7

Stereo Vision and its Application to Robotic Manipulation

Jun Takamatsu
Nara Institute of Science and Technology (NAIST)
Japan

1. Introduction

A robot is expected to provide a service to us in our daily-life environment. Thus, it is easy to find robots that can achieve a task, such as home cleaning performed by the iRobot *Roomba*[1] or entertainment provided by the Sony *Aibo*. Unlike the situation in a plant, the daily-life environment changes sequentially. Before performing a task, it is necessary to observe the environment. Since the motion and manipulation done by a robot occur in a 3D world, gathering 3D information and not 2D information is inevitable. For example, such information helps us achieve semi-automatic robot programming (Ikeuchi & Suehiro (1994); Kuniyoshi et al. (1994)).

There are many kinds of devices to obtain 3D information. They are roughly classified into two types: active sensors and passive sensors. What distinguishes these two types is the capacity of the sensor to output energy (*e.g.*, emit light) to the outer world. As an example of active sensors, a laser sensor measures the distance using the duration since the light is emitted until it captures the reflected light.

Among the passive sensors, stereo vision is the simplest device to obtain 3D information. A stereo vision system employs at least two separate imaging devices, as in the case of human vision. Generally, the active sensors are better in accuracy than the passive sensors[2]. Although this may mean that human stereo vision suffers from inaccuracy of the obtained 3D information, we unconsciously employ prior knowledge to compensate for this.

Les us consider estimation of a 6-DOF object trajectory, which is required in various kinds of robotic manipulation, such as pick-and-place and assembly. To reduce the inaccuracy in the estimation, some constraints about the trajectory are necessary. For example, if it is previously known that the trajectory is a straight line, we reduce the inaccuracy by minimally deforming the trajectory to align it to the line. We will introduce prior knowledge into an actual robot application.

In this chapter, we will first present an overview of the stereo vision system and a method for localization using 3D data (Section 2). We will describe a method for using the contact relation (Section 3) as prior knowledge; in real world, rigid objects do not penetrate each other. Also,

[1] http://www.irobot.com

[2] However, the progress of the computer vision technology and a better benchmark data set fill the gap between them. See http://vision.middlebury.edu/stereo/.

we will describe a method for using constraints by some mechanical joint (Section 4); the type of mechanical joint defines the type of trajectory. Besides, we will introduce 3D modeling using implicit polynomials, which is robust to noise in modeling of primitive shape (Section 5). Finally we will present a conclusion for this chapter (Section 6).

2. Reviews

2.1 Stereo vision

Generally, an image represents the 3D world by projecting it onto a 2D image plane. In this sense, 3D shape reconstruction from a single image is an ill-posed problem. To simply solve this problem, it is possible to use at least two images captured from different viewpoints, *i.e.*, stereo vision. If the simple pin-hole camera model is assumed and the location of an object on both images is known, the triangulation technique outputs the 3D position or depth of the object. Note that geometric properties of both cameras are calibrated (Tsai (1986); Zhang (2000)).

Usually, it is required to obtain depths in all pixels of one image. Which means that is necessary to estimate the correspondence among all pixels. Unfortunately, this task is quite difficult. Assuming that the photometric properties of the camera are already calibrated[3], the corresponding pixels tend to have the same pixel values. In other words, the difference between these values is regarded as the degree of correspondence. If the range of depth is assumed, the candidates of the correspondences are restricted by searching the minimum differences within the range.

It is difficult to estimate the correspondence only from the single pixel observation. There are two types of solution methods. The first method consists of estimating the correspondence from the observation of a small region around the pixel. The other method consists of using prior knowledge, such as depths on the image that are usually smooth except on occluding boundaries of objects. Although these two methods are efficient for resolving the ambiguity in the correspondence, it is necessary to pay attention to the handling of the occluding boundaries. Further, the first method suffers from poor estimation of the correspondences in the texture-less region.

In summary, the depth image is estimated by the following steps (Scharstein & Szeliski (2002)):

1. matching cost computation
2. cost (support) aggregation
3. disparity computation or optimization
4. disparity refinement

The first step corresponds to calculating the difference in pixel values and the second step corresponds to the first method for solving the ambiguity. The second method is included in Step 3. Generally, the use of the second method achieves better performance in stereo vision. Please see the details in Scharstein & Szeliski (2002).

[3] Roughly speaking, when two calibrated cameras capture the same Lambertian object under the same illumination, the pixel values of the two images are the same.

When estimating the correspondences by the second method, it is necessary to minimize the following energy function:

$$E = \min_{d} \sum_{i \in I} C_1(i, d(i)) + \sum_{i,j \in I, i \neq j} C_2(i, j, d(i), d(j)), \tag{1}$$

where the term $d(i)$ represents the depth in the pixel i, and the set I represents the image region. The term C_1 is determined by one pixel. On the other hand, the term C_2 is determined by the relationship of two pixels. It is possible to think about the relationship of more than two pixels (referred to as *higher order term*).

Graph cuts (Boykov & Kolmogorov (2004)) and belief propagation (Felzenszwalb & Huttenlocher (2006)) are very interesting methods for minimizing the function as shown in Eq. (1). Although both methods suffer from difficulties in handling the higher order terms, recently these difficulties have been circumvented (Ishikawa (2009); Lan et al. (2006); Potetz (2007)). Although the Graph cuts may achieve a better performance in the optimization, it is very useful to take advantage of parallel computing by a Graphic Processing Unit (GPU) in the belief propagation since it easily accelerates the calculation (Brunton et al. (2006)).

2.2 Localization

Although the methods described in Section 2.1 provide us with the 3D information of the world, it is further needed to localize the target objects in order to estimate the interaction of the objects, which is very important especially in robotics applications. We assume that the 3D model of the target object is previously given.

There are two kinds of localization: rough localization and fine localization. The rough localization includes object detection and estimates rough correspondence between the model and the observed 3D data. The result of the rough localization is usually used as the input of the fine localization. The fine localization obtains the location of the object by precisely aligning the model and the observed 3D data.

The rough localization can be classified into two types: one that uses local descriptors and the other one that uses global descriptors. The first method calculates a descriptor of a point to distinguish it from the other points based on its local shape. The descriptor should be invariant to rigid transformation. The correspondence is estimated by comparing two descriptors. The idea is very similar to the descriptors in 2D images, such as SIFT (Lowe (2004)) and SURF (Bay et al. (2008)). For example, geometric hashing (Wolfson & Rigoutsos (1997)) calculates the descriptors from the minimum number of neighborhood points to satisfy the invariant. Spin images (Johnson & Hebert (1999)) uses point distribution in polar coordinates as descriptors. Once the correspondences are given, the rigid transformation is calculated by Umeyama (1991). Also, RANSAC (Fischler & Bolles (1981)) is useful to detect the erroneous correspondences.

The latter methods calculate one descriptor in each object. From the definition, a change of the descriptor by the rigid transformation is easily calculated. The rough localization is estimated by matching two descriptors. Geometric moment (Flusser & Suk (1994)) aligns the two objects by matching principal axes. Extended Gaussian Image (EGI) (*e.g.*. Kang & Ikeuchi (1997)) uses the distribution of the surface normal directions as a descriptor. The use of the spherical harmonics accelerates the localization (Makadia et al. (2006)). Spherical Attribute

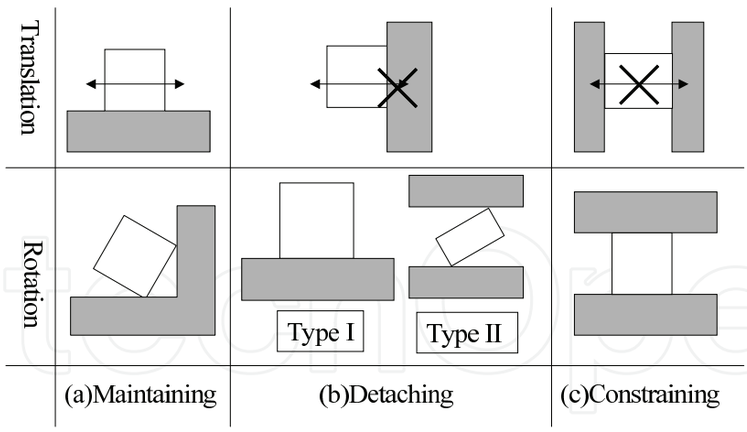

Fig. 1. Types of displacement. Considering the horizontal translation in the upper row, translation from left to right columns are unconstrained, partially constrained, and fully constrained. Considering the rotation in the lower row, rotation from left to right columns are unconstrained, partially constrained, and fully constrained as similar to the upper case. To reduce the vision errors while avoiding drastic changes in the original data, these three types should be distinguished.

Image (SAI) (Hebert et al. (1995)) represents the distribution of curvatures on the spherical coordinates.

In the fine localization, the key is how to estimate the correspondences in a fine resolution. The iterative closest point (Besl & McKay (1992); Chen & Medioni (1991)) is a pioneer work on the fine localization and regards the closest point as the correspondence. There are many variants with respect to the calculation of the correspondence and the evaluation function for the registration. Please see Rusinkiewicz & Levoy (2001). The output from the stereo vision system is relatively inaccurate, it is very important to make the fine localization robust to noise. To do so, a robust estimator, such as M-estimator (Huber (1981)), is normally used, such as Wheeler & Ikeuchi (1995).

3. Vision error correction using contact relations

Consider a moving object that comes into contact with another object or the environment. Due to the vision errors, in the imaginary world of the robot, the object possibly penetrates another object. The difference between the real world and the imaginary world makes more difficult to estimate the interaction of the objects. We introduce the use of contact information to reduce the vision errors. We assume that all the objects are polyhedral and concentrate on the two-object relationship; one object (referred to as *moving object*) moves and the other object (referrer to as *fixed object*) is fixed. Even by such simplification, the vision error correction is difficult due to the non-linearity (Hirukawa (1996)).

However, as shown in Fig. 1, local displacement along one direction (horizontal translation in the upper row and rotation in the lower row) is classified into three types: no constraint, partially constraint, and fully constraint. In order to reduce the vision errors while avoiding drastic changes in the original data, it would be better to keep the information corresponding to unconstrained direction.

We propose two types of methods for vision error correction using the contact relations. The contact relation represents a set of pairs of contacting elements (vertex, edge, and face). One method (Takamatsu et al. (2007)) relies on the non-linear optimization and often contaminates the unconstrained displacement. The other method (Takamatsu et al. (2002)) employs only the linear method. Although at least one solution which satisfies the contact relation is required, the optimality holds in this method.
The overview of the method is as follows:

1. Calculate the object configuration which satisfies the constraint on the contact relation using the non-linear optimization method (Takamatsu et al. (2007)). Note that we accept any configurations.
2. Formulate the equation of feasible infinitesimal displacement using the method (Hirukawa et al. (1994))
3. Calculate the optimum configuration by removing the redundant displacement that is derived from the non-linear optimization.

Hirukawa *et al.* proposed a method for introducing the constraint on the contact relation between two polynomial objects (Hirukawa et al. (1994)). They proved that the infinitesimal displacement that maintains the contact relation can be formulated as Eq. (2), where N is the number of pairs of contacting elements, $\mathbf{p}_i$ is the position of the i-th contact in the world coordinates, $\mathbf{f}_{ij}$ $(\in R^3)$ is the normal vector of the separate plane[4], $M(i)$ is the number of separate planes of the i-th contact, and the 6D vector $[\mathbf{s}_0, \mathbf{s}_1]$ represents infinitesimal displacement in the screw representation (Ohwovoriole & Roth (1981)).

$$\bigcap_{i}^{N} \bigcap_{j}^{M(i)} \mathbf{f}_{ij} \cdot \mathbf{s}_1 + (\mathbf{p}_i \times \mathbf{f}_{ij}) \cdot \mathbf{s}_0 = 0. \tag{2}$$

In the screw representation, the vector $\mathbf{s}_0$ represents the rotation axis. Introducing the constraint only about the term $\mathbf{s}_0$ gives us the range of the feasible rotation axis as Eq. (3).

$$\bigcap_{i}^{n} \mathbf{g}_i \cdot \mathbf{s}_0 = 0. \tag{3}$$

The non-linearity is only derived from the non-linearity in the orientation. If the optimum orientation is already known, the issue on the vision error correction is simply solved using the least linear minimization. The method for calculating the optimum orientation varies according to the rank of Eq. (3), because the constraint is semantically varied. If the rank is three, the optimum orientation is uniquely determined. We only use the orientation obtained by the non-linear optimization. If the rank is zero, the original orientation is used.
Figure 2 shows the case where the rank is two. The upper left and the upper right images represent the orientation before and after the vision error correction by the non-linear optimization, respectively. The rotation about the axis shown in the lower right image is the redundant displacement, because this displacement does not change the contact relation. The optimum orientation is obtained by removing this displacement.

[4] For example, in the case where some vertex on the moving object make contact with some face on the fixed object, the vector $\mathbf{f}_{ij}$ is equal to the outer normal of the face.

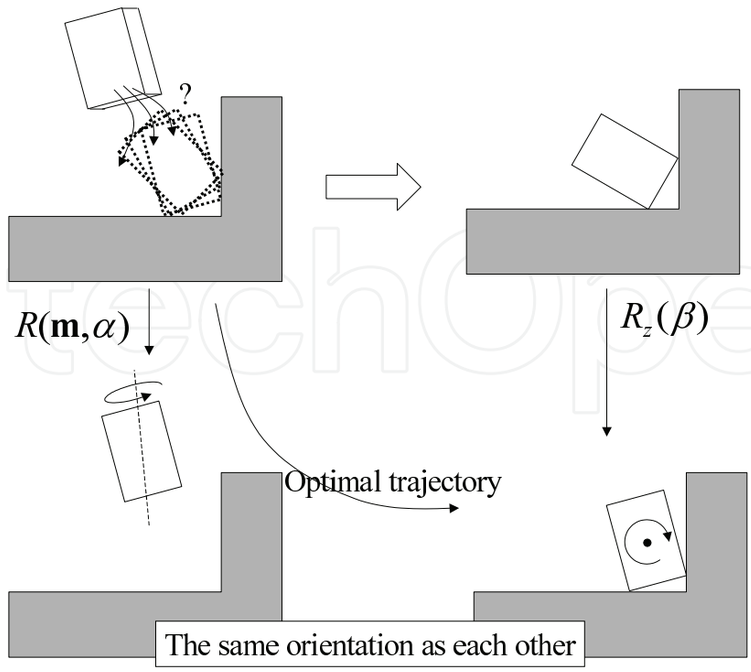

Fig. 2. Redundant orientation in the case where the rank is two. The upper left and the upper right images represent the orientation before and after the vision error correction by the non-linear optimization, respectively. The lower right image represents the optimum orientation; the rotation about the axis shown in the lower right image is the redundant displacement, because this displacement does not change the contact relation.

We define the local coordinates A, where the z-axis is defined as the axis of the redundant displacement, which is obtained from Eq. (3). Let ${}^{A}\Theta_{E}$ and ${}^{A}\Theta_{S}$ be the orientation before and after the vision-error correction in the local coordinates. The orientation ${}^{A}\Theta_{E}$ is translated to the orientation ${}^{A}\Theta_{S}$ by the following two steps:

1. rotation about the z-axis while maintaining the contact relation
2. rotation about the axis $\mathbf{m}$ which is on the xy-plane.

These two steps are formulated as Eq. (4), where $\mathbf{R}_{*}(\theta)$ $(\in \mathrm{SO}(3))$ is a θ [rad] rotation about $*$-axis, $\mathbf{R}(\mathbf{m}, \alpha)$ is a α [rad] rotation about the axis $\mathbf{m}$.

$$\mathbf{R}(\mathbf{m}, \alpha)\,{}^{A}\Theta_{S} = \mathbf{R}_{z}(\beta)\,{}^{A}\Theta_{E}. \tag{4}$$

By solving this equation, the terms $\alpha, \beta, \mathbf{m}$ are calculated. The first rotation is the redundant displacement and the optimum orientation ${}^{A}\Theta_{opt}$ in the local coordinates is obtained by

$${}^{A}\Theta_{opt} = \mathbf{R}(\mathbf{m}, \alpha)\,{}^{A}\Theta_{S}. \tag{5}$$

Figure 3 shows the case where the rank is one. We define the local coordinates A, where the z-axis is the constrained DOF in rotation, which is obtained from Eq. (3). Let ${}^{A}\Theta_{E}$ and ${}^{A}\Theta_{S}$ be the orientation before and after the vision-error correction in the local coordinates. Similarly in the case where the rank is two, the orientation is translated to the orientation ${}^{A}\Theta_{S}$ by the following two steps:

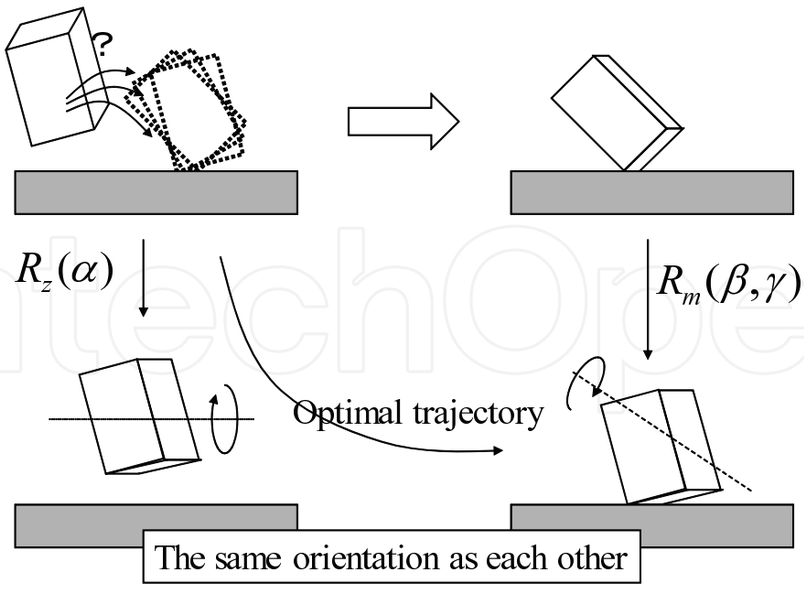

Fig. 3. Redundant displacement in the case where the rank is one. The upper left and the upper right images represent the orientation before and after the vision error correction by the non-linear optimization, respectively. The lower right image represents the optimum orientation; the rotation about the axis shown in the lower right image is the redundant displacement, because this displacement does not change the contact relation.

1. rotation while maintaining the contact relation
2. rotation about the z-axis

These two steps are formulated as Eq. (6),

$$\mathbf{R}_z(\alpha)\,{}^A\Theta_S = \mathbf{R}_m(\beta,\gamma)\,{}^A\Theta_E, \tag{6}$$

where $\mathbf{R}_m(\beta,\gamma)$ is the rotation to maintain the contact relation and has two DOF. The DOF of Eq. (6) is three and thus is solvable. The optimum orientation ${}^A\Theta_{opt}$ in the local coordinates is obtained by

$${}^A\Theta_{opt} = \mathbf{R}_z(\alpha)\,{}^A\Theta_S. \tag{7}$$

Unfortunately, the formulation of $\mathbf{R}_m(\beta,\gamma)$ varies case-by-case and there is no general rule. We assume that the rank becomes two, only when (1) some edge of the moving object makes contact with some face of the fixed object or when (2) some face of the moving object makes contact with some edge of the fixed object. These are common cases.
Consider the case 1 (see Fig. 4), a β [rad] rotation about the axis 1 followed by a γ [rad] rotation about the axis 2 maintains the contact relation. Thus the term $\mathbf{R}_m(\beta,\gamma)$ is formulated as:

$$\mathbf{R}_m(\beta,\gamma) = \mathbf{R}(\mathbf{n},\gamma)\mathbf{R}(\mathbf{l},\beta), \tag{8}$$

where $\mathbf{n}$ is the surface normal and $\mathbf{l}$ is the edge direction.
Consider the case 2 (see Fig. 5), a β [rad] rotation about the axis I followed by a γ [rad] rotation about the axis II maintains the contact relation. Thus the term $\mathbf{R}_m(\beta,\gamma)$ is formulated as:

$$\mathbf{R}_m(\beta,\gamma) = \mathbf{R}(\mathbf{l},\gamma)\mathbf{R}(\mathbf{n},\beta). \tag{9}$$

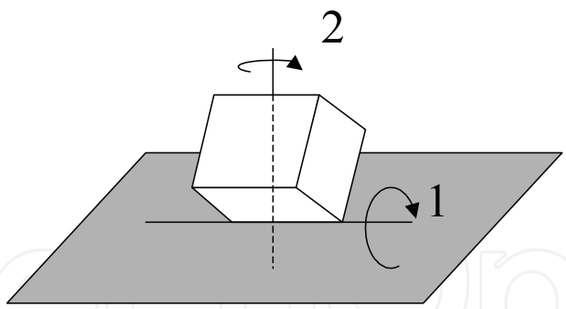

Fig. 4. Case 1: some edge of the moving object makes contact with some face of the fixed object

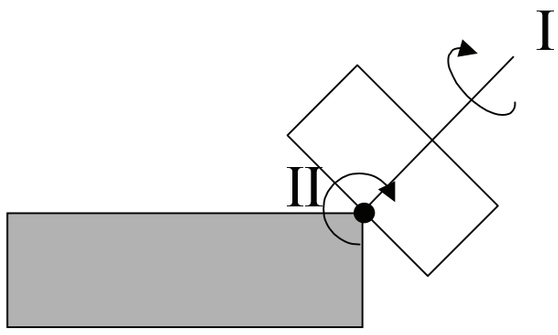

Fig. 5. Case 2: some face of the moving object makes contact with some edge of the fixed object

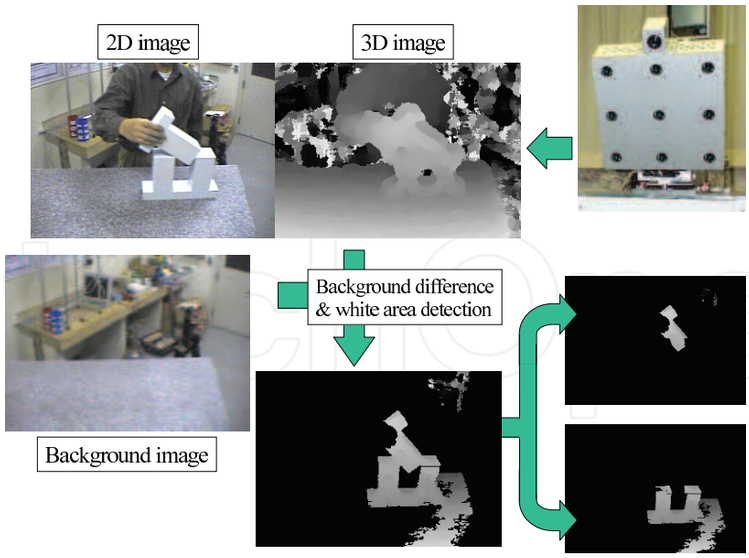

Fig. 6. Vision system and the overview of the vision algorithm

In both cases, Eq. (6) can be solved.

Result In experiments in this section, we use the vision system in Fig. 6. Since the depth image is obtained in real-time, this vision system uses only the first method to solve the

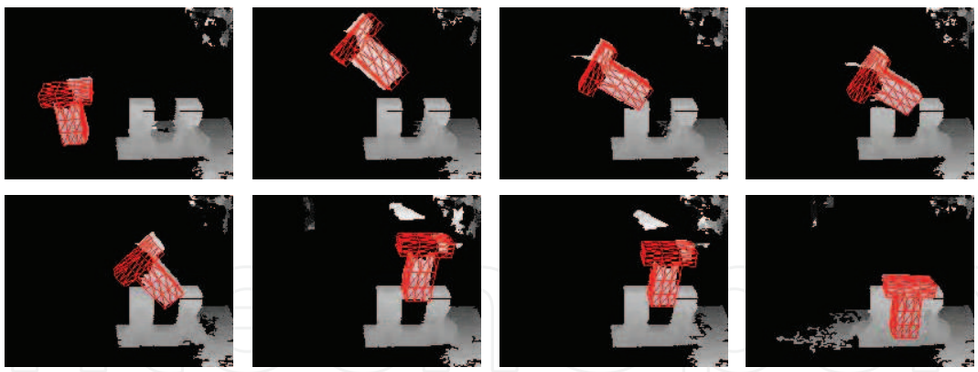

Fig. 7. Tracking result

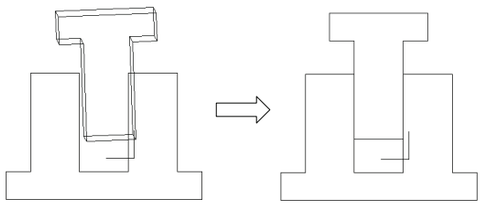

Fig. 8. Vision error correction by the non-linear methods. Left and right images show the result before and after the correction.

ambiguity mentioned in Section 2.1 and the calculation is implemented on the hardware. Using background subtraction and color detection, we only extract the target objects. By histogram of depths in each pixel, we roughly distinguish the moving and the fixed objects. We employ the method by Wheeler & Ikeuchi (1995) to extract the 6-DOF trajectory of the moving object. Figure 7 shows the results. And Figure 8 shows one example of the vision error correction by the non-linear method.

Figure 9 shows the result of applying the optimum vision error correction to the tracking result. The upper right and lower right graphs show the vision-error correction by the non-linear optimization (Takamatsu et al. (2007)) and the combination of the non-linear and linear optimization. Since translational displacement along the vertical direction is not constrained by any contacts, it is optimum that the displacement along the direction by the vision error correction is zero. In other words, the projected trajectories before and after the error correction should be the same. It is difficult to obtain the optimum error correction by using only the non-linear optimization, but it is possible to obtain it by combining the non-linear and linear methods. The lower left graph shows the trajectory projected on the xy-plane. The trajectory during the insertion is correctly adjusted as a straight line.

4. Estimating joint parameters

We often find the objects with several rigid parts which are connected by joints as shown in Fig. 10. These objects range from human body to daily-life artificial objects, such as door knobs and taps. Even in the constraint-free space, motion which seems to be virtually constrained by a joint can be seen. A constraint generated by a joint is useful for reducing vision errors. Even if the type of joint is known, the vision error correction involves estimation of joint parameters from a noise-contaminated observation. In this section, we describe the estimation of the

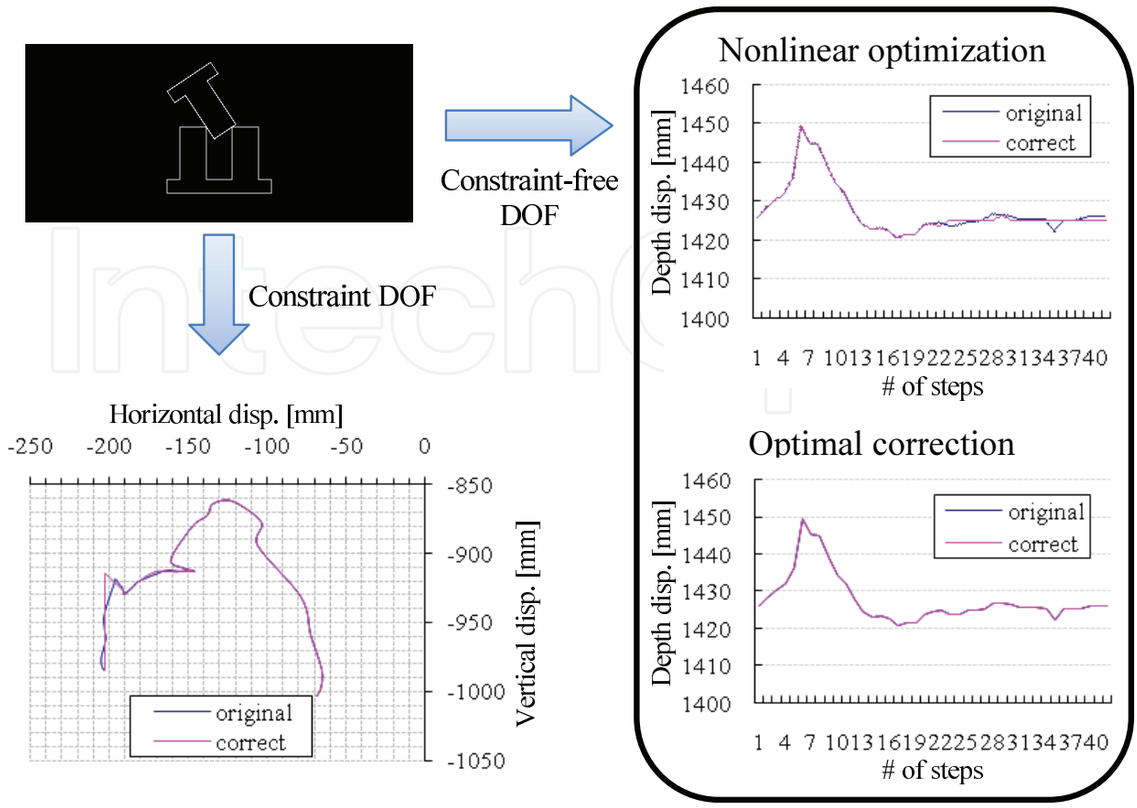

Fig. 9. Result of the vision error correction. The upper right and lower right graphs show the vision-error correction by the non-linear optimization (Takamatsu et al. (2007)), and combination of the non-linear and linear optimization. Displacement along the unconstrained direction is zero after the vision error correction (see the lower right graph), while trajectory during the insertion is correctly adjusted as the straight line (see the lower left graph).

revolute joint parameters as well as the vision error corrections. The estimation in the other types of joints is seen in Takamatsu (2004).

The trajectory of Link A with respect to coordinates of Link B, $({}^{B}\mathbf{t}_{A}(t), {}^{B}\Theta_{A}(t))$, is given as input, where the term ${}^{B}\mathbf{t}_{A}(t)$ $(\in \mathrm{R}^3)$ is the location and the term ${}^{B}\Theta_{A}(t)$ $(\in \mathrm{SO}(3))$ is the orientation at time t. As shown in Fig. 11, the joint parameters in the revolute joint are composed of the direction of revolute axis in coordinates of both Link A and Link B, ${}^{A}\mathbf{l}$, ${}^{B}\mathbf{l}$, and their location, ${}^{A}\mathbf{c}$, ${}^{B}\mathbf{c}$. Note that $|{}^{A}\mathbf{l}| = |{}^{B}\mathbf{l}| = 1$ holds. These terms must satisfy the following conditions:

$$ {}^{B}\mathbf{l} = {}^{B}\Theta_{A}(t)\ {}^{A}\mathbf{l}, \tag{10} $$

$$ {}^{B}\mathbf{c} = {}^{B}\Theta_{A}(t)\ {}^{A}\mathbf{c} + {}^{B}\mathbf{t}_{A}(t). \tag{11} $$

Considering the observation noise $\Delta\Theta(t)$ in orientation, Eq. (10) is reformulated as

$$ {}^{B}\mathbf{l} = \Delta\Theta(t)\ {}^{B}\Theta_{A}(t)\ {}^{A}\mathbf{l}. \tag{12} $$

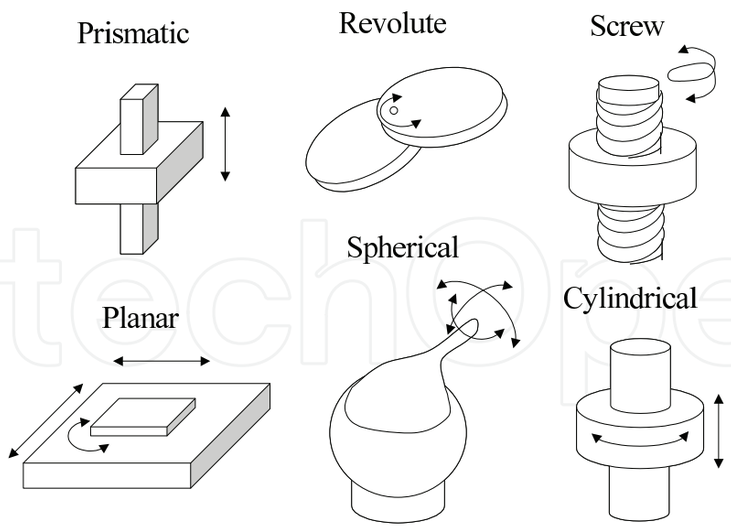

Fig. 10. Examples of joints

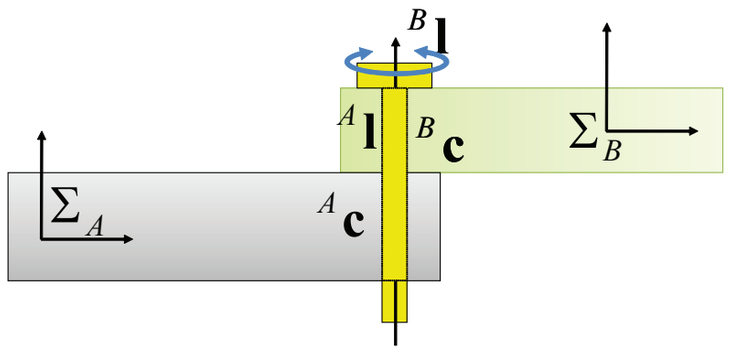

Fig. 11. Coordinates in revolute joint

We estimate the parameters in the least square manner, *i.e.*, estimate the parameters while minimizing the sum of the norm of $\Delta\Theta(t)$. The rotational displacement $\Delta\Theta$ is represented as a θ [deg] rotation about some axis $\mathbf{l}$, *i.e.*. $\Delta\Theta = \mathbf{R}(\mathbf{l}, \theta)$. Then, we define its norm as $1 - \cos\theta$. Note that θ is small enough, $1 - \cos\theta$ is approximated as $\frac{\theta^2}{2}$.
We decompose the noise term $\Delta\Theta(t)$ into a multiplication of two rotation matrices as shown in Eq. (13). One is a $\theta_1(t)$ [deg] rotation about the axis ${}^B\mathbf{l}$ and the other is a $\theta_2(t)$ [deg] rotation about the axis $\mathbf{l}(t)$, where $\forall t, {}^B\mathbf{l} \cdot \mathbf{l}(t) = 0$ holds.

$$\Delta\Theta(t) = \mathbf{R}({}^B\mathbf{l}, \theta_1(t))\mathbf{R}(\mathbf{l}(t)\theta_2(t)). \tag{13}$$

By substituting Eq. (13) into Eq. (12), Eq. (14) is obtained.

$$\mathbf{R}({}^B\mathbf{l}, \theta_1(t))\ {}^B\mathbf{l} = \mathbf{R}(\mathbf{l}(t), \theta_2(t)){}^B\Theta_A(t)\ {}^A\mathbf{l}. \tag{14}$$

The left part of this equation is constant for any $\theta_1(t)$, since ${}^B\mathbf{l}$ does not change after the rotation about the axis ${}^B\mathbf{l}$. Following the least square manner, we assume that $\theta_1(t) = 0$.

Thereafter, by multiplying the term ${}^{B}\mathbf{l}^{T}$ on the both sides from the right in Eq. (14), Eq. (15) is obtained. Note that we simply denote $\theta_2(t)$ as $\theta(t)$.

$$\mathbf{R}(-\mathbf{l}(t),\theta(t))\,{}^{B}\mathbf{l}\;{}^{B}\mathbf{l}^{T} = {}^{B}\Theta_{A}(t)\;{}^{A}\mathbf{l}\;{}^{B}\mathbf{l}^{T}. \tag{15}$$

The left side of Eq. (15) is written as follows:

$$(I - \sin\theta(t)[\mathbf{l}(t)]_{\times} + (1-\cos\theta(t))[\mathbf{l}(t)]_{\times}^{2})\,{}^{B}\mathbf{l}\;{}^{B}\mathbf{l}^{T},$$

where the matrix $[(x,y,z)]_{\times}$ is the skew symmetry matrix and is defined as

$$[(x,y,z)]_{\times} = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix}.$$

Through the actual calculation, it is proved that the following equations always hold:

$$\begin{aligned} \mathrm{Tr}({}^{B}\mathbf{l}\;{}^{B}\mathbf{l}^{T}) &= 1, \\ \mathrm{Tr}([\mathbf{l}(t)]_{\times}\;{}^{B}\mathbf{l}\;{}^{B}\mathbf{l}^{T}) &= 0, \\ \mathrm{Tr}([\mathbf{l}(t)]_{\times}^{2}\;{}^{B}\mathbf{l}\;{}^{B}\mathbf{l}^{T}) &= ({}^{B}\mathbf{l}\cdot\mathbf{l}(t))^{2} - 1 = -1, \end{aligned}$$

where $\mathrm{Tr}(\mathbf{M})$ returns the trace of the matrix $\mathbf{M}$. By using them, we obtain the following equation:

$$\mathrm{Tr}(\mathbf{R}(-\mathbf{l}(t),\theta(t))\,{}^{B}\mathbf{l}\;{}^{B}\mathbf{l}^{T}) = \cos\theta(t) \tag{16}$$

When the sum of $1-\cos\theta(t)$ is minimized, the sum of the norm of the noise term $\Delta\Theta(t)$ is minimized. We estimate the direction by minimizing the following equation.

$$({}^{A}\hat{\mathbf{l}},{}^{B}\hat{\mathbf{l}}) = \underset{{}^{A}\mathbf{l},{}^{B}\mathbf{l}}{\mathrm{argmin}} \sum_{t}(1 - \mathrm{Tr}({}^{B}\Theta_{A}(t)\;{}^{A}\mathbf{l}\;{}^{B}\mathbf{l}^{T})). \tag{17}$$

After estimating the direction, the orientation after the vision error correction ${}^{B}\hat{\Theta}_{A}(t)$ is obtained from the outer product of ${}^{B}\Theta_{A}(t){}^{A}\mathbf{l}$ and ${}^{B}\mathbf{l}$. The displacement for the vision error correction corresponds to the matrix with minimum norm that matches the vector ${}^{B}\Theta_{A}(t)\;{}^{A}\mathbf{l}$ with the vector ${}^{B}\mathbf{l}$.
After estimating the corrected orientation, the location is estimated by the linear least square method, where $A(t) = \left(-{}^{B}\hat{\Theta}_{A}(t)\;I\right)$.

$$\left(\sum_{t} A(t)^{T}A(t)\right)\begin{pmatrix} {}^{A}\mathbf{c} \\ {}^{B}\mathbf{c} \end{pmatrix} = \sum_{t} A(t)^{T}\;{}^{B}\mathbf{t}_{A}(t). \tag{18}$$

Since the matrix $\sum_{i} A(t)^{T}A(t)$ is not a full-rank matrix, the singular value decomposition is used to solve this equation.
Result We showed the estimation result from the observation using a real-time stereo vision system in Section 3. In this experiment, we used two LEGO parts, which are connected by the revolute joint. Figure 12 shows the tracking result.

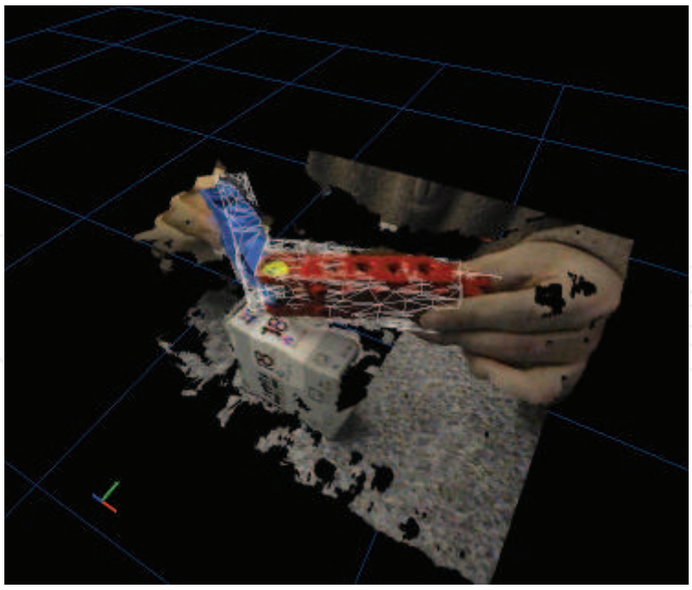

Fig. 12. Tracking result

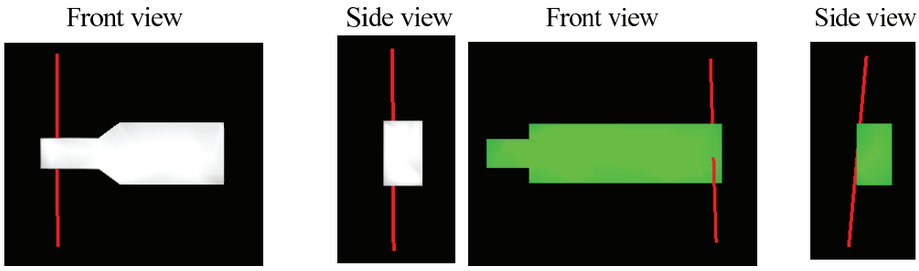

Fig. 13. Estimation result

Figure 13 shows the estimation result. The red line indicates the estimated revolute axis. The joint parameter of the blue LEGO block (corresponds to the white block in Fig. 13) is relatively accurate. If the noise distribution is accurately modeled, the maximum likelihood (ML) inference may improve the estimation. Unfortunately, it is difficult, perhaps impossible, to model the noise distribution. Outlier detection dissolves this poor estimation.

5. Modeling by implicit polynomial

To recognize the interaction of the objects, the differential properties of the object surface are often required. Generally, the differential operation amplifies the noise, and thus the object modeling method robust to the noise is highly demanded. Although we did not directly use the method mentioned in this section for the 3D data obtained from the stereo vision system, we would like to introduce modeling using an implicit polynomial which is very robust to noise.

Representation in implicit polynomial (IP) is advantageous in robustness against noise and occlusion, compactness of representation, and differentiability. And thus, many applications using IP's exist (*e.g.*Taubin & Cooper (1992)). Unlike the other parametric representations,

such as B-spline and NURBS, it is very easy to estimate the parameters of IP, given the target model.
IP with n-degree can be defined as follows:

$$\begin{aligned} f_n(\boldsymbol{x}) &= \sum_{0\leq i,j,k;i+j+k\leq n} a_{ijk}x^i y^j z^k \\ &= (\underbrace{1 \; x \; \ldots \; z^n}_{\boldsymbol{m}(\boldsymbol{x})^T})(\underbrace{a_{000} \; a_{100} \; \ldots \; a_{00n}}_{\boldsymbol{a}})^T, \end{aligned} \tag{19}$$

where $\boldsymbol{x} = (x \; y \; z)$ represents coordinates in 3D space. In the IP representation, the object surface is modeled by a zero level set of the IP, *i.e.*, $\{\boldsymbol{x}|f_n(\boldsymbol{x}) = 0\}$. The IP's parameter corresponds to the coefficient $\boldsymbol{a}$. Given the target model in point cloud representation, such as $\{\boldsymbol{x_i}\}$, the parameters are estimated by the following steps:

1. manually assign the IP's degree.
2. solve the following simultaneous linear system, where M is the matrix whose i-th row corresponds to $\boldsymbol{m}(\boldsymbol{x_i})$[5]:

$$M\boldsymbol{a} = \boldsymbol{b}. \tag{20}$$

3. Compare the modeling result to the target object. If it is not so accurate, change the degree and go back to Step 1.

Since it is not intuitive to select the appropriate degree n for the complicated shapes, this selection wastes time unnecessarily. Further, instability in higher degree IP is also problematic. We propose a method to adaptively select the appropriate degree by incrementally increasing the degree, while keeping the computational time. Incrementability of QR decomposition by the Gram-Schmidt orthogonalization plays a very important role in the proposed method (Zheng et al. (2010)). QR decomposition decomposes the given matrix M into two matrices Q, R as $M = QR$, where $Q^TQ = I$ holds and the matrix R is an upper triangle matrix. Since the Gram-Schmidt orthogonalization is conducted in an inductive manner, it offers the incrementability to the proposed method. To solve the coefficient $\boldsymbol{a}$ in each degree, we simply solve the upper triangle linear system, resulting in reducing the computational time. Eigenvalues provide information about stability in the calculation. Fortunately, eigenvalues of the upper triangle matrix are simply obtained by just checking the diagonal elements of the matrix.
We convert Eq. (20) to fit the QR decomposition. In the linear least square manner, the coefficient $\boldsymbol{a}$ should satisfy the following condition:

$$M^TM\boldsymbol{a} = M^T\boldsymbol{b}. \tag{21}$$

By substituting $M = QR$, the following equation is obtained:

$$R^TQ^TQR\boldsymbol{a} = R^TQ^T\boldsymbol{b} \Rightarrow R\boldsymbol{a} = Q^T\boldsymbol{b} \stackrel{\text{def}}{=} \widetilde{\boldsymbol{b}}. \tag{22}$$

[5] Generally, the condition about the zero level set generates a constraint where $\boldsymbol{b} = \boldsymbol{0}$. Thus, the eigen method is used for the estimation (*e.g.*, Taubin (1991)). In order to increase calculation stability, other additional constraints are added (*e.g.*, M. Blane & Cooper (2000)), resulting in $\boldsymbol{b} \neq \boldsymbol{0}$. This can be solved using a simple linear solver.

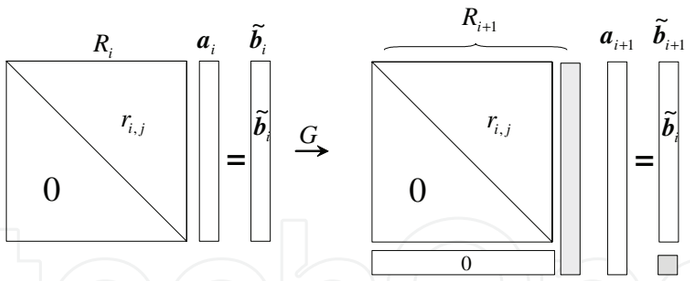

Fig. 14. Necessary calculation for going from the i-th step to the $i+1$-th step. Calculation results are reusable, except for the shaded part.

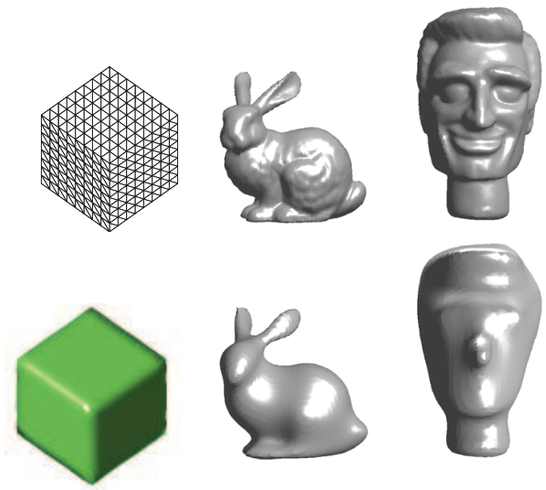

Fig. 15. Result of IP modeling. First row: original data. Second row: IP model.

As described above, QR decomposition is done by gradually applying the Gram-Schmidt orthogonalization to columns of the matrix M from left to right. When going from the i-th step to the $i+1$-th step, we only need to calculate the shaded part in Fig. 14; the other part is kept constant from the previous calculation. As a result, the calculation is totally accelerated. Regarding the numerical stability, we pay attention to the case where the conditional number of the matrix R becomes worse. The conditional number is usually defined as the ratio between the maximum and the minimum eigenvalues. Since the eigenvalues of the matrix R corresponds to the diagonal elements themselves, we simultaneously evaluate the numerical stability. If it tunes to be unstable, we ignore the corresponding column, which is added at this step, or partially apply the RR method (Sahin & Unel (2005); Tasdizen et al. (2000)). We increase the IP's degree until the modeling accuracy is sufficient.

Result Figure 15 shows the result of IP modeling. The cube consists of six planes so an IP with six degrees is appropriate. The paper (Zheng et al. (2010)) includes other IP modeling results. Since an IP models the shape considering global consistency, the model is useful for object recognition.

6. Conclusion

In this chapter, we described the vision error correction using various constraints, such as contact relation and mechanical joint. Further, we introduced the modeling method using an implicit polynomial which is very robust to noise. Stereo vision is simple, but potential technique to obtain 3D information. One disadvantage is the accuracy. We believe that vision error correction using prior knowledge becomes necessary research stream in real-world robot applications.

7. Acknowledgment

We would like to thank members of Ikeuchi laboratory (http://www.cvl.iis.u-tokyo.ac.jp) for providing the software, the experimental environment, and the fruitful comments. We would like to thank Mr. Garcia Ricardez Gustavo Alfonso for his useful feedback on this document.

8. References

Bay, H., Ess, A., Tuytelaars, T. & Gool, L. V. (2008). SURF: Speeded up robust features, *Comp. Vis. and Image Understanding* 110(3): 346–359.

Besl, P. J. & McKay, N. D. (1992). A method for registration of 3-d shapes, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 14(2): 249–256.

Boykov, Y. & Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 26(9): 1124–1137.

Brunton, A., Shu, C. & Roth, G. (2006). Belief propagation on the gpu for stereo vision, *Proc. of Canadian Conf. on Comp. and R. Vis.*

Chen, Y. & Medioni, G. (1991). Object modeling by registration of multiple range images, *Proc. of IEEE Int'l Conf. on R. and Auto. (ICRA)*.

Felzenszwalb, P. & Huttenlocher, D. (2006). Efficient belief propagation for early vision, *Int'l J. of Comp. Vis.* 70: 41–54.

Fischler, M. A. & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM* 24: 381–395.

Flusser, J. & Suk, T. (1994). A moment-based approach to registration of images with affine geometric distortion, *IEEE Trans. on Geoscience and Remote Sensing* 32(2): 382–387.

Hebert, M., Ikeuchi, K. & Delingette, H. (1995). A spherical representation for recognition of free-form surface, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 17(7): 681–690.

Hirukawa, H. (1996). On motion planning of polyhedra in contact, *WAFR* .

Hirukawa, H., Matsui, T. & Takase, K. (1994). Automatic determination of possible velocity and applicable force of frictionless objects in contact from a geometric model, *IEEE Trans. on Robotics and Automation* 10(3): 309–322.

Huber, P. J. (1981). *Robust statistics*, Wiley-Interscience.

Ikeuchi, K. & Suehiro, T. (1994). Toward an assembly plan from observation part i: Task recognition with polyhedral objects, *IEEE Trans. on Robotics and Automation* 10(3): 368–385.

Ishikawa, H. (2009). Higher-order clique reduction in binary graph cut, *Proc. of Comp. Vis. and Patt. Recog. (CVPR)*.

Johnson, A. E. & Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3d scenes, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 21(5): 433–449.

Kang, S. B. & Ikeuchi, K. (1997). The complex egi: New representation for 3-d pose determination, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 15(7): 707–721.

Kuniyoshi, Y., Inaba, M. & Inoue, H. (1994). Learning by watching: Extracting reusable task knowledge from visual observation of human performance, *IEEE Trans. on Robotics and Automation* 10(6): 799–822.

Lan, X., Roth, S., Huttenlocher, D. P. & Black, M. J. (2006). Efficient belief propagation with learned higher-order markov random fields, *Proc. of Euro. Conf. on Comp. Vis. (ICCV)*.

Lowe, D. (2004). Distinctive image features from scale-invariant key points, *Int'l J. of Comp. Vis.* 60(2): 91–110.

M. Blane, Z. L. & Cooper, D. (2000). The 3l algorithm for fitting implicit polynomial curves and surfaces to data, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 22(3): 298–313.

Makadia, A., IV, A. P. & Daniilidis, K. (2006). Fully automatic registration of 3d point clouds, *Proc. of Comp. Vis. and Patt. Recog. (CVPR)*.

Ohwovoriole, M. S. & Roth, B. (1981). An extension of screw theory, *J. of Mechanical Design* 103: 725–735.

Potetz, B. (2007). Efficient belief propagation for vision using linear constraint nodes, *Proc. of Comp. Vis. and Patt. Recog. (CVPR)*.

Rusinkiewicz, S. & Levoy, M. (2001). Efficient variants of the icp algorithm, *Proc. of Int'l Conf. on 3-D Digital Imaging and Modeling*.

Sahin, T. & Unel, M. (2005). Fitting globally stabilized algebraic surfaces to range data, *Proc. of Int'l Conf. on Comp. Vis. (ICCV)*.

Scharstein, D. & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithm, *Int'l J. of Comp. Vis.* 47(1/2/3): 7–42.

Takamatsu, J. (2004). *Abstraction of Manipulation Tasks to Automatically Generate Robot Motion from Observation*, PhD thesis, the University of Tokyo.

Takamatsu, J., Kimura, H. & Ikeuchi, K. (2002). Calculating optimal trajectories from contact transitions, *Proc. of IEEE Int'l Conf. on Intell. R. and Sys. (IROS)*.

Takamatsu, J., Ogawara, K., Kimura, H. & Ikeuchi, K. (2007). Recognizing assembly tasks through human demonstration, *Int'l J. of Robotics Research* 26(7): 641–659.

Tasdizen, T., Tarel, J.-P. & Cooper, D. B. (2000). Improving the stability of algebraic curves for applications, *IEEE Trans. on Image Proc.* 9(3): 405–416.

Taubin, G. (1991). Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 13(11): 1115–1138.

Taubin, G. & Cooper, D. (1992). *Symbolic and Numerical Computation for Artificial Intelligence*, Computational Mathematics and Applications, Academic Press, chapter 6.

Tsai, R. Y. (1986). An efficient and accurate camera calibration technique for 3d machine vision, *Proc. of Comp. Vis. and Patt. Recog. (CVPR)*.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 13(4).

Wheeler, M. D. & Ikeuchi, K. (1995). Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 17(3): 252–265.

Wolfson, H. J. & Rigoutsos, I. (1997). Geometric hashing: An overview, *Computing in Science and Engineering* 4(4): 10–21.

Zhang, Z. (2000). A flexible new technique for camera calibration, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 22(11): 1330–1334.

Zheng, B., Takamatsu, J. & Ikeuchi, K. (2010). An adaptive and stable method for fitting implicit polynomial curves and surfaces, *IEEE Trans. on Patt. Anal. and Mach. Intell.* 32(3): 561–568.

Advances in Stereo Vision

Edited by Prof. Jose R.A. Torreao

ISBN 978-953-307-837-3
Hard cover, 120 pages
Publisher InTech
Published online 19, July, 2011
Published in print edition July, 2011

Stereopsis is a vision process whose geometrical foundation has been known for a long time, ever since the experiments by Wheatstone, in the 19th century. Nevertheless, its inner workings in biological organisms, as well as its emulation by computer systems, have proven elusive, and stereo vision remains a very active and challenging area of research nowadays. In this volume we have attempted to present a limited but relevant sample of the work being carried out in stereo vision, covering significant aspects both from the applied and from the theoretical standpoints.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Jun Takamatsu (2011). Stereo Vision and Its Application to Robotic Manipulation, Advances in Stereo Vision, Prof. Jose R.A. Torreao (Ed.), ISBN: 978-953-307-837-3, InTech, Available from: http://www.intechopen.com/books/advances-in-stereo-vision/stereo-vision-and-its-application-to-robotic-manipulation

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen