Acquisition of Voxel Data for Human Body Models

Richard Peroutka
Johns Hopkins University
MERIT program
guano313@aol.com

Abstract

3D modeling of human shape and motion is a challenging problem with widespread applications, such as motion capture and human-computer interaction. We create an integrated system that executes, in close to real time, all the steps needed to create a voxel representation of a human using 2D data acquired from multiple cameras. A robust background model and background subtraction algorithm are included in this system to create a 2D silhouette of the human subject. A voxel based feedback to the system to refine the 2D silhouettes is also considered.

1. Introduction

The 3D reconstruction of a human model in a scene has vast applications in the fields of virtual reality, surveillance, human machine interface, and motion capture. Building a 3D model is the basis for 3D human tracking, where the model parameters need to be estimated in time to reflect the motion of the person. Though it is possible to achieve a 3D reconstruction of a human through the use of markers or other devices attached to the human body, this is not natural and therefore not practical for applications such as surveillance. Instead, it is preferable to use a multiple camera system that is calibrated in a way to allow an efficient representation of the human from multiple angles. This is an area of research known as markerless motion capture [8].

Before obtaining a 3D reconstruction of a human, the object has to be segmented from the background in all of the camera images. This process is known as background subtraction, or foreground segmentation [10]. In indoor capture environments, it involves creating a stable background model that compensates for changes in illumination as well as noise from the image acquisition process. This background model is subtracted from the foreground images, which consist of the human located in the same background, to create a silhouette of the human's body.

Silhouettes created through background subtraction form the basis for creating a volumetric model that represents both the shape and position of the human in world space. This is done through the principle of perspective projection, by which the object has to lie within the bounding volume formed by the silhouette and the camera viewpoint [11]. Therefore, by using multiple silhouette images from different viewpoints, the 3D shape of an object can be reconstructed by intersecting all of the bounding volumes [11]. This method is a form of shape from silhouettes, or voxel carving, which can be used to obtain an accurate 3D human reconstruction due to the smooth nature of human beings.

In the following sections we present a system for obtaining both silhouettes and voxel data in close to real time. The structure of the system is discussed in section 3. The background subtraction and voxelization algorithms are discussed in sections 4 and 5, respectively. Finally, a summary of the results and a discussion of future ideas is discussed in sections 6 and 7.

2. Capture Environment

Most of the image capture was done in the Keck Lab at the University of Maryland. The lab contains 16 stations at different viewpoints that each have 2 cameras mounted on them, for a total of 32 different cameras. As of now, this lab is going through an upgrade which involves the acquisition of 8 to 10 Pixelink 7A142 color firewire cameras. Currently only two cameras are operational and they provide synchronized images at 512x640x3 resolution and 15 fps. Due to the lack of cameras and calibration, though, acquisition of complete voxel data was not possible.

Due to the inability to obtain voxel data from the Keck Lab, the system that was created was simulated using images courtesy of the Stanford BioMotion Laboratory. These images were taken from 8 cameras at different viewpoints in good lighting. They were 494x656 ppm RGB images taken in an indoor background.

The initial development of algorithms for background subtraction and voxel acquisition was done using MATLAB. After successful implementation in MATLAB, the algorithms were then coded in C with the aid of functions provided by the Intel OpenCV library. This allowed for a faster implementation in order to achieve the goal of a system that performs in close to real time.

3. System Design

The goal of the system is to perform a method of acquiring voxel data quickly and online to form the basis for later exploration into parameter estimation of a human model and tracking of this human model. A robust background subtraction algorithm is presented which is focused towards indoor, cluttered environments. The system also implements a method of voxel carving similar to the one used by Cheung et. al[11].

The structure of the system in theory involves the use of 5 computers. 4 of the computers each write to disk the captured images from 2 cameras at different viewpoints. An external trigger is sent out from the main computer in order to synchronize all eight of these cameras. Each computer then performs background subtraction and partial voxelization for the two cameras pertaining to it. A fifth computer computes the final, complete voxelization from the partial voxel data obtained from the other computers. This design is structured so that parallel processing can be used in order to make the implementation as fast as possible.

It was not possible for this system to be fully realized in time for the completion of this project, but the programs to perform background subtraction and compute voxel data are modeled after this theoretical design.

4. Silhouette Generation

4.1 Previous Work

Previous work in this subject has focused on eliminating some of the key underlying problems that are associated with background subtraction. Illumination changes, shadows, and highlights are the most common problems in background subtraction that cause pixels to be misclassified. Other problems which occur include backgrounds that vacillate, such as waving trees, foreground objects that are camouflaged, and background objects that have been moved [12]. The latter set of problems primarily occur in outdoor environments, which are more variable, and therefore require more complex algorithms for detecting objects. For example, there are algorithms that use single or multiple

Gaussian distributions to model each pixel[1,4]. With these algorithms, distributions of pixels in the background are compared to distributions of pixels in run-time images to classify a pixel as foreground or background. Other algorithms for outdoor environments use adaptive kernel density estimation techniques to build statistical representations of the background and foreground [5,13]. These techniques deal with the uncertainty that is inherent in outdoor scenes.

The above algorithms serve no purpose in indoor environments because they offer a complexity that is unnecessary for the circumstances. For instance, any illumination changes dealt with indoors are much more gradual, unlike the abrupt changes in illumination that sunlight can create. Even in a very still outdoors background there is still slight movement, such as the wind moving objects and cloud movement. Indoors, these things don't occur, and the main concern is only dealing with shadow removal. Therefore, a complex statistical background model is not necessary for algorithms focused on indoor capture environments.

Previous background subtraction algorithms dealing with indoor environments include the Pfinder system [1]. This system uses a Gaussian model of each pixel and classifies a given run-time pixel as foreground or background based on the mean and covariance of that pixel's distribution over time. The Pfinder uses the YUV color space to model the scene surrounding the human as a texture surface, with each point on the surface associated with a mean color value and a distribution about that mean [1]. The W4 system uses the YUV color space in an indoor environment as well. In this system, the background is modeled by representing each pixel by its minimum and maximum intensity values, as well as its maximum intensity difference between consecutive frames. Once this model is obtained, foreground segmentation is done through a four stage process: thresholding, noise cleaning, morphological filtering, and object detection [3].

The algorithm presented here for background subtraction is most similar to that of Cheung et al [11]. Their algorithm uses a small number of region- based thresholds for each camera in order to classify pixels. The thresholds are based on the angle and the intensity difference between the RGB color vectors in the background model and in the current images. They have an upper and lower threshold for the intensity difference for each pixel. If a pixel is above the upper threshold, it is part of the foreground, and if it is below the lower threshold, it is part of the background. Those pixels that have an intensity difference in between the upper and lower thresholds are subjected to a third threshold which measures the angle, or color difference, between RGB vectors. This third threshold is used in order to eliminate shadows, which can often be misclassified as part of the object. This is because a shadow will have lower intensity than the background even though it has the same color information. The color angle between the shadow and the background is therefore small, and is classified as background if it does not pass the third threshold.

Cheung et. al also use a region based approach to their thresholds, since shadows tend to occur in certain regions (i.e. floor) and not others (i.e. above floor) [11]. They use these two separate regions to implement their thresholds, and therefore have a total of 6 thresholds for each of their five cameras.

4.2 Background Modeling

The background images in our system are obtained during a period of capture in which there is no person in the scene. 10 images, or one second worth of images at 10 frames/s, are captured and averaged to create a mean background image. This model is sufficient for the system, since it works on an indoor environment where there are not drastic illumination changes or moving background objects. This background modeling is done for all 8 cameras before the background subtraction process begins.

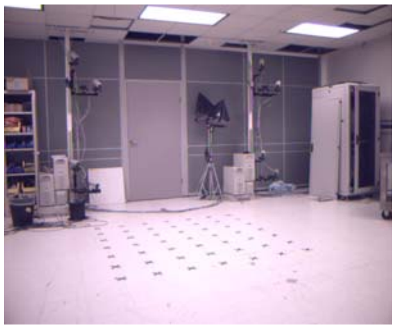


Figure 1: Background model

4.3 Background Subtraction

The silhouette generation that in the system involves the same color information used to determine thresholds in Cheung et. al, but a smaller number of thresholds. Only two thresholds are used for each camera, one that uses intensity difference and one involving color angle.

For a background image pixel, where the color vector for the (i,j)th pixel is denoted by **B(i,j),** and for a run-time foreground image, whose color vector is **F(i,j)**, the formula to obtain the silhouette image **S(i,j)** is given by:

1. Calculate the magnitude of the intensity difference

$\mathbf{D(i,j)} = \| \mathbf{F(i,j)} - \mathbf{B(i,j)} \|$

```
If D(i,j) > DIFF_THRESH
        then the (i,j)th pixel is foreground.
        else the (i,j)th pixel is background.
```

2. Calculate the angle

$\boldsymbol{\theta} = [((\mathbf{F(i,j)} \bullet \mathbf{B(i,j)}) / (\|\mathbf{F(i,j)}\| \, \|\mathbf{B(i,j)}\|)]$

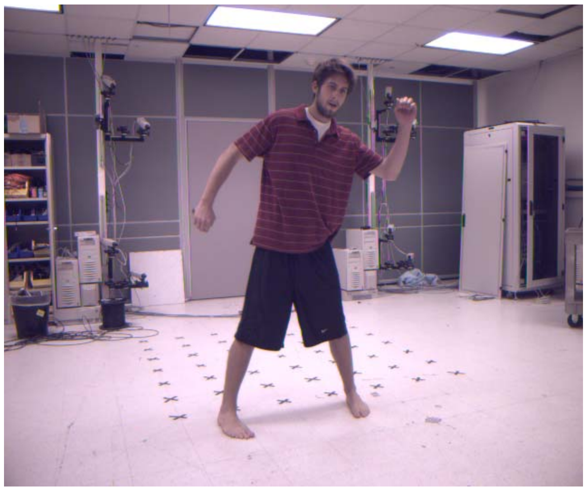
```
If θ > ANGLE_THRESH
        then the (i,j)th pixel is foreground.
        else the (i,j)th pixel is background.
```

3. Do the following AND operation:

$\mathbf{S(i,j)} = \mathbf{D(i,j)}$ AND $\boldsymbol{\theta}\mathbf{(i,j)}$

4. Dilate image

Here, ||F|| and ||B|| represent the norm of F and B, respectively, and •·is the dot product operator. In the first threshold, the color vector for each pixel in the current image is being checked to see whether it differs enough in magnitude from the background pixel to be considered part of the object. The second threshold checks to see if the angle between the background and current image is large enough for the pixel to be considered part of the object, since an angle that is small is most likely a shadow. The threshold values, DIFF_THRESH and ANGLE_THRESH, are predetermined manually by studying the color and shadows of the room before starting the system. This is preferable because a pixel-based method



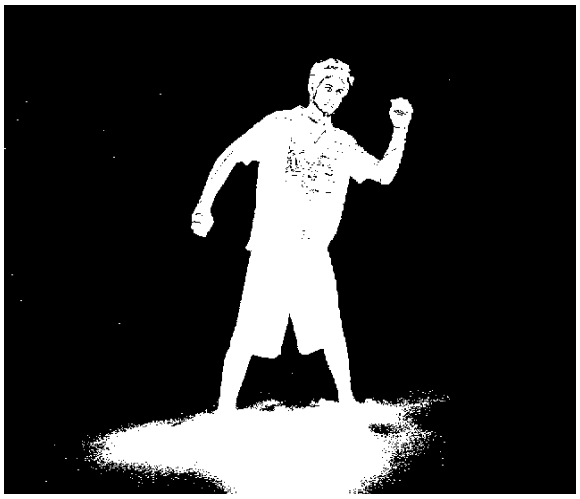


Figure 2c: Difference Image

uses a large number of thresholds that are usually determined by the color variances and can’t be fine tuned individually [11].

After each threshold is computed, the two thresholded binary images are ANDed together in order to remove the shadows and noise. The final silhouette image is then dilated in order to fill any small holes that appear in the body. The resulting images that are formed after thresholding are shown for one frame in figure 2.

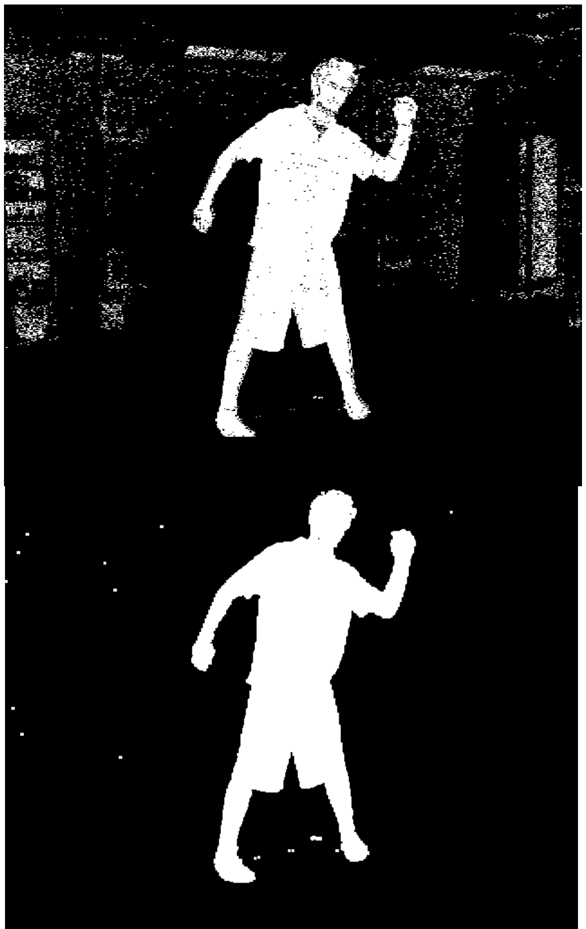


Figure 2b: Angle Image

Figure 2d: Final Image

5 Voxel Acquisition

5.1 Previous Work

Voxel carving, or shape from silhouettes, is a process that uses camera calibration parameters to produce a 3D volumetric model of an object based on 2D silhouette images. As stated in [11], when a silhouette is reverse-projected perspectively into the world space, a conic surface is obtained starting at the camera viewpoint, intersecting the silhouette, and finally enclosing the entire object. Here the voxels have a conic shape and the size varies with distance from the camera. Intersecting the conic volumes from all the images creates the visual hull, defined as the maximal object that gives the original object's silhouette from any viewpoint [14]. In practice, however, this is not a preferable method of constructing a volumetric model because it is computationally expensive [8,11].

A more preferable method of voxel acquisition involves the use of a rectangular bounding box that encloses the relevant world space [8]. For example, in the case of human subjects, this could be a 2m x 2m x 2m cubic grid of voxels that surrounds the human. Each voxel is then projected to the silhouette images using calibration data which contains the internal and external camera parameters that allow the transformation between 3D world coordinates and 2D image coordinates. Once this is done, the index of the pixel that corresponds to its voxel is checked to see if it lies within the silhouette of the object. A voxel whose projection resides in the silhouette of all or most of the cameras is considered to belong to the object, and those that do not are carved away from the grid.

Before voxels are projected, however, a decision must be made on which and how many vertices of the voxel should be projected to the image. One method is to perspectively project all 8 vertices of the voxel to each image plane and compute the convex hull of the eight projected points on the image. By testing all of the pixels inside the convex hull for all of the silhouette images, the voxel can then be classified as part of the object or not. Cheung et. al devised an algorithm that followed this method but instead tested Q uniformly distributed pixels inside the convex hull of each voxel for each image [11]. If enough of these chosen pixels were inside the silhouette in all of the cameras, the voxel would be part of the foreground object.

A third method of computing voxels was presented by Szeliski and is similar to the previous one in that it starts with a predefined volume of interest with which to begin shaping. It involves an octree representation in which the bounding box is recursively subdivided into smaller voxels until a voxel is fully occupied or empty [7]. This creates a voxel resolution that matches the object resolution exactly [8].

Many of the techniques used for acquiring voxels use voxels of equal size because this assumption has computational benefits. For instance, having the voxel position and size predetermined allows for look-up tables that contain depth and projection information to be built off-line. This speeds up the on-line processing time and is useful for real-time applications.

5.2 Voxelization Method

To be able to test the voxel acquisition process, it was necessary to use images from the Stanford BioMotion Lab along with the calibration data used for those cameras. Silhouette, foreground, and background images from multiple cameras are shown in figure 4 in Appendix A.

The method used by this system involves a fixed number and size of voxels. The algorithm finds the 2D and 3D center of mass of the object in order to create a 2m x 2m x 2m bounding box around the object at any given time. Once the region of interest is determined, a grid of voxels that are each 10mm x 10mm x 10mm in size is produced. The center point of each voxel is then perspectively projected to the silhouette images using the calibration data from all 8 cameras. The pixel on the image that is the projection of the center of the voxel is then tested to see if it lies within the silhouette. This is done for each voxel for all 8 camera images and a running sum of the number of cameras that a voxel agrees with is kept. This number is then checked to see if it meets a threshold, which in our case is seven (8 cameras must agree). The voxels that meet this threshold are considered true voxels and therefore part of the actual object, and the ones below the threshold are not included in the final reconstruction. The final reconstruction from multiple angles is illustrated in figure 3.

The threshold for the voxel computations depends greatly on the background subtraction algorithm that is used. If the background subtraction algorithm leaves holes in the body of the human, then the threshold for computing the voxels must be lowered or holes will show up in the 3D reconstruction. Our background subtraction strives to minimize the holes at the expense of blobs. This is why we choose a very strict threshold in the voxelization process. We dilate our final images in the background subtraction process, which lessens holes yet creates a less accurate figure. The voxel acquisition process then cuts down the thick figure to a more accurate shape in the 3D reconstruction.

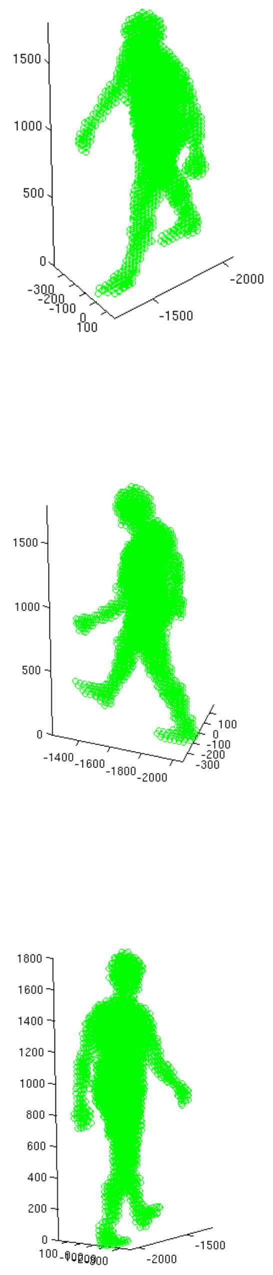


Figure 3: Voxel Reconstructions

6 Results

The background subtraction algorithm in MATLAB ran at .78 frames/s for 2 cameras. The partial voxel acquisition took about .71 frames/s for 2 cameras. The total frame rate for the entire system in MATLAB was about .36 frames/s.

The background subtraction algorithm ran at close to 2 frames/s for 2 cameras when implemented in C. The voxel representation was not able to be fully implemented in C due to time constraints, so it is not certain what speed the entire system ran at using C.

7 Discussion

Our algorithm can deal with indoor, stationary environments that are cluttered. It effectively removes noise that is brought upon by the image acquisition process. It also is efficient in removing shadows caused by diffused lighting, such as the lighting that would be found in labs that have multiple light sources. The algorithm has difficulties dealing with sharp shadows caused by a single light source.

The voxel acquisition process was accurate but not perfect. Problems still arose due to the dilation of the images in the background subtraction phase of the system. This is acceptable, though, since we were trying to build a close to real time system and therefore sacrificed some accuracy for speed.

The original intention of the system was to create an initial rough voxel model to use as feedback to the system. The feedback would be used to adjust thresholds in the background subtraction process and therefore make sharper silhouettes for more accurate voxel representations. Obviously, this would be very difficult to do in real time, but it could certainly be an effective method at obtaining very accurate voxel reconstructions.

Unfortunately, the complete system was not able to be completed using the C language. Problems arose in using the OpenCV library functions when trying to convert the voxel algorithm from MATLAB to C. The background subtraction algorithm was also not as fast as expected. The clean images it produces, however, are a good sign that it may be possible in the future to reduce some computations in the algorithm and still obtain accurate silhouettes. Better experience with the OpenCV library will also go a long way in making the algorithm faster.

8 Conclusion

Overall, there were a few complications that arose in debugging and creating the system. Only being able to test a system in theory was a drawback, since it isn't possible to be sure that everything in a system will work together as planned until you test it. It was also difficult to complete the entire system in the allotted time for the MERIT program. Speeding up of the algorithms was not able to occur in time, so the results are based on programs that are not as fast as they could be.

From the accurate images produced by the algorithms it seems like the approach that is taken is a good start to building a better and faster system. What is needed is a refinement of the system so that all of its parts work together better so that it runs faster. A better understanding of the OpenCV library and coding in C would help this greatly, and lead to a more efficient system.

9 References

[1] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. "Pfinder: Real Time Tracking of the Human Body." IEEE Trans. PAMI, 19(7):780-785, July 1997

[2] Mikic,I. 2002. "Human body model acquisition and tracking using multi-camera voxel data," Ph.D. Dissertation, University of California, San Diego.

[3] I. Haritaoglu, D. Harwood and L. Davis, "W4: Who? When? Where? What? A Real time system for detecting and tracking people." In Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998.

[4] M. Harville. "A Framework for High-level Feedback to Adaptive, Per-pixel, Mixture-of-gaussian Background Models." In ECCV, page III: 543 ff., Copenhagen, Denmark, May 2002.

[5] A. Elgammal, D. Harwood, and L. Davis. "Non-parametric Model for Background Subtraction." In ECCV, pages II:751-767, Dublin, Ireland, May 2000.

[6] W.E.L. Grimson,C. Stauffer, "Adaptive Background Models for Real-Time Tracking," in CVPR, 1999.

[7] R. Szeliski. "Real-time octree generation from rotating objects." Technical Report CRL 90/12 Digital Equipment Corporation, Cambridge Research Lab, Dec 1990.

[8] A.O. Balan. "Voxel Carving and Coloring – Constructing a 3D Model of an Object from 2D Images." Brown University, Providence RI, December 2003.

[9] I. Mikic, M. Trivedi, E. Hunter, P. Cosman. "Human Body Model Acquisition and Tracking Using Voxel Data." International Journal of Computer Vision 53(3), 199-223, 2003.

[10] A.M. McIvor. "Background Subtraction Techniques." Remuera, Aukland, New Zealand.

[11] G.K.M. Cheung, T. Kanade, M. Holler, J. Bouguet. "A Real Time System for Robust 3D Voxel Reconstruction of Human Motions." CVPR, vol. 02, no. 2, p. 2714, 2000.

[12] K. Toyama, J. Krumm, B. Brumitt, B. Meyers. "Wallflower: Principles and Practice of Background Maintenance." International Conference on Computer Vision, September 1999, Corfu,Greece.

[13] A. Mittal, N. Paragios. "Motion Based Background Subtraction using

Adaptive Kernel Density Estimation." CVPR 2004, vol. 2, II-302 – II-309.

[14] W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, L. McMillan. "Image-Based Visual Hulls." International Conference on Computer Graphics and Interactive Techniques,p369-374, 2000.

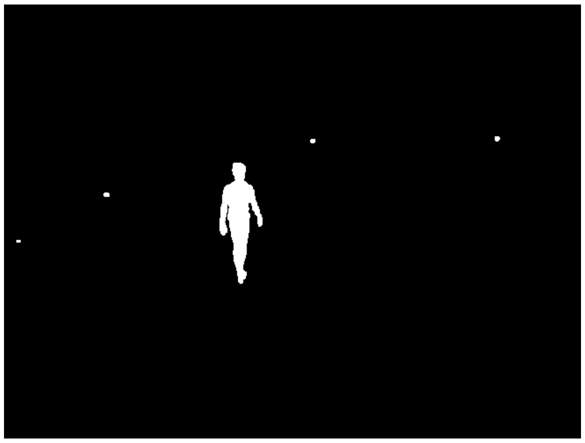
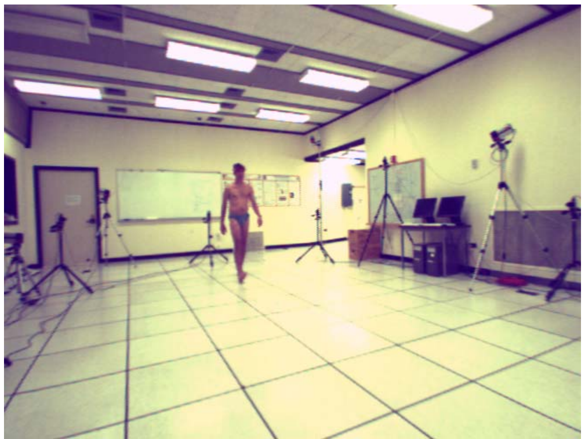


Figure 4a: Camera 1

Appendix A



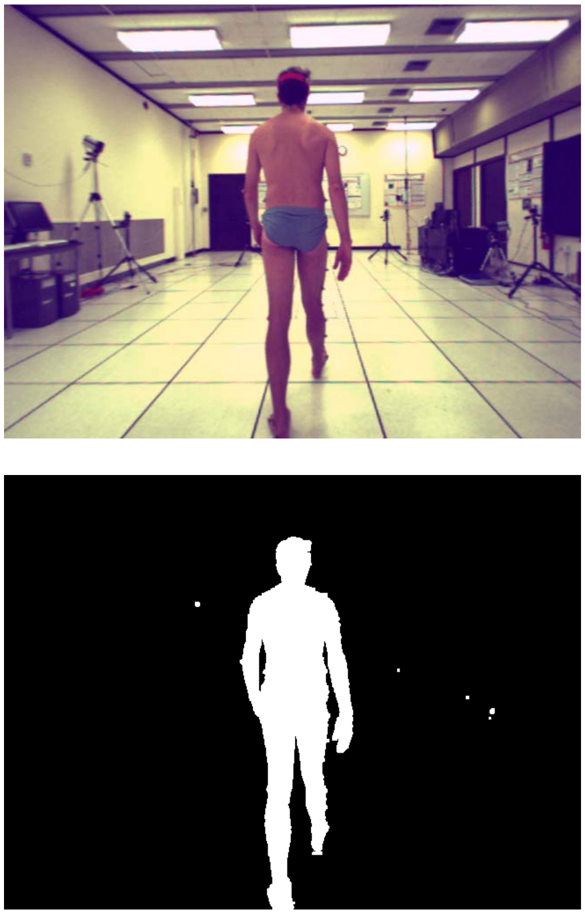


Figure 4b: Camera 4