International Journal of Technology 10(5): 942-952
ISSN 2086-9614

A NOVEL APPROACH IN LOW-COST MOTION CAPTURE SYSTEM USING COLOR DESCRIPTOR AND STEREO WEBCAM

Muhammad Imanullah[1*], Eko Mulyanto Yuniarno[2], Adri Gabriel Sooai[1,3]

[1]*Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Sukolilo, Surabaya 60111, Indonesia*
[2]*Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Sukolilo, Surabaya 60111, Indonesia*
[3]*Department of Computer Science, Universitas Katolik Widya Mandira, Kupang City 85225, Indonesia*

(Received: January 2019 / Revised: May 2019 / Accepted: October 2019)

ABSTRACT

Animated motion has become crucial in some electronic entertainment business products, such as games, animated movies, and simulations. Making such animated motion is associated with high cost and hard setup requirements. This research proposes a low-cost system for capturing motion using stereo webcam (two webcams placed side by side) and some daily house-grade tools. The system test has been specifically designed for shadow puppet theaters. The setup consists of two identical webcams placed side by side to acquire the depth of a marker by stereo camera triangulation. Image processing is needed to improve object detection and feature matching. Once images are captured from two webcams, they will be inverted and color-filtered to detect the markers and set those markers as features to be matched. Each feature is packed with a unique descriptor based on its color composition. Features in both images are then compared to get their related matches. When matched, their depth and position can be calculated and recorded as a 3D representation that is ready to be processed as motion data. The proposed system is reasonably efficient since we can get an average accuracy of 83.5% using webcams that cost around $7.69.

Keywords: Feature matching; Marker; Motion capture; Stereo camera

1. INTRODUCTION

Electronic entertainment business products, such as games, animated movies, and simulations, have recorded innovations every year. Some basic asset requirements like 3D models, 3D environments, and animated motion for such products have also become crucial in their development process. To acquire such basic asset requirements, especially animated motion, we may need motion capture devices that have a range variety of cost and accuracy, but most of them are expensive. Many attempts, such as those by Huang et al. (2018) and Zecca et al. (2013), have focused on the main issue of capturing motion without any cost-related consideration. They use small IMU sensors attached to human body joints to acquire their orientation and position displacement, which facilitates motion capturing. Other attempts to solve the problem of cost have been made by Budiman et al. (2005), Chao et al. (2009), and Guarisa et al. (2016). Although their attempt to solve the problem was hampered by cost requirement, unlike what Huang et al. and Zecca et al. did with their latest technological approach, it provided alternatives for project scale, method, and affordance

*Corresponding author's email: ptrusted@gmail.com, Tel. +62-857-55665651
Permalink/DOI: https://doi.org/10.14716/ijtech.v10i5.2789

consideration.

Budiman et al. (2005) used a mean-shift algorithm to track detected objects and used black curtain as background with a white circular marker placed in the lower body part to make it easier to detect. He also used a camera calibration step to get each extrinsic camera matrix needed in finding the global coordinate of each marker from two webcams. Chao et al. (2009) used a dynamic background subtraction technique to ease the segmentation of human silhouette needed in 3D motion data reconstruction. Unlike Budiman et al., Chao et al. used four cameras and a color-marker-based spatial calibrating technique for fast and easier camera calibration to get the fundamental matrices and calculate the relative coordinate system. When it comes to the amount of camera used, Guarisa et al. (2016) used only one webcam that holds its aim for a low-cost motion capture feature. They developed their motion capture specifically to recognize the facial pose with markers of contrast color other than black and white. They successfully developed the low-cost and open-source facial motion capture even though the lack of depth value is unavoidable since only one camera is used.

Following the examples of the researches above, we decided to use two low-cost webcams (cost below $5 each) and hire a stereo camera triangulation system to estimate the depth instead of using intrinsic and extrinsic camera matrices calibration since the camera will be set in an unfixed position. The markers are made with various color differences to improve the feature matching process in a white background. We chose not to use black background as Budiman et al. (2005) did because most rooms in typical houses are painted white. We also proposed an image processing method to improve the detection of markers and color descriptors to match detected markers in both cameras.

2. SYSTEM SETUP AND METHODS

As a standalone system, it needs to be presented in a simple manner so users can use it by plug and play. The order of use is presented as sequential steps for users. The system will start from input device selection and produce the output of 3D points that can be visualized in the 3D world or in real-time. As can be seen in Figure 1, the system will run as long as input devices are connected and sending data. The iterative process starts with obtaining a signal for visualization. As an iteration gate, the obtain-signal step will determine whether the following steps (from feature extraction until visualization) should be executed. Once the visualization step has been successfully executed at the end of an iteration round, it will lead to another round of iteration process through the obtain-signal step again.

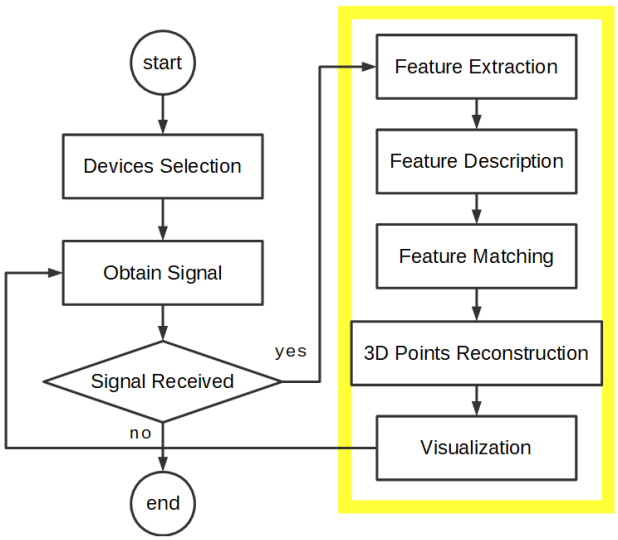


Figure 1 Architecture of low-cost motion capture system

2.1. Feature Extraction

The uses of triangulation, as stated by Heale and Forbes (2013), are various and common. But the term triangulation in this research is specifically about finding the unknown value of point from two or more known points that have some form of relationship. The two known points here refer to stereo camera. The stereo camera was made to mimic the human eyes, which see objects in two different positions. Due to this characteristic, we can perceive the depth of an object by knowing that, the more distinct its position as seen from the left and right eyes, the closer that object is. This characteristic can be used to determine the depth of a marker in our low-cost motion capture setup.

Two webcams that have known characteristics are positioned side by side at a determined distance (b) from each other, as can be seen in Figure 2. A manual focused webcam is recommended to easily determine the value of the focal length (f). When the object is captured by two cameras, its position in each camera image should be different (x and x'). The difference in position (x' - x) is called disparity (d).

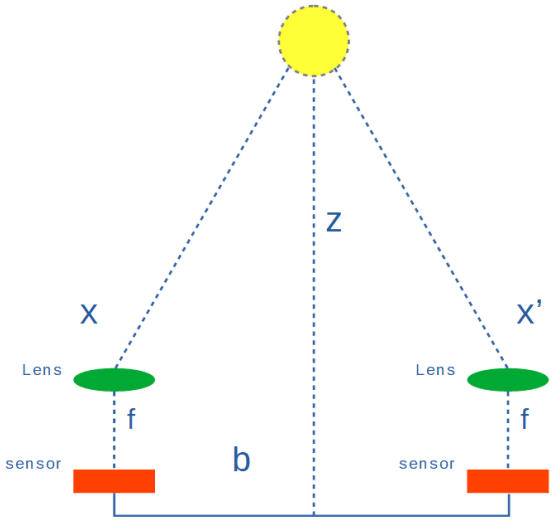


Figure 2 Stereo camera triangulation

The relation of each parameter can be seen in Equation 1 and can be rearranged into Equation 2 to obtain the depth.

$$\frac{b}{Z} = \frac{d}{f} \tag{1}$$

$$Z = \frac{f \times b}{(x' - x)} \tag{2}$$

It is shown in Shen et al. (2012), Kamencay et al. (2012), and Muhlmann et al. (2002) that the stereo camera triangulation is possible and doable. Through the calculation of disparity between two images, we can get the accurate depth of an object.

Images taken from two cameras should be processed for further use. We want to obtain the marker images and separate them from the background. According to Cunha (2009), the task category of image processing we should do is called segmentation. There are many methods proposed in the theme of background segmentation. Li and Lihong (2014) used the deviation information method based on the Gaussian mixture model to join the chroma and brightness of background segmentation. Their method works well for static and special environments, such as reflective ice and moving object reflection. Bindu et al. (2014) used histogram equalization to enhance the image, color thresholding for distinct unrelated colors, and Canny edge detector to

segment the object. Das and Saharia (2014) evaluated three background subtraction techniques of distinctly moving an object from its background and found that a statistical approach for real-time robust background subtraction and shadow detection is the most appropriate method due to its average computation cost and average result. As clearly stated, there are various ways to segment the image and distinguish the object from its background using image processing. So, in this research, our aim is clear: separating the colored markers from the white background and counting the identified markers using some combination of image processing methods.

As we need to develop a standalone application for a more efficient motion capture project, we need an image processing framework such as Aforge.NET. This open-source C# framework is designed specifically for developers and researchers who work in the field of computer vision and artificial intelligence (Aforge.NET, 2012).

Raw image (I) obtained from a webcam should be processed (I') to increase distinctness between markers and white backgrounds. This can be done by applying inversion (Equation 3) and color filtering (Equations 4, 5, and 6) to each pixel in raw images.

$$I'\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - I\begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{3}$$

$$I'(r) = \begin{cases} I(r), & if\ minR \le I(r) \le maxR \\ 0, & if\ I(r) \le minR \wedge I(r) \ge maxR \end{cases} \tag{4}$$

$$I'(g) = \begin{cases} I(g), & if\ minG \le I(g) \le maxG \\ 0, & if\ I(g) \le minG \wedge I(g) \ge maxG \end{cases} \tag{5}$$

$$I'(b) = \begin{cases} I(b), & if\ minB \le I(b) \le maxB \\ 0, & if\ I(b) \le minB \wedge I(b) \ge maxB \end{cases} \tag{6}$$

Ideally, inversion turns white pixel into black while colored pixel remains colored. But in reality, there will be shades of gray among white backgrounds due to noise and shadow. We can turn those shades of gray into black by using color filtering within certain minimum and maximum values.

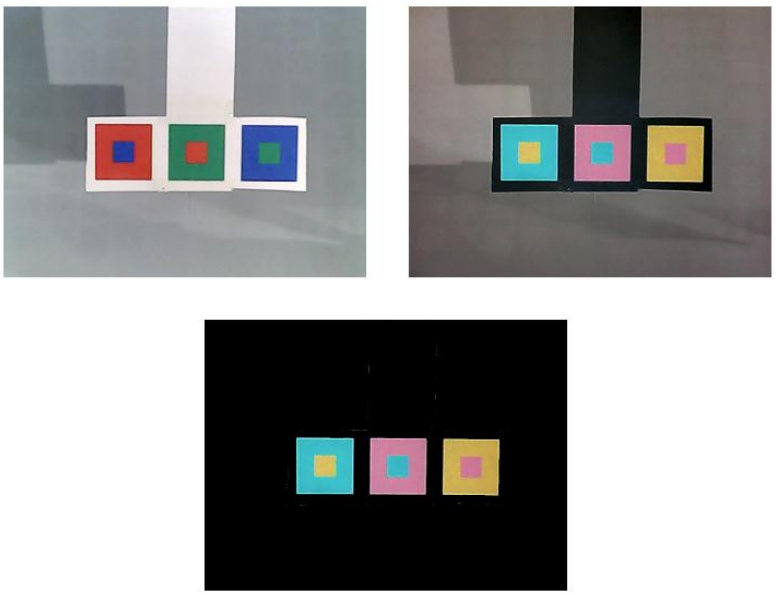


Figure 3 Upper left: raw image; upper right: inverted image; bottom: color-filtered image

Inversion and color filtering will give us an optimized image for object detection, as shown in Figure 3. As the background turns black and markers are clearly seen, the object detection that works under Aforge.NET framework will work more accurately. From this point, objects detected by the system will be treated as extracted features.

2.2. Feature Description

It differs from common disparity calculation in the stereo camera system as Ko and Ho (2016) and Lee et al. (2016) did, which use Census Transform (CT) to find matches between each pixel in both images, which takes more calculation time for our real-time stereo system. We need to find matches only for detected markers instead of all pixels in both images. Detected markers we obtain from the image processing phase act as features. To calculate disparity (d), we need to match each feature from the first image to its corresponding feature in the second image. There are many ways to do feature matching and one method to improve it is using a feature descriptor. Many feature descriptors are well-known, including SIFT and SURF. According to Kangas (2011), a SURF descriptor produces better results than any other contender. A SURF descriptor is perfect for dealing with a feature that varies in scale and transforms. Since the markers we use are made with simple shapes and color variations, we choose to use our own descriptor that can describe their characteristics with a low-cost calculation based on color composition.

Extracted features we got from the previous step are stored in the array of descriptors. This descriptor is a form of structured data that describes the unique characteristics of a feature's color. It contains parameters like color percentages, color class, world position, etc. Parameters contained within the structure are aimed to help us match the same feature in different images (left and right). Figure 4 shows an example of the features' color percentages.

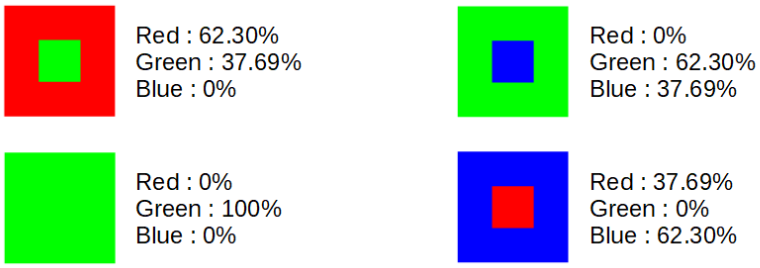


Figure 4 Features' color percentages

Those percentage values (p(x)) are needed in color classification index assignment. The percentage values of red, green, and blue are calculated with Equation 5, where x is a specific color (red, green, or blue), c is the total of colored pixel detected around the feature's boundary, and 100 is a constant value to make up the percentage scale. As can be seen in Figure 5, the feature descriptor is assigned using the raw image, so the existence of white pixels around the feature cannot be anticipated, especially when the marker is rotated. To distinguish between colored and white pixels, we can use Equation 6 where max is the maximum value between red, green, and blue in that specific pixel. The constant value of 10 is chosen to avoid noisy data after all values of red, green, and blue are subtracted by max.

$$p(x) = \frac{\sum x}{\sum c} \times 100 \tag{5}$$

$$c=\begin{cases}0, & if \quad |r-max|<10 \wedge |g-max|<10 \wedge |b-max|<10 \\ 1 & \end{cases} \tag{6}$$

As we obtained the needed parameters, we can assign the color classification index based on the feature's color composition. For example, if the percentage of red is 90%, with green 10% and blue 0%, then it is classified as index 1. Also, if the percentage of red is 60%, with green 0% and blue 40%, then it is classified as index 4. The index classification based on color percentages is entirely custom-able, so we can actually make markers with various colors in it.

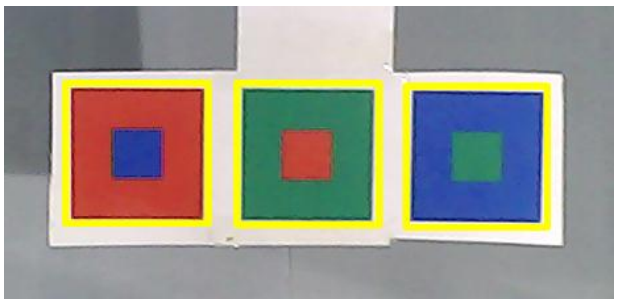


Figure 5 Color descriptor assignment (with white pixel inside the feature's boundary)

2.3. Feature Matching

The index classification assignment in the feature description process is making feature matching a lot easier. We can match two features in the left and right images by comparing the color class indexes. If the color indexes are similar, then it is matched. So, we can continue to the next processes.

2.4. 3D Points Reconstruction

Matched features give us the values needed to find disparity. According to Equation 2, the disparity can be calculated by subtracting the x position value of feature right (F') and the x position value of feature left (F). But due to the elevation difference between two cameras that cannot be anticipated, we need to use Euclidean distance formula (Equation 7). F_x and F_x' represent the x position of the left and right features in pixel unit, while F_y and F_y' represent the y position of the left and right features in pixel unit.

$$d(F',F)=\sqrt{(F_x'-F_x)^2+(F_y'-F_y)^2} \tag{7}$$

As we have obtained depth (Z) using Equation 2, we can calculate the world position of the feature in the horizontal axis (X) and vertical axis (Y). The value of X and Y will represent the position of the feature with respect to Z from the point in between two cameras as can be seen in Equations 8 and 9. We need the width (w) and height (h) of the image in pixel and divide the calculation by 1000 in the end to set the result to the centimeter measurement unit. To obtain rotation (R), we need the positions of, at least, two features. By using Equation 10, we get the rotation value of the line between two features with respect to the horizontal line in radians.

$$X=\frac{(F_x-(\frac{w}{2}))\times Z-(\frac{b}{2})}{f}\times\frac{1}{1000} \tag{8}$$

$$Y=\frac{(F_y-(\frac{h}{2}))\times Z-(\frac{b}{2})}{f}\times\frac{1}{1000} \tag{9}$$

$$R = \arctan\left(\frac{(Y_2 - Y_1)}{(X_2 - X_1)}\right) \tag{10}$$

2.5. Visualization

A legend from over 2000 years ago in China tells about an emperor who loses his beloved and a shaman brings her back to him in the form of shadow puppetry (Miettinen, 2018). This shadow puppetry is a kind of storytelling that became very popular due to its simplicity and powerful effect. It needs a light source, an object that casts a shadow, and a translucent material in between the puppeteer and the audience. Many countries in Asia have been adapting the form, such as wayang (Indonesia), nang talung (Thailand), and tolpavakoothu (India).

We chose to test our motion capture system by borrowing the concept of shadow puppetry. Detected features act as puppets in digital representation. Like in the shadow puppet theater, we can move and rotate the puppets. We can also change the form, background, or foreground of puppets anyway we want. The idea of putting shadow puppetry into digital representation is worth considering.

3. RESULTS AND DISCUSSION

The webcam used in this research cost around $3.51. It has a manual focus and 6 lights feature. The webcams setup can be seen in Figure 6. It is known that even two similar types of webcams may have different characteristics (focal length and sensor wide), which can lead to a scale issue, even when those webcams are put side by side. To overcome that, we propose a simple method that needs some guidelines printed in the output screens of both webcams, as shown in Figure 7. Those guidelines will help us adjust the position and tilt of the webcams until we get the same scale factor of the object in the output screens of both webcams.

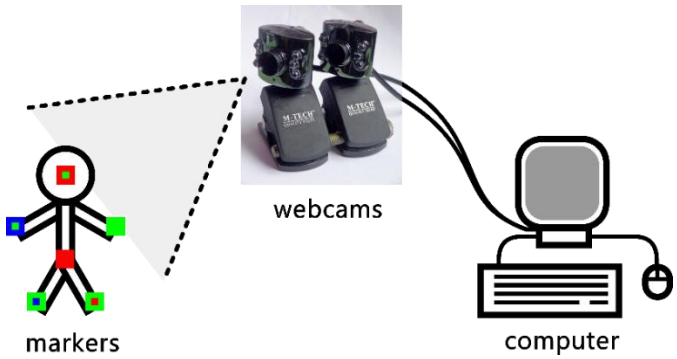


Figure 6 Webcam setup

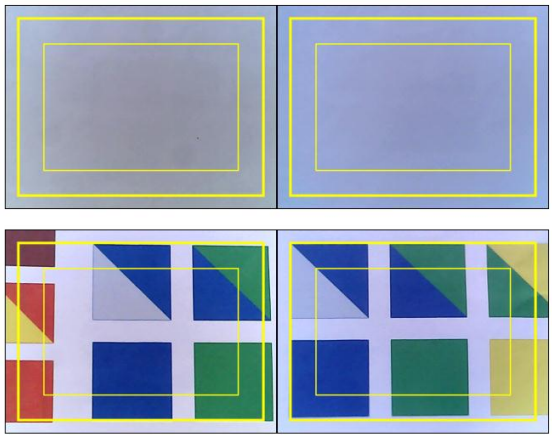


Figure 7 Top picture shows the yellow guidelines with no observed object; bottom picture shows observed objects assumed to have a similar scale factor

There have been many attempts to find the best combination of parameter values in the feature extraction process. It is known that the best value for some parameters in color filtering are minR = minG = minB = 150 and maxR = maxG = maxB = 255. With these values, we can clearly distinguish markers from the background, as can be seen in Figure 8. All nine markers have successfully been detected and extracted as features.

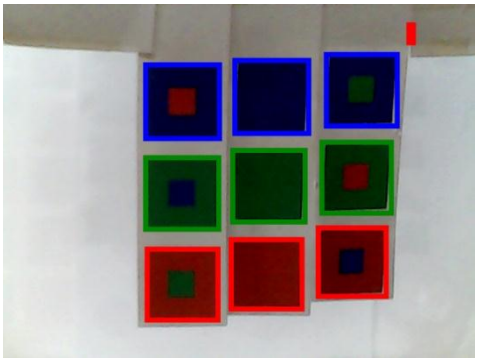


Figure 8 Feature extraction process

We use only 9 markers made with the solid colors of red, green, and blue for easier matching. Figure 9 shows that feature description has been successfully assigned 9 different color class indexes. Each feature comes with a numbering index and a color class index separated by a "|" character. Color percentages are shown below it, starting from red, followed by green and blue.

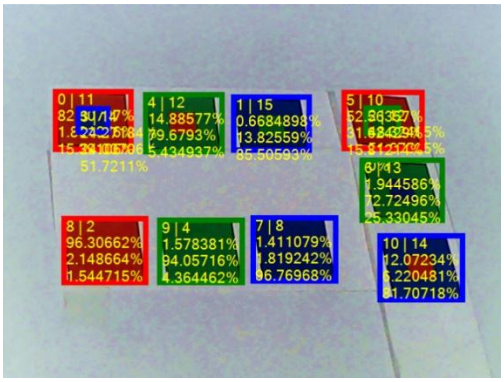


Figure 9 Feature description process

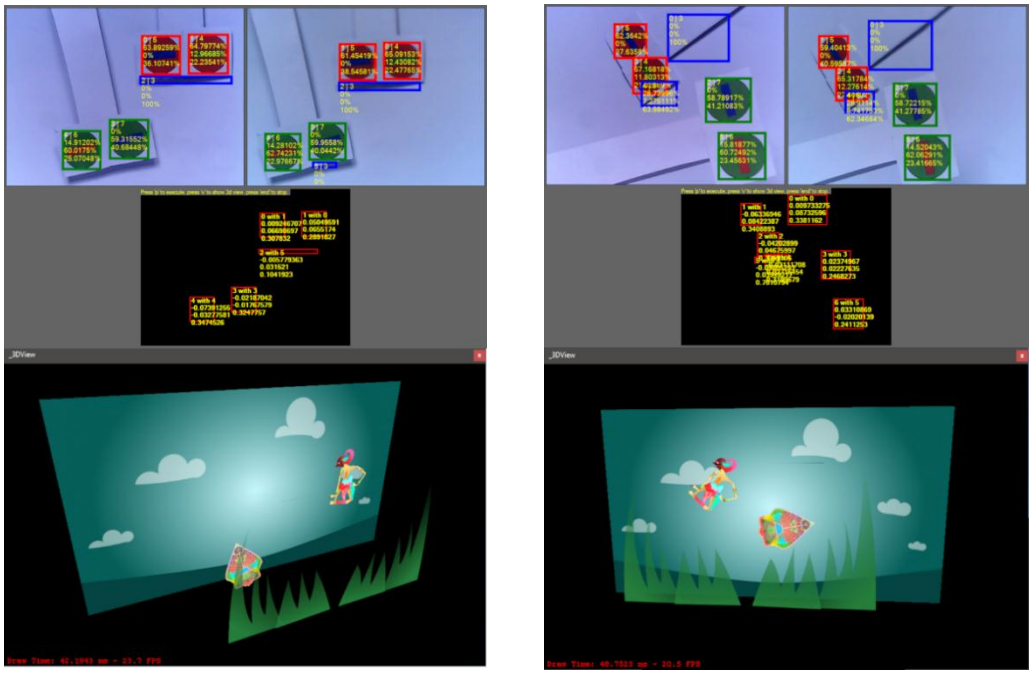
Figure 10 shows the look of a system that contains three viewpoints. The top part is for left and right webcam view, the middle part is for matched features view, and the bottom part is for 3D representation view. Features matched in the middle view are shown as red rectangles with their correspondent index numbers and position values (X, Y, and Z) printed below them. As we can see in the top view of Figures 10a and 10b, sharp shadows are also detected as features. We can anticipate this issue by giving more distance between markers and background so the shadows will not render too sharply. Another issue found in the top view image is that it is rendered in bluish color. This is unanticipated due to the webcam's automatic exposure and white balance capabilities. Since the system is supposed to detect features based on their color composition, the bluish color issue will add more blue value to the color descriptor and yield an ambiguous calculation.

Since position reconstruction in the 3D world is crucial in motion capture system, we need to test our system. A test has been done to measure the error of reconstructed depth values and it is presented in Table 1. The values presented within the Real column were obtained by a manual

measurement with a ruler from the marker position to the camera baseline. The values presented within the Reconstructed column are obtained from our low-cost motion capture system output. The error values are obtained by calculating the absolute values of subtracted Real and Reconstructed values. From these error values, we can determine the accuracy of our low-cost motion capture system, which ranged from 0.75 to 0.95 and has an average of 0.835 or 83.5%.

Table 1 Depth reconstruction test

Test	Real	Reconstructed	Error	Accuracy
1	20	19.84	0.16	0.84
2	21	20.80	0.20	0.80
3	22	22.11	0.11	0.89
4	23	23.11	0.11	0.89
5	24	24.19	0.19	0.81
6	25	25.05	0.05	0.95
7	26	26.23	0.23	0.77
8	27	27.25	0.25	0.75
9	28	28.15	0.15	0.85
10	29	29.20	0.20	0.80
Means				0.835



(a) (b)

Figure 10 System views

4. CONCLUSION

The use of stereo camera triangulation to find depth in a low-cost motion capture setup is reasonably effective since we can get an accuracy of 83.5% using webcams that cost around $7.69. It serves us as a forgivable result in obtaining the 3D point reconstruction data with some calculated error shown in Table 1. Parameter adjustment, such as baseline (b) and focal length (f), is crucial to getting better output.

The combination of white background, colored markers, and image processing methods, such as inversion and color filtering, effectively helps our attempt in features extraction (as shown in

Figure 8 where all markers can be extracted) even though the existence of shadows and the issue of white balance will worsen the result, as can be seen in Figure 10b. Those image processing methods may be considered as suitable to distinguish colored markers from white background. With those image processing methods, object detection within Aforge.NET framework is working more accurately.

Along with various colored markers, color descriptor effectively helps us distinguish each marker and improve the feature matching process since all markers in Figures 9 and 10 can be successfully distinguished. Feature matching is a crucial part of this low-cost motion capture setup since it lets us locate the corresponding features needed in the calculation of disparity (d). We hope our proposed methods would be useful in motion capture-related projects to improve effectiveness or performance.

5. REFERENCES

Aforge.NET, 2012. *Aforge. NET Framework*. Available Online at http://aforgenet.com/framework/

Bindu, S., Prudhvi, S., Hemalatha, G., Sekhar, N.R., Nanchariah, V., 2014. Object Detection from Complex Background Image using Circular Hough Transform. *International Journal of Engineering Research and Applications*, Volume 4(4), pp. 23−28

Budiman, R., Bennamoun, M., Huynh, D., 2005. Low Cost Motion Capture. *In*: B. McCane (Ed.), Proceedings of IVCNZ 05 (Dunedin N.Z. Edition, Volume 1). Dunedin: I & VCNZ.

Chao, S.-P., Chen, Y.-Y., Chen, W.-C., 2009. The Cost-effective Method to Develop a Real-time Motion Capture System. *In:* Fourth International Conference on Computer Sciences and Convergence Information Technology: IEEE

Cunha, A., 2009. *A Brief Introduction to Image Processing*. Center for Advanced Computing Research California Institute of Technology

Das, D., Saharia, S., 2014. Implementation and Performance Evaluation of Background Subtraction Algorithms. *International Journal on Computational Sciences & Applications (IJCSA)*, Volume 4(2), pp. 49−55

Guarisa, G.P., Angonese, A.T., Judice, S.F.P.P., 2016. Low-cost and Open Source Optical Capture System of Facial Performance. *In:* SBC – Proceedings of SBGames, ISSN: 2179-2259, Brazil

Heale, R., Forbes, D., 2013. Understanding Triangulation in Research. *BMJ Journals Evidence-Based Nurs*, Volume 16(4), pp. 98

Huang, Y., Kaufmann, M., Aksan, E., Black, M.J., Hilliges, O., Pons-moll, G., 2018. Deep Inertial Poser: Learning to Reconstruct Human Pose from Sparse Inertial Measurements in Real Time. *Journal ACM Transactions on Graphics*, Volume 37(6), pp.1−15

Kamencay, P., Breznan, M., Jarina, R., Lukac, P., Zachariasova, M., 2012. Improved Depth Map Estimation from Stereo Images based on Hybrid Method. *Radioengineering*, Volume 21(1), pp. 70−79

Kangas, V., 2011. A Comparison of Local Feature Detectors and Descriptors for Visual Object Categorization. *In:* Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan

Ko, J., Ho, Y.-S., 2016. Stereo Matching using Census Transform of Adaptive Window Sizes with Gradient Images. *In:* 2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA), Wangju Institute of Science and Technology, Republic of Korea

Lee, J., Jun, D., Eem, C., Hong, H., 2016. Improved Census Transform for Noise Robust Stereo Matching. *Optical Engineering*, Volume 55(6), pp. 1−10

Li, S., Lihong, H., 2014. Research of Background Segmentation Method in Sports Video. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, Volume 12(6), pp. 4274−4282

Miettinen, J.O., 2018. *Shadow and Puppet Theatre.* Asian Traditional Theatre and Dance, ISBN 978-952-7218-23-5, Theatre Academy of the University of the Arts Helsinki

Muhlmann, K., Maier, D., Hesser, J., Manner, R., 2002. Calculating Dense Disparity Maps from Color Stereo Images, an Efficient Implementation. *In:* Proceedings IEEE Workshop on Stereo and Multi-baseline Vision (SMBV 2001), August 2002, Kauai, HI, USA

Shen, Y., Peng, P., Gao, W., 2012. 3D Reconstruction from a Single Family Camera. *In:* IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI), October 2012, Nanjing, Jiangsu, China

Zecca, M., Saito, K., Sessa, S., Bartolomeo, L., Lin, Z., Cosentino, S., Ishii, H., Ikai, T., Takanishi, A., 2013. Use of an Ultra-miniaturized IMU-based Motion Capture System for Objective Evaluation and Assessment of Walking Skills. *In:* The 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), September 2013, Osaka, Japan