

LOW COST MOTION DETECTION AND 3D OBJECT MAPPING

R.G. Ranaweera¹, G.C. Sandeepani², N.Y.L. Udayakantha³, R.L. Vitharana⁴, Chathura De Silva⁵

¹ Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka. Email: ruchiragayan@gmail.com

² Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka, Email: chanika1118@gmail.com

³ Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka. Email: nyludaya@gmail.com

⁴ Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka, Email: lanka.vitharana@gmail.com

⁵ Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka, Email: chathura@uom.lk

ABSTRACT

This paper presents an image processing solution to animate 3D models according to an actor's motion. This solution detects motion using passive markers and then those detected motion data are recorded in appropriate format which enables mapping motion data to the model such that the model animates according to those animation data. A 3D human model is generated by creating human skeleton, mesh grid, skin and rig the mesh on the skeleton. The model is then used for animation process which can be rendered into a video footage.

Key words: Motion Detection, Motion Tracking, Quaternion Format

1. INTRODUCTION

Motion tracking is one of the key technologies in film industry, synthetic environments, security devices and other fields that require real-time information about the motion of a human[1]. Among them, the film industry uses new trends and concepts for their creations. Most of the times, image processing techniques are used extensively for their products. Film industry at present is rapidly approaching an era of using computer generated animations of 3D models according to motion of a real actor[2]. For this purpose most production companies have built their own model animation products. Almost all of them are commercial and most of the techniques developed by those companies are highly technical and need so many expensive hardware elements[2][3]. As a result, the production cost increases. This solution was proposed to solve this problem by providing a solution which has the ability to map detected motion data to a generated model only using image processing techniques.

This solution basically detects motions according to actor's movements and records them for later usage of the application. Then human models are generated by generating human skeleton, mesh grid, skin and then rig the mesh on the skeleton to come up with generated human model which is used for animation. Animation is done by mapping detected motion and generated model.

2. METHODOLOGY

Selecting a suitable motion capturing technique is essential. There are several existing methods and algorithms to capture motion. One method is using either active markers or passive markers. The actor wears markers near each joint of his body. But motion capturing using markers requires specific environments such as wearing a plain color suit[4]. There are marker-less techniques which can be used to capture motion, such as considering optical flow[1] of moving object. There are several drawbacks when using markers such as obtrusive and robustness. Even though that, it will be simpler to use marker based technique because it needs less computational complexity and those drawbacks have the potential to be reduced easily as well. Though it is more appropriate to use marker less techniques when concerned about reducing objects placed on user, this solution did not choose marker less or vision based techniques because they will increase the complexity of the solution. Passive markers do not require the user to wear wires or electronic equipment[5]. But instead it requires the user to wear garments which do not contain the color of the marker. Active markers uses optical systems triangulate positions by illuminating one or more LEDs[6]. Active markers increase complexity and cost than passive markers. Therefore, passive markers were selected.

Markers were used in such a way that total number of markers is equal to number of joints in the skeleton of the model used in this solution. Those markers were placed on actor's body in the same places as of the joints of the model skeleton since there should be a one to one mapping between them.

Environment for the capturing input videos of this solution needs to be carefully setup as point detection and tracking process solely depends on the quality of image stream. In order to map points, synchronized videos are required. Therefore same model digital cameras are used in order to record videos with same frame rate. 3D coordinates of points have to be detected and calculated as those detected points have to be mapped to a 3D human model. As digital cameras provide only 2D image sequences, two cameras are used in such a way that they are perpendicular to each other.

3. POINT DETECTION

Marker Positions have to be detected by the system. After user provides two input video files it extracts image frames as sequences from those video files. These frames are scanned pixel by pixel to identify colors of each pixel. White color pixels are detected by checking whether the pixel's bias color from RGB lies within all of the previously specified ranges for intensity levels for white color through iterations. These ranges were obtained on experimental basis by using the system several times. Then according to the marker color it filters every pixel and generates a gray scale image by assigning white color value for detected pixels and black color for other pixels. This is a simple algorithm which iterates throughout the whole image pixel by pixel and check whether the color values of the pixel lies within the predefined range. If so that pixel will be identified as a pixel in a point.

As coordinates are required for animation purposes centroid of color patches has to be found. For that boundaries of them are marked. Contours over color patches are found using the contour tracking algorithm and then centroid values are calculated. Contour tracking algorithm is an algorithm which searches image from top to bottom and left to right until the first point pixel is found and then iteratively find the boundary of that point. The same process is repeated for multiple color patches and for entire two frame sequences. A screen shot of an instant of multi point detection of the motion of actor is shown in Figure 1.



Figure 1: Left: original image. Right: tracked points

The resultant gray scale image is not smooth enough to identify white color patches of markers correctly. Because of that it creates multiple contours over a single marker in the frame. This issue has been occurred because of noise occurred due to light and illumination changes. These noises should be reduced in order to have good quality point detection. This solution does Image Convolution operations using Gaussian filter [7] in order to eliminate noises. The size of the Gaussian filter is called as the kernel. A kernel is usually a smallish matrix of numbers that is used in image convolutions. First 3x3 kernel is used but it didn't give expected results. That means the smoothness of the color patches were not in a level that can be used to identify them. Then next smallest matrix which is 5x5 kernel is used and near expected results were given by that. The expected values and resultant values are plotted in a graph (Figure 2).

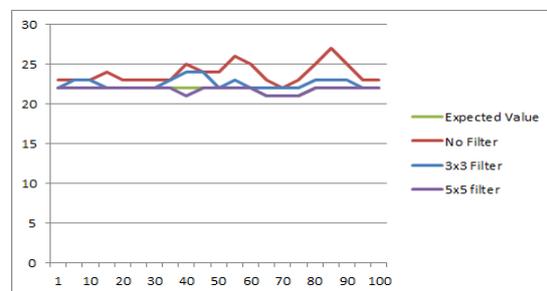


Figure 2: Total count of detected points with no filter; 3x3 Gaussian filter; 5x5 Gaussian filter with expected values

Even though increasing the size of kernel has the ability to increase the smoothness of images, it increases the processing overhead as well.

4. POINT TRACKING

For motion tracking, points in each frame should be tracked to identify corresponding points in each frame sequences. For that two methods are found, using different color markers for different points and then track points using color differences, using a filtering mechanism to filter out other points and being interest on the

particular point in entire frame sequence.

In the first approach, to identify the pixels values at the motion detection stage it needs different higher and lower threshold values. As this frame consists of RGB layers, it requires those two threshold values for each three layers. It increases noises and adds an additional constraint on actor's garments as those colors should not appear in the background.

As the second approach which is to track a point throughout the entire frame sequence, Kalman filter is used. Kalman filter is a well-known filter[8] which has the potential to be used to filter out areas of an image in an image sequence[9]. It takes predictions in to account while it is filtering the frame sequence.

Kalman Filter is an algorithm which uses a series of measurements obtained over time, and produces estimates of unknown variable or variables that tend to be more precise than those that would be based on a single measurement alone[10]. In order to make sure that null values are not assigned to any point, the area that we are interested on is predefined. The nearest point which has been detected is found as the first step. Then a range is given to check whether that nearest point lies within that range. Therefore the length of the line which is drawn between the predicted point and the actual point is calculated in order to find the nearest point. Then that value is compared with the minimum value and then comes to the decision of what should be added. Then all the points are tracked using Kalman Filter. Figure 3 illustrates the path that points were gone at an instance. Then those tracked points are saved for later usage.



Figure 3: An instant of object tracking using Kalman Filter. Here the actor is waving his arms and that motion is tracked

5. MODEL GENERATION

The next major part is to generate a 3D human model. There are model generation tools available such as Maya, 3Ds Max, Lightwave, Softimage, Blender and Cinema4D. As the first step the polygon model has to be created. The polygon model is created from a standard object. Box is used in this solution as the starting object. The actual shapes are generated in the body by deforming it and creating other boxes around it. For the texturing purpose, the model has to be unwrapped. Unwrapping leads to some unexpected deformation in character skin called stretching and seams. So stretch and seams are removed from character model. After unwrapping the model it has to be textured in order to have a realistic view. Textures are taken from photos of existing person and are used to texture the unwrapped image. After texturing, the character is rigged for animation. This is the process of preparing the character for animation. The idea is to use special helper objects and modifiers to prepare a set of tools that make the animating process easy.

6. ANIMATING THE MODEL

To animate the model, available animation libraries are considered. They are: XNA, Jmonkey, Crytec. Among them this solution chooses to use XNA.

The final stage is to map the motion data and the model data. In XNA each point is having its own axis system which is given as the orientation in quaternion format[11] in joint definition data. Each point's, both translation and orientation are given relative to its parent's orientation. It maintains same relative orientation between each child and parent even there is a motion. Therefore the axis system should be rotated by that rotation angle which has to be calculated in such a way that the original orientation is unchanged.

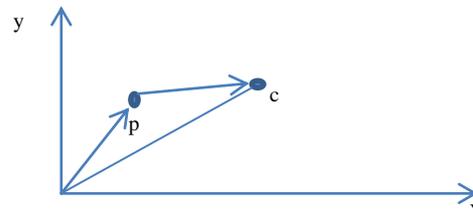


Figure 4: Relative vector of point wrt parent

That rotation angle is calculated as a number in quaternion format which is a number system that extends the imaginary number system by having three imaginary components. Detected points

which are relative to same x,y,z coordinate system are mapped to the joints in the model which are oriented relative to its parent.

In figure 4 parent point and child point are indicated as 'p' and 'c' respectively. The relative rotation of 'c' with respect to 'p' is calculated in quaternion format. This process is continued for all points in every frame which is later saved in a file. The resultant points which are in quaternion format are fed to XNA to animate the model.

7. CONCLUSION

This solution detects and tracks motion of a human animator and maps that motion into a 3D human model. It uses two cameras to get motion data in 3D space. White color passive markers are used as the motion detection technique in order to reduce the complexity of the solution.

Videos are separated into frames and then pixels with white color intensity are detected. Noise is removed by using 5x5 Gaussian filter which gives 95% accurate results. Kalman filter is used to track points frame wise. Models are generated using model generation tools by creating the polygon model, unwrapping the model, texturing using images and rigging the model. Model is animated using XNA and coordinates in x,y,z axis system is converted to represent as rotation of each point with respect to its parent in quaternion format.

8. REFERENCES

- [1] D. Moore, "A real-world system for human motion detection and tracking," California Institute of Technology, 2003.
- [2] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 491–500, 2002.
- [3] J. Lee and K. H. Lee, "Precomputing avatar behavior from human motion data," *Graphical Models*, vol. 68, no. 2, pp. 158–174, 2006.
- [4] B. Rosenhahn, T. Brox, U. Kersting, A. Smith, J. Gurney, and R. Klette, "A system for markerless motion capture," *Künstliche Intelligenz*,(1), pp. 45–51, 2006.
- [5] A. C. Sementille, L. E. Lourenço, J. R. F. Brega, and I. Rodello, "A motion capture system using passive markers," in *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*, 2004, pp. 440–447.
- [6] L. P. Maletsky, J. Sun, and N. A. Morton, "Accuracy of an optical active-marker system to track the relative motion of rigid bodies," *Journal of biomechanics*, vol. 40, no. 3, pp. 682–685, 2007.
- [7] G. Deng and L. W. Cahill, "An adaptive Gaussian filter for noise reduction and edge detection," in *Nuclear Science Symposium and Medical Imaging Conference*, 1993., 1993 IEEE Conference Record., 1993, pp. 1615–1619.
- [8] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: particle filters for tracking applications*. Artech House, 2004.
- [9] B. Stenger, P. R. S. Mendonça, and R. Cipolla, "Model-based hand tracking using an unscented kalman filter," in *Proc. British Machine Vision Conference*, 2001, vol. 1, pp. 63–72.
- [10] G. Bishop and G. Welch, "An introduction to the kalman filter," *Proc of SIGGRAPH*, Course, vol. 8, pp. 27599–3175, 2001.
- [11] D. Murtagh, "Pose-Space Deformation on top of Dual Quaternion Skinning.," Dissertation presented to the University of Dublin, Trinity College, 2008.