Fast Implementation of Level Set Method and Its Real-time Applications *

Yumi Iwashita, Ryo Kurazume,
Tokuo Tsuji and Tsutomu Hasegawa
Department of Intelligent Systems
Kyushu University
Fukuoka City, JAPAN
{yumi,kurazume,tsuji,hasegawa}@irvs.is.kyushu-u.ac.jp

Kenji Hara
Department of Visual Communication Design
Kyushu University
Fukuoka City, JAPAN
hara@design.kyushu-u.ac.jp

Abstract – *The level set method (LSM) has been widely used for various applications such as motion tracking and 3D geometrical modeling. However, the calculation cost of reinitialization and updating of an implicit function is considerably expensive as compared with conventional active contour models such as "Snakes". To tackle this problem, we propose an efficient algorithm of the LSM named the Fast Level Set Method(FLSM). This paper introduces some experiments based on the FLSM, including 2D real-time tracking of moving objects in video images, and 3D simultaneous motion capture system of multiple targets using stereo range images.*

Keywords: Active contour, Snakes, Deformable surfaces, Level Set, Motion capture, Tracking.

1 Introduction

The level set method(LSM) proposed by S. Osher and J. A. Sethian[7] has attracted much attention as a method that realizes a topology free active contour modeling by detecting and tracking a boundary. Various applications have been proposed in the past including motion tracking[5] and 3D modeling[8][9].

However, the calculation cost of reinitialization and updating of the implicit function is considerably expensive as compared with conventional active contour models such as "Snakes"[2]. Thus, this has been considered as a heavy burden for applying the level set method in real-time applications such as motion tracking in video images. To overcome this problem, this paper presents a new high speed execution technique of the LSM named the Fast Level Set Method (FLSM), which makes it possible to apply the LSM for real-time applications. This paper introduces two examples of the FLSM applications: 2D real-time tracking of moving objects in video images and 3D real-time tracking of multiple human bodies using stereo range images.

2 The Level Set Method

The LSM utilizes an implicit function Φ which is defined in a space one dimensional higher than that of where a contour of interest is described. This function Φ, which is defined as a distance function from a current contour in general, is updated according to a next PDE (Partial Differential Equation).

$$\Phi_t = -F(\kappa) \mid \nabla\Phi \mid \tag{1}$$

where, κ is a local curvature of Φ, and F is a speed function. The contour to be tracked is detected as the cells with a value of zero of the implicit function (zero level set), that is, the contour line of $\Phi = 0$. In the implementation of the LSM, the space is uniformly split by cells, and Eq.(1) is solved iteratively using numerical schemes such as the upwind scheme.

To solve Eq.(1), the speed function $F(\kappa)$ has to be determined at each cell for every update process of Φ. The distribution of the $F(\kappa)$, which is known as the extension velocity field[1], is constructed as follows: i) at the current zero level set cell, F is calculated according to the intensity of the current image at first; ii) next, at each cell except the zero level set cell, the speed function F is copied from the nearest zero level set cell. However, finding the nearest zero level set cell needs large calculation cost.

In addition, since integral errors are accumulated during the calculation of updating, the reinitialization process is required in which the proper quantity (the distance from the current zero level set cell) at each cell is re-calculated periodically after several update processes. The constructed field is called the distance field. However, this reinitialization process is time consuming, because the nearest zero level set has to be determined at each cell. Therefore, this is also a major obstacle for the high speed implementation of the LSM.

To overcome these problems, several techniques have been proposed in the past, such as the Narrow Band Method, the Fast Marching Method, SFA (Sparse Field Algorithm), and HERMES. The most popular and efficient method is the one

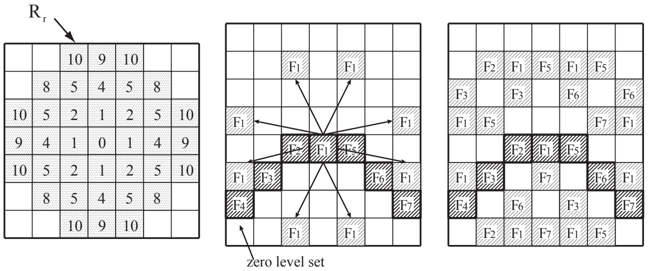
---

*0-7803-8566-7/04/$20.00 © 2004 IEEE.

proposed by Adalsteinsson and Sethian[1] in 1999, which combines the Narrow Band Method and the Fast Marching Method. The calculation cost of this method is $O(n^2 \log n)$ (n is the number of cells along a side of the voxel space in 3D space).

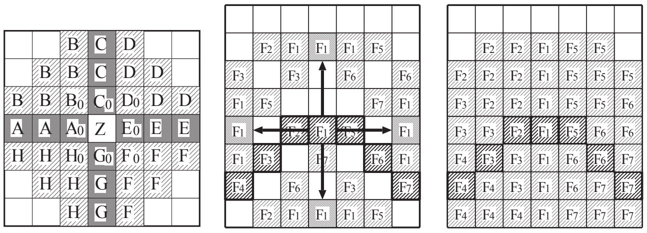
3 Fast Level Set Method

In this section, we propose a new efficient algorithm named the Fast Level Set Method (FLSM) with the calculation cost of $O(n^2)$. The key idea of the proposed FLSM is the use of a reference map (Figure 1(a)). This map indicates classification in which each cell is categorized according to a distance from a center cell. For example, the class R_r consists of cells which are located $\sqrt{r}$ away from the center cell.



		10	9	10		
	8	5	4	5	8	
10	5	2	1	2	5	10
9	4	1	0	1	4	9
10	5	2	1	2	5	10
	8	5	4	5	8	
		10	9	10		

(a) Refence map (1) Rr = 10 (2) Rr = 10



		B	C	D		
	B	B	C	D	D	
B	B	B_0	C_0	D_0	D	D
A	A	A_0	Z	E_0	E	E
H	H	H_0	G_0	F_0	F	F
	H	H	G	F	F	
		H	G	F		

(3) Rr = 9 (4) Final result

(b) Divided reference map (c) Procedure of constructing extension velocity field

Figure 1: Reference map and the construction process of the extension velocity field.

The extension velocity field is constructed efficiently using the reference map. Here, we assume that the speed function at the zero level set cell has already been determined. At first, one zero level set cell is chosen. Then, using the class $R_{\delta(\delta+1)}$ in the reference map, all cells which are located $\sqrt{\delta(\delta+1)}$ away from this zero level set cell are selected, and the speed function at the zero level set cell is registered to these cells tentatively. This procedure is repeated for all the zero level set cells. Next, the same procedure is performed using the next class $R_{\delta(\delta+1)-1}$. This process is repeated until all the classes in the reference map are investigated. At the end of the registration processes for all the classes, the speed function of the nearest zero level set cell is registered at each cell, and the extension velocity field is constructed consequently(Figure 1(c)).

In this process, the speed function is registered in order of a distance from the zero level set. Thus, by overwriting the distance stored in the reference map at the same time, the distance field can also be created. Additional cost by this reinitialization process is only memory access time, and total calculation cost is the almost same as the one without the reinitialization process. Therefore, the reinitialization process can be performed frequently, even up to every update process, with trivial calculation cost.

Moreover, the method can be improved by using a divided reference map(Figure 1(b)). For example, let us consider a zero level set cell Z. If a left cell A_0 adjacent to the cell Z is also a zero level set cell, there must be no cells which are nearer to Z than A_0 in the left side region of the cell Z (A,B, and H). Therefore, by skipping the registration process in these regions, it is possible to reduce the number of useless overwriting and execute the construction process of the extension velocity field more efficiently.

The comparison of the calculation cost in 3D ($n \times n \times n$) is shown in Table 1. The cost of the conventional level set method using the fast marching method[1] is $O(n^2 \log n)$. On the other hand, the calculation cost of our method is $O(n^2)$. Moreover, the additional cost of the reinitialization process is trivial in our method.

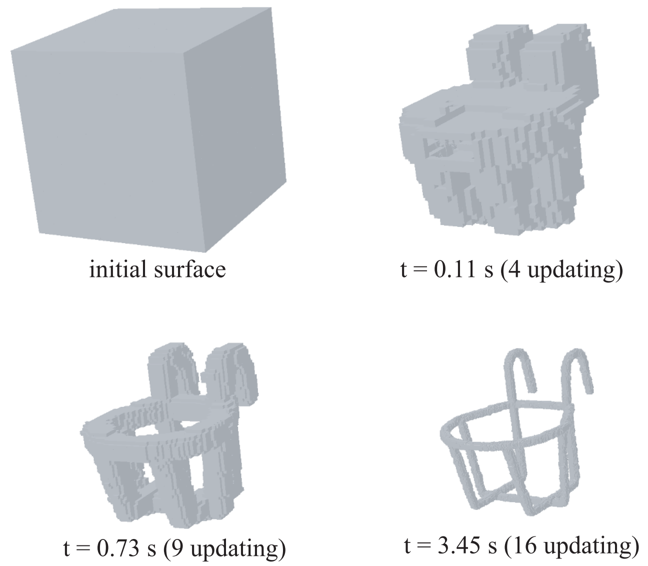
Table 1: Comparison of calculation costs. $\delta(\ll n)$ is the width of the narrow band.

	LSM	FLSM
Construction of velocity field	$O(\delta n^2 logn)$	$O(\delta^3 n^2)$
Reinitialize	$O(\delta n^2 logn)$	-
Updating process	$O(\delta n^2)$	$O(\delta n^2)$
Detection of zero level set	$O(\delta n^2)$	$O(\delta n^2)$

Figure 2 shows the 3D surface construction process of a wired basket. The range data consists of 5530 points. Comparison of calculation time is shown in Figure 3 for two conventional level set methods, that is, the no extension and extension velocity fields[1], and the proposed FLSM. We performed several experiments for various 3D models with complex shapes. These results suggest that the FLSM is executed two or three times faster that the methods proposed so far.

4 2D real-time tracking of moving objects in video images

We developed a real-time contour detection and tracking system of moving objects in video images using the FLSM. In this system, the size of the image is 320 × 240 pixels, and the speed of image acquisition is 30 frames/second. The



initial surface

t = 0.11 s (4 updating)

t = 0.73 s (9 updating)

t = 3.45 s (16 updating)

Figure 2: The process of 3D surface reconstruction.

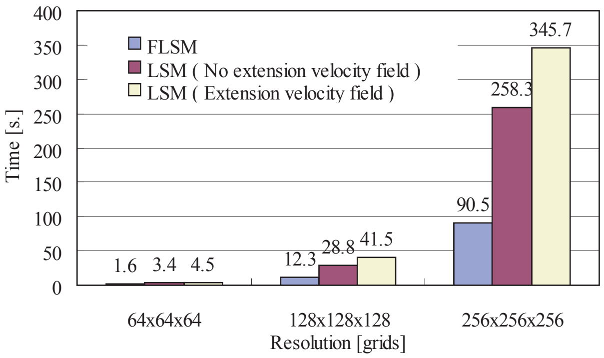


Figure 3: The comparison of the processing time (Pentium IV, 2.0GHz).

process of the FLSM is calculated at 60 frames/second using Pentium IV 2.0 GHz processor.

The speed function F is determined as

$$F(\kappa) = k(a - b\kappa) \tag{2}$$

where κ is the local curvature of the implicit function $\Phi(x, y, t)$, and a and b are constant values. k is determined by intensity $I(x, y)$ and its subtraction of the background image $D(x, y) = I(x, y) - I_{org}(x, y)$ as

$$k = \frac{\pm F_c}{1 + min(\mid \nabla I(x,y) \mid, \mid \nabla D(x,y) \mid)} \tag{3}$$

The maximum value of the speed function, F_c, was 1 cell per 1 update in the experiments.

Figure 4 shows the results of detecting and tracking multiple objects in real scene. This shows that boundary detection and real-time tracking of moving objects which move up to 100 pixels/second are successfully realized. Though the detected contours are separated at first, they are merged to a single contour when one object occludes the other. When the objects split, the contours separate again.

t = 2.93 s

t = 3.70 s

t = 4.53 s

t = 4.76 s

Figure 4: Simultaneous tracking of multiple objects.

5 3D real-time tracking of multiple human bodies

We also implemented the proposed method for the 3D tracking of human bodies. The multi-view stereo techniques[6] and the visual hull[3][4] are well-known 3D all-round reconstruction methods of a real object using multiple cameras. However, when two or more objects are in the scene at once, the reconstruction process of these objects separately is quite difficult because of the mutual occlusion. In addition, for the visual hull technique, to extract a silhouette of the object robustly is quite laborious, and reconstruction of a non-convex object is also impossible.

This paper introduces a prototype model of a 3D motion tracking system of multiple targets using the FLSM with two stereo cameras. Characteristics of this system are as follows: i) By maintaining shapes of objects obtained previously, closed and smooth surfaces are given even if parts of the depth images cannot be captured temporarily. Therefore, though the mutual occlusion occurs when multiple objects are in the scene, we can reconstruct each object separately. Thus, the proposed system is considered to be robust for mutual occlusion; ii) by using the FLSM, complex models with arbitrary topology can be reconstructed; iii) by restricting the maximum value of the speed function and updating of an implicit function frequently, the proposed system is robust

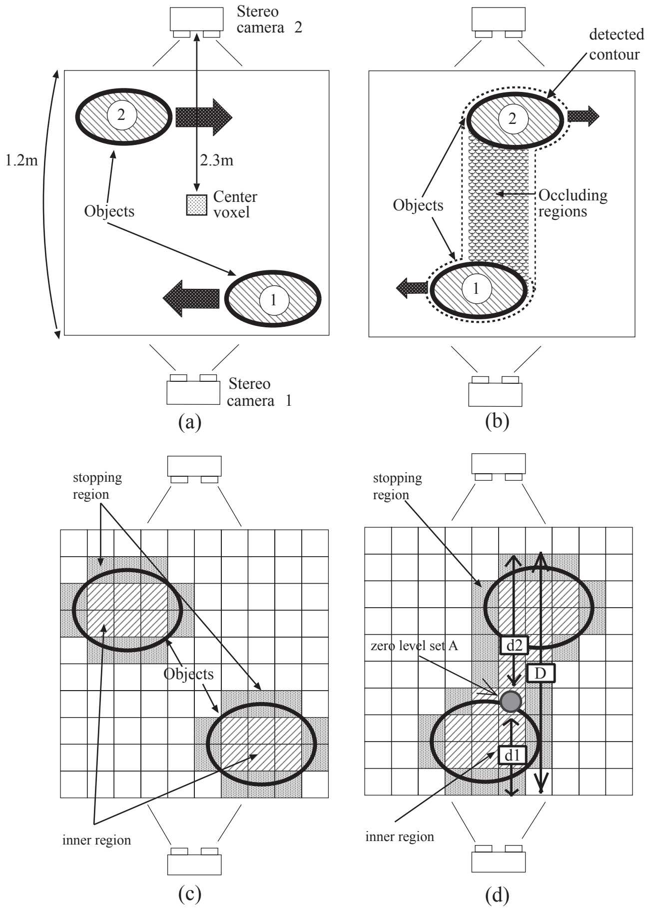


Figure 5: Occluding boundary.

for spike noise within stereo range data; iv) the calibration process of mutual camera position is quite easy by utilizing iterative registration algorithms such as ICP algorithm for obtained 3D points.

5.1 Tracking of multiple targets using the FLSM

Figure 5(a) shows the camera and workspace configuration of our system. Two stereo cameras(Point Grey Research, Bumblebee) are placed at a distance of $2.3m$ from the center of the workspace. The workspace is divided into small voxels (hereafter denoted the voxel space). After capturing depth images using stereo cameras, multiple depth images are combined in the voxel space.

Each voxel in the voxel space can hold one of two status, IN(inner human body) or OUT(outer human body). At first, IN is assigned to all voxels. Then, each pixel (x,y) of a depth image is backprojected onto the voxel space, and OUT is voted until the distance from the camera is equal or smaller than the depth value of (x,y). This process is repeated for all depth images, and a region entirely composed of voted IN is extracted, named a region-S .

Next, we apply the FLSM for the region-S. Here, we define voxels of the region-S which are adjacent to voxels voted OUT as stopping region, and the rest of the region-S as inner region(Fig.5(c)). A closed and nonintersecting initial surface including the voxel space is set at first, and it is shrunk and split until the front reaches to the stopping region.

If more than two objects are in the scene, mutual occlusion might occur between them(Fig.5(b)). In addition, a part of the body may be occluded by his hands or legs. In these cases, a connected 3D model shown as a dotted line in Fig.5(b) is reconstructed along to the occluding boundary. To avoid the production of connected objects and to reconstruct each models separately, a quite small value is stored to the speed function of the cells which are considered to be located in the occluding regions. To judge whether a cell is in the occluding regions or not, we use a significantly simple method in this approaches. Here, $d1$ and $d2$ are distances from inside zero level set cells at previous time to the outside zero level set cells as shown in Fig.5(d). D is the sum of $d1$ and $d2$, and rd is the distance radio defined as the following equation;

$$r_d = \frac{min(d1, d2)}{D} \tag{4}$$

The procedure for determining $F(\kappa)$ is as follows:

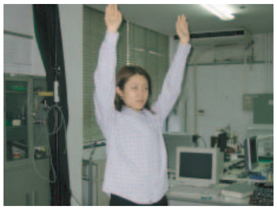
1. When zero level set voxels are in the stopping region, $k = 0$;
2. When zero level set voxels are in the inner region and $D > \alpha$;
    (a) $r_d > \beta$, then $k = C_0$
    (b) $r_d \leq \beta$, then $k = C_1$
3. When zero level set voxels are in the inner region and $D \leq \alpha$, $k = C_1$;
4. When zero level set voxels are not in the stopping and inner region, $k = -C_2$.

Here, $C_0 \ll C_1 < C_2$, and we used $\alpha = n/2$, $\beta = n/8$ in experiments. By this procedure, the position of the previous zero level set cells are maintained and two separate region can be extracted even if mutual occlusion occurs.

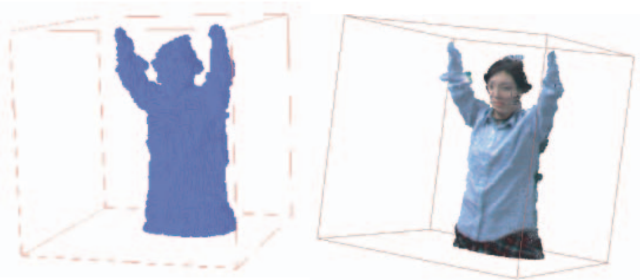
5.2 Experiments

The experiments of detecting and tracking multiple human bodies using the proposed system are carried out. In the experiment, the size of the depth images is 640 × 480 pixels, and the FLSM calculation is done by the Pentium IV PC, 2.8 GHz.

Figure 6 shows an example of the 3D modeling of a human body. We examined two kinds of resolutions of the voxel space, that is, $50 \times 50 \times 50$ voxels and $200 \times 200 \times 200$ voxels.



(a) Object



(b) Voxel data (200x200x200)

(c) Texture mapping

Figure 6: 3D reconstruction of human body.

The processing cycle is 15.4 Hz($50 \times 50 \times 50$ voxels), and 0.84 Hz($200 \times 200 \times 200$ voxels), respectively.

Figure 7 shows an example of the 3D motion tracking of two human bodies with a box. The resolutions of the voxel space is $50 \times 50 \times 50$ voxels. Though the detected surfaces are separated at first, they are merged to a single surface when passing the box. After the box is passed, the surfaces separate again.

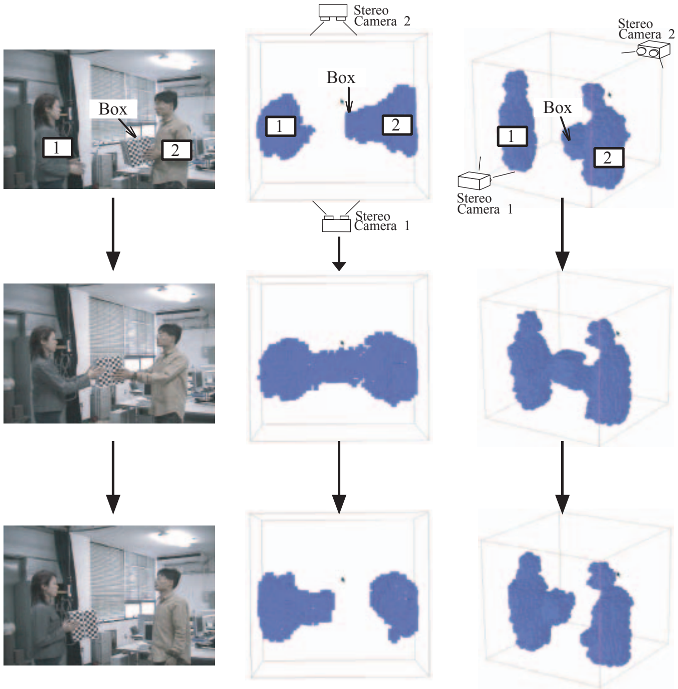
Figure 8 shows the examples when 2 people walk across a scene. Though the mutual occlusion between them occurs when one of them is occluded by the other, separate shapes are detected and kept tracking. These examples suggest that the proposed system is capable of detecting and tracking multiple human bodies in a scene individually even if a part of them is occluded.

6 Conclusions

We have proposed a new high speed execution technique of the level set method named the Fast Level Set Method(FLSM). The efficiency of the FLSM was verified through 2D real-time tracking of moving objects in video images and 3D real-time tracking of multiple human bodies in stereo range images. Our future work includes the implementation of the 3D real-time tracking system on a PC-cluster, and the development of the high speed and high quality 3D real-time tracking system.

Acknowledgment

This research was partly supported by ”Ministry of Public Management, Home Affairs, Posts and Telecommunications” and Grants-in-Aid for Scientific Research(16700185).



(a) Objects

(b) Top view

(c) Cross view

Figure 7: 3D reconstruction of multiple human bodies with a box.

References

[1] D. Adalsteinsson and J. A. Sethian: *The fast construction of extension velocities in level set methods*, J. Computational Physics, 148:2–22, 1999.

[2] M. Kass, A. Witkin and D. Terzopoulos: *Snakes, active contour models*, Int. J. Computer Vision, 1(4):321–331, 1988.

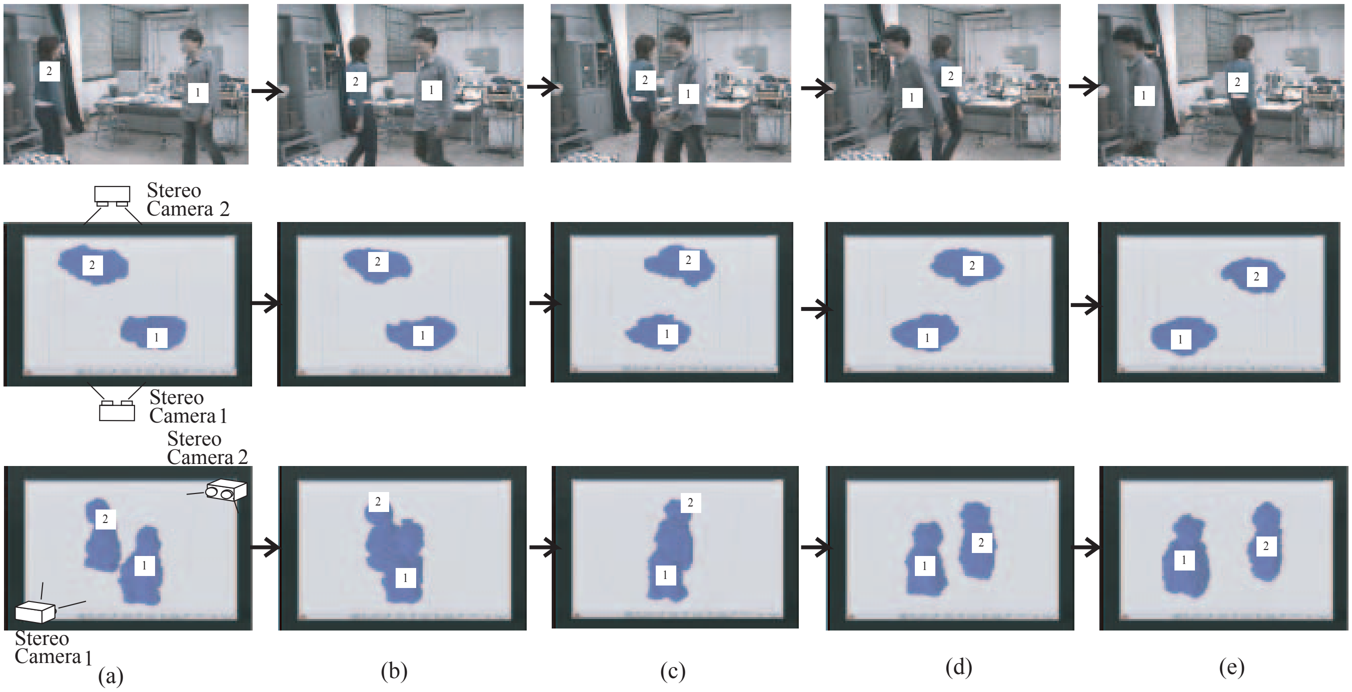
[3] A. Laurentini: *The visual hull concept for silhouette-based image understanding*, IEEE Trans. on PAMI, 16(2):150–162, 1994.

[4] W. N. Martin and J. K. Aggarwal: *Volumetric description of objects from multiple views*, IEEE Trans. on PAMI, 5(2):150–158, 1983

[5] N. Paragios and R. Deriche: *Geodesic active contotraurs and level sets for detection and tracking moving objects*, IEEE Trans. on PAMI, 22:266–280, 2000

[6] H. Saito, S. Baba, M. Kimura, S. Vedula and T. Kanade: *Appearance-based virtual view generation of temporally-varying events from multi-camera images in the 3d Room*, Proc. 3DIM99,516–525, 1999.

[7] J. A. Sethian: *Level Set Method and Fast Marching Methods*, second ed. Cambridge University Press, UK,1999



(a) (b) (c) (d) (e)

Figure 8: 3D reconstruction of multiple human bodies.

[8] S. Yui, K. Hara, H. Zha and T. Hasegawa: *A fast narrow band method and its application in topology-adaptive 3-D modeling*, Proc. ICPR02, 4:122–125, Aug, 2002

[9] H. Zhao, S. Osher and R. Fedkiw: *Fast surface reconstruction using the level set method*, Proc. 1st IEEE Workshop on Variational and Level Set Methods in Computer Vision, 2001.