Extraction of Light Stripe Centerline Based on Self-adaptive Thresholding and Contour Polygonal Representation

Qingguo Tian[1,2], Yujie Yang[1,2], Xiangyu Zhang[1,2], and Baozhen Ge[1,2]

[1] College of Precision Instrument and Opto-electronics Engineering,
Tianjin University, Tianjin 300072, China
[2] Key Laboratory of Opto-electronics Information Technology of the Ministry of Education,
Tianjin 300072, China
`Tianqingguo@tju.edu.cn`

Abstract. Extracting light stripe centerline is the key step in the line-structure light scanning visual measuring system. It directly determines the quality of three-dimensional point clouds obtained from images. Due to the reflectivity and/or color of object surface, illumination condition change and other factors, gray value and curvature of light stripe in image will vary greatly that makes it very difficulty to completely and precisely extract sub-pixel centerline. This paper presents a novel method for light stripe centerline extraction efficiently. It combines the integral image thresholding method, polygon representation of light stripe contour and adaptive center of mass method together. It firstly locates light stripe region and produces binary image no matter how change gray values of light stripe against background. Then the contour of light stripe is extracted and approximately represented by polygon. Based on the local orthogonal relationship between directions of light stripe cross-section and corresponding polygon segment, the direction of light stripe cross-section is calculated quickly. Along this direction, sub-pixel centerline coordinates are calculated using adaptive center of mass method. 3D scanning experiments with human model dressed colorful swimsuit on a self-designed line laser 3D scanning system are implemented. Some comparisons such as light stripe segmentation using 3 thresholding methods, the time used and the smoothness are given and the results show that the proposed method can acquire satisfying data. The mean time used for one image is not beyond 5 ms and the completeness and smoothness of point clouds acquired by presented methods are better than those of other two methods. This demonstrates the effectiveness and practicability of the proposed method.

Keywords: centerline extraction, light stripe, integral image thresholding, polygon representation, adaptive center of mass.

1 Introduction

Extraction of light stripe centerline is a key technique in the line-structure light scanning visual measuring system [1]. It directly determines the quality of three-dimensional (3D)

V.G. Duffy (Ed.): DHM/HCII 2013, Part II, LNCS 8026, pp. 292–301, 2013.

world coordinate points obtained from images. Light stripe gray image records distorted light stripe information which is modulated and scattered by the measured object surface and captured by CCD detector. On one hand due to the reflectivity and color of object surface and illumination conditions, the gray value of light stripe in image will be uneven. On the other hand due to change of object surface curvature and tilt angle of object surface to the light plane the curvature and width of light stripe in image may be vary largely. So it's very difficult to precisely and fast extract light stripe centerline.

In order to efficiently extracting light stripe centerline, three steps must be tackled carefully. First is to locate the light stripe region in virtue of the gray difference between light stripe (foreground) and background. When gray value has bimodal distribution in image series, one can use global single thresholding method (GSTM) to segment light stripe region [2, 3]. If reflectivity of the object surface under test is uneven and/or illumination condition changes greatly, the gray distribution is uneven too, so the GSTM will lose some useful light stripe segments whose gray values are not remarkably larger than background's, and eventually result in losing some 3D point clouds. WU Qing-yang, et al. [4] and Zhang Lei, et al. [5] apply OTSU method [6] to adaptive compute optimal threshold for segmentation. Second step is to calculate center coordinate. Most common methods include maximum value method, thresholding method, Gaussian approximation, center of mass, linear approximation, Blais and Rioux detector, and parabolic estimator. Although first two methods have faster speed, they can only provide pixel precision. Ref. [7] provides a systematic comparison of latter five methods and concludes that all of them display performance within the same sub-pixel precision range. Ref. [8] indicates that center of mass has characteristics of fast speed and high precision. Third step is to calculate the direction of light stripe cross-section. Because only along this direction, the center coordinate calculated by center of mass method has more accurate. Forest, et al. [9] scan each image column to calculate the center coordinate, which can have a right result only when light stripe is approximately horizontal. If the curvature of object surface changes greatly, the curvature of light stripe will change greatly too, column scanning method will result in losing some useful data. There are many references researching how to calculate the direction of light stripe. Steger, et al. [10] present Hessian matrix method to calculate the direction of light stripe, HU Kun, et al. [11] apply tremendous template Gaussian convolution to compute Hessian matrix so that speed up the calculation of direction of light stripe. WU Qing-yang, et al. [4] defines several alterable direction templates to calculate direction of light stripe. BAZEN, et al. [12] combine gray gradient and principal component analysis method to calculate direction of light stripe. JIA Qian-qian, et al. [13] applies light stripe contour gray gradient approximation to calculate its direction. The existing light stripe direction calculating algorithms have some drawbacks, such as calculating complicatedly, time-consuming, etc. They can hardly meet the real time requirement for light stripe centerline extraction.

This paper presents a fast and efficient light stripe centerline sub-pixel extraction method which is especially suitable for object surface with changeable reflectivity and/or color. The rest of this paper is organized as follows: The proposed algorithm is developed in Section 2, its experimental results and quantitative and qualitative comparisons with other methods are discussed in Section 3, and finally, conclusions are presented in Section 4.

2 The Proposed Method for Centerline Extraction

The proposed method for light stripe centerline extraction includes 3 main steps: localization of light stripe region; computation of light stripe cross-section direction and calculation of sub-pixel center coordinates.

2.1 Apply Integral Image Thresholding Method to Locate Light Stripe Region

As discussed above, if the reflectivity and/or color of tested object surface change greatly, gray values of light stripe will become uneven greatly too. Fig. 4(a) is a real light stripe image captured by CCD camera from a colorful swimsuit model. The gray values of light stripe within dash line rectangle are obviously larger than those within dash line ellipse. In order to solve this problem and locate whole light stripe region, this paper adopts the method in Ref. [14] to segment gray image and produce corresponding binary image.

Let *II* be an integral image having same size with the original gray image $I_{\mathrm{m\times n}}$ where m is the number of rows and n is the number of columns. Let *I*(x, y) be gray value of pixel (x, y) in $I_{\mathrm{m\times n}}$, so the function value *f*(x, y) in *II* can be calculated using formula (1)

$$f(x,y)=I(x,y)+f(x,y-1)+f(x-1,y)-f(x-1,y-1) \tag{1}$$

The sum function *s*(x, y) within local square window with center (x, y) and window size $2l+1$ in image *I* can be fast calculated by formula (2). Providing left-up coordinate of window is (x_1, y_1), right-down coordinate is (x_2, y_2) and $x_2-x_1=y_2-y_1=2l+1$, so

$$\begin{aligned} s(x,y)&=\sum_{i=x-l}^{x+l}\sum_{j=y-l}^{y+l} I(x_i,y_j) \\ &=f(x_2,y_2)+f(x_1,y_1)-f(x_2,y_1)-f(x_1,y_2) \end{aligned} \tag{2}$$

Finally, through formula (3) one can judge whether a pixel (x, y) belongs to light stripe region or belongs to background.

$$bw(x,y)=\begin{cases}0, count\times I(x,y)<T(x,y)=ratio\times s(x,y)\\ 1, count\times I(x,y)>=T(x,y)=ratio\times s(x,y)\end{cases} \tag{3}$$

Where count is sum of pixels in window, *ratio* is a predefined coefficient and $ratio<1$. By formula (1) to (3) one can extract light stripe completely and produce a corresponding binary image.

2.2 Calculation of Local Light Stripe Cross-Section Direction

Referring to Fig. 1, the direction $\mathbf{n}$ of light stripe cross-section is perpendicular to the direction of light stripe contour $\overline{pq}$ in a small region. So we can quickly calculate direction of local light stripe cross-section through calculating the direction of local light stripe contour. The detail calculation process is presented in our previous paper [15], here only give brief steps.

Firstly, the contour C of light stripe is tracked in binary image using method based on Ref. [16]. The contour C has 3 characteristics: a) contour is enclosed and corresponds to a light stripe segment, b) it can recognize light stripe with holes automatically, and c) points on C are anticlockwise sorted.

Secondly, using polygon P_i to represent contour C_i satisfying that distance from any pixel on contour C_i to polygon P_i is less than a preliminary distance threshold d_{th}.

Thirdly, for an edge $\overline{pq}$ in polygon P_i as illustrated in Fig. 1, providing the coordinate of p and q is (x_p, y_p) and (x_q, y_q) respectively. The direction vector $\mathbf{v}_{\overline{pq}}$ can be easy computed as $(x_q - x_p, y_q - y_p)$.

Fourthly, due to direction $\mathbf{n}_{cross}$ of cross-section of light stripe between point p and q are orthogonal to $\mathbf{v}_{\overline{pq}}$, it can be calculated as $-{}^{1}\!/\!{}_{\mathbf{v}_{\overline{pq}}}$.

2.3 Calculation Sub-pixel Center Coordinates Using Adaptive Center of Mass

As illustrated in Fig. 1, let s_b be a start point, s_e be corresponding end point, and $\mathbf{n}$ be direction vector, a scanning line l_{scan} can be defined. The sub-pixel center coordinate can be computed using formula (4) and (5).

$$x_{cen} = \frac{\sum\limits_{s_j \in l_{scan} \& s_j \in Inner(C)} x_{s_j} \times I(s_j)}{\sum\limits_{s_j \in l_{scan} \& s_j \in Inner(C)} I(s_j)} \tag{4}$$

$$x_{cen} = \frac{\sum\limits_{s_j \in l_{scan} \& s_j \in Inner(C)} y_{s_j} \times I(s_j)}{\sum\limits_{s_j \in l_{scan} \& s_j \in Inner(C)} I(s_j)} \tag{5}$$

where s_j is on scanning line l_{scan}, $s_j \in Inner(C)$ represents that s_j is inside contour C. From Fig. 1 one can see that the cross-section's width is changeable. Using formula (4) and (5) to compute center coordinate can not only acquire sub-pixel precision, but be adaptive to the width change of cross section.

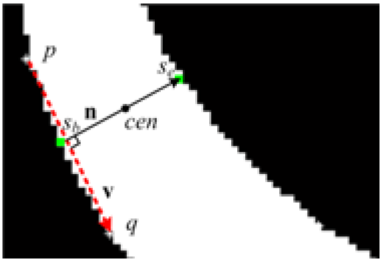


Fig. 1. The relationship between directions of local contour and cross-section

3 Experiments and Result Analyses

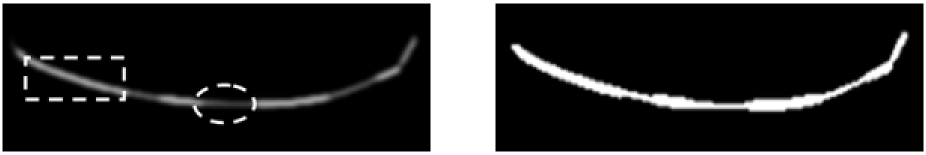
3D scanning experiments with human upper body model dressed colorful swimsuit on a self-designed line laser 3D scanning system [17] are implemented. The experimental setup is shown in Fig. 2. The light source is line-structure red laser source. Four pillars make a square, and a scanning head consisting of an upper scanning unit and a lower scanning unit moves up and down along each pillar driven by servo motor with micrometer accuracy, so the scanning system includes 8 scanning units. The measurement volume is a cylinder of diameter 1000mm by height 2000mm, and the width and depth resolution is 1mm and 2mm respectively. The photos from 4 views of tested human upper body model dressed colorful swimsuit are shown in Fig. 3.

Fig. 2. The experimental setup

Fig. 3. Photos of colorful swimsuit human model

3.1 The Light Stripe Localization Results

Firstly we compare light stripe localization results using different thresholding methods. Fig. 4(a) is one original light stripe image. Due to the color of swimsuit, gray values of light stripe are very uneven. Fig. 4(b) is the light stripe localization result using OTSU. For this image, the calculated threshold value is 45. The useful light stripe region has been localized, but the cross section width varies at different position because of uneven gray change. Fig. 4(c) and (d) are results using GSTM. Fig. 4(c) adopts threshold value 40; the shape of light stripe region is similar to that using OTSU. Fig. 4(d) adopts threshold value 70, light stripe has been broken. This inevitably results in losing some 3D point data. Fig. 4(e) and (f) are light stripe localization results using IITM with parameters window size = 7, *ratio* = 0.86 and window size = 5, *ratio* = 0.75 respectively. The shapes of two results are very similar and cross section width changes not very obvious. All these demonstrate that the IITM is not sensitive to parameters and can obtain good light stripe localization result.



(a) Initial light stripe gray image (b) Using OTSU

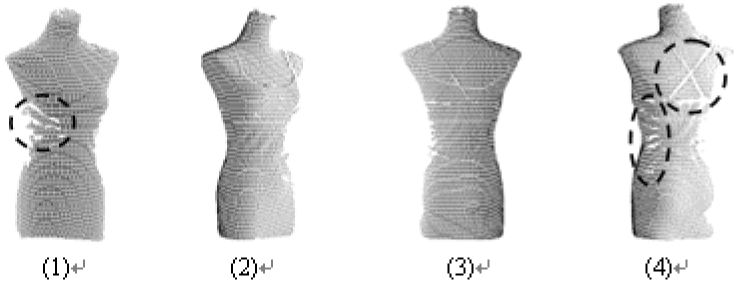
(c) Using GSTM, Threshold = 40 (d) Using GSTM, Threshold = 70

(e) Using IITM, winsize = 7, *ratio*=0.86 (f) Using IITM, winsize = 5, *ratio*=0.75

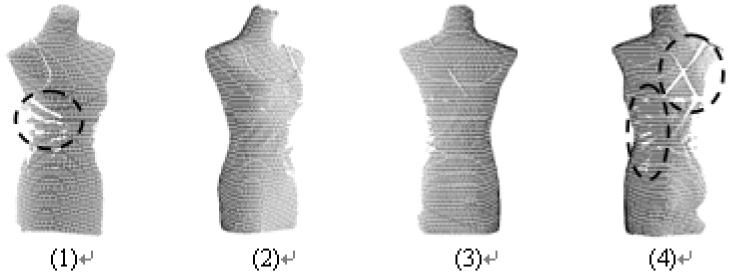
Fig. 4. Light stripe segmentation results using different methods

3.2 3D Point Cloud Model Integrality Analysis

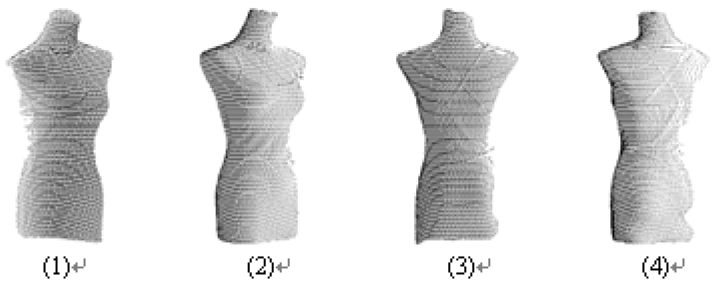
We indirectly evaluate light stripe centerline extraction results, such as completeness, time used and smoothness using 3D point clouds obtained from light stripe images. Because we can acquire whole 3D point clouds of colorful model shown in Fig. 3 utilizing upper 4 scanning units, subsequent analysis is based on point clouds obtained by upper 4 scanning units and labeled with (1), (2), (3), (4). With distance threshold value for polygon representation d_{th} equals 3.0, 3D point clouds after deleting outlier noises are displayed in Fig. 5. Fig. 5(a) shows results using GSTM and threshold value for all images is 40. Due to surface color change, ambient light change and CCD camera's performance difference, there is a lack of points in point clouds, especially in dash-line ellipse regions of point clouds labeled with (1) and (4). This indicates that GSTM with threshold = 40 can't extract all light stripe region. Fig. 5(b) shows point clouds using OTSU. The threshold calculated by OTSU for any image may be different. Similar to Fig. 5(a), some points have not been calculated. Fig. 5(c) shows point clouds using IITM with parameters window size = 5 and *ratio* = 0.85. Every pixel's threshold may be different when using IITM to localize light stripe. The fact that there has no lack of points in Fig. 5(c) demonstrates that using IITM can localize all light stripes.



(a) Using GSTM, threshold = 40



(b) Using OTSU



(c) Using IITM, winsize = 5, ratio = 0.85

Fig. 5. Point clouds obtained by scanning units using different thresholding method

3.3 Time Analysis

The algorithm was coded in C++ and compiled for 32 bits using the Visual C++ compiler under Windows XP SP3 and the computer used has 2.8G dual core CPU and 4G memory. Tab. 1 is the statistics of time and number of processed images. The numbers of processed images are not equal to each other for 3 methods because we assume that an image is valid only if the light stripe was extracted from it. The number of processed images using OTSU is 1696 that is more than that using other 2 methods. Contrast to point clouds in Fig. 5, this implies again that method using OTSU process more useless images, produce more noises, but lose many useful light stripe information. The mean time used with GSTM, OTSU and IITM is 3.495ms, 3.831 and 4.347ms respectively. The standard deviation of time using three thresholding methods is 1.493ms, 1.668ms and 1.585ms. Total time is 4987.434ms, 6497.944ms and 5885.799ms respectively. From these results, one can see that although the time used for one image with IITM is a little larger than those with other 2 methods, due to the number of processed images is minimal, the total time used is not maximum.

Table 1. The time statistics for different light stripe centerline extraction methods

Items / Methods	Frames of Processed images	Min time (ms)	Max time (ms)	Mean time (ms)	Std (ms)	Total time (ms)
GSTM threshold=40	1427	1.683	14.222	3.495	1.493	4987.434
OTSU	1696	2.785	19.076	3.831	1.668	6497.944
IITM winsize=5, ratio=0.85	1354	3.177	14.551	4.347	1.585	5885.799

3.4 Smooth Contrast of Light Stripe Centerlines

We reconstruct mesh models [18] from point clouds shown in left column of Fig. 5, and apply the smooth characteristics of mesh model to contrast the smoothness of light stripe centerline extracted. Fig. 6 shows the reconstructed mesh model results. The smoothness of mesh model in Fig. 6(b) is better than mesh model in Fig. 6(a), and the smoothness of mesh model in Fig. 6(c) is best. These results show that in the process of light stripe localization, method using IITM has minimal influence on smoothness of light stripe centerline, at the same time the smoothness of light stripe centerline extracted by method using GSTM is worst. These are consistent with results shown in Fig. 4.

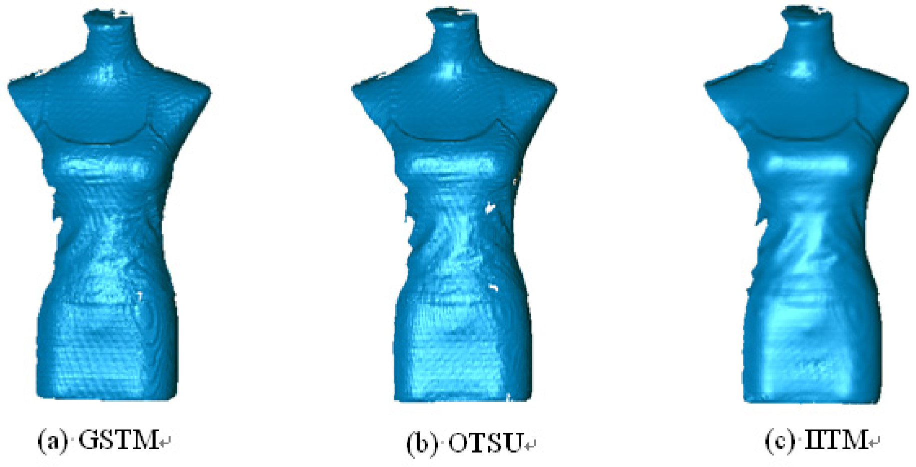


Fig. 6. The mesh model of point clouds obtained using different thresholding method

4 Conclusions

For 3D digitising of object surface with large reflectivity and/or color change using line-structure laser scanning vision measuring system, extracting light stripe sub-pixel centerline is key and hard problem, but must be resolved firstly. The method proposed in this paper combines the integral image thresholding image segmentation method, polygon representation of light stripe contour and self-adaptive center of mass together. From time statistics one can conclude that the proposed method can probably extract light stripe centerline in real time if we deeply optimize the program to improve its run efficiency. From the 3D point clouds and corresponding mesh models it is obvious that the proposed method can acquire whole and smooth 3D data that is better than those obtained by other 2 methods. Our further work is focus on research of 3D color modeling of object surface.

Acknowledgements. The paper is supported by National natural science foundation of China (No. 61027012, No. 61177002).

References

1. Gao, S., Zhao, M., Zhang, L., et al.: Dual-Beam Structured Light Vision System for 3D Coordinates Measurement. In: Proceedings of the 7th World Congress on Intelligent Control and Automation, pp. 3687–3691. Chongqing (2008)
2. Yang, S., Cho, M., Lee, H., Cho, T.: Weld line detection and process control for welding automation. Meas. Sci. Technol. 18(3), 819–826 (2007)
3. Liu, W., Jia, Z., Wang, F., et al.: An improved online dimensional measurement method of large hot cylindrical forging. Measurement 45, 2041–2051 (2012)

4. Wu, Q.-Y., Su, X.-Y., Li, J.-Z., et al.: A New Method for Extracting the Centre-line of Line Structure L ight-stripe. Journal of Sichuan University (Engineering Science Edition) 39(4), 151–155 (2007) (in Chinese)
5. Zhang, L., Xu, Y., Wu, C.: Features Extraction for Structured Light Stripe Image Based On OTSU Threshold. In: Proceedings 4th International Symposium on Knowledge Acquisition and Modeling, KAM, pp. 92–95 (2011)
6. Otsu, N.: A Threshold Selection Method from Gray level Histogram. IEEE Transactions on SMC (9), 62–67 (1979)
7. Fisher, R.B., Naidu, D.K.: A comparison of algorithms for subpixel peak detection. Image Technology: Advances in Image Processing, Multimedia and Machine Vision (1996)
8. Haug, K., Pritschow, G.: Robust laser-stripe sensor for automated weld-seam-tracking in the shipbuilding industry. In: IECON Proc. Ind. Electron. Conf., vol. 2, pp. 1236–1241 (1998)
9. Forest, J., Salvi, J., et al.: Laser stripe peak detector for 3D scanners. A FIR filter approach. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004), vol. 3, pp. 646–649 (2004)
10. Steger, C.: An unbiased detector of curvilinear structures. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(2), 113–125 (1998)
11. Hu, K., Zhou, F., Zhang, G.: Fast extrication method for sub-pixel center of structured light stripe. Chinese Journal of Scientific Instrument 27(10), 1326–1329 (2006) (in Chinese)
12. Bazen, A.M., Gerez, S.H.: Systematic methods for the computation of the directional fields and singular points of fingerprints. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7), 905–918 (2002)
13. Jia, Q.-Q., Wang, B.-X., Luo, X.-Z.: Extraction of central positions of light stripe in sub-pixel in 3D surface measurement based on light sectioning method. Optics and Precision Engineering 18(2), 390–396 (2010) (in Chinese)
14. Bradley, D., Roth, G.: Adaptive Thresholding Using the Integral Image. Journal of Graphics 12(2), 13–21 (2007)
15. Tian, Q.-G., Ge, B.-Z., Li, Y.-P., et al.: Real-time extraction of light stripe central line based on contour polygonal representation. Optics and Precision Engineering 18(2), 1357–1362 (2010) (in Chinese)
16. Haig, T.D., Attikiouzel, Y., et al.: Border following: new definition gives improved borders. IEEE Proceedings-I 139(2), 206–211 (1992)
17. Ge, B.-Z., Tian, Q.-G., Young, K.D., Sun, Y.-C.: Color 3D digital human modeling and its applications to animation and anthropometry. In: Duffy, V.G. (ed.) HCII 2007 and DHM 2007. LNCS, vol. 4561, pp. 82–91. Springer, Heidelberg (2007)
18. Tian, Q.-G., Ge, B.-Z., Yu, D.-Y., et al.: An Incremental Projection-based Surface Reconstruction Algorithm from Scattered Data. Journal of Image and Graphics 11(10), 1409–1413 (2006) (in Chinese)