A Study of 3D Feature Tracking and Localization Using A Stereo Vision System

Li-Wei Zheng
Department of Information and Computer Engineering
Chung Yuan Christian University
Chung-Li, Taiwan, R.O.C.
e-mail: tize@cycu.org.tw

Yuan-Hsiang Chang
Department of Information and Computer Engineering
Chung Yuan Christian University
Chung-Li, Taiwan, R.O.C.
e-mail: changyh@ice.cycu.edu.tw

Zhen-Zhong Li
Department of Information and Computer Engineering
Chung Yuan Christian University
Chung-Li, Taiwan, R.O.C.
e-mail: bakuryu19@hotmail.com

***Abstract*—With the advances of image processing and computer vision, techniques being developed are no longer limited in images acquired with single camera (namely plane vision). A stereo vision system with two cameras has become the research of interest in many areas because its ability to yield the depth information similar to human vision. In this study, the objective was to develop a system that can automatically track and localize a 3D feature in motion using left and right video sequences. The system design included feature definition, feature tracking, feature localization, and depth computation. In addition, we evaluated our system with several research parameters, including various depths, video tracking in plane vision or stereo vision, use of kernel functions, and integer-pixel vs. sub-pixel accuracy. Our results demonstrated that the system could track and localize the given feature in motion, leading to yield reasonable results of depth information. In addition, the video tracking in stereo vision with sub-pixel accuracy clearly outperformed the video tracking in plane vision with integer-pixel accuracy. In summary, our system yielded a potential solution in tracking and localizing 3D feature that could be incorporated in a large variety of video applications.**

Keywords-computer vision, depth information, feature tracking, stereo vision, sub-pixel accuracy.

I. INTRODUCTION

With progress of computer technology and cost reduction of video cameras, the studies of image processing and computer vision have become important in recent years. To date, development of image processing is no longer limited to single image with single camera. Instead, stereo vision systems that mimic the nature of human vision have attracted the attention of many researchers [1,2]. The applications may include autonomous vehicle, virtual reality, video surveillance system, and 3D television, etc.

Computer vision can be categorized into: (a) plane vision, and (b) stereo vision. The major difference is that the depth information (i.e., distance from camera) of objects, while not detectable with single camera, can be estimated with two cameras. Barnard and Fischler [3] proposed that "computation stereo" should cover the following areas: image acquisition, image matching, feature extraction, and depth information. As a result, stereo vision systems can be used with two cameras that simultaneously capture left and right images in order to acquire the necessary depth information for a variety of applications (e.g., 3D tracking of video objects).

In past research, many techniques for video tracking have been developed. These included: Optical flow [4], kernel-based tracking [5], contour tracking [6], blob tracking [7], mean shift tracking [8,9], Kalman filtering [10,11], and 3D tracking [12,13], etc. Many of these techniques, while effective for tracking motions using video sequences, could be limited in tracking 3D motions with plane vision acquired from single camera.

In this paper, our goal is aimed to study the 3D feature tracking and localization using a stereo vision system. The system includes two identical cameras with their optical axes parallel to each other and to the floor. After a user define a feature to track, the system is designed to automatically track and localize the given feature in motion and estimate the corresponding depth information. In addition, we evaluated the accuracy of the estimated depth information with several research parameters studied, e.g., various depths, video tracking in plane vision or stereo vision, use of kernel functions, and integer-pixel vs. sub-pixel accuracy.

II. METHOD

In this section, the method for the system design is presented. Our system is designed using two identical cameras that are set to use the same camera settings (i.e., focal length, ISO value, and shutter speed) for the stereo vision. The two cameras are installed such that their optical axes are parallel to each other and to the floor. Their vertical heights with respect to the floor are the same.

Fig.1 shows the block diagram of our system for the 3D feature tracking and localization using stereo vision. The main processes include: (a) Feature Definition; (b) Feature Tracking; (c) Feature Localization; and (d) Depth Computation. Given left and right video sequences, a user is first asked to indicate a 3D feature point of interest to track using the first frame in the sequences. Then, the system is designed to automatically track the 3D feature in the left video sequence, and to localize the corresponding 3D feature in motion using the right video sequence. After obtaining the two feature locations with feature tracking and localization, the system computes the depth information using triangulation and prompts the results.

978-1-4244-7640-4/10/$26.00 ©2010 IEEE

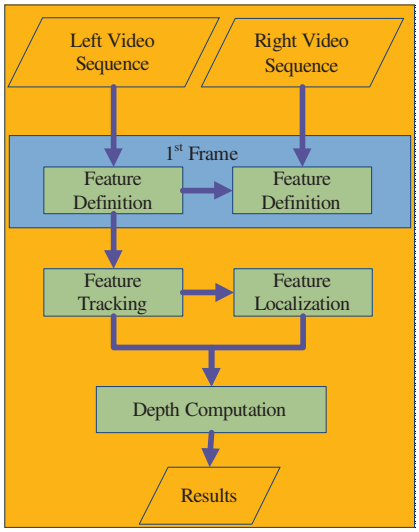


Figure 1. A block diagram of our system for the 3D feature tracking and localization using stereo vision.

A. *Feature Definition*

First, a user is asked to indicate (mark) a feature point of interest to track in the first frame for both the left and right video sequences. An example of the feature definition is shown in Fig. 2.

(a) (b)

Figure 2. An example of the "feature definition" in the first frame of the (a) left, and (b) right video sequences. The selected features are marked as "red".

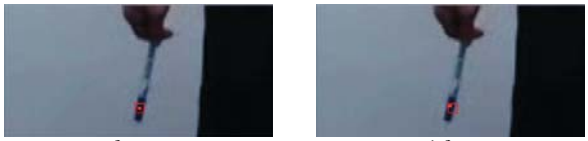
B. *Feature Tracking*

Based on the feature obtained in the first frame (or current frame t), our system is designed to automatically track the feature point in the next frame (frame $t + 1$) using a full-search method. That is, given the feature location in the current frame as the center for a block with the block size of $N \times N$ pixels, the system opens a search area with $R \times R$ pixels in the next frame, and finds the best-matched feature point. The best-matched feature point is determined by minimizing the mean squared error (M.S.E.) between two blocks in both frames. An example of the feature tracking is given in Fig. 3. As a result, a two-dimension motion vector can be computed.

To localize the feature for video tracking, the system applies a different weight to each pixel in the block such that larger weights are assigned to pixels closer to the center prior to the computation of the M.S.E. In this study, a kernel function for the weights is defined by:

$$k(r) = \begin{cases} 1 - r^2, & \text{if } |r| < 1 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $r = \frac{\sqrt{\left(i-\frac{N}{2}\right)^2+\left(j-\frac{N}{2}\right)^2}}{\sqrt{2}\times\frac{N}{2}}$, and (i, j) is the image coordinate in the block. The size of the block is with $N \times N$ pixels. An example of the kernel function is shown in Fig. 4.



t frame $t + 1$ frame

Figure 3. An example of the feature tracking. Our system uses the feature point in the current frame t as the center of a block, and opens a search area with $R \times R$ pixels in the next frame $t + 1$. The feature being tracked is marked as "red".

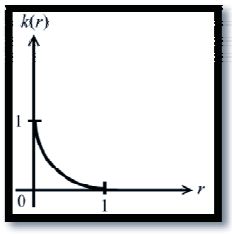
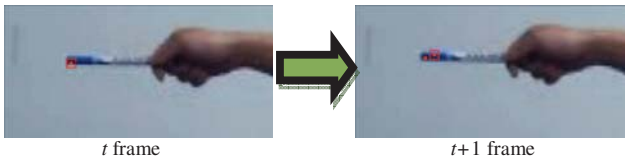


Figure 4. The kernel function for video tracking.

Our system is designed with a relatively small search area (i.e., $R = 9$) initially to maintain a reasonable speed for video tracking. However, if any fast motion occurs, the initial search area may be too small, resulting in mis-tracking. An example of mis-tracking is given in Fig. 5.



t frame $t+1$ frame

Figure 5. An example of mis-tracking is shown when the initial search area is too small for the feature with fast motion.

To resolve the problem, our system is further incorporated with an adaptive search method. That is, the search area is enlarged until the following criterion is met, i.e.,

(a) If the sum of absolute differences of RGB values between the two blocks is less than a pre-defined threshold T_{RGB}; and

(b) If the average hue values between the two blocks is less than a pre-defined threshold T_{Hue}.

C. *Feature Localization*

The objective of this process is to localize the corresponding feature point in the right video sequence. Because the motion of the given feature point in the left and right video sequences is similar, our system is designed to obtain the motion vectors as acquired in the left video sequence and to estimate the corresponding motion vectors for the feature in the right video sequence. An example for the feature localization is shown in Fig. 6.

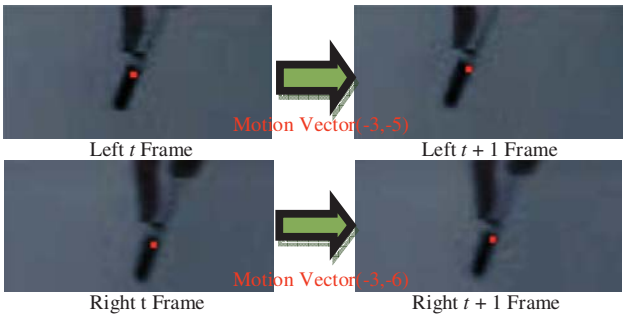


Figure 6. An example for the feature localization. The motion vector from the t-th frame to the t + 1-th frame in the left video sequence was (-3, -5). This motion vector can be used as the center for the search area in the right video sequence. The resulting motion vector from the t-th frame to the t + 1 th frame in the right video sequence was (-3, -6).

D. Depth Computation

A stereo vision with two identical cameras is established for the computation of depth information. Fig. 7 shows a stereo vision model based on the pin-hole camera model. As shown in the figure, f is the distance from the CCD center to the image plane (or focal length). p_l and p_r are the projected pixels in the left image plane and right image plane, respectively. d_l and d_r are the horizontal distances to the center of the left and right image plane, respectively. d_L and d_R are the horizontal distances from the target feature point P to the left CCD center and right CCD center, respectively. L is the distance between the two cameras. Finally, Z is the distance from the target feature point P to the center of the two cameras, which is the depth information to compute.

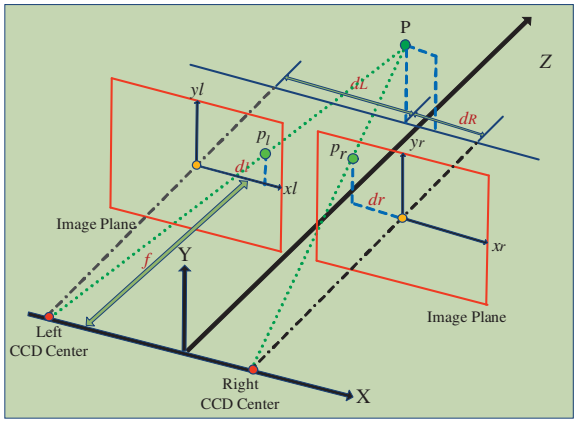


Figure 7. A stereo vision model.

The stereo vision model can be simplified in Fig. 8. Based on triangulation, the geometric relationship can be described by:

$$\frac{d_l}{d_L} = \frac{d_r}{d_R} = \frac{f}{Z} \tag{2}$$

or

$$d_L = \frac{d_l \times Z}{f}, d_R = \frac{d_r \times Z}{f} \tag{3}$$

Because $L = d_L + d_R$,

$$L = \frac{d_l \times Z}{f} + \frac{d_r \times Z}{f} = \frac{(d_l + d_r) \times Z}{f} \tag{4}$$

resulting in:

$$Z = \frac{L \times f}{(d_l + d_r)} \tag{5}$$

As seen in Fig. 8, the target feature point P is located between the two cameras. Because the target point P may also be located at either the left side or right side of the two cameras, we further derive the equations for the depth computation in the following.

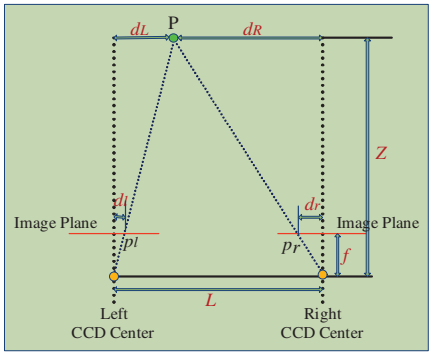


Figure 8. A simplified stereo vision model.

Fig. 9 and 10 show two different conditions when the target point P is located at either the left side or the right side of the two cameras, respectively. The equations for the depth computation become:

$$Z = \frac{f \times L}{(d_r - d_l)} \tag{6}$$

and

$$Z = \frac{f \times L}{(d_l - d_r)} \tag{7}$$

From equations (5), (6), and (7), we observe that the equations for the depth computation are somewhat different with respect to the location of the target point. Further, Fig. 8, 9, and 10 can be simplified in Fig. 11, 12, and 13, respectively, where nr is the image width in pixels. (l_x, l_y) is the image coordinate of the target point in the left image, and (r_x, r_y) is the image coordinate in the right image.

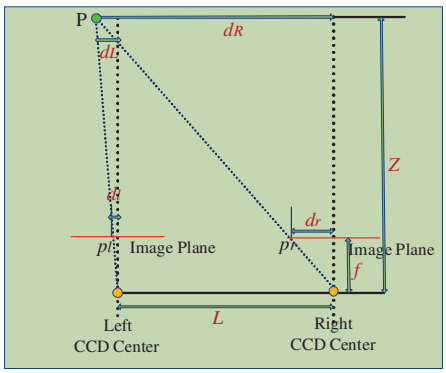


Figure 9. A simplified stereo vision model when the target point P is located at the left side of the two cameras.

In Fig. 11, we can derive that

$$d_l = l_x - \frac{nr}{2}, d_r = \frac{nr}{2} - r_x \tag{8}$$

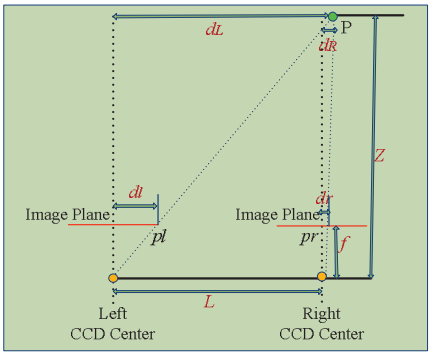


Figure 10. A simplified stereo vision model when the target point P is located at the right side of the two cameras.

Therefore,

$$d_l + d_r = \left(l_x - \frac{nr}{2}\right) + \left(\frac{nr}{2} - r_x\right) = l_x - r_x \quad (9)$$

Similarly in Fig. 12, we can derive that:

$$d_r - d_l = \left(\frac{nr}{2} - r_x\right) - \left(\frac{nr}{2} - l_x\right) = l_x - r_x \quad (10)$$

In Fig. 13,

$$d_l - d_r = \left(l_x - \frac{nr}{2}\right) - \left(r_x - \frac{nr}{2}\right) = l_x - r_x \quad (11)$$

Finally, we derive the depth information that can be computed by:

$$Z = \frac{f \times L}{(l_x - r_x)} \quad (12)$$

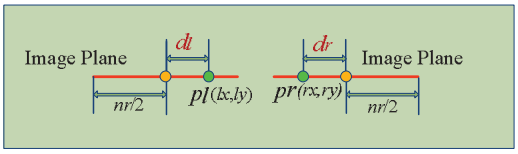


Figure 11. The target point P is located between the two cameras.

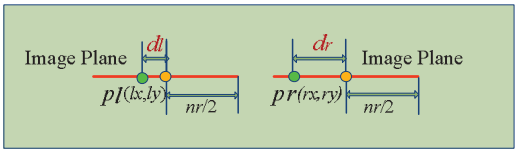


Figure 12. The target point P is located at the left side of the two cameras.

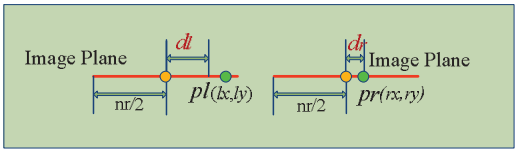


Figure 13. The target point P is located at the right side of the two cameras.

E. Evaluation

In this study, we studied the system performance in terms of several research parameters.

Various Depths: Three different depths (i.e., 150cm, 200cm, and 250cm) were chosen for the system evaluation.

Video Tracking in Plane Vision or Stereo Vision: To study the effect of plane vision or stereo vision in video tracking and depth computation, we evaluated the results of tracking and localization in two different modes. In plane vision, the feature in left or right video sequence is tracked independently. That is, the left feature point in the left frame $t + 1$ is tracked independently from the left frame t, while the right feature point in the right frame $t + 1$ is tracked independently from the right frame t. Therefore, the two motion vectors are evaluated independently. In stereo vision, the left feature point is tracked first that yields a motion vector, followed by the localization of the right feature point (an example is given in Fig. 6).

Use of Kernel Functions: Our system essentially incorporates the kernel functions when computing the mean squared errors for video tracking. Here, we also evaluated the effect of system performance if the kernel function is not used.

Integer-Pixel vs. Sub-Pixel Accuracy: During the depth computation, either integer-pixel or sub-pixel accuracy is considered. For the integer-pixel accuracy, the original image resolution (i.e., 640 × 480 pixels in this study) as captured by the camera is used. For the sub-pixel accuracy, the image is magnified by k (e.g., k = 2) in both dimensions before video tracking and localization. Therefore, the depth information can be evaluated in terms of 1 / k fraction of the pixel.

In this study, the error rate for the depth computation is measured by:

$$\left| \frac{d_{measured} - d_{actual}}{d_{actual}} \right| \times 100\ (\%)$$

where $d_{measured}$ is the measured depth, and d_{actual} is the actual depth based on the experimental settings.

III. Results

A. Research Environment

The two cameras used for the stereo vision were identical (Fuji Film F31fd). Both cameras were set with ISO 400, f-stop of 5.6, and white balance. The video files being captured were with 30 frames per second (fps) and 640 × 480 pixels. The computer for the system development was Intel Core ™ 2 Duo P8400 with 4GB memory. The software development was in C/C++. In addition, the experimental environment was controlled under sufficient illumination with simple background. The distance between the two cameras is L = 25cm.The focal length of the camera is f = 8mm.

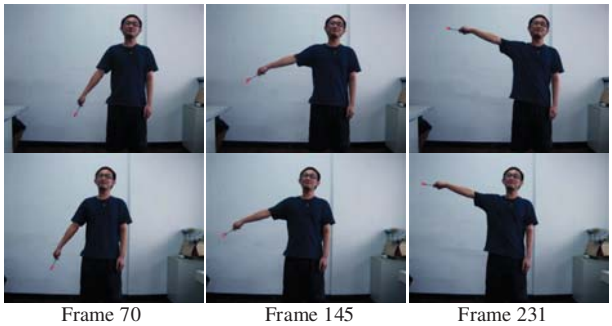
B. Video Database

In this study, a total of four video sequences were collected: (1) The human object was with a distance of 150 cm to the cameras; (2) The human object was with a distance of 200 cm to the cameras and at the left side of the two cameras; (3) The human object was with a distance of 200 cm to the cameras and at the right side of the two cameras; and (4) The human object was with a distance of 250 cm to the cameras. The human object carried a marker with a specific color (blue) such that the feature being

tracked can be well distinguished from its surrounding background.

C. Video Tracking in the Stereo Vision

Fig. 14 shows partial results of the video tracking using the second database as described above. Results (i.e., frame 70, 145, and 231) in the upper row and lower row correspond to the left and right video sequences, respectively, where the features being tracked are marked (red). The system parameters were selected with the video tracking in stereo vision with the use of kernel functions and sub-pixel accuracy. The results demonstrated that our system could successfully track and localize both in left and right video sequences.



Frame 70 Frame 145 Frame 231

Figure 14. Partial results of the video tracking. The features being tracked are marked (red).

D. Depth Computation

Fig. 15 shows the results of error rates with respect to various depths. The computed depths (average) and error rates at 200cm depth are summarized in Table II. The error rates at the depth of 200cm were found to be relatively small than other depths.

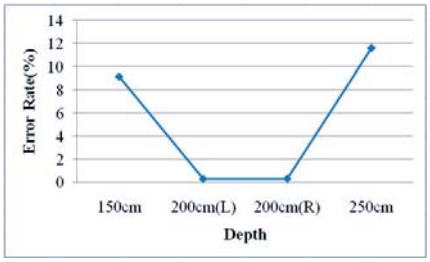


Figure 15. Results of error rates with respect to various depths (150, 200, and 250cm).

TABLE I. RESULTS OF COMPUTED DEPTHS AND ERROR RATES WITH RESPECT TO VARIOUS DEPTHS (150, 200, AND 250CM).

Depth	Computed Depth	Error Rate
150cm	163.732cm	9.15%
200cm(L)	199.398cm	0.3%
200cm(R)	200.628cm	0.31%
250cm	278.995cm	11.6%

Fig. 16 shows the results of error rates with respect to various depths for video tracking in plane vision or stereo vision. The computed depths (average) and error rates are summarized in Table III. Our results demonstrated that the computed depths using the stereo vision are more accurate than using the plane vision.

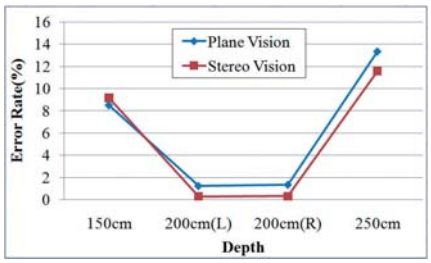


Figure 16. Results of error rates with respect to various depths in video tracking with plane and stereo vision.

TABLE II. RESULTS OF COMPUTED DEPTHS AND ERROR RATES WITH RESPECT TO VARIOUS DEPTHS IN VIDEO TRACKING WITH PLANE AND STEREO VISION.

Depth		Plane Vision Tracking	Stereo Vision Tracking
150cm	Computed Depth	162.78cm	163.732cm
	Error Rate	8.52%	9.15%
200cm(L)	Computed Depth	197.491cm	199.398cm
	Error Rate	1.25%	0.3%
200cm(R)	Computed Depth	202.726cm	200.628cm
	Error Rate	1.36%	0.31%
250cm	Computed Depth	283.375cm	278.995cm
	Error Rate	13.35%	11.6%

Fig. 17 shows the result of error rates with respect to various depths, where the kernel function was either used or not used. Table IV shows the computed depths (average) and error rates. Despite that the kernel function was intended to centralize the feature point during feature localization, our results demonstrated that the effect on the accuracy of depth computation was minimal.

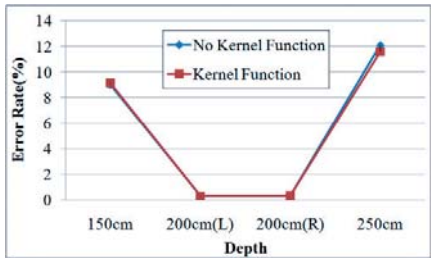


Figure 17. Results of error rates with respect to various depths, where the kernel function was either used or not used.

TABLE III. RESULTS OF COMPUTED DEPTHS AND ERROR RATES WITH RESPECT TO VARIOUS DEPTHS, WHERE THE KERNEL FUNCTION WAS EITHER USED OR NOT USED.

Depth		No Kernel Function	Kernel Function
150cm	Computed Depth	163.531cm	163.732cm
	Error Rate	9.02%	9.15%
200cm(L)	Computed Depth	199.323cm	199.398cm
	Error Rate	0.34%	0.3%
200cm(R)	Computed Depth	200.637cm	200.628cm
	Error Rate	0.32%	0.31%
250cm	Computed Depth	280.276cm	278.995cm
	Error Rate	12.11%	11.6%

Fig. 18 shows the results of error rates with respect to various depths, where either integer-pixel accuracy or sub-pixel accuracy was used. Table V shows the corresponding computed depths (average) and error rates. Our results clearly demonstrated that the computed depths using the sub-pixel accuracy was relatively accurate than using the integer-pixel accuracy.

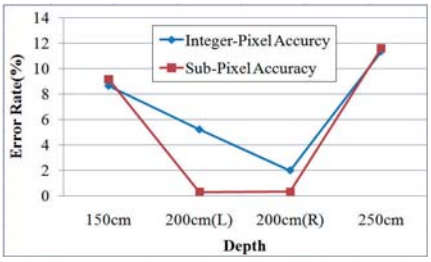


Figure 18. Results of error rates with respect to various depths, where either integer-pixel accuracy or sub-pixel accuracy was used.

TABLE IV. RESULTS OF COMPUTED DEPTHS AND ERROR RATES WITH RESPECT TO VARIOUS DEPTHS, WHERE EITHER INTEGER-PIXEL ACCURACY OR SUB-PIXEL ACCURACY WAS USED.

Depth		Integer-Pixel Accuracy	Sub-Pixel Accuracy
150cm	Computed Depth	162.952cm	163.732cm
	Error Rate	8.63%	9.15%
200cm(L)	Computed Depth	210.415cm	199.398cm
	Error Rate	5.2%	0.3%
200cm(R)	Computed Depth	203.985cm	200.628cm
	Error Rate	2%	0.31%
250cm	Computed Depth	278.364cm	278.995cm
	Error Rate	11.34%	11.6%

IV. CONCLUSION

In this paper, we have developed a system for the 3D feature tracking and localization using stereo vision. The system includes the methods for feature definition, feature tracking, feature localization, and depth computation. With sufficient illumination and simple background in a controlled environment, our experimental results demonstrated that the system could reasonably track and localize the feature in motion.

To further evaluate the system performance in depth computation, we have also studied the effect on depth computation. Several research parameters were considered such as various depths, video tracking in plane vision or stereo vision, use of kernel functions, and integer-pixel accuracy vs. sub-pixel accuracy. Our studies showed relatively accurate results at the 200cm depth than other depths (150cm or 250cm depth). In addition, the video tracking in stereo vision with sub-pixel accuracy clearly outperformed the video tracking in plane vision with integer-pixel accuracy. Furthermore, while the use of kernel functions doesn't dramatically affect the system accuracy, the sub-pixel accuracy was shown to be relatively accurate than the integer-pixel accuracy.

Our system was designed with the assumption that the environment was well controlled with sufficient illumination and simple background. In addition, the system assumed that the 3D world coordinate system was well-aligned with the center of the stereo vision system and the optical axes of the two cameras were parallel to each other and to the floor. Furthermore, the intrinsic parameters (e.g., geometric distortions due to lens) of the cameras were not considered. Overall, these conditions may affect the system accuracy in depth computation. To improve the system accuracy, these conditions should be taken into consideration, but this is beyond the scope of this study.

In summary, our results clearly showed the feasibility of yielding a relatively accurate measure in depths given a stereo vision system. Furthermore, while preliminary, our system showed promising results for video tracking and localization. Ultimately, our system for the depth computation using stereo vision could be incorporated in a large variety of applications (e.g., virtual reality, autonomous automobile, 3D TV, etc.). Future investigations may warrant in this regard.

ACKNOWLEDGMENT

This project was supported in part by the National Science Council, Taiwan, under contract NSC 98-2221-E-033-057.

REFERENCES

[1] S. Bahadori, L. Iocchi, G. R. Leone, D. Nardi and L. Scozzafava, "Real-time people localization and tracking through fixed stereo vision," Applied Intelligence, Vol. 26, No. 2, pp. 83-97, 2007.

[2] S. Se and P. Jasiobedzki, "Stereo-Vision Based 3D Modeling and Localization for Unmanned Vehicles," International Journal of Intelligent Control and Systems, Vol. 13, No. 1, pp.46-57, March 2008.

[3] S. T. Barnard and M. A. Fischler, "Computation Stereo," Computing Surveys, Vol 14, pp. 553-572, 1982.

[4] B. K. P. Horn and B. G. Schunck, "Determine Optical Flow," Artificial Intelligence,Vol. 17, pp. 285-204, 1981.

[5] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 5, pp. 564-577, May 2003.

[6] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," Lecture Notes in Computer Science, 1996 – Springer.

[7] R. T. Collins, "Mean-shift blob tracking through scale space," Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 2003.

[8] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, May 2002.

[9] D. Comaniciu and V. Ramesh, "Real-time tracking of non-rigid objects using mean shift," Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 142-149, June 2000.

[10] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," University of North Carolina at Chapel Hill, Chapel Hill, NC, 1995.

[11] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," Proc. SPIE, vol. 3068, pp. 182-193, 1997.

[12] B. Stenger, P. Mendonca, and R. Cipolla, "Model based 3D tracking of an articulated hand," CVPR, vol. 2, pp. 310-315, 2001.

[13] L. Vacchetti, V. Lepetit, and P. Fua, "Stable real-time 3D tracking using online and offline information," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 26, no. 10, pp. 1385-1391, 2004.