Motion Capture and Animation Using Digital Image Processing

Eliza Mae A. Saret and Jaime Samaniego

We present a practical method for human posture recognition and animation through the use of digital image processing as an alternative for expensive means of motion capture. Given an input video of a moving subject wearing colored markers, the application identifies the position of the body parts and maps its movements into a cartoon character.

1 Introduction

Significance of the Study

According to an article by Steven Dent, motion capture or mocap is defined as the process of transferring complex actions of a model subject into an animated character. It can be traced back in 1914 where Animator Max Fleischer invented rotoscoping which uses the method of tracing the movements and facial gestures frame-by-frame from live-action footages. Creating this form of digital animation by hand takes a lot of time and human effort to complete. Wanting to automate this process, bio-kinetic researcher Tom Calvert from Simon Fraser University invented mechanical capture suites which used electronic sensors to output computer animated figures. This paved way to the use of more advanced methods in motion capture like the ones that are commonly used in animated films nowadays. On-set performace capture requires actors to wear tight suites with markers placed on different parts of the body and be surrounded by

several special cameras while they act out their scenes. Various software applications are then used to map these markers on 3D characters such as Autodesks Maya, MotionBuilder and 3D Max. Recently, Xbox had also released KINECT 2 sensor, a hardware which allows bone and facial gesture tracking through infrared vision and could be used in navigating and interacting with the console. [1] [2]

Alexandre Szykman and João P. Gois mentioned on their study last 2014 that the cost of motion capture can be a huge barrier for people who want to create animations for the sake of art and science. They further stated that animations that are the same level as Hollywood films are over-priced and that independent professionals who want to create such animations often do not make enough profit to pursue mocap-based animation. It can also be seen from the previously given examples that using motion capture can be impractical for independent creatives especially for those who are only starting in the field of animation, those who aim to map human motions for scientific studies, or those who just want to explore motion capture for entertainment purposes. Due to these reasons, it is reasonable to look for a more convenient and less expensive means of motion tracking that will allow more people to use motion capture in animation. [3]

A possible way of tracking human gesture can be through the use of digital image processing. Instead of purchasing special cameras or electronic motion sensors to track the movements of the human body, the computer itself could be the one detecting the required anatomical features of a moving person by applying human body detection algorithms into a given input video. Color detection and the application of human body geometric constraints will be used in the identification of these anatomical features. Once the needed features are successfully detected, a skeletal figure of the moving human could be outputted and mapped into a character making the whole process more convenient.

The result of this study created a way to extend motion capture to people who are unable to afford the special equipments needed for this technology. By testing the use of digital image

processing in determining human body movements, the feasibility of using these techniques in animating human motions was confirmed. These results can be used in the development of a more practical means of motion capture and the reduction of the production cost in film and game animation.

Objectives

This study aims to create a computer program that can be used to facilitate motion capture and animation by fulfilling the following objectives:

1. To use digital image processing and knowledge in human body geometric constraints in detecting and identifying human body parts
2. To use the detected human body parts in drawing a moving skeletal figure representing the motion executed
3. To map the motions of the moving skeletal figure in animating a character

Review of Related Literature

Few studies have been conducted on extracting data from human body images through the use of digital image processing. These informations were applied in various fields such as in surveillance systems, motion analysis and medical analysis. Researchers were able to separate detected human figures from photos and identify their body parts using various algorithms.
Park et al. (1999) reconstructed an image of a human body into a 2D model with labeling on ten main body parts (the head, two upper arms, two forearms, the torso, two thighs, and two shins), twelve joints (two shoulders, two elbows, two wrists, two hips, two knees, two ankles) and two pseudo joints (the neck and pelvis). Watershed segmentation algorithm and region merging was used in determining the regions in the image. Skin region extraction is then applied by marking

prominent skin color regions. Curve segments are then used to enclose the detected skin color regions in order to create 2D ribbons that can fit a human body. To determine the posture of the body, the 2D ribbons were labeled into what parts they represent. After these steps, a stick figure representing the model could be generated. [4]

A study conducted by Chowdary et al. (2014) explored and discussed different gesture recognition methods in detecting the number of fingers held out by a hand on a plain background. These methods were the use of pixel count algorithm, detection of circles, morphological operations, and scanning method. All the given methods involved converting the input into a binary image representing the skin region as white pixels and the background as black pixels. In pixel count algorithm, the number of white pixels are counted and classified according to predefined ranges representing the number of fingers. Marking the fingers with circles before taking the image and counting these markers in the program can also be used in getting the needed information. The use of morphological operations involve using hit or miss operation which will result to an image with only one side of every finger. Dilation is used to be able to further identify and count the objects that remained in the image. The scanning method could be classified into two ways: linear horizontal scanning and linear vertical scanning which divides the image into two. The number of detected transitions from 1 to 0 in the binarized image is equal to the number of fingers held out. At the end of the study, the researchers were able to conclude that the best way to detect the number of fingers was through the scanning method which gave accurate results for 82.47% of the input images. [5]

On the same year, Toshev and Szegedy used deep neural networks (DNN) in human pose estimation. Researchers were able to detect four most challenging limbs: lower arms, upper arms, lower leg, and upper leg given an input image of a human body. Prediction of poses were used to identify the position of body parts that are not visible on the image. At the end of the research, it was concluded that the use of a DNN-based approach gives an advantage of capturing

pose in a holistic manner and outputting a more realistic models for input images. [6]

These studies showed that the use of digital image processing techniques in determining human body parts in images and identifying their positions is possible through various algorithms. Since a video can be defined as only a series of images, some of these concepts were also applied in capturing motion in videos. The study aimed to identify the different poses done by a human in an input video and represent it in animating a model.

Date and Place of Study

This study was conducted from January to April, 2018 at the University of the Philippines Los Baños, Institute of Computer Science.

2 Materials and Methods

OpenCV 3.3.0 C++ and Qt Creator 4.5.1 were used in creating the application. OpenCV C++ was used in preprocessing the input videos, identifying body parts and in creating the character animation. Meanwhile, Qt Creator was used to create the user interface for playing and configuring the animations.

Since the study aims to create an application that will minimize the cost of motion capture, there was no need to use special cameras or electronic motion sensors in capturing the human subjects. Any device with a camera can be used to take the input video for the application. A total of 11 colored markers were placed on the human subject in order to track its movements. Seven colors respresenting different body parts are recognized by the application: red, blue, yellow, green, orange, purple and pink for the head, neck, hips, left hand and foot, left elbow and knee, right hand and foot, and right elbow and knee respectively. The captured videos were inputted in the application where its frames were subjected to preprocessing, body part identification, and animation using OpenCV C++.

Setup

The initial set of input videos used in the application consists of the moving subject wearing black with different colored markers and a plain black background. This was taken inside a room with only a single bulb as the source of light. The resulting videos in this kind of setup had dark and inconsistent lighting which became a problem in the color detection part of the application. In order to come up with better input videos, a mini-studio was setup inside the room to create a brighter and more consistent lighting which allowed the application to better detect the colored markers. The studio consisted of two small led lights positioned on both sides of the phone tripod to provide a balanced amount of light on all parts of the frame and a plain white background to prevent any wrong color detections that do not belong to the subject to be captured. All input videos were captured using a cellphone camera.

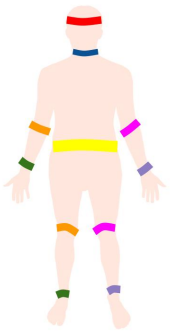


Figure 1: Positioning of Colored Markers on Human Subject

Preprocessing

Once the video was loaded into the application, its contrast, brightness and saturation were adjusted to allow the application to detect the colored markers better. The dimensions of the video was also reduced to allow the program to process each frames faster. Next step was to

convert the frames from BGR (Blue-Green-Red) to HSV (Hue-Saturation-Value) colorspace which is more suitable for color based segmentation.

In the application, the user can change the values used in adjusting the contrast, brightness and saturation of the program as well as the threshold values used for detecting the colored markers. The user can also choose to rotate the video to fix the orientation of the subject.

Body Part Identification

The first step in body part identification was to detect the individual colored markers in the frame using the HSV threshold values mentioned earlier. This returned an image containing blobs representing the parts of the frame falling under the specific color. Initially, there were only seven uniquely colored body markers each representing a single body part (head, neck, hips, left hand, right hand, left foot and right foot) but due to the need for a more accurate posture detection and the lack of other color options, four colors were used twice to represent different body parts. The right elbow was paired to the right knee, right hand to the right foot, left elbow to the left knee and left hand to the left foot. This then lead to the challenge in identifying the body part corresponding to the detected colored markers which were reused. There were two possible body parts to be assigned to two points of the same color. The problem was managed by assuming that the hand and elbow were always located at the upper part of the frame and that the knee and foot were always at the bottom.

The program looked for the largest blob if the color was used for only one body part and the two largest if the color was used for two body parts. Through these blobs, the program was able to identify the coordinates of the detected colors. The program can immediately assign the coordinates to the respective body part when the color was only used once. If not then the two detected coordinates for the given color is compared and the one located above is considered as the hand or the elbow and the one below is the knee or the foot.

Finally, the identified coordinates of the body parts will be saved into a structure to be used in the animation process. The detected coordinates are also connected accordingly in order to create a stick figure representation of the body movements.

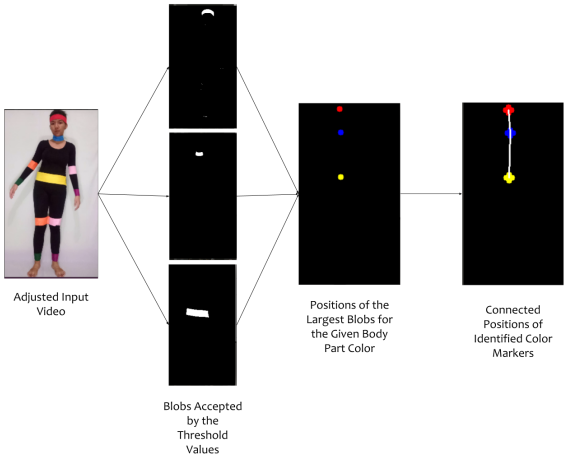


Figure 2: From Input Video to Stick Figure (head, neck, and hip)

Algorithm 1: Locating Body Parts

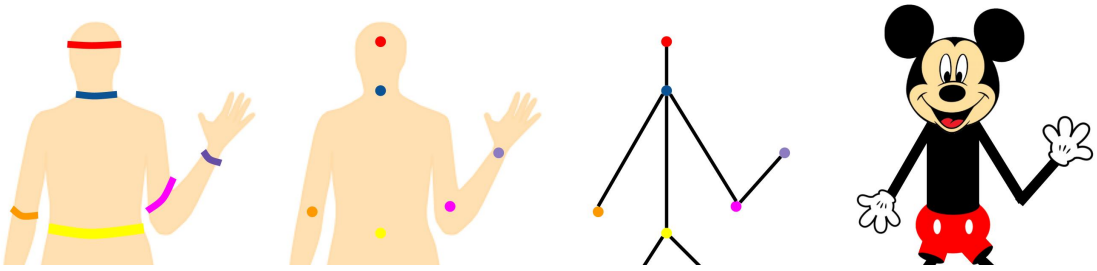
Input: Mat **img** (current frame), struct Pose **currPose** (current positions of body parts)

Output: Mat **imgDetected** (body part representations)

```
for each color:
  scan img for blobs that fall under the current color;
  if current color was only used once:
    blobPosition = largestBlobXY;
    correspondingBodyPart = blobPosition;
  else if the current color was used twice:
    blob1Position = firstLargestBlob;
    blob2Position = secondLargestBlob;
    if (blob1Pos.y < blob2Pos.y)
      upperLimbBodyPart = blob1Position;
      lowerLimbBodyPart = blob2Position;
    else
      upperLimbBodyPart = blob2Position;
      lowerLimbBodyPart = blob1Position;
end
draw colors on the points representing the body parts and connect
    them accordingly;
```

Animation

Images of body parts that were too detailed to draw using the built-in functions in opencv were loaded at the start of the animation process depending on the character chosen by the user. The angle between two endpoints were also taken to know how much the loaded image should be rotated to match the movements. From the coordinates of the body parts detected on the previous step, the images were inserted into the position of their correspnding body parts. Lastly, the limbs were represented by using built-in opencv functions to draw lines from one joint to another.



(a) Colored Markers (b) Color Detection (c) Model Construction (d) Mapped Animation

Figure 3: Body Part Identification Stages

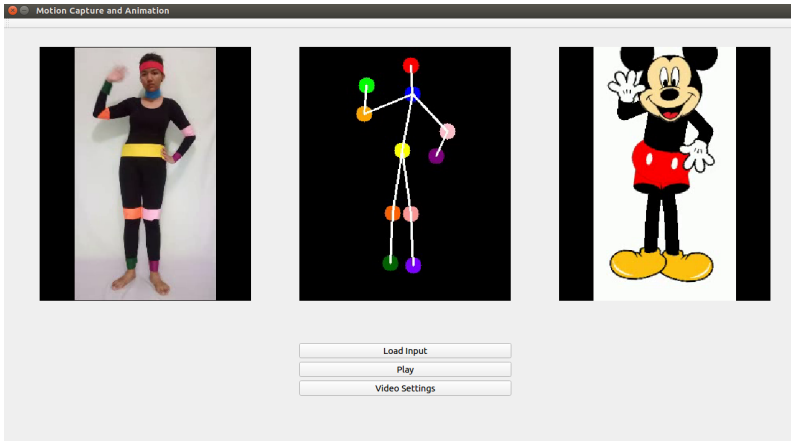


Figure 4: Final output page of the application

Three videos were displayed in the application window. The first one displayed the adjusted input video. The second contained a stick model of the subject's movements. The last video showed the animated movement of the input video. The user can choose which character to be animated inside the video settings. At the same time, the user can also adjust settings such as contrast, brightness, saturation, and the threshold values for the colored markers used.

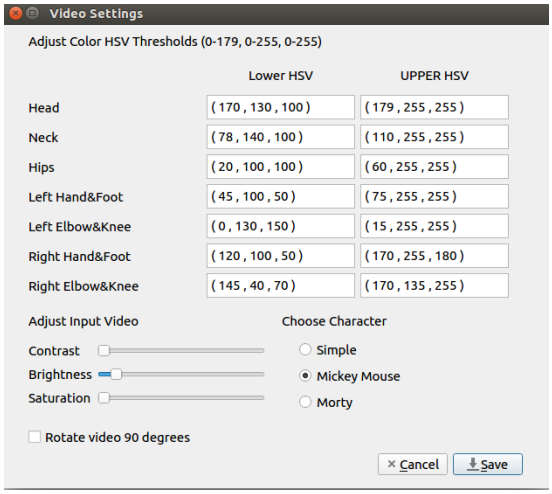


Figure 5: Video configuration options

3 Results and Discussion

Three sets of input videos were used to test the ability of the application to detect body movements and animate them. The first set being the ones taken with black background and inconsistent lighting. Colored markers located at the bottom part of the frame could not be classified correctly because its color value didn't fall under the given threshold values. Expanding the given upper and lower threshold values made the program accept more positions of colors that are present in the frame. This resulted in a lot of misdetected body parts and in an inaccurate representation of the subject's movements for the animation part.

The second set of videos were the ones taken in the mini-studio setup with a balanced amount of

light throughout the frame. The good amount of lighting gave way to much easier classification of colored markers. The upper and lower threshold values for some of the given colors could also be narrowed because there was already a uniform set of colors used in taking the video. Detected body parts were more correct with a little misdetections on colors that are almost similar. The animation output was more acceptable and the movements of the subject can be seen clearer in the animation than the previous set of videos. There were only a few frames with misdetected body parts and misrepresented posture.

The mini-studio setup was still used for the third set of input videos. The only difference is that the movements of the subject were made to purposely break the body part assumptions of the program. This included videos when some colored markers were not visible in the frame, when the human subject's body is facing a different direction, when the input videos were rotated so that the human body's orientation was not correct and when the posture allows the lower limbs to be located above the upper limbs. Instead of removing the body parts when the markers are not present in the frame, the program identified other colored markers as that body part resulting in more misdetections than the first and second set. When lower limbs such as the knee and feet were found above the upper limbs, the resulting body part identification and animation assigned them as their upper limb counterparts.

On running the input videos on a machine with Intel Celeron processor N3160, it takes the application a few seconds to a minute to produce the outputs depending on the length and the quality of the input video.

For videos of lengths five seconds, ten seconds, fifteen seconds and twenty seconds all with 720p video quality, the application took an average of 8.3 seconds, 13.9 seconds, 22.4 seconds, and 30.5 seconds to process and write the outputs respectively. This is due to the amount of processes done on every frame of the input videos before finally displaying it to the user once all of the movement detection and animation is done.

Input Length	Trial 1	Trial 2	Trial 3	Average Running Time
5s	8.3s	8.3s	8.3s	8.3s
10s	13.9s	14.0s	13.7s	13.9s
15s	23.2s	22.1s	22.0s	22.4s
20s	30.5s	30.7s	30.3s	30.5s

Table 1: Average time it takes the program to detect the body parts from the input video and display the animated output to the user

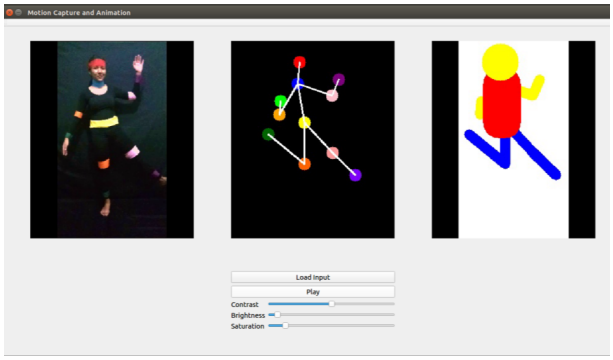


Figure 6: Final outputs from input video with poor lighting

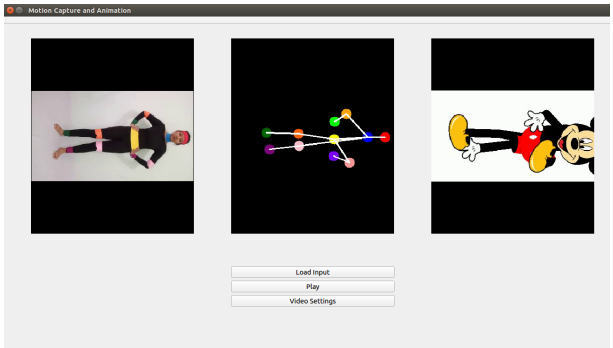


Figure 7: Final outputs from input video with wrong orientation

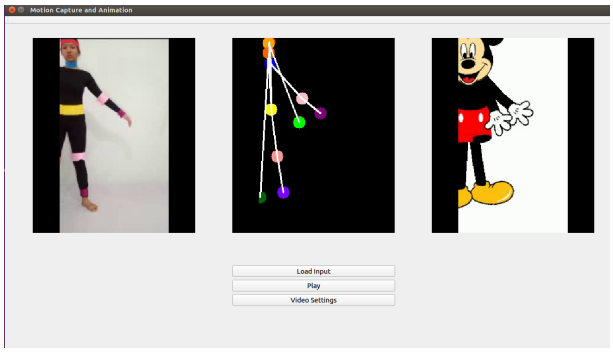


Figure 8: Final outputs from input video with missing markers

4 Conclusion and Future Work

The detection of colors using digital image processing worked well in the identification of body parts and in keeping track of the subject's movements. The output of the demo videos proved that digital image processing can be used as an alternative for expensive means of motion capture based animation. The application created can be used for studying the movements of a subject or for creating small animation projects.

Since the program uses threshold values for color detections, it is important that there is a consistent amount of light throughout the whole frame to identify the colors better and to be able to assign them into their respective body parts more correctly. Having colorful backgrounds for the input video is also discouraged because some of these colors might have been used as colored markers and assigned to some body parts. Using plain and neutral background is more appropriate in order to avoid wrong identification of body parts. Some issues in the animation, such as overlapping body parts and wrong posture detections, also arise when not all the markers are visible inside the frame. This is because the program tries to find the color corresponding to each body part and assigns the upper left corner of the frame to that body part when no color is detected. On other cases when the marker is not present but its color value is close to the color values of other markers, the program assigns the wrong body markers to the body parts with

markers that are not present.

Other improvements to the application can be done such as fixing the animation problems when the markers can't be seen on the frame or when the subject's body is not facing the camera. Using better human body constraint definitions can also help in increasing the ammount of movements a subject can do.

demo videos:

1. `https://drive.google.com/file/d/16WUcdiERpY_tmkyTIbcj1brSWtFplAHv/view?usp=sharing`
2. `https://drive.google.com/file/d/1w3JnSumqBOnleQeoQKxJUXnkWWxrVJZx/view?usp=sharing`

repository:
`https://github.com/lizsaret/Motion-Capture-Animator.git`

References

[1] S. Dent, What you need to know about 3d motion capture.

[2] S. Dent, How i turned my xbox's kinect into a wondrous motion-capture device.

[3] A. G. Szykman, J. P. Gois (2014).

[4] J. Park, H. Oh, D. Chang, E. Lee (1999).

[5] R. Chowdary, N. Babu, T. Subbareddy, B. Reddy, V. Elmaran (2014).

[6] A. Toshev, C. Szegedy, *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014).