Jurnal
Teknologi

Full Paper

PARAMETER IDENTIFICATION OF DEPTH-DEPTH-MATCHING ALGORITHM FOR LIVER FOLLOWING

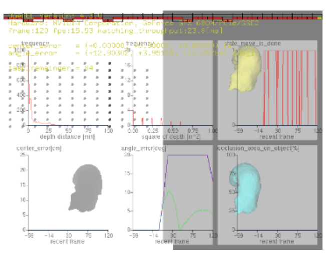
Kaoru Watanabe[a*], Masahiro Yagi[a], Kento Ota[a], Katsuhiko Onishi[a], Masanao Koeda[a], Shigeki Nankaku[a], Hiroshi Noborio[a], Masanori Kon[b], Kousuke Matsui[b], Masaki Kaibori[b]

[a]Department of Computer Science, Osaka Electro-Communication University, Kiyotaki 1130-70, Shijo-Nawate, Osaka, 575-0063, Japan
[b]Medical School, Kansai Medical University, 2-5-1 Shin-machi, Hirakata City, Osaka, 573-1010, Japan

Article history
Received
19 June 2015
Received in revised form
26 June 2015
Accepted
10 July 2015

*Corresponding author
kaoru@wtlab.osakac.ac.jp

Graphical abstract



Abstract

We proposed a depth-depth-matching algorithm as a fast motion transcription algorithm from a real liver to a virtual liver in a surgical navigation. The real is always captured by 3D depth camera, and the virtual is represented by a polyhedron with STL format via DICOM captured by MRI/CT. In our algorithm, we firstly compare a 2D depth image in a real world and the Z-buffer in a virtual world, and secondly search 3 translation and 3 rotation movements by matching both depth images in order for a virtual liver to move against a real liver. In this paper, the performance of our algorithm is ascertained in a PC simulation by changing several parameters and/or the evaluation function.

Keywords: Graphics processing unit, z-buffer, depth camera, depth image, robust estimation

Abstrak

Kami telah mencadangkan algoritma '*depth-depth matching*' adalah sebagai pergerakan transkripsi algoritma yang pantas daripada 'hati nyata' ke 'hati maya' dalam navigasi pembedahan. Hati nyata selalunya diambil dengan kamera kedalaman 3D manakala hati maya dibentangkan daripada polihedron dalam format STL melalui DICOM yang diambil dengan menggunakan MRI/CT. Di dalam algoritma kami, pertama, kami bandingkan imej kedalaman 2D di dalam dunia nyata dan penampan-Z di dalam dunia maya, kedua, pencarian 3 penterjemahan dan 3 gerakan putaran dengan memadankan kedua-dua imej iaitu hati maya ditindihkan ke hati nyata. Di dalam kertas kerja ini, prestasi algoritma kami ditentukan di dalam simulasi PC dengan mengubah beberapa parameter dan/atau fungsi penilaian.

Kata kunci: Graphics processing unit, z-buffer, kedalaman kamera, kedalaman imej, anggaran teguh

1.0 INTRODUCTION

In the surgical navigation, we firstly need the following two functions: (1) A virtual liver should be coincident with its real liver by a medical doctor in a surgical operation room. The virtual liver is converted from DICOM captured from the real liver by CT/MRI. (2) After that, while a real liver moves by a lot of doctor's operations, its virtual liver follows the real liver with the same position and orientation. In our research, the matching 3D real and virtual livers can be attained by searching the best coincidence of their 2D depth images in the GPU.[1-2] In this paper, we focus on the latter function and try to construct a prototype.

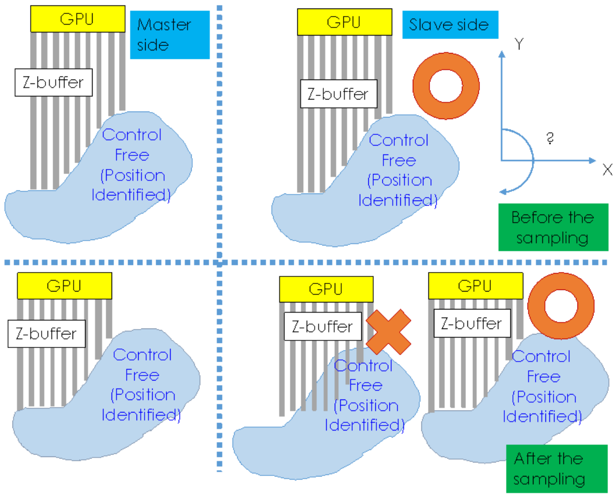


Figure 1 By matching master and slave depth images, we can find a better position/orientation of slave liver against the master liver.

So far, many peoples have designed approaches matching closest pairs of a huge number of 3D points for surface registration.[1-5] In the surgical navigation, motions of real liver are copied to motions of virtual liver of PC in an operating room is necessary between real and virtual livers. However, because 3D points are too many, the registrations are to be time consuming.

To overcome this problem, we propose a new algorithm matching a few number of corresponding 2D depths. Especially, the depth comparison can be very fast by the parallel processing of all pixels in GPU. Therefore, the registration is not to be time consuming.

In our algorithm, we successively find a position/posture which is likely to exist most by searching minimum, median, or average of difference distribution between a depth image in a real world and the Z-buffer in a virtual world. For this purpose, we propose such an algorithm and also evaluate its performance by master and slave virtual livers in PC (Figure 1). The master liver corresponds to a real liver captured by a 3D depth camera, on the other hand, the slave liver corresponds to a virtual liver calculated by the Z-buffer of GPU.

In this framework, we can design many types of 2D depth-depth comparison algorithms by changing evaluation functions such as minimum, median, average, and/or many parameters such as the number of pixels and/or the number of images.

2.0 MODELLING

In this algorithm, a liver of patient is captured as DICOM data by MRI/CT, and the DICOM data is converted into polyhedron with STL format. The reason using STL is to maintain visible quality efficiently and also to calculate a depth image as the Z-buffer in GPU. In this research, in order to evaluate our algorithm performance (position/orientation errors, calculation time, and so on), we use a master liver with STL format and also its slave liver with STL format (Figure 2).

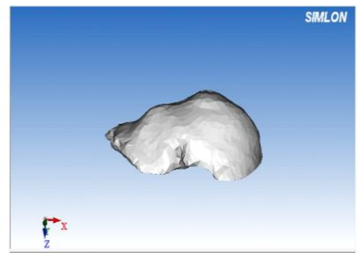


Figure 2 Master and slave virtual livers which are the same polyhedron with STL format, which is converted from DICOM captured by CT/MRI against a patient.

While a sequence of motions of master liver is completely fixed, the pixel number M and also its evaluation is changed by the minimum, median or average differences in all the selected pixels. Also, the image number N and also its evaluation is changed by the minimum, median or average differences in all the captured images.

3.0 PROPOSED EFFICIENT ALGORITHM

Our algorithm consists of the following five steps. First of all, master and slave virtual livers are initially to be the same position/posture. The underline part differs from our previous algorithm.[1]

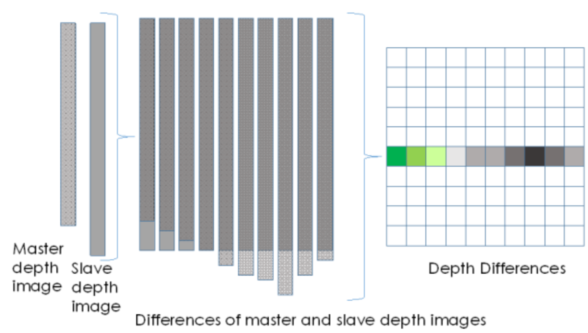


Figure 3 The sum of differences in master and slave depth images whose pixels are calculated in parallel based on Z-buffer of GPU.

[Step 1] A doctor operates a master virtual liver STL in the simulation (a real liver in the experiment).

[Step 2] The master virtual liver STL is captured as rectangular parallelepiped group by Z-buffer of GPU (The real liver is captured as rectangular parallelepiped group by the 3D depth camera) (Figure 3).

[Step 3] The slave virtual liver STL is moved to the best position/posture of 26 or 728 ones neighbouring the present position/posture (Figure 4).

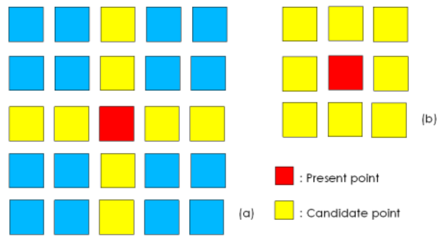


Figure 4 The least descendent algorithm always selects the best neighbor position of present one by the evaluation value explained previously. In this research, we prepare two types of neighbor candidates (a) 2*2*6= 24 or (b) 3^6-1= 728, whose dimension is 6 represented by 3 translational degree-of-freedoms and 3 rotational ones.

Each virtual liver moves among motion space with 6 degrees-of-freedom (X, Y, Z translation freedoms and pitch, yaw, roll rotation freedoms). In our algorithm (steepest descendent algorithm), we always select one of three kinds of motions such as positive step, stop, negative step along each freedom. Therefore, 26(=6*2*2) candidates or 728(=3^6-1) candidates exist around a present candidate (Figure 4).

(3-1) For each candidate, the slave virtual liver STL is captured by Z-buffer of GPU.

(3-2) For each candidate, we firstly select the minimum, median, or average of difference distribution between rectangular parallelepiped groups of master and slave Z-buffers in randomly selected pixels (Figure 3). Secondly, we select the minimum, median, or average of value distribution selected in many pairs of master and slave Z-buffers. In addition, the number of randomly selected pixels is denoted as M in the first stage, and also the number of used images is denoted as N in the second stage. Here, we change M and N as 10, 50, 100, respectively (Figure 5).

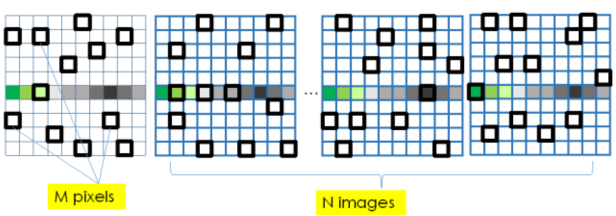


Figure 5 In our algorithm, we randomly select a set of pixels whose number is as M in each image, and calculate minimum, median, or average of difference distribution between their master and slave depths. Furthermore, we select minimum, median, or average of value distribution in many images whose number is as N.

(3-3) We select the candidate whose evaluation is to be the minimum as the next position/posture (Figure 6).

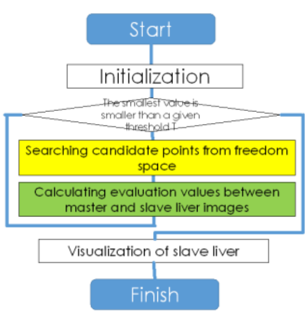


Figure 6 The least descendent algorithm always finds the best neighbor by the smallest evaluation value which is smaller than the threshold T.

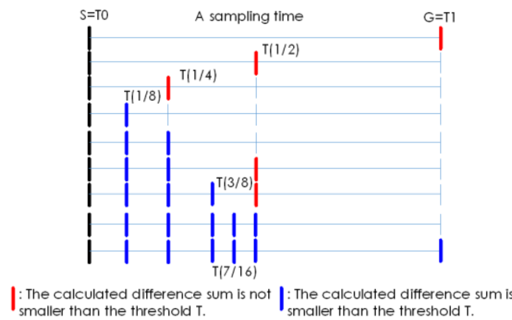


Figure 7 If the difference sum is not small enough, we divide a time interval of successive frames. Then, the steepest descendent algorithm starts recursively until the difference sum is small enough.

[Step 4] PC moves a slave virtual liver STL (a truly virtual liver) according to the selected position/posture.

[Step 5] We return to [Step 1].

In [Step 3] (the classic steepest descent algorithm), if the difference sum becomes larger than a given threshold T, the sampling time is adaptively changed smaller and smaller by the binary search of time interval, and consequently the width between present and next candidates becomes smaller and smaller (Figure 7). By this function, the smaller the sampling time is, the better the transcription precision from a master (or real) liver to a slave (or virtual) liver is. However, because the sampling time becomes smaller, following speed becomes slow. Therefore, a doctor sometimes stops his operation in the surgery.

4.0 EXPERIMENTAL RESULTS

In this section, when a doctor always gives the same and reasonable sequence of angular motions shown in the Figure 8(a) to a master virtual liver, we

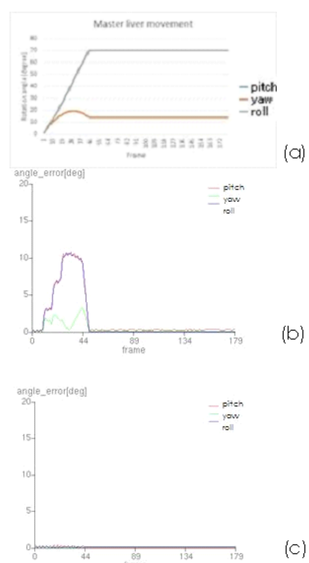


Figure 8 (a) Rotational movement of master liver, (b) a sequence of angular errors in our algorithm with 26 neighbor candidates,(c) a sequence of angular errors in our algorithm with 728 neighbor candidates.

check whether a slave virtual can precisely follow the master liver or not by using many types of our 2D depth-depth matching algorithms. First of all, we checked all the algorithms whose neighbors are 26 or 728, whose pixels are randomly selected by the minimum, median, or average value in their distribution, and simultaneously whose images are selected as the minimum, median, or average value in their distribution. Simultaneously, the number M of pixels selected randomly is changed as 10, 50, 100, and also the number N of images is changed as 10, 50, 100.

As a result, in both algorithms with 26 or 728 neighbors, the median-image-average-pixel type of depth-depth matching algorithms is totally better than the others for all the combinations of M and N concerning to speed and accuracy (Figure 8(b) and (c)). Especially, the combination (M,N)=(10,100) in 26 neighbor type and the combination (M,N)=(50,10) in 726 neighbor type are the best for the accuracy. The former's calculation time per frame is to be 19.3 milli-second within the video-rate and the letter's calculation time per frame is 77.6 milli-second which is twice or more of the video-rate (33 milli-second). This problem will be quickly solved by rapid speedup of GPU (graphics board).

caption: Average [ms]

100	19.3	19.5	19.7
50	18.9	19.1	19.3
10	19.2	18.7	18.9
	10	50	100

(a)

caption: Sum of RotateX [deg]

100	[332.45]	1012.13	891.29
50	787.68	991.08	767.44
10	756.65	638.77	944.93
	10	50	100

caption: Sum of RotateY [deg]

100	[86.78]	152.45	133.96
50	129.18	140.65	155.64
10	183.23	139.30	162.05
	10	50	100

caption: Sum of RotateZ [deg]

100	[301.98]	1001.60	874.46
50	766.20	977.05	762.96
10	728.34	620.78	944.54
	10	50	100

(b)

Figure 9 (a) Average frame rate in our median-image-average-pixel algorithms with 26-neighbors, whose pixel number M is changed along the horizontal axis and image number N is changed along the vertical axis, (b) Integration of errors in our median-image-average-pixel algorithms with 26-neighbors.

5.0 CONCLUSION

In this paper, by matching 2D depth-depth images in many types of algorithms whose pixel and image selections are changed, we improved our motion transcription algorithm faster. In succession, we evaluated all types of algorithm's efficiencies and their motion precision precisely in the high-speed GPU-based PC simulation.

In future, there are some occlusion of real liver by human body in general liver surgical operations. Therefore, when such an occlusion of real liver by human body exists, we will check whether same properties are maintained in our algorithm. Furthermore, we are now planning for capturing a real sequence of liver motions achieved by many medical doctors. If such sequences are captured and the master liver is controlled by them, we will investigate more reasonable properties in our algorithm.

Acknowledgement

This is supported in part by 2014 Grants-in-aid for Scientific Research (No. 26289069) from the Ministry of Education, Culture, Sports, Science and Technology, Japan. Also, this is supported in part by 2014 Collaborative Research Fund for Graduate Schools (A) in Osaka Electro-Communication University.

caption: Average [ms]

100	79.8	85.3	92.4
50	77.9	82.0	86.8
10	78.2	77.6	78.3
	10	50	100

(a)

caption: Sum of RotateX [deg]

100	20.13	18.67	20.93
50	79.39	21.75	20.95
10	24.04 [	18.27]	18.53
	10	50	100

caption: Sum of RotateY [deg]

100	27.06	25.99	25.53
50	38.16	33.77	25.53
10	24.45 [	18.57]	20.91
	10	50	100

caption: Sum of RotateZ [deg]

100	19.50	18.31	21.27
50	73.05 [	18.23]	21.91
10	21.09	24.07	21.95
	10	50	100

(b)

Figure 10 (a) Average frame rate in our median-image-average-pixel algorithms with 728-neighbors, whose pixel number M is changed along the horizontal axis and image number N is changed along the vertical axis, (b) Integration of errors in our median-image-average-pixel algorithms with 728-neighbors.

References

[1] H.Noborio, K.Onishi, M.Koeda, K.Mizushino, M.Yagi, M.Kaibori, M.Kwon. 2014. Motion Transcription Algorithm By Matching Corresponding Depth Image and Z-buffer. *In Proceedings of the 10th Anniversary Asian Conference on Computer Aided Surgery,* Kyusyu University, Japan. 60-61.

[2] H.Noborio, T.Kunii, K.Mizushino. 2014. GPU-Based Shortest Distance Algorithm For Liver Surgery Navigation. *In Proceedings of the 10th Anniversary Asian Conference on Computer Aided Surgery,* Kyusyu University, Japan. 42-43.

[3] K.Onishi, H.Noborio, M.Koeda, K.Watanabe, K.Mizushino, T.Kunii, M.Kaibori, K.Matsui, M.Kwon. 2015. Virtual Liver Surgical Simulator By Using Z-Buffer For Object Deformation. *In Proceedings of the 17th Int. Conf. on Human-Computer Interaction (HCII2015),* Los Angeles, CA, USA (to appear).

[4] M.Koeda, A.Tsukushi, H.Noborio, K.Onishi, K.Mizushino, T.Kunii, K.Watanabe, M.Kaibori, K.Matsui, M.Kwon. 2015. Depth Camera Calibration and Knife Tip Position Estimation for Liver Surgery Support System*. In Proceedings of the 17th Int. Conf. on Human-Computer Interaction (HCII2015),* Los Angeles, CA, USA. (to appear).

[5] P. J. Besl and N. D. McKay. 1992. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 14(2): 239-256.

[6] Z. Zhang. 1994. Iterative Point Matching for Registration of Free-Form Surfaces. *Int. Journal of Computer Vision,* 13(2):119–152.

[7] S. Granger, X. Pennec. 2002. Multi-Scale EM-ICP: A Fast and Robust Approach for Surface Registration. *In Proceedings of the 7th European Conference on Computer Vision.* 4:69-73.

[8] Y. Liu. 2006. Automatic Registration of Overlapping 3D Point Clouds Using Closest Points. *Image and Vision Computin.* 24(7): 762-78.

[9] J.Salvi, C.Matabosch, D.Fofi, J.Forest. 2007. A Review Of Recent Range Image Registration Methods With Accuracy Evaluation. *Journal of Image-and-Vision-Computing,* 25: 578-596.

[10] R. B. Rusu and S. Cousins. 2011. 3D is here: Point Cloud Library (PCL), *The IEEE Int. Conf. Robotics and Automation*, 1-4.

[11] Y.F.Wu, W.Wang, K.Q.Lu, Y.D.Wei, Z.C.Chen. 2015. A New Method For Registration Of 3D Point Sets With Low Overlapping Ratios. *Procedia CIRP 27 (The 13th CIRP Conference on Computer Aided Tolerancing*). 202-206.

[12] H.Noborio, K.Watanabea, M.Yagi, Y.Ida, K.Onishi, M.Koeda, H.Nankaku, M.Kaibori, K.Matsui, M.Kwon. 2015. Image-based Initial Position/Orientation Adjustment System between Real and Virtual Livers. *In Proceedings of the Medical Engineering Conference,* Johor Bahru, Malaysia, 22 - 23 August. (to appear).