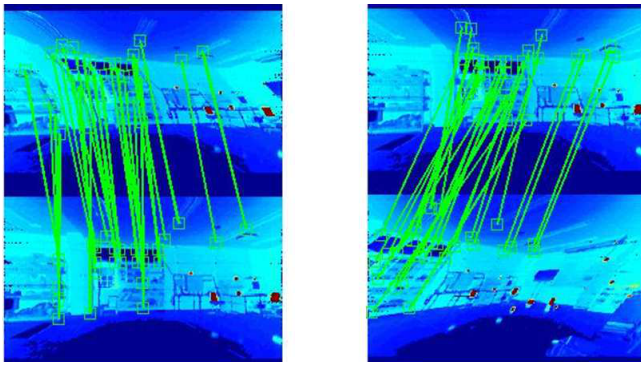

Fig. 27 Manual correspondence of distorted data from the indoor scene.

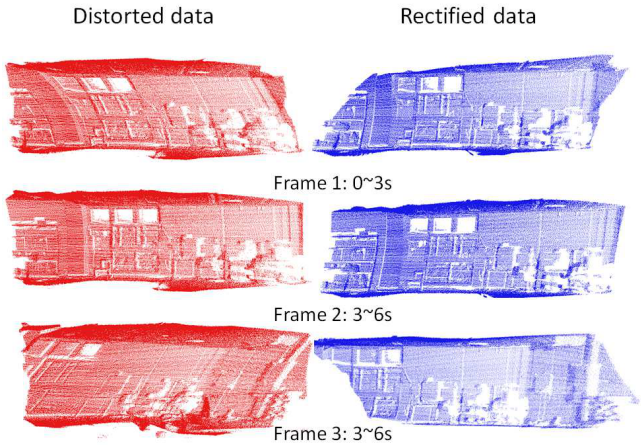

Fig. 28 Rectification result of the indoor scene.

$$d_m = d(\mathbf{X}_m, \{\mathbf{S}\}_1^N) = \min_{n\in[1,N]} \| \mathbf{X}_m - \mathbf{S}_n \| \qquad (29)$$

Similarly, we can calculate closest point distance for point in rectified data set to the ground truth:

$$d_k = d(\mathbf{Y}_k, \{\mathbf{S}\}_1^N) = \min_{n\in[1,N]} \| \mathbf{Y}_k - \mathbf{S}_n \| \qquad (30)$$

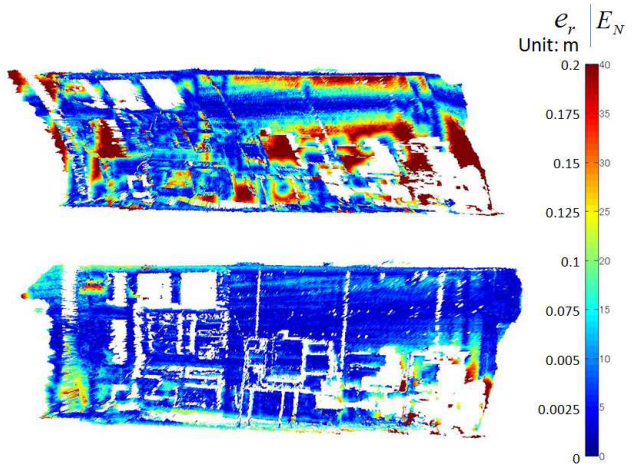

Fig. 29 Rectification error of the indoor scene.

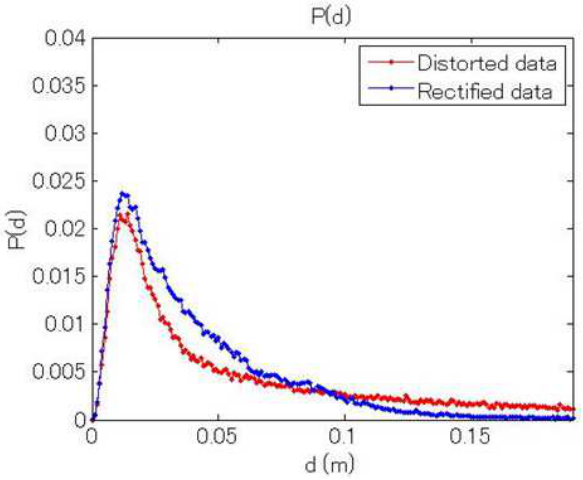

Fig. 30 Comparison of error distribution curves between distorted data and rectified data of the indoor scene.

d_m and d_k are mapped as colors in Fig.29. Point distances increase from blue to red. Using these distances, we can obtain two probability density error curves based on kernel density estimation (KDE) [32]. As shown in Fig.30, $p(d)$ of rectified data has more data with small error.

Another way to evaluate the point cloud rectification, is utilizing prior knowledge of indoor environment, that angles between walls, floor and ceiling should be 90 degree, as shown in Fig.31. We estimate plane normals of the subset of point clouds which only contain the wall,floor and ceiling parts using RANSAC. We then calculate the angle between two normals:

$$\theta_{i,j} = \arccos \hat{n}_i \cdot \hat{n}_j, \tag{31}$$

where $\hat{n}_i$ and $\hat{n}_j$ are normals for the two planes.

As shown in Table 4, the angles after rectification are much closer to the ground truth.

6.2 Application in Large Scale Culture Heritage Digitization

6.2.1 Bayon Project

One of the most important and comprehensive applications of

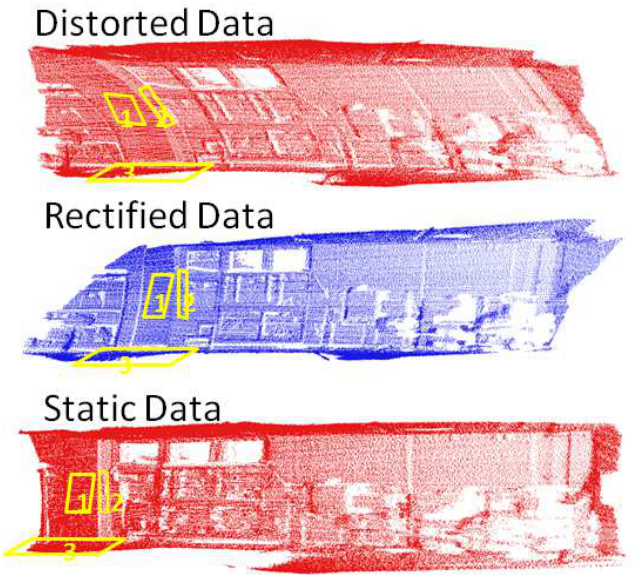

Fig. 31 Comparison of plane angles between distorted data and rectified data of the indoor scene.

Table 4 Evaluation based on plane angles.

	$\theta_{1,2}$	$\theta_{1,3}$	$\theta_{2,3}$
Distorted data	87.0°	78.4°	52.2°
Rectified data	86.1°	84.4°	87.3°
Static data	88.9°	89.9°	88.8°

3D technologies is modeling cultural heritage objects. It has great significance in many aspects. Modeling technologies can provide digital archive of object shapes of culture heritage. Digital data enables us to restore the original shapes of the heritage objects, even if some unfortunate disasters destroy them, like natural weathering, fire, or wars. Moreover, researchers can do further analysis on digital data applying modern computer vision and graphics technologies. In addition, normal people can get access to digital culture heritages through the Internet from any corner around the world.

In order to preserve and study one important UNESCO World Heritage, the Bayon temple, which is located in the center of Angkor Thom in Cambodia, the Bayon Digital Archive Project [33] is started.

6.2.2 Flying Laser Range Sensor

Several novel sensors are developed to observe this huge ($160m \times 140m \times 45m$) and complex structure. One of them is flying laser range sensor (FLRS), as shown in Fig.32. Using a balloon platform instead of previous helicopter platform, FLRS is safer and more flexible to measure large culture heritages in outdoor condition.

6.2.3 Results and Discussion

This system is certainly free from high frequency vibration such as that of a helicopter engine. However, the obtained range data are still distorted because the sensor itself is moving during the scanning processes, as shown in Fig.33.

We apply our method to the distorted FLRS data using manual

Fig. 32 The FLRS and the Bayon Temple

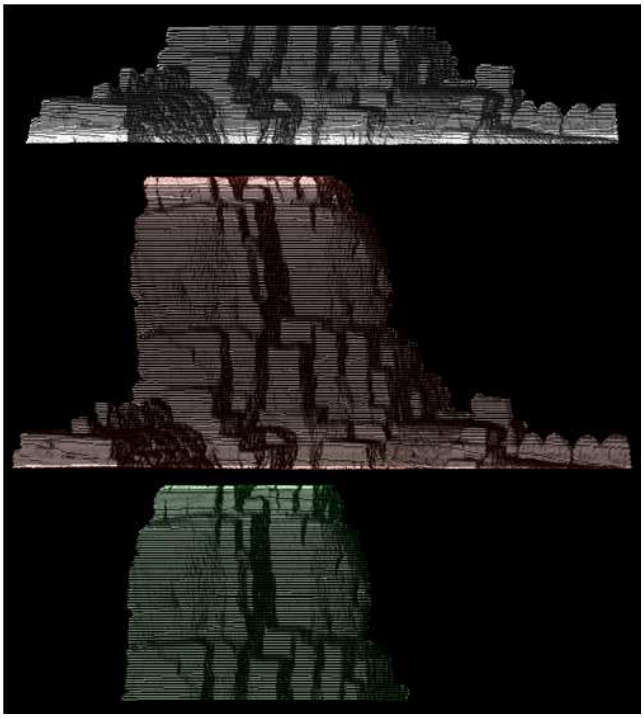

Fig. 33 Distorted data of Bayon temple gathered by flying laser range sensor.

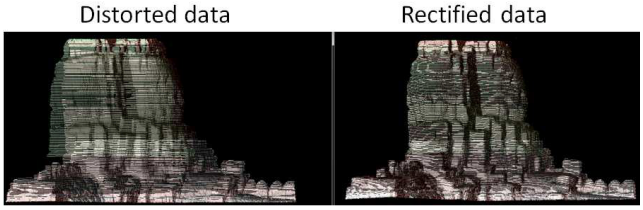

Fig. 34 Rectified data of Bayon temple.

correspondences. Rectified data is shown in Fig.34.

7. Conclusion

We propose a feature based polynomial fitting method to estimate 6 DOF motion parameters of moving range sensor. The 3D model can be reconstructed according to a highly accurate sensor motion estimation. When using a 2D scanning sensor instead of a 1D scanning sensor, the same region is measured in

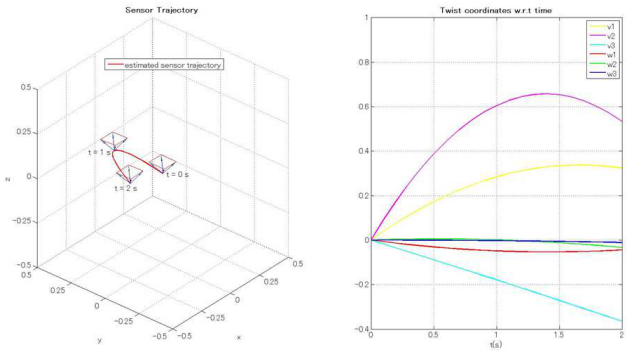

Fig. 35 Estimated trajectory of flying laser range sensor.

multiple times when the sensor moves. We show that we can reconstruct the sensor motion and the scene from only the measured coordinates and times of the same set of points. This reconstruction is intrinsic, which relies on only the intrinsic properties of the distortion, and not relying on the extrinsic information from other sensors. Firstly, for robust estimating sensor movement, we model the sensor motion using polynomial with respect to time. Secondly, to estimate the parameters of sensor motion model, we utilize the 3D corresponding points extracted from the overlapped parts between consecutive frames. This method doesn't need the secondary sensor and is not limited with specific environment features. Without linearization of constraint and discretization of trajectory, distorted data is accurately rectified.

To obtain the corresponding constraints, we propose a novel 3D affine invariant feature detection and matching method which is designed for the deformed 3D data collected by moving range sensor. The basic idea is utilizing Morse theory to extract topological information from information about critical points of a function. We design two types of Morse function for different conditions. The one based on implicit polynomial fitting of 3D mesh is more robust to data noise and suitable for sparse data set. The other one based on difference of Laplacian has the advantages to the problem of data changes, like occlusion or the change of view point and scene. We then find the maximal stable energy basins from the extracted topological tree by introducing energy landscape and disconnected graph here. To describe the features, we apply an affine invariant normalization to the extracted regions. We then attach a multiple-scale description based on spin image to each critical point thus their similarities are compared in different scales. The simulation experiments prove the verification and robustness of the proposed algorithm, especially for the distorted data gathered from moving sensor.

References

[1] Banno, A., Masuda, T., Oishi, T. and Ikeuchi, K.: Flying Laser Range Sensor for Large-Scale Site-Modeling and Its Applications in Bayon Digital Archival Project, *International Journal of Computer Vision*, Vol. 78, pp. 207–222 (online), available from ⟨http://dx.doi.org/10.1007/s11263-007-0104-6⟩ (2008). 10.1007/s11263-007-0104-6.
1

[2] Birk, A., Vaskevicius, N., Pathak, K., Schwertfeger, S., Poppinga, J.

and Buelow, H.: 3-D perception and modeling, *Robotics Automation Magazine, IEEE*, Vol. 16, No. 4, pp. 53 –60 (2009).
1

[3] Yano, H., Miyamoto, Y. and Iwata, H.: Haptic interface for perceiving remote object using a laser range finder, *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pp. 196 –201 (2009).
1

[4] Rana, M., Setan, H., Majid, Z. and Chong, A.: Computer Assisted Surgical Planner for Craniofacial Reconstruction - Imaging Techniques, *Geometric Modeling and Imaging–New Trends, 2006*, pp. 215 –220 (2006).
1

[5] Kagami, S., Hanai, R., Hatao, N. and Inaba, M.: Outdoor 3D map generation based on planar feature for autonomous vehicle navigation in urban environment, *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 1526 –1531 (2010).
1

[6] Ryde, J. and Hu, H.: 3D mapping with multi-resolution occupied voxel lists, *Autonomous Robots*, Vol. 28, pp. 169–185 (online), available from ⟨http://dx.doi.org/10.1007/s10514-009-9158-3⟩ (2010).
10.1007/s10514-009-9158-3.
1

[7] Jeong, Y., Bok, Y., Kim, J.-S. and Kweon, I.-S.: Complementation of cameras and lasers for accurate 6D SLAM: From correspondences to bundle adjustment, *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3581 –3588 (2011).
1

[8] Abuhadrous, I., Ammoun, S., Nashashibi, F., Goulette, F. and Laurgeau, C.: Digitizing and 3D modeling of urban environments and roads using vehicle-borne laser scanner system, *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, Vol. 1, pp. 76 – 81 vol.1 (2004).
1

[9] Banno, A. and Ikeuchi, K.: Determination of motion parameters of a moving range sensor approximated by polynomials for rectification of distorted 3D data, *Machine Vision and Applications*, Vol. 22, pp. 889–897 (online), available from ⟨http://dx.doi.org/10.1007/s00138-010-0318-4⟩ (2011).
10.1007/s00138-010-0318-4.
1

[10] Harrison, A. and Newman, P.: High quality 3D laser ranging under general vehicle motion, *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 7 –12 (2008).
2

[11] Bosse, M. and Zlot, R.: Continuous 3D scan-matching with a spinning 2D laser, *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 4312 –4319 (2009).
2

[12] Ringaby, E. and Forssén, P.-E.: Scan Rectification for Structured Light Range Sensors with Rolling Shutters, *IEEE International Conference on Computer Vision*, Barcelona, Spain, IEEE, IEEE Computer Society (2011).
2

[13] Baker, S., Bennett, E., Kang, S. B. and Szeliski, R.: Removing rolling shutter wobble, *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 2392 –2399 (online), DOI: 10.1109/CVPR.2010.5539932 (2010).
2

[14] Forssen, P. and Ringaby, E.: Rectifying rolling shutter video from hand-held devices, *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 507 –514 (online), DOI: 10.1109/CVPR.2010.5540173 (2010).
2

[15] Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces, *Int. J. Comput. Vision*, Vol. 13, No. 2, pp. 119–152 (online), DOI: 10.1007/BF01427149 (1994).
2, 12

[16] Rusinkiewicz, S. and Levoy, M.: Efficient variants of the ICP algorithm, *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145 –152 (online), DOI: 10.1109/IM.2001.924423 (2001).
2

[17] Chui, H. and Rangarajan, A.: A new point matching algorithm for non-rigid registration, *Computer Vision and Image Understanding*, Vol. 89, No. 2a?g3, pp. 114 – 141 (online), DOI: 10.1016/S1077-3142(03)00009-2 (2003).
¡ce:title¿Nonrigid Image Registration¡/ce:title¿.
2

[18] Myronenko, A., Song, X. and Carreira-Perpinan, M.: Free-Form Nonrigid Image Registration Using Generalized Elastic Nets, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1 – 8 (2007).
2

[19] Zeng, Y., Wang, C., Wang, Y., Gu, X., Samaras, D. and Paragios, N.: Dense non-rigid surface registration using high-order graph matching, *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 382 –389 (online), DOI: 10.1109/CVPR.2010.5540189 (2010).
2

[20] Fujiwara, K., Nishino, K., Takamatsu, J., Zheng, B. and Ikeuchi, K.: Locally rigid globally non-rigid surface registration, *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 1527 –1534 (online), DOI: 10.1109/ICCV.2011.6126411 (2011).
2

[21] Lipman, Y. and Funkhouser, T.: Mobius Voting for Surface Correspondence, *ACM Transactions on Graphics (Proc. SIGGRAPH)*, Vol. 28, No. 3 (2009).
2

[22] Wu, H.-Y., Zha, H., Luo, T., Wang, X.-L. and Ma, S.: Global and local isometry-invariant descriptor for 3D shape comparison and partial matching, *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 438 –445 (online), DOI: 10.1109/CVPR.2010.5540180 (2010).
2

[23] Murray, R. M., Sastry, S. S. and Zexiang, L.: *A Mathematical Introduction to Robotic Manipulation*, chapter 2, CRC Press, Inc., 1st edition (1994).
2

[24] Zheng, B., Takamatsu, J. and Ikeuchi, K.: An Adaptive and Stable Method for Fitting Implicit Polynomial Curves and Surfaces, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, Vol. 32, No. 3, pp. 561 –568 (online), DOI: 10.1109/TPAMI.2009.189 (2010).
4

[25] Gu, S., Zheng, Y. and Tomasi, C.: Critical nets and beta-stable features for image matching, *Proceedings of the 11th European conference on computer vision conference on Computer vision: Part III*, ECCV'10, Berlin, Heidelberg, Springer-Verlag, pp. 663–676 (online), available from ⟨http://dl.acm.org/citation.cfm?id=1927006.1927058⟩ (2010).
4

[26] Doye, J. P. K., Miller, M. A. and Wales, D. J.: Evolution of the potential energy surface with size for Lennard-Jones clusters, *The Journal of Chemical Physics*, Vol. 111, No. 18, pp. 8417–8428 (online), DOI: 10.1063/1.480217 (1999).
5

[27] F, C., J, L., J, Morel. P, M. and F, S.: *A Theory of Shape Identification*, Springer Verlag (2008).
6

[28] Johnson, A.: Spin-Images: A Representation for 3-D Surface Matching, PhD Thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (1997).
6

[29] Salti, S., Tombari, F. and Stefano, L.: A Performance Evaluation of 3D Keypoint Detectors, *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pp. 236 –243 (online), DOI: 10.1109/3DIMPVT.2011.37 (2011).
7

[30] Repository, S. D. S.: *http : //graphics.stanford.edu/data/3Dscanrep/*.
Accessed: 07/02/2013.
8

[31] Repository, A. S.: *http : //shapes.aimatshape.net/*.
Accessed: 07/02/2013.
8

[32] Wand, M.: *Kernel smoothing*, Chapman & Hall, London New York (1995).
13

[33] Project, B.: *http : //www.cvl.iis.u – tokyo.ac.jp/research/bayon/index.shtml*.
13