Human Segmentation from Video in Indoor Environments Using Fused Color and Texture Features

Robert H. Luke, *Student Member*, *IEEE*, Derek Anderson, *Student Member*, *IEEE*, James M. Keller, *Fellow*, *IEEE*, and Marjorie Skubic, *Member, IEEE*

***Abstract*--A requirement for robust human activity analysis from video in complex and dynamic environments involves the reliable segmentation of individuals. This paper describes a system that has been built to segment moving people in video using the strengths of both color and texture features. In addition, a new algorithm was developed for the detection and removal of shadows from change detection images. To preserve the privacy of the individual, the output of this system is a binary map segmentation that distinguishes the individual's silhouette from his or her background.**

***Index Terms*--Human segmentation, color computer vision, video surveillance, eldercare.**

1. INTRODUCTION

HUMAN segmentation, with respect to a fixed camera location, is a classical image processing problem with countless applications to video surveillance. Not only must foreground objects be segmented, but a background model must be acquired and updated given potential changes in lighting or object manipulation. In complex and dynamic environments, there is an increased need for incorporating multiple features that perceive the problem in unique ways. The system described in this paper utilizes multiple feature sets including texture, color and shadow information to improve the reliability of segmentation in real indoor environments.

Silhouette extraction, namely, segmenting the human body from the background with the camera at a fixed location, is the initial stage in activity analysis. Before foreground silhouette extraction can occur, an accurate background model must be acquired. The background is defined as any non-human, static object. After the background model is initialized, regions in

Manuscript received November 12, 2007. R. Luke and D. Anderson are pre-doctoral biomedical informatics research fellows funded by the National Library of Medicine (T15 LM07089). This work is also supported by the National Science Foundation (ITR award llS-0428420) and the Administration on Aging (90AM3013).

The authors are with the Electrical and Computer Engineering Department at the University of Missouri-Columbia, Columbia, MO 65201. E-mail: {rhl3db, dtaxtd}@mizzou.edu, {kellerj, skubicm}@missouri.edu).

subsequent images with significantly different characteristics from the background are considered as foreground objects. As well, areas classified as background are used to update the background model.

There have been numerous algorithms proposed for background representation update and foreground segmentation. A short list includes Mean and Covariance, Least Median Squared, Mixtures of Gaussians [15], and Eigen Backgrounds [13]. Each of these techniques models the background and makes decisions only on pixel level information such as color or intensity. The Wallflower [17] algorithm is a notable exception using a Wiener prediction filter for pixel level decisions, logical filling at the region level, and multiple background models at the image level.

The major contribution of this paper is the use of both color and texture information to build a robust background model and determine change. Texture information is extracted into histograms of gradients from an extended YCbCr color space. As well, color histograms are built for each pixel using a modified Hue, Saturation, and Value (HSV) color space. Shadows cast by people moving through the scene modify the color information of background pixels and are incorrectly classified as foreground pixels; therefore, a separate algorithm to detect shadowed areas will be described.

Section 2 outlines the silhouette segmentation procedure. This is followed by the description of the texture-based features in section 3 and the color-based features with shadow removal in section 4. Next, section 5 presents the rules for change detection. Section 6 describes the update of the background model. Segmentation results for a variety of sequences are shown in section 7. Section 8 gives the conclusions of the paper and work that needs to be accomplished in the future.

2. System Overview

The context for our video sensor network is an eldercare facility. This facility is an assisted living community for elderly residents. The sensor network in each living space consists of binary motion detectors using infrared, bed sensors, stove temperature sensors, and web cameras for video processing. If a resident falls and is injured, the system needs to recognize the event and take appropriate action. As well, longer term activities such as watching television, cooking, and bathroom visits can be tracked to build daily patterns. Changes in these daily patterns could be a symptom of deeper illness in the resident. Therefore, the goal of the system is to both classify aberrant behavior and predict future impairments of the individual. The consistency and reliability of silhouette segmentation is critical to the well being of a resident. If segmentation is not consistently operating properly or if there are too many false alarms generated, accidents could be missed or in the extreme, caregivers and residents might become irritated and disregard the system.

Silhouette segmentation is a classification task. The classifier determines if a location in an image belongs to a known background or if it belongs to something introduced to the scene. Using a stationary camera, a model of the background is built using the first T frames of the sequence. With each new frame, the background model is updated using the pixel values segmented as background. The background model is shown in Fig. 1.

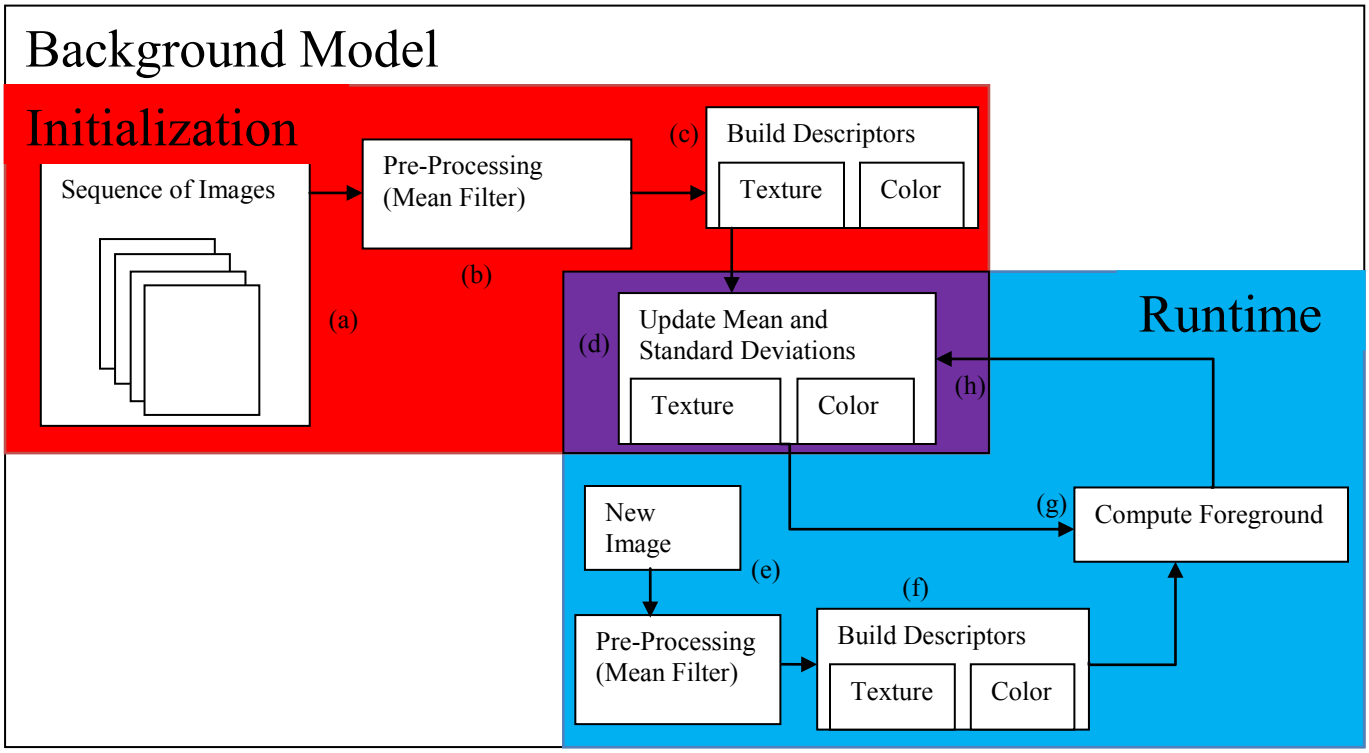


Fig. 1: A graphical representation of the background model. (a) A sequence of ten images is input to the system. (b) A 3x3 mean filter is passed over the red, green and blue color planes of each input image. (c) Color and texture descriptors are then extracted from these images. (d) Finally, the means and standard deviations of the descriptors at each pixel are found over the input sequence. (e) During runtime, a 3x3 mean filter is passed over each new image. (f) Color and texture descriptors are then extracted from the image. (g) The foreground is then found for the new image using the background model. (h) The background model is then updated.

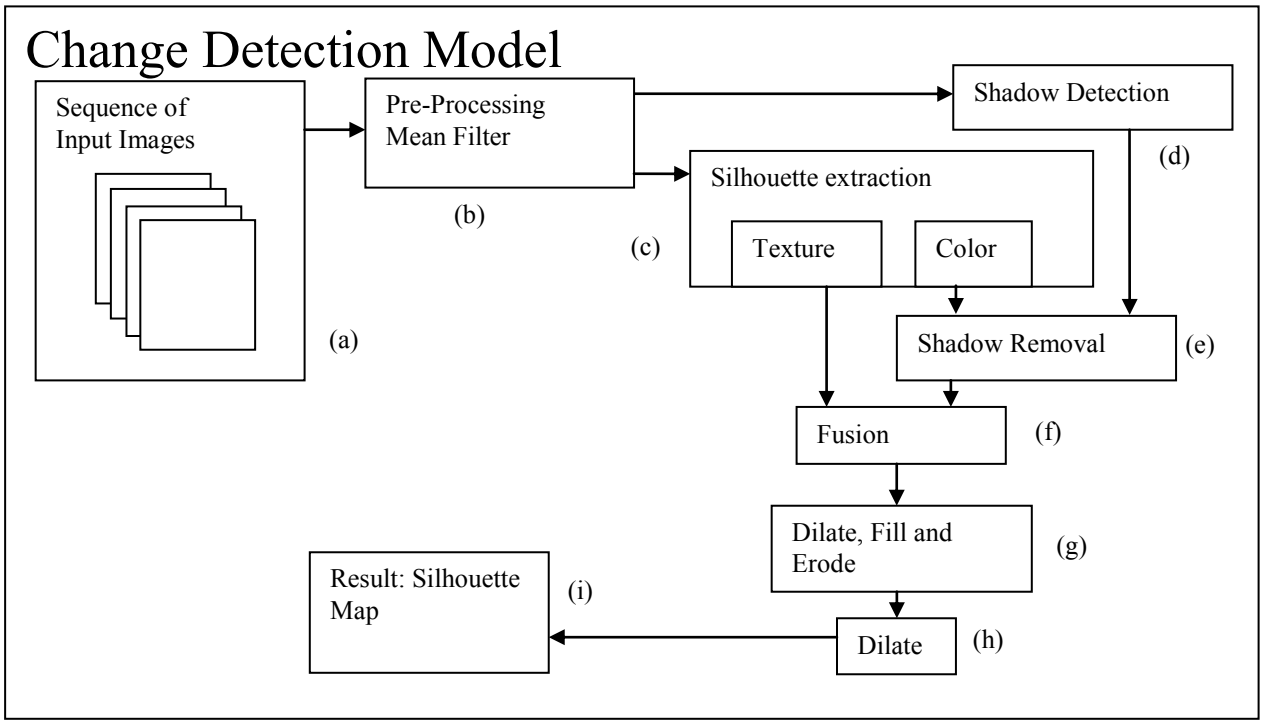


Fig. 2: Silhouette extraction procedure. (a) Images are captured from a video camera in the sensor network. (b) A 3x3 mean filter is passed over the red, green and blue color planes of each input image. (c) Silhouettes are found in color and texture features. (d) Shadow regions are identified. (e) Shadows are removed from the output of silhouettes found from color features. (f) The results of the texture and color silhouettes are fused. (g) Morphological and logical operators are applied to the output of silhouettes to remove false alarms and fill-in silhouettes. (h) The fused data is then dilated to correspond to the size of the subjects in the image. (i) The final silhouette output image.

An overview of the change detection model is shown in Fig. 2. After the background model is built, change detection can be performed for silhouette segmentation. Each input image is first smoothed with a 3x3 mean filter. Pixels that differ significantly from the background model are detected using color and texture features. The change detection algorithm is susceptible to shadows in the color features. Therefore, shadows are detected in parallel to the detection of change. Any change due to shadow is removed from the color silhouette. The changes acquired using texture and color features are then fused, creating a binary image representing change. Morphological and logical operations are then performed on the change detection images to remove noise and to form solid, compact silhouettes. A final morphological dilation is executed so that the detected silhouette is approximately the same size as the true silhouette of the person the scene.

3. TEXTURE FEATURES

Texture is a reliable source of information for scene description and change detection. Previous research in texture features includes [1, 8, 9, 11, 16]. A smaller amount of research has been focused on texture in color spaces [4, 5, 10, 14]. We believe color texture features are more robust than their monochromatic counterparts and are necessary for the modeling of a complex background.

Numerous factors go into the choice of color space. Because lighting is an uncontrolled variable in the scene, the color space chosen to extract textural information must be robust to lighting changes. The red, green and blue gamma corrected channels output from the camera make up the $R'G'B'$ color space. Though $R'G'B'$ color space is easily produced, features in this space are susceptible to changes in brightness. The HSV color space; hue, saturation,

value; has advantages over $R'G'B'$ space. Most importantly, it separates chroma, (color), and luma, (brightness), components. However, there are still some shortcomings. First, the luma component is unreliable for texture because changes in lighting or hard shadows cause drastic changes in the signals received. Also, when luma is low, the hue is unreliable. In addition, though the chroma and luma are separated, experimentation shows that the saturation is correlated with light intensity.

The YC_bC_r color space was selected for this system due to its reliability in a wide range of lighting situations. This color space again has a luma component; defined using the constants K_b , K_r , and K_g which are based on perceived brightness of human vision to blue, red and green; but it separates the blue and red chroma components, C_b and C_r . We have extended the YC_bC_r space to also include a green component. This C_g component is extracted in a fashion similar to the C_b and C_r components. These three chroma quantities describe how much the red, green and blue components contribute to luma. After experimentation, it was determined that better results were achieved after normalization by dividing the C_b , C_r and C_g components by the luma Y . The new $C_b^{'}$, $C_r^{'}$ and $C_g^{'}$ components used to extract feature descriptors are defined as

$$R^{'}, G^{'}, B^{'} \in [0,1]$$

$$K_b = .114$$

$$K_r = .299$$

$$K_g = 1 - (K_b + K_r) = .587$$

$$Y = K_r R' + K_g G' + K_b B'$$

$$C_b^{'} = \frac{\left(\frac{B' - G' - \frac{K_r}{1 - K_b}(R' - G')}{2}\right) + .5}{Y + 1}$$

$$C_r' = \frac{\left(\frac{R' - G' - \frac{K_b}{1 - K_r}(B' - G')}{2}\right) + .5}{Y + 1}$$

$$C_g' = \frac{\left(\frac{\left(-\frac{K_r}{K_b + K_r}R'\right) + G' - \left(\frac{K_b}{K_b + K_r}B'\right)}{2}\right) + .5}{Y + 1}$$

The silhouette change detection features calculated using the $C_b' C_r' C_g'$ space are based on the Edelman descriptor, [7]. The practical use and robustness of the Edelman descriptor have been displayed in the Scale Invariant Feature Transform, (SIFT) [12]. First, gradients are computed for each pixel in the C_b', C_r' and C_g' components. For each pixel in each gradient image, an eight dimensional histogram, h, is built using a five by five window of gradients. The orientation of each pixel's gradient, g_o, determines into which bin, b_r, the gradient resides. The bin index b_r is the floor of the gradient orientation g_o divided by 45^o. The neighboring bin, b_n, is identified as the second nearest bin to the gradient orientation g_o. The gradient's magnitude, g_m, is then linearly interpolated between the nearest bin and the neighboring bin and added to both.

The linear interpolation value $\propto$ represents how close the gradient's direction is to the center of the nearest bin. The value $\propto$ is found by first computing the modulus of the gradient direction by 45^o. The absolute difference of this value with 22.5^o represents how many degrees this gradient is off from the center of the bin. Finally, this value is divided by 45^o to compute the linear interpolation.

Hence formally, for each color component,

$$g_o \in [0,360)$$

$$g_m \in [0, \sqrt{2}]$$

$$g_o = \begin{cases} mod\left(\tan^{-1}\frac{d_y}{d_x} + 2\pi, 2\pi\right) & if\ d_x > 0 \\ \tan^{-1}\frac{d_y}{d_x} + \pi & if\ d_x > 0 \\ \frac{\pi}{2} & if\ d_x = 0\ and\ d_y > 0 \\ \frac{3\pi}{2} & if\ d_x = 0\ and\ d_y < 0 \\ 0 & else \end{cases}$$

$$g_m = \sqrt{{d_x}^2 + {d_y}^2}.$$

For each gradient in a window for a given pixel, the associated histogram h is updated as

$$b_r \in [0,7]$$

$$b_n \in [0,7]$$

$$\propto \in [0, .5].$$

$$b_r = \left\lfloor \frac{g_o}{45} \right\rfloor$$

$$b_n = \begin{cases} (b_r + 1)\ mod\ 8 & if\ g_o mod\ 45\ \geq 22.5 \\ (b_r + 7)\ mod\ 8 & else \end{cases}$$

$$\propto = \frac{|g_o mod\ 45 - 22.5|}{45}$$

$$h(b_r) = h(b_r) + (1-\propto) g_m$$

$$h(b_n) = h(b_n) + \propto g_m$$

The gradient magnitude is the Euclidean distance of the change in the horizontal and vertical direction and therefore has a maximum value when the horizontal and vertical change are both 1, such that

$$g_{m\,max} = \sqrt{1^2 + 1^2} = \sqrt{2}$$

The results from the process outlined above are three images where each pixel is associated with an eight bin histogram descriptor. Each eight bin histogram represents the conglomeration of gradient magnitudes in eight directions for a window of texture. The strength of this

descriptor is its matching ability and robustness to pixel jitter. Pixel jittering phenomenon refers to how consistently a camera registers pixel values through time. This jitter is due to noise in the perceived intensity of red, green and blue elements in the CCD of the camera. The cameras used in this system are low cost web cameras that more accurately capture intensity than color information. Human vision is more sensitive to intensity changes than color changes, making these cameras suitable for communication over the web. Higher quality cameras with less pixel jitter could be used, but because this system is to be deployed in an assisted living community, the most economically viable option is to use cheaper web cameras.

4. COLOR HISTOGRAM FEATURES

Our experiments have shown that the use of both color and texture information is more robust than using either alone. This system uses the HSV color model to build a color histogram at each pixel. Using hue, H, and saturation, S, is preferred over the $R'G'B'$ color space as explained earlier, because this representation separates chroma from luma. The HSV model is defined as

$$\vee - Max$$

$$\wedge - Min$$

$$H \in [0,360),$$

$$S, V, R', G', B' \in [0,1],$$

$$B_{max} = R' \vee G' \vee B',$$

$$B_{min} = R' \wedge G' \wedge B',$$

$$H = \begin{cases} 0, & if\ B_{max} = B_{min} \\ 60\left(\frac{g-b}{b_{max}-b_{min}}\right), & if\ B_{max} = R'\ and\ G' \geq B' \\ 60\left(\frac{g-b}{b_{max}-b_{min}}\right) + 360, & if\ B_{max} = R'\ and\ G' < B' \\ 60\left(\frac{b-r}{b_{max}-b_{min}}\right) + 120, & if\ B_{max} = G' \\ 60\left(\frac{r-g}{b_{max}-b_{min}}\right) + 240, & if\ B_{max} = B' \end{cases},$$

$$S = \begin{cases} 0 & if\ B_{max} = 0 \\ 1 - \frac{B_{min}}{B_{max}} & else \end{cases},$$

$$V = B_{max}.$$

Similar to the texture descriptor, for each pixel in the image, a histogram is built using the local color information. The 360^o hue component is discretized into eight bins, similar to the gradient orientation in the texture feature, resulting in a feature vector of length eight at each pixel.

As mentioned earlier, if the light intensity is low, then hue is unreliable. In an extreme example, the $R'G'B'$ value of (.01,0,0) has a hue of 0^o and a saturation of 1, while the $R'G'B'$ value of (0,.01,0) has a hue of 120^o and a saturation of 1. So, though these colors are nearly identical to a human observer, they are represented as very different values in HSV color space. Due to saturation being a measure of color purity, both of these values make sense in HSV space, but cause great difficulties when trying to compute similarity. We therefore define a new term, "brightness saturation", S_v, as

$$S_v = S * V = \left(1 - \frac{B_{min}}{B_{max}}\right) B_{max} = B_{max} - B_{min}.$$

This brightness saturation value is linearly interpolated over the two nearest bins from the hue discretization, similar to the gradient magnitude in the texture feature.

5. CHANGE DETECTION

Before calculating the occurrence of change, a background model must be built and maintained. For this system, the background is modeled with a single Gaussian distribution, with standard deviation modeling the pixel jitter from each pixel's mean. The first T images,

(we use 10), of a sequence are used to initialize the background model. The texture and color feature vectors described above are built for each of these images. The means of the texture and color vectors; $\mu_{h_{C_b'}}$, $\mu_{h_{C_r'}}$, $\mu_{h_{C_g'}}$ and $\mu_{h_{HS_v}}$; represent the average background while the standard deviations; $\sigma_{h_{C_b'}}$, $\sigma_{h_{C_r'}}$, $\sigma_{h_{C_g'}}$ and $\sigma_{h_{HS_v}}$; represent pixel jitter.

To detect change at each pixel, the absolute values of the difference between the current frame and the mean vectors are calculated. It is assumed that any values of change less than two standard deviations from the mean are noise and are therefore ignored. Beyond two deviations, the new observation is assumed to be a significant change from the background. It is this value beyond the noise range that we want to keep for change detection. Therefore, two standard deviations are subtracted from the amount of change at each bin. Subtracting the standard deviation, instead of the more common operation of dividing the change value by the standard deviation, has the added benefit of not possibly causing a divide by zero error.

The differencing method is performed on the C_b', C_r', C_g' and HS_v histogram images to compute

$$C = 2$$

$$\Delta_{h_{C_b'}}(x,y,i) = 0 \vee \left(\left|\mu_{h_{C_b'}}(x,y,i) - h_{C_b'}(x,y,i)\right| - C * \sigma_{h_{C_b'}}(x,y,i)\right),$$
$$\Delta_{h_{C_r'}}(x,y,i) = 0 \vee \left(\left|\mu_{h_{C_r'}}(x,y,i) - h_{C_r'}(x,y,i)\right| - C * \sigma_{h_{C_r'}}(x,y,i)\right),$$
$$\Delta_{h_{C_g'}}(x,y,i) = 0 \vee \left(\left|\mu_{h_{C_g'}}(x,y,i) - h_{C_g'}(x,y,i)\right| - C * \sigma_{h_{C_g'}}(x,y,i)\right),$$
$$\Delta_{h_{HS_v}}(x,y,i) = 0 \vee \left(\left|\mu_{h_{HS_v}}(x,y,i) - h_{HS_v}(x,y,i)\right| - C * \sigma_{HS_v}(x,y,i)\right).$$

These differences are summed across the eight histogram bins to find the total change at each pixel of the image

$$\Delta'_{h_{C_b'}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{C_b'}}(x,y,i),$$
$$\Delta'_{h_{C_r'}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{C_r'}}(x,y,i),$$

$$\Delta'_{h_{C'_g}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{C'_g}}(x,y,i),$$
$$\Delta'_{h_{HS_v}}(x,y) = \sum_{i=0}^{7} \Delta_{h_{HS_v}}(x,y,i),$$

where (x,y) is the pixel location, i is the i[th] histogram bin, and h_{CS} is the histogram of color space CS.

At every pixel in each color space, the change is a single scalar and the resulting picture can be thought of as a difference image. The four difference images for a single frame are shown in Fig. 3.

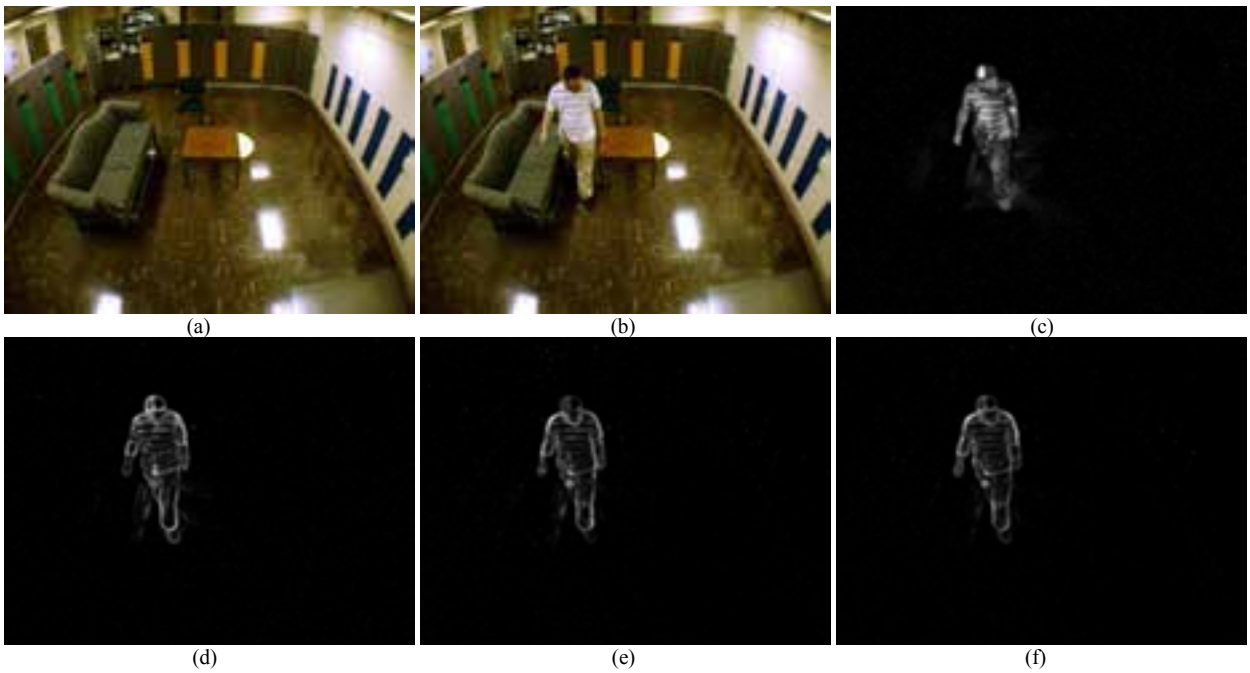


Fig. 3. Images representing the amount of change according to each texture and color descriptors. (a) Background image (b) Test image (c) HS_v features Diff Image (d) C'_b features Diff Image (e) C'_r features Diff Image (f) C'_g features Diff Image

The fusion of the changes of the C'_b, C'_r and C'_g components require special care because in this color space, changes to secondary colors; cyan, magenta and yellow; register in multiple color planes. Hence, their responses to change are smaller in each plane than for primary colors. A Yager union [18] is used to fuse the texture changes. The operator is defined as

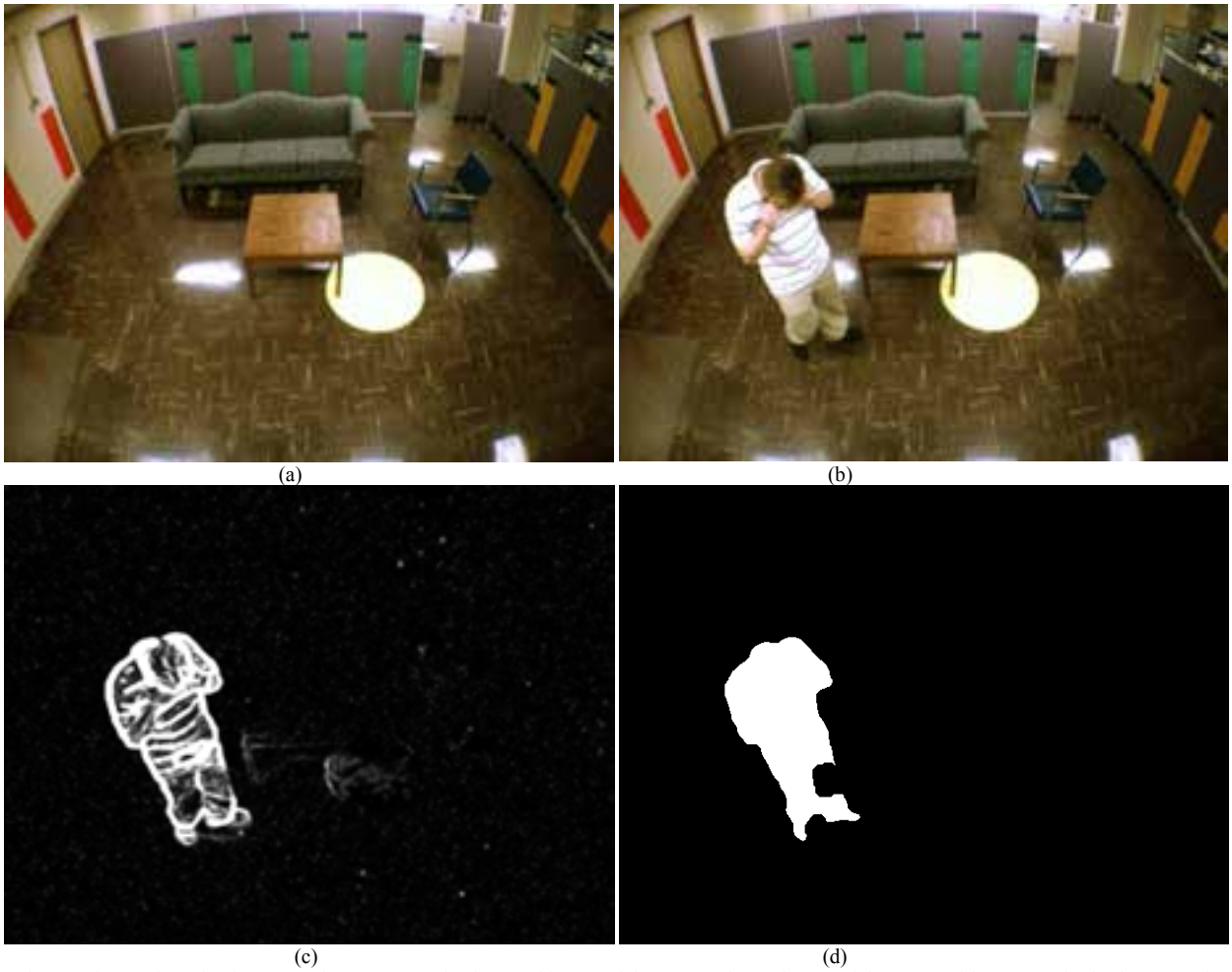
$$\Delta'_{h_{C'_{brg}}}(x,y) = 1 \wedge \left(\Delta'_{h_{C'_b}}(x,y)^w + \Delta'_{h_{C'_r}}(x,y)^w + \Delta'_{h_{C'_g}}(x,y)^w\right)^{\frac{1}{w}}.$$

The variable w is a tunable parameter, which can be manually assigned or learned from

training data, which set to 2 here. Values of $\Delta'_{h_{C'_{brg}}}(x,y)$ above a given threshold, 0.4, represent pixels that differ from the background.

$$\varphi_{h_{C'_{brg}}}(x,y) = \begin{cases} 1 & \Delta'_{h_{C'_{brg}}}(x,y) > 0.4 \\ 0 & else \end{cases}$$

The result of this fusion is a binary image, where the pixels along the contour of the person and areas of large textural change are classified as foreground. Multiple stages of change detection with respect to $C'_{brg}(x,y)$ are shown in Fig. 4.



(a) (b) (c) (d)

Fig. 4. Change detection in texture features. (a) A background image of the scene. (b) An image of the scene with a person in the foreground. (c) Change detection confidence using the texture descriptors. (d) The pixels of image (c) above the threshold of .4 with morphological and logical post processing operations.

Values of the color descriptor change, $\Delta'_{h_{HS_v}}$, above a given threshold, 2, represent pixels of

color change. These pixels are defined as

$$\varphi_{h_{HS_v}}(x,y) = \begin{cases} 1 & \Delta'_{h_{HS_v}}(x,y) > 2 \\ 0 & else \end{cases}$$

Fig. 5 shows the change detected from the HS_v features. Unfortunately, color change detection in HS_v is vulnerable to shadows cast by moving objects, as is shown in Fig. 5a. This problem with shadows is again due to the correlation between saturation and luma. It is therefore necessary to detect and remove shadows.

The shadow detection algorithm used in this paper is an extension to an earlier model proposed by Blauensteiner et al., [6]. In [6], circular statistics are built from hue and saturation mapped into a two dimensional space. If the luma, Y', drops by a reasonable amount from the mean luma, $\mu_{Y'}$, while the hue and saturation change very little, a shadow is detected. For this paper, an extension to this algorithm was made using our HS_vV space.

The first shadow condition is based on change in luma. When a background pixel goes into shadow, the luma of that pixel is expected to decrease. To be considered shadow, the luma of a pixel must drop below 95% of its average value.

The second condition is based on the chroma of the pixel. When a surface is shadowed, the colors of the pixels in that area are nearly unchanged. Therefore, to determine pixel color change, the mean location, μ_{HS_vV}, is determined for each pixel in HS_vV space and updated as part of the background model. As brightness decreases, the color of each pixel typically moves along an approximately linear path toward the origin in HS_vV space. Similarity is determined using the dot product of the angle between a pixel's current location $f_{HS_vV}(x,y)$ and its mean location $\mu_{HS_vV}(x,y)$,

$$d(x,y) = \frac{f_{HS_vV}(x,y) \cdot \mu_{HS_vV}(x,y)}{\left\|f_{HS_vV}(x,y)\right\| \left\|\mu_{HS_vV}(x,y)\right\|}$$

If $d(x,y)$ is greater than .99, the pixel's color is assumed to be unchanged. The threshold of .99 was found empirically to work well in most lighting conditions and coincides with an eight degree angle in HS_vV space.

As mentioned previously, when luma is low, color information becomes unreliable. Therefore, the final condition is that the luma of each pixel must be above .2 to be considered to be a shadow. This also handles a special case of black areas moving through the scene. Without this condition all new black areas would be discarded as shadow and never selected as foreground.

If the pixel's color is unchanged, its luma has decreased, and its luma is not too dark, the pixel is classified as being in shadow. The output of shadow detection is an image $L(x,y)$ defined as

$$L(x,y) = \begin{cases} 1 & d(x,y) > .99 \; and \; Y(x,y) < .95\mu_Y(x,y) \; and \; Y(x,y) > .2 \\ 0 & else \end{cases}$$

The color change detection image is

$$\varphi'_{h_{HS_v}}(x,y) = \varphi_{h_{HS_v}}(x,y) \; \wedge \big(1 - L(x,y)\big).$$

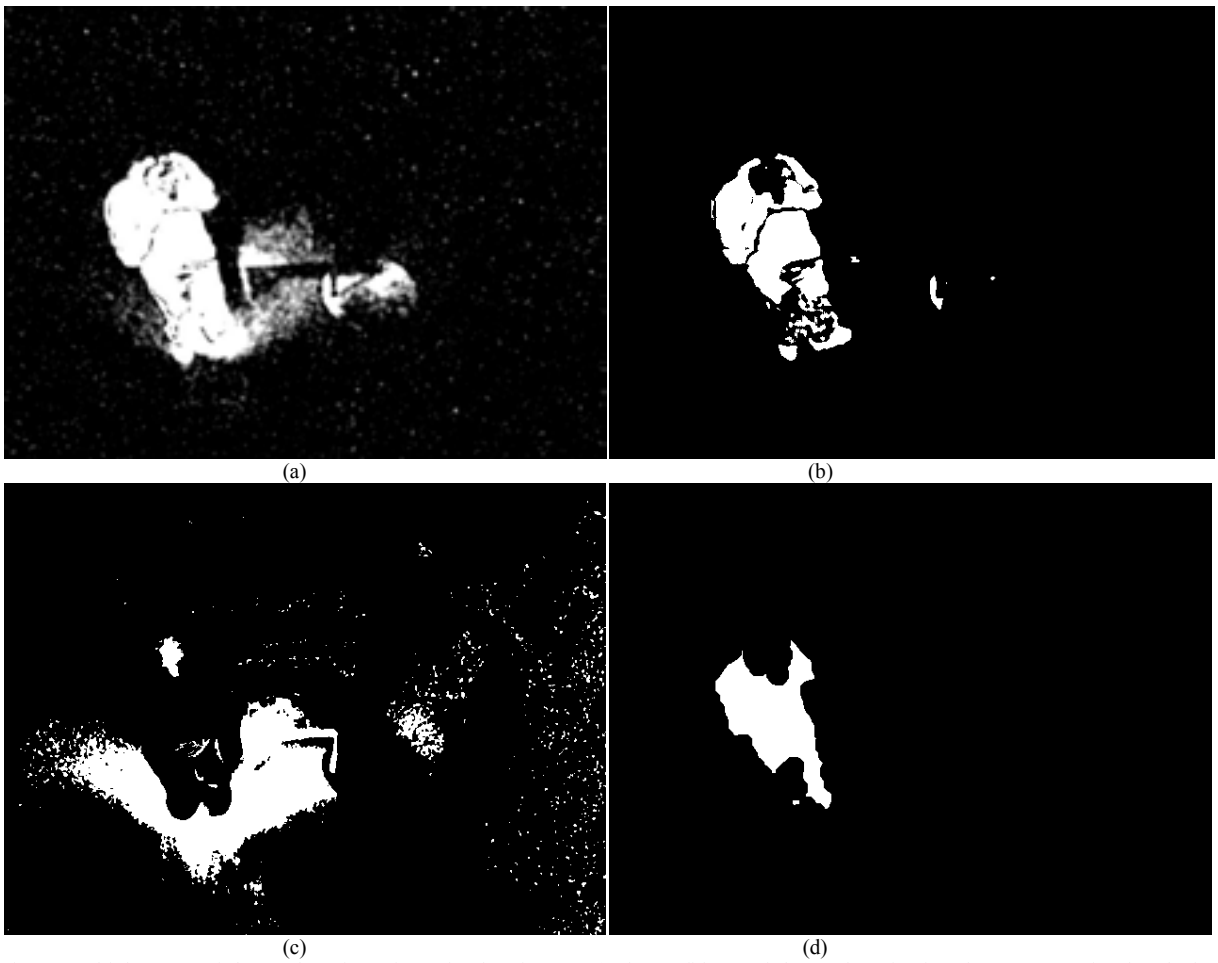
The final step is the fusion of texture and color difference which is performed using a union operator. This image, $F(x,y)$, represents the change detected from both texture and color.

$$F(x,y) = \left(\varphi_{C'_{brg}}(x,y) \vee \varphi_{HS_v}(x,y)\right)$$

Because the contour of the person often has segments missing from change detection, the fused image $F(x,y)$ is morphologically dilated by a circular kernel of radius 3, k_3. Regions of pixels with value zero surrounded by pixels of value one are then filled with ones. The image is then morphologically eroded with a circular kernel of radius 6, k_6, to eliminate noise points. One final morphological dilation with the k_3 kernel is performed to return the silhouettes to their proper size. The operation is defined as

$$O(x,y) = \Big(\big(fill(F(x,y) \oplus k_3)\big) \ominus k_6\Big) \oplus k_3.$$

The shadow removal process is shown visually in Fig. 5.



(a) (b) (c) (d)

Fig. 5. Multiple stages of change detection using color descriptors. (a) The confidence of change detection in color space. Notice that shadows are detected as change. (b) Pixels that have registered a change above the threshold of 2. (c) Shadows in the scene. (d) The change with respect to color descriptors after removing shadows and performing morphological operations. Although the output does not match the silhouette closely, it fills in parts of the silhouette missed by the texture descriptor.

6. Background Update

The background, even in a constrained indoor environment, is not constant. Changes in lighting or manipulation of objects in the scene must be taken into account for a robust system. As mentioned in the introduction, well known algorithms such as Mixtures of Gaussians and Wallflower have been developed to handle background adaptation. Because our eldercare

tracking system is a conglomeration of many smaller systems, algorithms with greater complexity are too computationally expensive to run in real time. It was therefore decided to update just a single mean and standard deviation for each feature dimension at each background pixel.

It is assumed that regions of change correspond to moved objects, or a person. Because the living quarters of the eldercare environment house only a single person, it is assumed that there will be at most only one person in the scene at any given time. Furthermore, our supposition is that the person is larger than any object moved in the scene. Therefore, the largest foreground region is recognized as the person. That area is then dilated by six pixels and is not used in the background update. All other pixels are used to update the background model.

An alpha update similar to that used in [15] is used to update the background model. It is too expensive to store and recompute the 32 means and standard deviations otherwise. The mean values are updated using a linear interpolation of the old value and new value.

$$\mu_{h_{C_b'}}(x,y,i) = (1-\alpha)\mu_{h_{C_b'}}(x,y,i) + \propto h_{C_b'}(x,y,i)$$

$$\mu_{h_{C_r'}}(x,y,i) = (1-\alpha)\mu_{h_{C_r'}}(x,y,i) + \propto h_{C_r'}(x,y,i)$$

$$\mu_{h_{C_g'}}(x,y,i) = (1-\alpha)\mu_{h_{C_g'}}(x,y,i) + \propto h_{C_g'}(x,y,i)$$

$$\mu_{h_{HS_v}}(x,y,i) = (1-\alpha)\mu_{h_{HS_v}}(x,y,i) + \propto h_{HS_v}(x,y,i)$$

Standard deviations are updated in a similar fashion using the absolute difference between the current value and the mean at each dimension.

$$\sigma_{h_{C_b'}}(x,y,i) = (1-\alpha)\sigma_{h_{C_b'}}(x,y,i) + \propto \left| h_{C_b'}(x,y,i) - \mu_{h_{C_b'}}(x,y,i) \right|$$

$$\sigma_{h_{C_r'}}(x,y,i) = (1-\alpha)\sigma_{h_{C_r'}}(x,y,i) + \propto \left| h_{C_r'}(x,y,i) - \mu_{h_{C_r'}}(x,y,i) \right|$$

$$\sigma_{h_{C_g'}}(x,y,i) = (1-\alpha)\sigma_{h_{C_g'}}(x,y,i) + \propto \left| h_{C_g'}(x,y,i) - \mu_{h_{C_g'}}(x,y,i) \right|$$

$$\sigma_{h_{HS_v}}(x,y,i) = (1-\alpha)\sigma_{h_{HS_v}}(x,y,i) + \propto \left| h_{HS_v}(x,y,i) - \mu_{h_{HS_v}}(x,y,i) \right|$$

Alpha determines the rate at which the system updates the background model. This parameter is associated with the frame rate of the camera and a user desired update rate for the system. We use an alpha of .01, for a system that captures images at a rate of 5 frames per second.

7. RESULTS

In order to calculate the accuracy of the proposed foreground detection system, the extracted silhouettes are compared to three hand-segmented ground truth sequences. The backgrounds in these sequences include a range of colors and intensities that remain static throughout each sequence. The clothing worn by the subjects demonstrate the need to use both texture and color features for change detection. The beginning of each sequence contains only the static background with no humans. The subject then walks to a specified location in the room and performs a random action. The subject then begins walking again.

Test sequence one consists of 148 images of a primarily white and yellow colored background. The subject is wearing a solid blue shirt and yellow-green shorts. Fig. 6 illustrates three frames from this sequence with the original image, the hand segmented silhouettes and the extracted silhouettes. In this sequence, the system correctly classified 99% of the hand segmented foreground. Also, only 2% of the pixels classified as foreground by the system were considered background by hand segmentation.

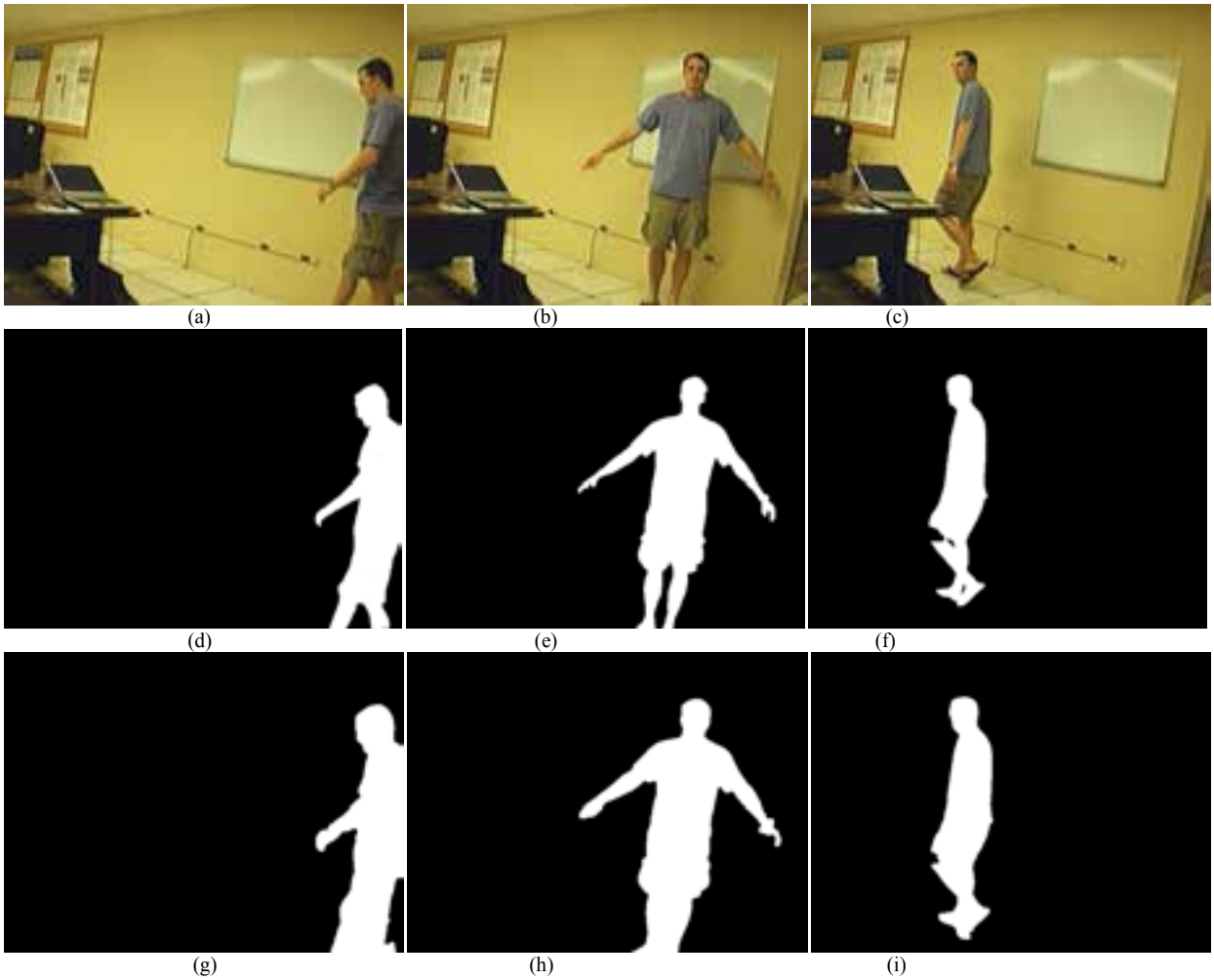


Fig. 6. A test sequence of images. (a), (b) and (c) are three unprocessed images from the test sequence. (d), (e) and (f) are the hand-segmented silhouettes of the person. (g), (h) and (i) are the three silhouette images output from this system.

The second test sequence has the same background as the first, but with a different subject. This sequence contains 192 frames. The subject in this sequence wears a striped shirt and blue jeans. Fig. 7 again shows three frames of the sequence. The system correctly classifies 98% of the hand segmented foreground, while only incorrectly classifying 1% of the background.

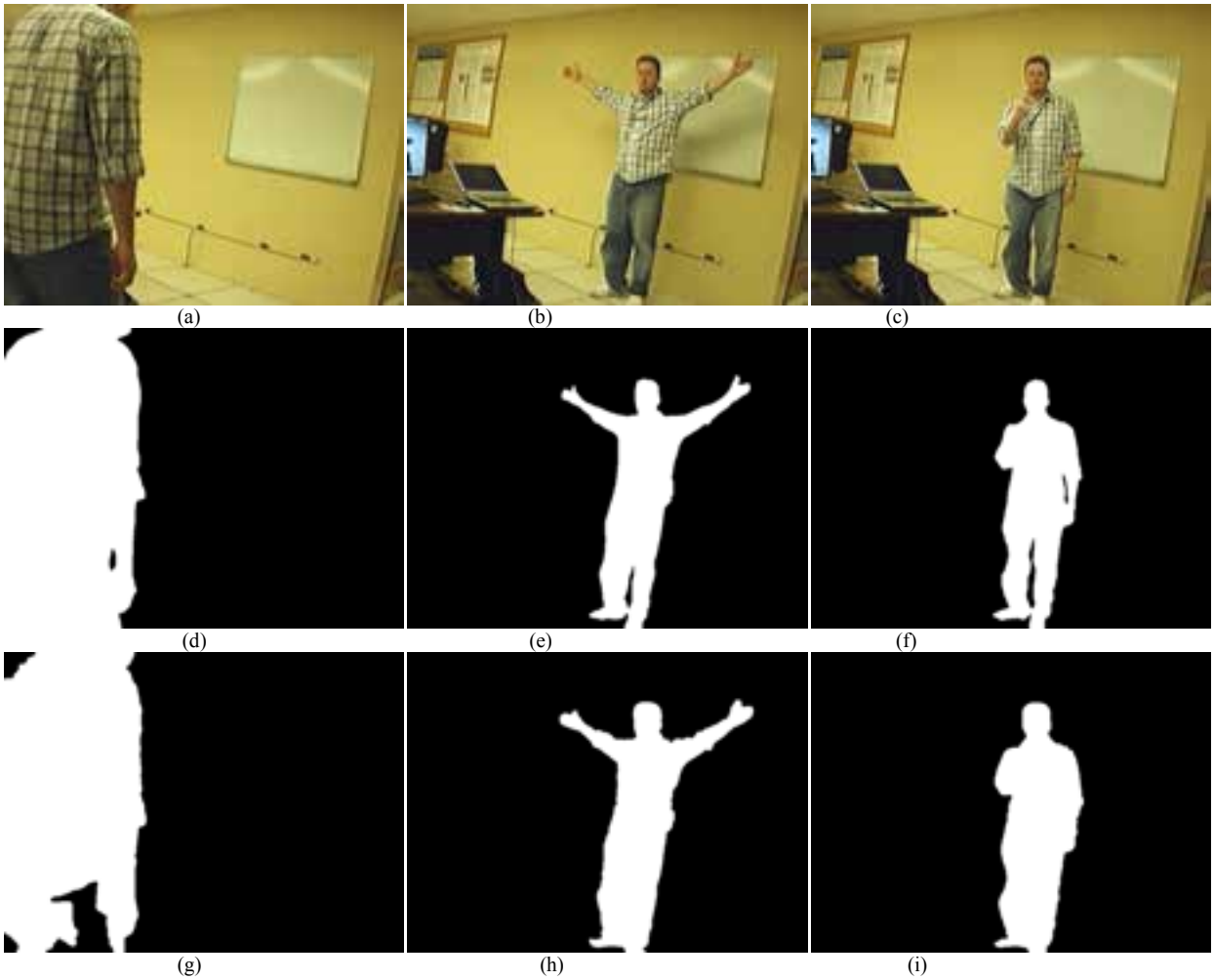


Fig. 7. A test sequence of images. (a), (b) and (c) are three unprocessed images from the test sequence. (d), (e) and (f) are the hand-segmented silhouettes of the person. (g), (h) and (i) are the three silhouette images output from this system.

The final sequence has the same flat colored wall, but hard edges on the floor. The subject wears a white shirt and blue jeans. This 70 frame sequence stresses the system's ability to segment the subject in a scene with little color information. Fig. 8 shows three representative frames from this sequence. Fig. 8 (h) shows the effect of reflection on silhouette segmentation as the subject is reflected off the floor. Because the reflection has color information, it is not considered shadow and is therefore part of the foreground segmentation. Even with this difficulty, 99% of the hand segmented foreground is found, while only 1% of the foreground classification is incorrect.

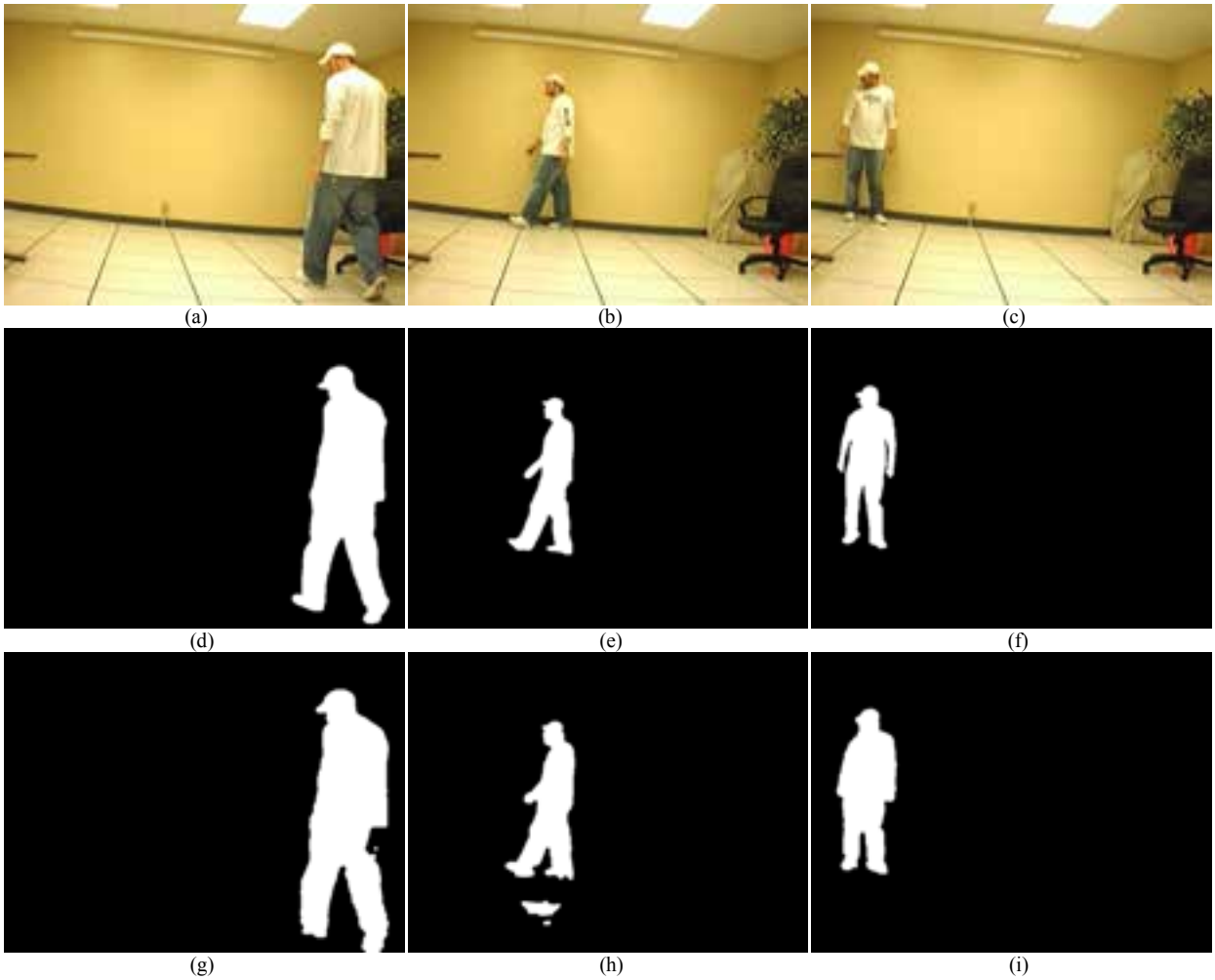


Fig. 8. A test sequence that stresses change detection with little color information. (a), (b) and (c) are three unprocessed images from the test sequence. (d), (e) and (f) are the hand-segmented silhouettes of the person. (g), (h) and (i) are the three silhouette images output from this system.

Over these three sequences, this system correctly classified 99% of the hand segmented foreground pixels. In addition, 98% of the areas classified as foreground by this system was segmented as foreground by a human. This level of accuracy makes the system suitable for many higher level intelligence processes such as human activity analysis.

The same three sequences were run through the Gaussian Mixture Model (GMM) defined in [15]. This system models the background as a misture of Gaussians at each pixel and classifies change, i.e. foreground, when a new pixel does not reside within a user specified number of standard deviations from one of the K Gaussians. A new Gaussian distribution is built for each

new pixel that is classified as foreground.

For testing, we used four models per pixel and three standard deviations as the model for the underlying pixel values. Pixel values outside of three standard deviations from all Gaussians are classified as change. The GMM was tested separately for R'G'B' and Cb'Cr'Cg' color spaces. The best results were found using the Cb'Cr'Cg' values. As displayed in table I, significantly higher accuracies were found using the system defined in this paper to the GMM system. Fig. 9 displays the results of the GMM and the system defined in this paper.

TABLE I
ACCURACY OF THE SYSTEM DESCRIBED IN THIS PAPER AND A GAUSSIAN MIXTURE MODEL.

	Current System		GMM System (Cb'Cr'Cg')		GMM System (R'G'B')	
	Percentage of Foreground Pixels Found	Percentage of Pixels Incorrectly Classified as Foreground	Percentage of Foreground Pixels Found	Percentage of Pixels Incorrectly Classified as Foreground	Percentage of Foreground Pixels Found	Percentage of Pixels Incorrectly Classified as Foreground
Sequence 1	99%	2%	77%	8%	76%	9%
Sequence 2	98%	1%	85%	3%	80%	5%
Sequence 3	99%	1%	63%	1%	70%	2%
Total	99%	2%	81%	5%	78%	6%

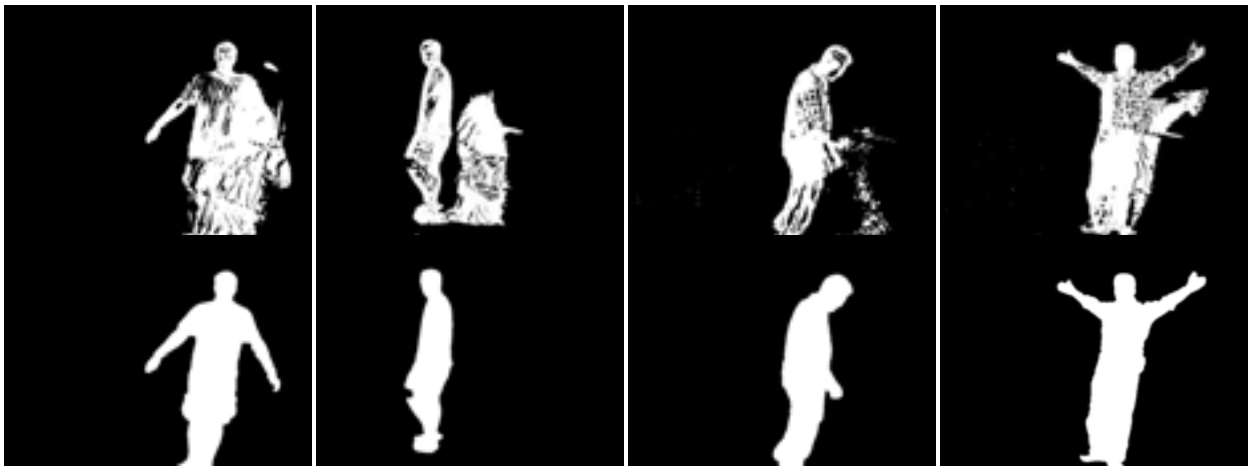


Fig. 9. Several images comparing the output of the Gaussian Mixture Model using Cb'Cr'Cg' color space, (top images), and those from the system defined in this paper, (bottom images).

Several more sequences have been collected and processed but not hand segmented. These sequences were collected in a scene that modeled a home setting. The environment includes a couch, two chairs, two tables, a rug and a lamp. There are also two blue mats on the floor for fall data collection. This furniture consists of flat colors, colored textured and monochromatic textures. Several images from one such sequence are shown in Fig. 10. The same thresholds and constants were used to process this sequence as were used on the hand-segmented data. These parameters could have been fine-tuned for this scene, but this sequence shows the robustness of this system even using a non-optimal set of thresholds. These images further illustrate the accuracy and reliability of the system across multiple background textures while removing shadows. Fig. 10 shows a scene using multiple cameras. In [2] and [3] we constructed a three-dimensional representation of these images to lower the false alarm rate and describe higher level activity.

A repository of image sequences, including the hand segmented sequences used in this paper, can be found at http://cirl.missouri.edu/sequences.

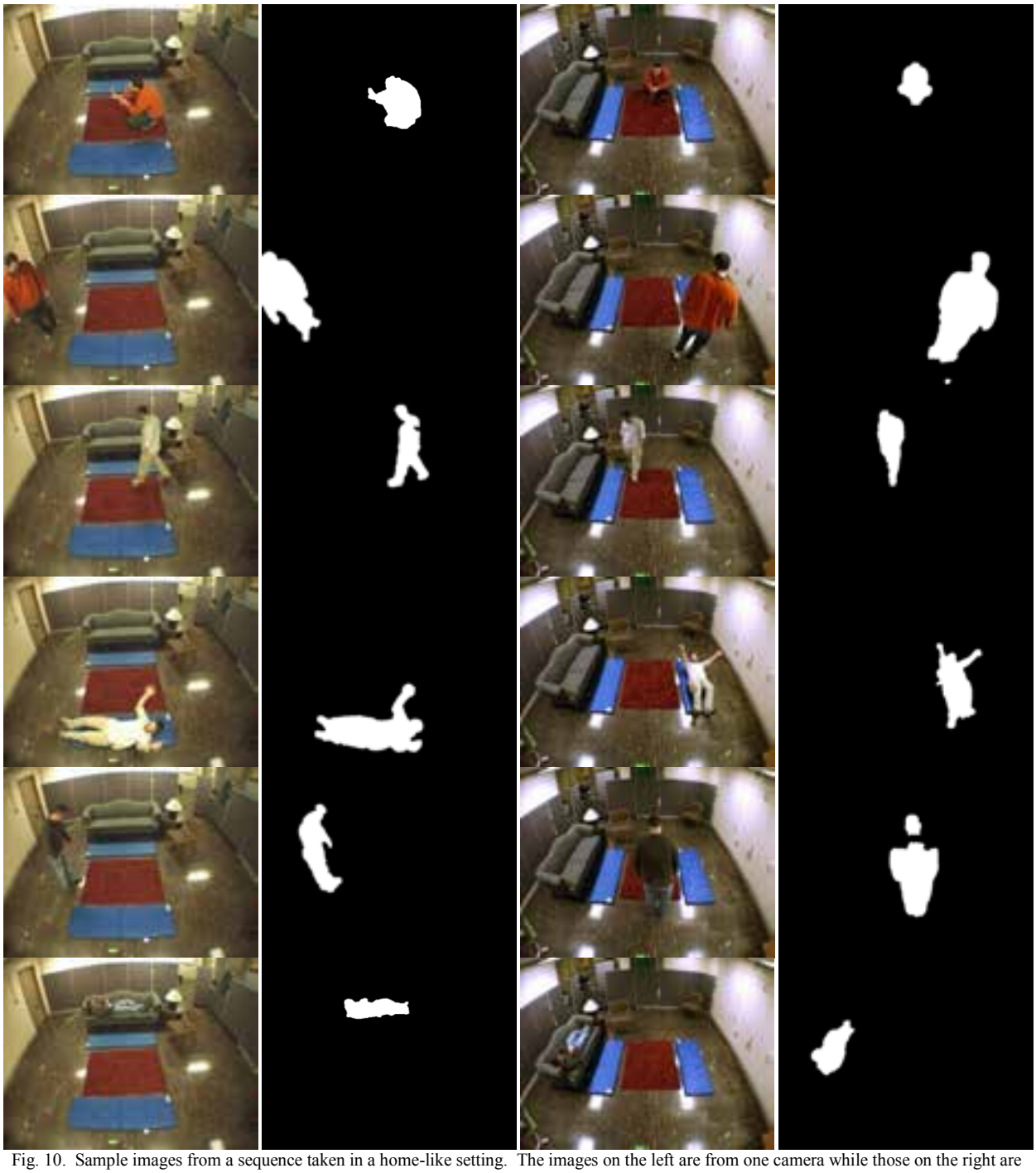


Fig. 10. Sample images from a sequence taken in a home-like setting. The images on the left are from one camera while those on the right are from another. By correlating the two points of view, many false alarms can be removed from three dimensional space.

There are some situations where the human goes undetected by this system. Fig. 11 produces a sample. These most frequently occur when the foreground has a highly similar color to the background. If there is no texture, and edges are undetectable then parts of the body are not

filled in. Also, foreground can be falsely classified as shadow when its intensity lower than that of the background color.

Fig. 11. Images displaying the shortcomings of this system. The human in the left images is wearing a shirt that has a darker, but similar color to the background, and is falsely detected as shadow.

8. CONCLUSION

Silhouette segmentation in complex and dynamic environments requires multiple computer vision algorithms running in parallel that process scene information in different ways. Fused color and texture features generate better silhouettes and avoid many conditions where segmented body regions would be otherwise disconnected. This results in improved features extracted from a better silhouette, and is being used for tracking humans for fall detection using higher level intelligence.

Exceptional correlation was found between this system's output and hand segmented ground truth image sequences. The classification of this system was significantly more accurate than the Gaussian Mixture Model. This paper also demonstrated results for longer sequences in a complex environment that modeled a home setting. Though hand segmentation was not performed on these sequences, output images were displayed at various parts of these sequences to show the qualitative accuracy of both the classification model and the background update model.

The procedure defined in this paper has shown exceptional accuracies in both classification

and background update for controlled indoor environments. For more complex changes to the background, extreme lighting changes, large object movement, and multiple objet tracking; higher level intelligent algorithms are needed to make more complex decisions. We plan on addressing these topics through object recognition and 3D modeling of the environment.

ACKNOWLEDGEMENT

The authors would like to thank Jun Liang for his help hand segmenting the sequences used for testing this system.

Robert Luke and Derek Anderson are pre-doctoral biomedical informatics research fellows funded by the National Library of Medicine (T15 LM07089). This work is also supported by the National Science Foundation (ITR award llS-0428420) and the Administration on Aging (90AM3013).

REFERENCES

[1] Amadasun, M., King, R.: Textural features corresponding to textural properties, IEEE Transactions on Systems, Man and Cybernetics. 19(5), 1264-1274, (1989)
[2] Anderson, D., Luke, R. H., Keller, J. K., Skubic, M.: Modeling Human Activity From Voxel Person Using Fuzzy Logic, Under Review by IEEE Transactions on Fuzzy Systems.
[3] Anderson, D., Luke, R. H., Keller, J. K., Skubic, M.: Linguistic Summarization of Activities from Video for Fall Detection Using Voxel Person and Fuzzy Logic, Under Review by ComputerVision and Image Understanding.
[4] Arivazhagan, S., Ganesan, L., Angayarkanni, V.: Color Texture Classification using Wavelet Transform, Sixth International Conference on Computational Intelligence and Multimedia Applications, 2005. 315- 320, (2005)
[5] Belongie, S., Carson, C., Greenspan, H., Malik, J.: Color- and texture-based image segmentation using EM and its application to content-based image retrieval, Proc. 6th International Conference on Computer Vision, 675-682 (1998)
[6] Blauensteiner P., Wildenauer H., Hanbury, A., Kampel, M.: Motion and Shadow Detection with an Improved Colour Model, International Conference on Signal and Image Processing, (2006)
[7] Edelman S., Intrator N., Poggio, T.: Complex Cells and Object Recognition, Unpublished Manuscript: http://kybele.psych.cornell.edu/~edelman/archive.html, (1997)
[8] Flicker, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D.,Yanker, P.: Query by image and video content: the qbic system, IEEE Computer, 28(9), 23-32, (1995)
[9] Haralick, R.: Statistical and Structural Approaches to Texture, Proceedings IEEE, 67(5), 786-804, (1979)
[10] Haindl, M., Grim, J., Somol, P., Pudil, P.,Kudo, M.: A Gaussian Mixture-Based Colour Texture Model. International Conference on Pattern Recognition, (3), 177-180, (2004)
[11] Iqbal, Q.,Aggarwal, J. K.: Cires: A system for content-based retrieval in digital image libraries, Proc. of International Conference on Control, Automation, Robotics and Vision, (2), 205-210, (2002)
[12] Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 60(2), 91-110, 2004.
[13] Oliver, N.Rosario, B.,Pentland, A.: A bayesian computer vision system for modeling human interactions. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(8), 831-843, (2000)
[14] Paschos, G.: Perceptually uniform color spaces for color texture analysis: An exeprimental evaluation. IEEE Trans. on Image Processing, 10(6), 932–937, (2001)
[15] Stauffer, C., Grimson, W.E.L.: Adaptive Background mixture Models for Real-time Tracking, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2), 243-252, (1999)

[16] Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception," IEEE Trans. on Systems, Man and Cybernetics, 8(6), 460-473, (1978)
[17] Toyama, K., Krumm, J., Brumitt, B., Meyers, B.:Wallflower: Principles and Practice of Background Maintenance. In Proceedings of ICCV'1999, 255-261 (1999)
[18] Yager, R. R.: On a General class of fuzzy connectives, Fuzzy Sets and Systems, 4(3), 235-242, (1980)