Fast Alignment of 3D Geometrical Models and 2D Color Images using 2D Distance Maps

Yumi Iwashita, Ryo Kurazume, Tsutomu Hasegawa
Graduate School of Information Science and Electrical Engineering
Kyushu University, Japan
yumi@irvs.is.kyushu-u.ac.jp

Kenji Hara
Graduate School of Design
Kyushu University, Japan

Abstract

This paper presents a fast pose estimation algorithm of a 3D free form object in 2D images using 2D distance maps. One of the popular techniques of the pose estimation of 3D object in 2D image is the point-based method such as the ICP algorithm. However, the calculation cost for determining point correspondences is expensive. To overcome this problem, the proposed method utilizes a distance map on the 2D image plane, which is constructed quite rapidly by the Fast Marching Method. For pose estimation of the object, contour lines of the 2D image and the projection of the 3D object are aligned using the distance map iteratively by the robust M-estimator. Some experimental results with simulated models and actual images of the endoscopic operation are successfully carried out.

1. Introduction

Pose estimation problem of a 3D free form object in a 2D image is one of the fundamental problems in the fields of computer vision, such as object recognition, virtual reality, and texture mapping. A lot of algorithms for 2D-3D registration have been proposed so far [1]- [4], [6]- [15], [18]- [23]. Brunie [2] and Lavallee [8] utilized a pre-computed 3D distance map of a free form object for 3D pose estimation. The error metric is defined as the minimum distance between the surface of the 3D model and projection rays, and the sum of the error is minimized using the Levenberg-Marquardt method. To make the registration process efficiently, 3D distance from the surface is precomputed and stored with the octree structure. Zuffi [23] applied this algorithm for the pose estimation of a knee joint in a single X-ray image for total knee replacement surgery.

For the alignment of a 2D color image taken by a color sensor and a 3D geometric model measured by a range sensor, Viola proposed a technique based on a statistical method [22]. This method evaluated the mutual information between the 2D image and the image of the 3D model based on the distribution of intensity. Allen [19] also proposed a method using 3D linear edges and the 2D edges. These methods work well on the surface with little albedo variance. But these are easily trapped in a local minimum on surfaces with rich textures.

Kurazume [7] proposed the simultaneous registration algorithm using the reflectance image and the 2D texture images. Range sensors often provide reflectance images as side products of range images. This 2D reflectance image is aligned with the 3D range image because both images are obtained through the same receiving optical device. Therefore, we can utilize the 2D reflectance image for solving the 2D-3D registration problem instead of the original 3D model. By using the 2D reflectance image, the 2D-3D problem is modified to the simple 2D-2D problem. In [7], a number of photometric edges extracted from reflectance image and texture image were registered and the relative pose is determined using a robust M-estimator. Epipolar constraints are also introduced to estimate relative poses of multiple texture images simultaneously. Elstrom [4] extracted feature points from the reflectance image and the texture image by a corner detector, and determined correspondence between these feature points. Umeda [21] also utilized the reflectance image, but they introduced the optical flow constraint between the reflectance image and the texture image. Intrinsic and extrinsic parameters are determined using the least squares method.

On the other hand, some registration techniques using silhouette images or contour lines in 2D image plane have been proposed. Lensch [9], [10], [11] proposed a silhouette-based approach. The size of XOR regions of silhouette images of a 2D image and a 3D model is defined as the similarity measure, and the optimum pose which minimizes the size of the XOR region is determined using the Downhill Simplex method. Neugebauer [15] proposed simultaneous registration algorithm for multiple texture images and a free form object. To determine pose parameters of multiple images (6n dof: n is the number of images) and camera focal length (2 dof), three objective functions are utilized, which are i) distance between the feature point in the image and its

corresponding 3D point projected in the image, ii) distance between the outline in the image and the outline of the projected 3D model, and, iii) similarity of the image attribute at corresponding points in different images. In the 3rd criteria, the distance map is introduced as an artificial image attribute. This distance map is defined as the Euclidean distance from edges extracted by the sobel operator, and established using a 3x3 mask operation iteratively. A blend of objective functions is minimized by the nonlinear least squares method, and optimum poses of multiple images are determined simultaneously.

In contour-based approach, the error is computed as the sum of distances between points on a contour line in a 2D image and points on a projected contour line of a 3D model, and the optimum pose is determined based on the ICP algorithm [3], [14]. Though these algorithms are simple and quite robust against noise, they are computationally expensive since a number of point correspondences between two contours have to be determined for calculating the registration error.

This paper presents a new registration algorithm of a 2D color image and a 3D geometric model utilizing a 2D distance map. In this method, firstly, the boundary of an object on the 2D color image is detected and the 2D distance map from the detected boundary is constructed by the Fast Marching Method. Next, an optimum pose of the 3D model is estimated to decrease the sum of the distance value on projected contour points of the 3D model iteratively using a robust M-estimator. Our 2D-3D registration algorithm belongs to the contour-based approach. However, we utilize a distance map for evaluating a registration error instead of the point correspondence. Since there is no need to determine the point correspondence, our algorithm works faster than the conventional point-based approach, after the distance map is created. In addition, since the distance map can be established quite rapidly using the Level Set Method named the Fast Marching Method, our method applies to the realtime tracking of a moving object. In this paper, we show some fundamental experiments including tracking of internal organs in endoscopic images for a navigation system of the endoscopic operation.

2. Fast Marching Method for rapid construction of distance map

Before describing our registration algorithm, the Fast Marching Method proposed by Sethian [16], [17] is briefly introduced, which is utilized for the fast construction of a distance map in our method.

The Fast Marching Method was initially proposed as a fast numerical solution of the Eikonal equations ($|\nabla T(p)|\ F = 1$), where $T(p)$ is arrival time of a front at a point p and F is a speed function. It generally takes a long time to obtain a proper solution since this equation is solved using a convergent calculation. However, by adding restriction such that a sign of the speed function F is invariant, the Fast Marching Method solves the Eikonal equation straightforwardly, and thus, quite rapidly. In this method, the arrival time of a front at each point is determined in the order from the old to the new.

At first, the Fast Marching Method modifies the Eikonal equation into the next difference equation.

$$(\max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 +$$

$$\max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2)^{\frac{1}{2}} = 1/F_{ij} \qquad (1)$$

Next, since the arrival time is propagated in one direction from the old to the new, a point which holds the oldest arrival time in a whole region is chosen, and the arrival time of the boundary at this point is determined from Eq.(1).

The concrete procedure is as follows:

Step.1 (Initialization) The whole region is divided into a number of grid points with a proper grid width. Before starting the calculation, all the grid points are categorized into three categories (*known*, *trial*, *far*) according to the following procedure.

1. The grid points belonging to the initial front (denote the boundary hereafter) are added to the category of *known* and the arrival time of these grid points is set to 0 ($T = 0$).

2. Among the 4 neighboring grid points of a grid point belonging to the category of *known*, the one which doesn't belong to the category of *known* is categorized into to the category of *trial*, and the arrival time of this grid point is calculated temporarily from the equation, $T_{ij} = 1/F_{ij}$. In addition, the arrival time of these grid points is stored in heap data structure using the heap sort algorithm in ascending order of T.

3. The grid points except for the above grid points are all categorized into the category of *far*, and the arrival time is set to infinity ($T = \infty$).

Step.2 Choose the grid point (i_{min}, j_{min}) which is placed at the top of the heap structure. This grid point has the smallest arrival time in the category of *trial*. Remove this grid point from the category of *trial* and the heap structure, and categorize it as the category of *known*. Run "downheap" algorithm to reconstruct the heap structure.

Step.3 Among the 4 neighboring grid points $((i_{min} - 1, j_{min}), (i_{min} + 1, j_{min}), (i_{min}, j_{min} - 1), (i_{min}, j_{min} + 1))$ of the selected grid point (i_{min}, j_{min}), the one which belongs to the category of *far* is changed to the category of *trial*.

Step.4 Among the 4 neighboring grid points of the selected grid point (i_{min}, j_{min}), the arrival time of the one which belongs to the category of *trial* is calculated using Eq.(1). Then, run "upheap" algorithm to reconstruct the heap area.

Step.5 If there is a grid point which belongs to the category of *trial*, then go to Step 2. Otherwise, the process is terminated.

This Fast Marching Method constructs a distance map of a whole region quite rapidly, which indicates the distance from a boundary (an initial position of the front) to a certain point. To construct the distance map, the speed function F_{ij} in Eq.(1) is set to 1 at first. Next, the arrival time of the front T is determined using the above procedure. Since the speed function is 1, T suggests the distance from the boundary to the point. Figure 1 shows an example of the calculated distance map.
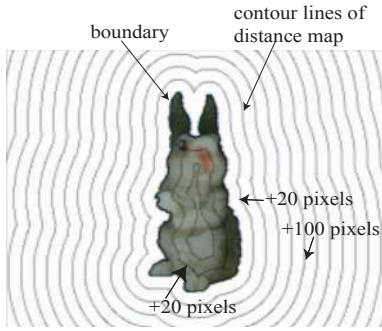

Figure 1. An example of the distance map calculation using the Fast Marching Method.

3. A New 2D-3D registration algorithm based on the distance map

This section describes our fast 2D-3D registration algorithm using the distance map in detail. We assume that a 3D geometric model of the object has been constructed beforehand, and is represented by a number of triangular patches. Camera internal parameters are also calibrated and obtained precisely.

The proposed algorithm is summarized as follows:

1. At first, a boundary of an object on the 2D color image is detected using an active contour model (ex. Snakes or the Level Set Method [5]).

2. Next, the distance map from the detected boundary on the 2D image plane is constructed using the Fast Marching Method.

3. The 3D geometric model of the object is placed at an arbitrary position and projected on the 2D image plane.

4. Contour points of the projected image and their corresponding patches on the 3D model are identified.

5. Force is applied to the selected patch of the 3D model in 3D space according to the distance value obtained from the distance map at each contour point.

6. Total force and moment around the COG (center of gravity) of the 3D model is determined using the robust M-estimator.

7. The pose of the 3D model is changed according to the total force and moment.

8. Repeat from step 1 to 7 until the projected image of the 3D model and the 2D image coincide each other.

The above procedure is explained in more details with some examples in the following sections.

3.1. Construction of distance map

Figure 2 shows the calculated distance map. At first, a boundary is extracted from the color image using the Level Set Method [17], [5]. Next, the distance map from this boundary is calculated using the Fast Marching Method explained in Section 2.
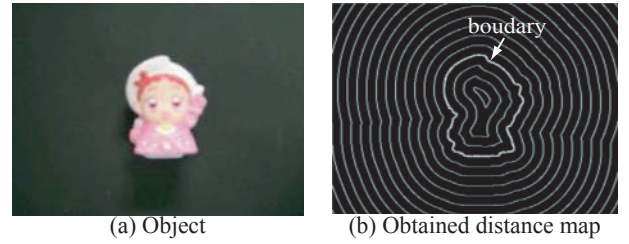

(a) Object   (b) Obtained distance map

Figure 2. Detected boundary and distance map.

3.2. Fast detection of triangular patches of contour points in 3D model

Figure 3 shows an example of the contour detection of the projected 3D geometric model. The contour detection and identifying triangular patches on the 3D model corresponding to points on the contour line are computationally expensive and time consuming. In our implementation, we utilize the high-speed rendering function of the OpenGL hardware accelerator and thus these procedures are executed quite rapidly.

The detailed algorithm is as follows: Initially, we assign different colors for all the triangular patches in the 3D model and draw the projected image of the 3D model on the image buffer using the OpenGL hardware accelerator. The contour points of the 3D model are detected by raster scanning of the image buffer. By reading colors of the detected contour points, we can identify the corresponding triangular patches on the 3D geometric model. Figure 4 shows the

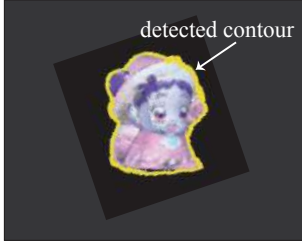color image of the projected 3D model with the patches of different colors.

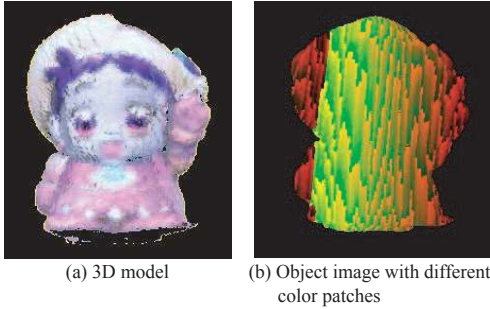**Figure 3. Contour detection of 3D geometric model.**

(a) 3D model   (b) Object image with different color patches

Figure 4. Detection of triangular patches of the contour points.

3.3. Force and moment calculation using the M-estimator and determination of optimum pose

After obtaining the distance map on the 2D image and the list of the triangular patches of the 3D model corresponding to the contour points, the force f_i is applied to the center of the triangular patches on the contour line (Figure 5). The force f_i is the vector perpendicular to the line of sight, and the projection of the f_i onto the 2D image plane coincides with the f_{DM}, which is the vector obtained by the following equation (Figure 6(a)).

$$f_{DM} = DM_{i,j} \frac{\nabla DM_{i,j}}{|\nabla DM_{i,j}|} +$$

$$k \cdot sign(n \cdot \frac{\nabla DM_{i,j}}{|\nabla DM_{i,j}|})(1 - |n \cdot \frac{\nabla DM_{i,j}}{|\nabla DM_{i,j}|}|)n \quad (2)$$

where DM_{ij} is the value of the distance map at (i, j), n is a normal vector of the projected contour point of the 3D model, k is an arbitrary constant, and $sign(\cdot)$ is a sign function which returns the sign of a number. The first term stands for the force toward the direction of the steepest descent of the distance map. We assume that the magnitude of this term is proportional to the value on the distance map . The second term stands for the force regarding the consistency of the normal vector of the contour line and the direction of the steepest descent of the distance map. The

effectiveness of this term is shown in Figure 7. This term works when the direction of the normal vector of the contour line does not coincide with the direction of the steepest descent of the distance map. Therefore, this is effective especially for a long, slender and symmetrical object, such as arms or legs of human body, since a part of normal vectors is almost perpendicular to the direction of the steepest descent during the alignment procedure as shown in Figure 7.

Then, total force and moment around the COG of the 3D model are calculated using the following equations as shown in Figure 6(b).

$$F = \sum_i \rho(f_i) \quad (3)$$

$$M = \sum_i \rho(r_i \times f_i) \quad (4)$$

where r is a vector from the COG to the triangular patch and $\rho(z)$ is a particular estimate function. Then, current position T and pose R are updated as follows;

$$T \leftarrow T + k_T F \quad (5)$$

$$R \leftarrow E_{k_R M} \cdot R \quad (6)$$

where k_T and k_R are constant gains and E_ω is the coordinate transformation matrix around the axis of ω. In practical scenario, a part of the object is occasionally occluded by the other parts, or the 2D image is corrupted by noise. In these cases, the obtained boundary does not coincide with the projected contour of the 3D model and the correct distance value cannot be obtained. Therefore, we introduces the robust M-estimator to ignore contour points with a large amount of errors. The robust M-estimator is generally utilized to reduce the effect of outliers by the weight estimation. By applying this method to the our registration algorithm, the registration process is executed stably even if a part of the contour of the 2D image is occluded and there are no corresponding points of the projected contour of the 3D model on the 2D image.

Let's consider the force f_i and the moment $r_i \times f_i$ as an error ϵ_i. Then, we modify Eqs.(3) and (4) as

$$E(P) = \begin{pmatrix} F \\ M \end{pmatrix} = \sum_i \rho(\epsilon_i) \quad (7)$$

where P is the pose of the 3D geometric model. The pose P which minimizes the $E(P)$ is obtained as the following equation.

$$\frac{\partial E}{\partial P} = \sum_i \frac{\partial \rho(z_i)}{\partial z_i} \frac{\partial z_i}{\partial P} = 0 \quad (8)$$

Here, we define the weight function $w(z)$ as the following equation in order to evaluate the error term.

$$w(z) = \frac{1}{z} \frac{\partial \rho}{\partial z} \quad (9)$$

From the above equation, we obtain the following weighted least squares method.

$$\frac{\partial E}{\partial P} = \sum_i w(z_i) z_i \frac{\partial z_i}{\partial P} = 0 \qquad (10)$$

In our implementation, we adopt the Lorentzian function for the estimation function $\rho(z)$, and gradually minimize the error E in Eq.(7) by the steepest descent method.

$$\rho(z) = \frac{\sigma^2}{2} \log(1 + (z/\sigma)^2) \qquad (11)$$

$$w(z) = \frac{1}{1 + (z/\sigma)^2} \qquad (12)$$

The pose P which minimizes the error in Eq.(7) is the estimated relative pose between the 2D image and the 3D geometric model.

Figure 8 shows an example of the total force F, the total moment M, and the registration error in the registration process. As shown in this figure, the force and the moment gradually decrease according to the decrease of the registration error. We repeated registration experiments for a number of objects with various shapes, and verified that the registration error is minimized by applying the force and the moment defined in section 3.3 in most of the cases.
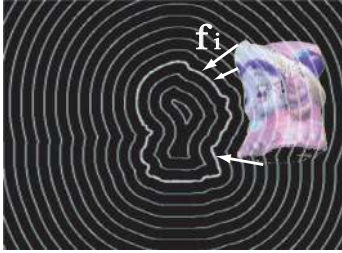

Figure 5. Applying the force f to all the triangular patches of the contour points.

3.4. Coarse-to-fine strategy and the "distance band"

Though the 2D distance map can be constructed efficiently by the Fast Marching Method, computing the distance value in whole region of the color image is wasteful. Therefore we adopt coarse-to-fine strategy and a band-shaped distance map named a distance band. The distance band is a narrow band along the image boundary and the precise distance map is constructed in this region using the Fast Marching Method. In other region except the distance band, the distance value is roughly estimated as the distance from the center of the image boundary. In the beginning of the registration process, we use a coarse image and a wide distance band in order to align the 2D image and the 3D model roughly. After several iterations, we change the image to the fine one and compute the distance band with narrower width. In experiments, we used the distance band
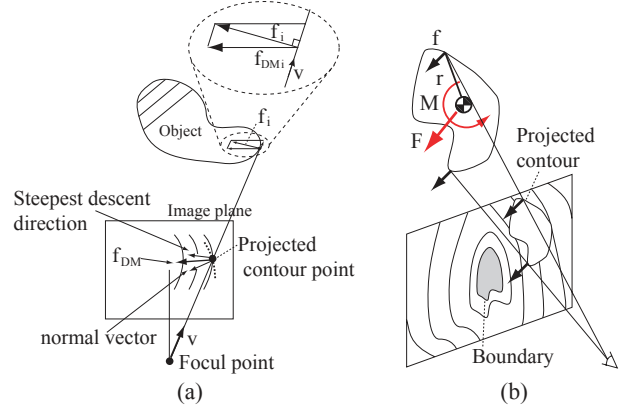

Figure 6. Force and moment around COG.

with 10 pixels and three images with difference resolution, that is, 160x120, 320x240, 640x480. Computation time of the distance band in each image is shown in Table 1.

Table 1. Computation time of the distance band by the Fast Marching Method.

Image size	Whole region [ms]	Distance band (10 pixels) [ms]
160x120	6.9	0.48
320x240	32.8	1.1
640x480	189.1	2.7

3.5. Characteristics of our method

Characteristics of our method are summarized as follows:

1. The computational cost is expensive in the conventional contour based approach, since a number of point correspondences have to be determined for calculating the registration error. However, after the distance map is created, our algorithm works faster than the conventional point-based approach, since there is no need to determine the point correspondence.

2. The Fast Marching Method is utilized to construct a distance map of a whole region quite rapidly.

3. The larger the number of points on the boundary becomes, the longer processing time is required for the conventional point-based approach. On the other hand, total processing time of the proposed method is almost constant even if the number of points on the boundary increases.

4. Experiments

In this section, we show some fundamental results of the registration and tracking experiments using simulated images of a doll as shown in Figure 3 and actual images of internal organs for the endoscopic operation.
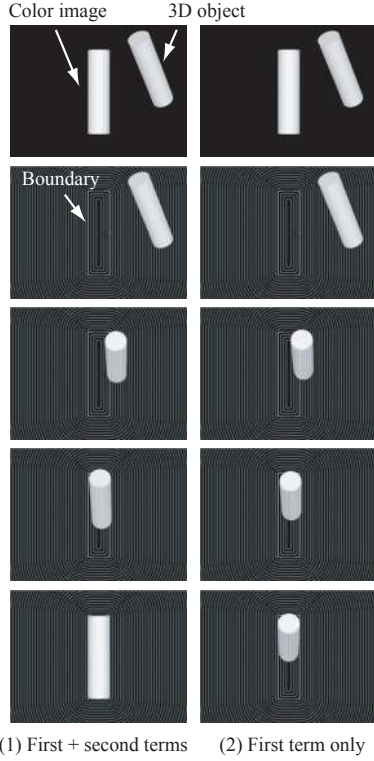
Figure 7. Effect of Eq.2

(1) First + second terms   (2) First term only

Figure 8. Total moment, total force and average of registration error of the registration process.

4.1. 2D-3D registration of simulated images

Firstly, Figure 9 shows how the proposed 2D-3D registration algorithm works with a model of a small doll (a height of 5cm).

The computation time of the force f_i for one patch on the 3D model is 0.30 $[\mu s]$ for Pentium IV processor, 3.06 GHz. The total processing time is 9.6 $[ms]$ for one update period including projected contour detection, calculation of force and moment, and the steepest descent method for Pentium IV processor, 3.06 GHz. The average and the variance of the registration error between the boundary and the projected contour are 1.19[pixels] and 0.90[pixels^2] respectively after 60 iterative calculation. Here, the number of points on the boundary is 652, image size is 640×480 pixels, and the 3D model is composed of 73,192 meshes with 219,576 vertices. On the other hand, the computation time of the force f_i for the conventional point-based method [7] is 1.15 $[\mu s]$, which uses the k-D tree structure to search the nearest points between the contour points of the 2D image and the projected 3D model.

Table 2 show the comparison of the processing time for several cases in which different numbers of points on the boundary were used. From these results, it is clear that the processing time of the proposed method is almost constant even if the number of points on the boundary increases. On the other hand, the point-based approach requires 3 to
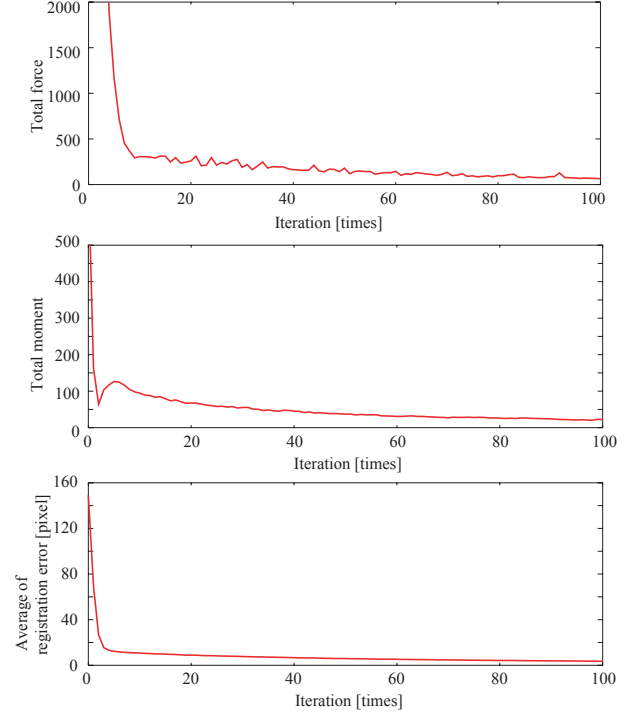
7 times longer processing time than the proposed method. Moreover, the larger the number of points becomes, the longer processing time is required for the point-based approach. Therefore, it is verified that the proposed 2D-3D registration algorithm has an advantage comparing with the conventional point based approach in views of execution time, especially when there are a large number of points on the boundary.

Next example is the tracking experiment of the moving object. Figure 10 shows the tracking results of the doll which moves on the image plane. As seen in this example, the 2D-3D registration can be performed even if the object moves up to 20.5 pixels/second.

Though the 3D model shown in Figure 4 has a colored texture, our method can be applied to any 3D models without texture since it does not use any color information on 3D models. The example of mapping the camera image instead of the original texture is shown in Figure 11.

4.2. 2D-3D registration using actual images of the endoscopic operation

We carried out a fundamental experiment for a navigation system of the endoscopic operation. In this system, a 3D organ model, which is constructed by CT or MRI images, is superimposing endoscopic images to guide opera-
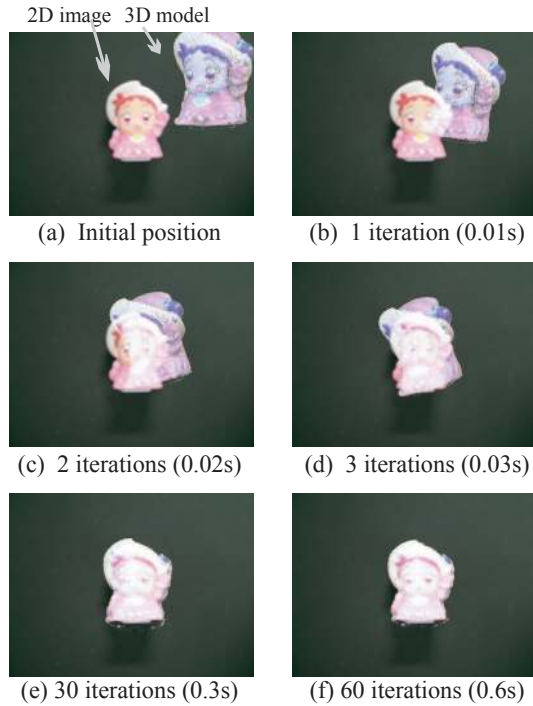
(a) Initial position   (b) 1 iteration (0.01s)

(c) 2 iterations (0.02s)   (d) 3 iterations (0.03s)

(e) 30 iterations (0.3s)   (f) 60 iterations (0.6s)

Figure 9. 2D-3D registration of simulation images.

Table 2. Comparison of processing time for one patch

Number of points on boundary	Proposed method $[\mu s]$	Point-based method [7] $[\mu s]$
628	0.30	1.15
1265	0.30	1.50
1868	0.30	1.70
2490	0.30	2.22

(a) t = 0 s   (b) t = 7.7 s

(c) t = 12.8 s   (d) t = 18.1 s

Figure 10. An example of moving object tracking.

Figure 11. An example of mapping the camera image on the 3D model.

tion procedure.

Figure 12 shows the tracking results of the gallbladder in video images of the endoscopic operation. Figure 12(a) and Figure 12(b) show the actual image and the 3D model taken by the endoscope and CT scanner, respectively. Figure 12(c) shows the experimental results of the tracking of the gallbladder image. As seen in this result, our algorithm is able to track the gallbladder image even if the position of the gallbladder is changed by the operation of the endoscope or respiratory motion effect.

5. Conclusion

This paper described a new registration algorithm of 2D color images and 3D geometric models. This method utilizes the 2D images and their distance map created by the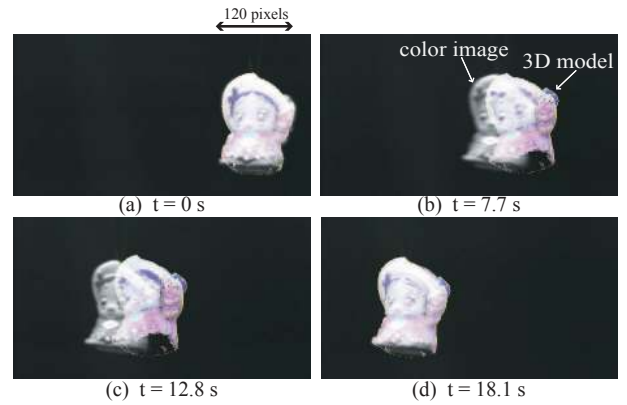 Fast Marching Method, and determines a precise relative pose between 2D images and 3D models using the robust M-estimator.

Our registration algorithm works faster than the conventional point-based approach by the use of the distance map. In addition, the distance map can also be constructed quite rapidly using the Fast Marching Method. The efficiency of the proposed algorithm was verified through the fundamental experiments using simulation images and actual images of internal organs for a navigation system of the endoscopic operation.

References

[1] S.A. Banks and W.A. Hodge: *Accurate measurement of three-dimensional knee replacement kinematics using single-plane fluoroscopy*, IEEE Trans. on Biomedical Engineering,43(6):638–648, 1996.
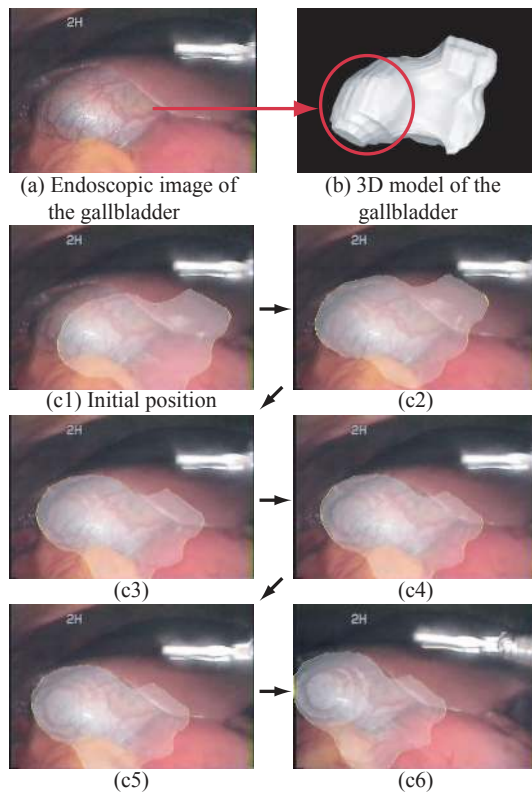
(a) Endoscopic image of the gallbladder

(b) 3D model of the gallbladder

(c1) Initial position

(c2)

(c3)

(c4)

(c5)

(c6)

Figure 12. 2D-3D registration in endoscopic video images of the gallbladder.

[2] L. Brunie, S. Lavallee and R. Szeliski: *Using force fields derived from 3D distance maps for inferring the attitude of a 3D rigid object*, Proceedings of the Second European Conference on Computer Vision, 670–675, 1992.

[3] Q. Delamarre and O. Faugeras: *3D Articulated Models and Multi-View Tracking with Silhouettes*, Proc. of the International Conference on Computer Vision, 2:716–721, 1999.

[4] M.D. Elstrom and P.W. Smith: *Stereo-Based Registration of Multi-Sensor Imagery for Enhanced Visualization of Remote Environments*, Proc. of the 1999 IEEE International Conference on Robotics and Automation, 1948–1953, 1999.

[5] Y. Iwashita, R. Kurazume, T. Tsuji, K. Hara and T. Hasegawa: *Fast Implementation of Level Set Method and Its Realtime Applications*, IEEE International Conference on Systems, Man and Cybernetics, 6302–6307, 2004.

[6] D. J. Kriegman and Jean Ponce: *On Recognizing and Positioning Curved 3-D Objects from Image Contours*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(12):1127–1137, 1990.

[7] R. Kurazume, K. Noshino, Z. Zhang and K. Ikeuchi: *Simultaneous 2D images and 3D geometric model registration for texture mapping utilizing reflectance attribute*, Proc. of Fifth Asian Conference on Computer Vision (ACCV), 99–106, 2002.

[8] S. Lavallee and R. Szeliski: *Recovering the Position and Orientation of Free-Form Objects from Image Contours Using 3D Distance Maps*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(4):378–390, 1995.

[9] H. Lensch, W. Heidrich and H.-P. Seidel: *Automated Texture Registration and Stitching for Real World Models*, In Pacific Graphics '00, 317–326, 2000.

[10] H. Lensch, W. Heidrich and H.-P. Seidel: *Hardware-accelerated silhouette matching*, In SIGGRAPH Sketches, 2000.

[11] H. Lensch, W. Heidrich and H.-P. Seidel: *A Silhouette-Based Algorithm for Texture Registration and Stitching*, Graphical Models, 63:245–262, 2001.

[12] D.G. Lowe: *Fitting parametrized 3D models to images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 13(5):441–450, 1991.

[13] C.-P. Lu, G.D. Hager and E. Mjolsness: *Fast and Globally Convergent Pose Estimation from Video Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(6):610–622, 2000.

[14] K. Matsushita and T. Kaneko: *Efficient and handy texture mapping on 3D surfaces*, Comput. Graphics Forum 18, 349–358, 1999.

[15] P.J. Neugebauer and K. Klein: *Texturing 3d models of real world objects from multiple unregistered photographic views*, Comput. Graphics Forum 18, 245–256, 1999.

[16] J. Sethian: *A fast marching level set method for monotonically advancing fronts*, Proc. of the National Academy of Science, 93:1591–1595, 1996.

[17] J. Sethian: *Level Set Methods and Fast Marching Methods, second edition*, Cambridge University Press, UK, 1999.

[18] I. Stamos and P. Allen: *3D Model Construction Using Range and Image Data*, Proc. of CVPR 2000, 531–536, 2000.

[19] I. Stamos and P.K. Allen: *Integration of Range and Image Sensing for Photorealistic 3D Modeling*, Proc. of the 2000 IEEE International Conference on Robotics and Automation, 1435–1440, 2000.

[20] S. Sullivan, L. Sandford and J. Ponce: *Using geometric distance fits for 3D object modeling and recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(12):1183–1196, 1994.

[21] K. Umeda, G. Godin and M. Rioux: *Registration of Range and Color Images Using Gradient Constraints and Range Intensity Images*, Proc. of 17th International Conference on Pattern Recognition, 12–15, 2004.

[22] P. Viola and W.M. Wells III: *Alignment by maximization of mutual information*, Int. J. of Computer Vision, 24(2):137–154, 1997.

[23] S. Zuffi and A. Leardini and F. Catani and S. Fantozzi and A. Cappello: *A Model-Based Method for the Reconstruction of Total Knee Replacement Kinematics*, IEEE Trans. on Medical Imaging, 18(10):981–991, 1999.