Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), Kauai, HI, December 2001.

Integrated Face and Gait Recognition From Multiple Views

Gregory Shakhnarovich, Lily Lee and Trevor Darrell

MIT Artificial Intelligence Laboratory
{gregory, llee, trevor}@ai.mit.edu

Abstract. We develop a view-normalization approach to multi-view face and gait recognition. An image-based visual hull (IBVH) is computed from a set of monocular views and used to render virtual views for tracking and recognition. We determine canonical viewpoints by examining the 3-D structure, appearance (texture), and motion of the moving person. For optimal face recognition, we place virtual cameras to capture frontal face appearance; for gait recognition we place virtual cameras to capture a side-view of the person. Multiple cameras can be rendered simultaneously, and camera position is dynamically updated as the person moves through the workspace. Image sequences from each canonical view are passed to an unmodified face or gait recognition algorithm. We show that our approach provides greater recognition accuracy than is obtained using the unnormalized input sequences, and that integrated face and gait recognition provides improved performance over either modality alone. Canonical view estimation, rendering, and recognition have been efficiently implemented and can run at near real-time speeds.

1 Introduction

Person tracking and recognition systems should ideally integrate information from multiple views, and work well even when people are far away. Two key issues that make this challenging are varying appearance due to changing pose, and the relatively low resolution of images taken at a distance. We have designed a system for real-time multi-modal recognition from multiple views that substantially overcomes these two problems.

To address the first issue we adopt a view-normalization approach and use an approximate shape model to render images for recognition at canonical poses. These images are sent to externally provided recognition modules which assume view-dependent input. For distant observations view-normalization must not presume accurate 3-D models are available; our system is designed for environments where relatively coarse-disparity stereo range images or segmented monocular views are provided. We have chosen to use shape models derived from silhouette information since they are practically computable in real time from these types of input data.

To overcome the second issue, we adopt a multi-modal recognition strategy. Low-resolution information makes it less likely that recognition using any single modality will be accurate enough for many desired applications. By combining cues together, we can obtain increased performance. A typical drawback of multi-modal approaches is that they presume different types of imagery as input. Face recognition usually works best with front-parallel images of the face, whereas gait recognition often requires side-view sequences of people walking. It can be difficult in practice to simultaneously acquire those views when the person is moving along a variable path. We propose a method for view-normalization which performs this automatically, generating appropriately placed virtual views for each modality.

We have implemented a system for integrated face and gait recognition using a shape model based on an image-based visual hulls. Our recognition algorithms were separately developed for view-dependent recognition. In our system a small number of static calibrated cameras observe a workspace and generate segmented views of a person; these are used to construct a 3-D visual hull model. Canonical virtual camera positions are estimated, and rendered images from those viewpoints are passed to the recognition methods.

In the following section we will review some of the previous work related to multi-view, pose-invariant face and gait recognition. We will consider different approximate shape models for virtual view rendering, and argue for the use of the image-based visual hull algorithm due to its appealing tradeoff of accuracy and computational efficiency. We will then present new methods for estimating canonical frames given visual hull representations, based on shape, appearance, and motion cues. Finally, we will show recognition results integrating face and gait cues with separately developed view-dependent recognition modules. The particular modules we have used for our current experiments are based on principle components analysis and spatio-temporal templates,

for face and gait respectively, but our framework is applicable to any view-dependent face or gait recognition method.

2 Previous work

To achieve pose-invariance, recognition models generally must incorporate information from multiple views of an object's pose. Broadly speaking, there are several classes of techniques for view-independent face recognition, including modular learning, elastic matching, view-interpolation, and geometric warping. Our visual hull approach is an instance of the last category, using multiple views and silhouette inputs.

Several authors have developed methods for recognition using a set of distinct view categories. The well-known eigenfaces paradigm was extended to recognize a set of different poses using an eigenspace for each view [17]. Rather than using replicated classifiers for distinct views, several authors have investigated elastic matching or view interpolation methods [21, 22]. Beymer and Poggio introduced a method for interpolating face views for recognition given dense correspondences, using a Radial Basis Function paradigm [2]. Seitz [19] developed a view morphing technique which used dense correspondences to interpolate rigid views of an object, but did not apply this technique to recognition.

Generalizing the notion of elastic matching, recognition based on principle components analysis of shape and texture distributions has been shown to be able to model and recognize a range of object poses[8]. When a model has been constructed fast optimization of shape and texture coefficients is possible. However, all these methods have generally presumed either knowledge of face pose and/or an accurate, dense depth or correspondence field during model training. This can be difficult to acquire in practice, so we have focused on geometric warping methods.

2.1 Geometric models

If we presume a model of the underlying geometry of the object, we can use that geometry to warp one view onto another view. For tracking faces, previous authors have used planar [3] and ellipsoidal [1] models to bring images into a canonical view. Several authors have used affine, cylindrical and ellipsoidal models for warping views during motion tracking [9, 6, 1].

Simple shape models are often inaccurate for view warping. More complex models may be used, such as warping with a depth map obtained from a laser range scanner. But as model detail increases, it

becomes difficult to precisely align a static model with dynamically changing observations. This negates the value of the detailed features. To overcome these problems, we would like to use a dynamic model of actual object shape, computed in real-time from the object being tracked. Dynamic models can be recovered from a variety of sources, but we will restrict ourselves to models recovered from a set of regular cameras.

We know the relative camera positions between the views, so if we accurately knew the depth at each pixel we could simply apply view morphing or traditional rigid motion warping. However, our source views are monocular and widely separated, so it is difficult to determine correspondences using traditional methods for multi-view matching.

With a rich statistical 3-D shape model of the object class, such as developed in [4], we could estimate a 3-D shape directly from the set of 2-D appearance images, and use that to render a high-quality image from the desired view. While this is an appealing idea, we would like our method to be general, and will not in practice assume such a statistical range model is available.

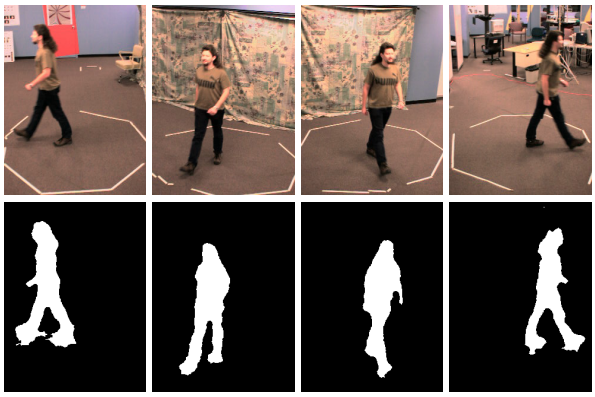
An equally appealing approach would be to apply voxel coloring or carving techniques [19, 11], to recover a discrete 3-D volumetric representation, and then use volume rendering techniques to generate the canonical view. However, these systems are computationally expensive, and require a specified discretization in 3-D which may not be optimal to re-render a given viewpoint.

We are interested in dynamic 3-D shape models that are computable without requiring dense correspondence or volumetric reconstruction. We will use a model which is computable solely from silhouette input, which we can obtain either from monocular analysis or segmentation of coarse-disparity range data.

2.2 Visual hulls

The concept of *visual hull* (VH) was introduced in [12]. A VH of an object is the maximal volume that creates all the possible silhouettes of the object. The VH is known to include the object, and to be included in the object's convex hull. In practice, the VH is usually computed with respect to a finite (often small) number of silhouettes.

An efficient technique consists of computing an *image-based VH* (IBVH) ([15]). For a desired viewpoint, for each pixel in the resulting image the intersection of the corresponding viewing ray and the VH is computed. The computation can be performed in 2D image planes, resulting in an algorithm that renders a desired view of n^2 pixels in



(a) Input

(b) Output

Fig. 1. (a) An example of rendering virtual views with an image-based visual hulls: the images obtained at the 4 cameras (top row) and their segmentation (bottom row). (b) The polyhedral VH model built from the input silhouettes in (a) (top pair), and synthetic views (bottom pair) rendered by a "virtual camera" corresponding to a frontal viewpoint. The view from the back has poor texture but reasonable shape.

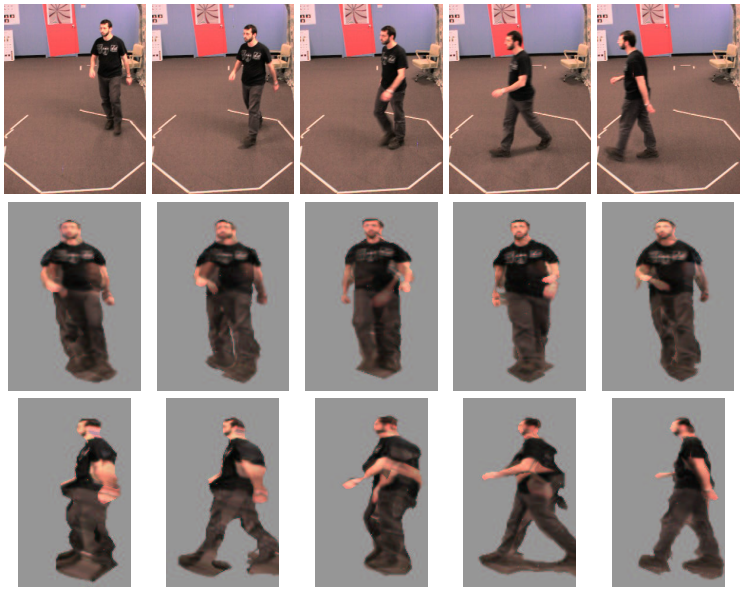


Fig. 2. An example of tracking body position and orientation using a Kalman Filter: input from one of the cameras (top row), synthetic frontal view (middle row) and synthetic side-view (bottom row).

$O(kn^2)$ where k is the number of input images (the number of views). A variant of this algorithm provides a polyhedral 3D approximation of the VH [14]. This $O(k^2n^2)$ algorithm represents contour of each silhouette as a polygon set, and computes in 2D image planes the pairwise intersections between every pait of cones, resulting in $k-1$ polygon sets for each silhouette. Intersection of these polygons set at each cone face defines the 3D polyhedron; this is the approximation of the surface of the VH with a polygonal mesh.

After the VH is constructed, its surface is texture-mapped based on the original images ([14]). Let θ_i be the angle between the viewing ray of the virtual camera for a given pixel p, and the viewing ray of the i-th camera for p. Then each view is assigned a weight $1-\theta_i/max_i\theta_i$, and the value of p in the synthetic view is a weighted sum of the values in the original images.

Figure 1 shows an example of the original images, and the resulting VH without and with texture. The VH allows us to render a synthetic view of the object from desired viewpoints, at a moderate computational cost, and also provides information about the object's 3D loca-

tion and shape. We use this information to track the position and pose of a user in the environment, and to reduce the complex task of view-invariant recognition to the simpler one of view-normalized recognition.

3 Tracking and estimating canonical views

To render virtual views for recognition, we need to determine the canonical pose of the camera which will generate the most discriminative view. In general, one could formulate the view selection process as part of the overall recognition framework, as in [5]. Indeed, given freedom to design the recognition method as well as to select the optimal view, a general optimization would be necessary. In our current work, however, we presume the use of external, black-box recognition engines for face and gait recognition. These methods have been constructed with the explicit assumption of a canonical view, so we use them directly. For faces we place the camera in the plane fronto-parallel to the face, and for gait sequences we place the camera so that it observes a side-view of the walking sequence. We have developed algorithms based on motion analysis and pattern detection to estimate these viewpoints. A strong assumption that we make is that the person is walking and generally facing forward; this allows us to use trajectory analysis to help constrain the search for canonical views.

3.1 Trajectory analysis

Without loss of generality we presume that the XZ-plane of our coordinate system is the ground plane, and the Y axis is the normal to the ground. We estimate the location of the centroid of the subject by taking the center of gravity of the VH $\mathbf{c} = \langle c_x, c_y, c_z \rangle$. The method of computing $\mathbf{c}$ depends on the VH algorithm. For the polyhedral VH, it is simply the centroid of the polyhedral model, which can be computed while building the model. This method was used in all the experiments described in this paper. For the sampled VH, one estimates the VH by integrating the volume enclosed within the endpoints of the ray intervals, and computes the zero-th moment of that volume. A third, more ad-hoc approach consisting of computation of the 3D bounding prism (which can be done directly from the silhouettes) and taking its centroid, was found by us to be inferior in practice.

Given the estimated centroids of the VH in two consecutive frames $\mathbf{c}_t$ and $\mathbf{c}_{t+1}$, we estimate the motion of the object between t and $t+1$ by $\Delta\mathbf{c} = \mathbf{c}_{t+1} - \mathbf{c}_t$. Under the assumption that the motion is parallel

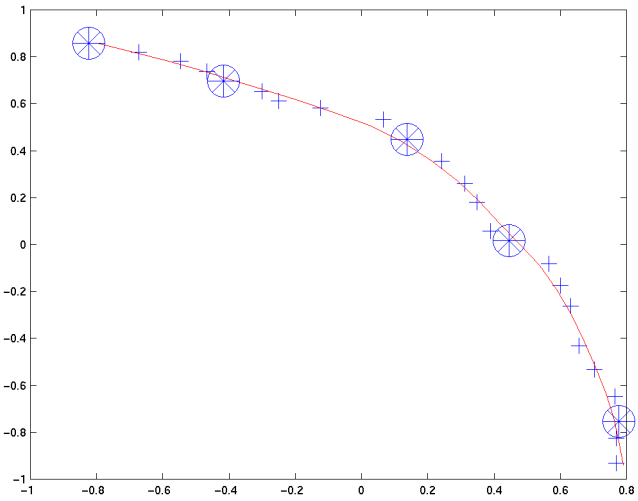


Fig. 3. Trajectory estimated from sequence in Figure 2. Frames shown in top row of Figure 2 are marked with an asterisk. Virtual views are generated along the tangent and normal to this trajectory for face and gait recognition, respectively.

to the XZ plane, we consider the projection of $\Delta\mathbf{c}$ on that plane as the motion vector.

We shall call the set of the synchronized views obtained at time t a *multiframe* f_t. The VH computed from f_t will be denoted by VH_t. Instantaneously, we need to fit a straight line $z = mx + b$ to the (noisy) centroid observations. This is done by solving a linear least-squares optimization problem, for the $\langle x_t, z_t \rangle$ in each multiframe VH_t. This gives us the unit vector $\mathbf{v}_t$ in the estimated direction of the person at time t. Once we have established the direction, we can place a "virtual camera", say, in front of the person, at a desired distance δ:

$$O_t = \mathbf{c}_t + \delta\mathbf{v}_t \tag{1}$$

For a general trajectory, we use a constant-velocity Kalman filter to recover the centroid path.

Figures 2 and 3 demonstrates the results of the method. Input from only one camera out of four is shown for reference. While the orientation estimate is not perfect, we keep track of the orientation after the person

turns at about 60 degrees, and can automatically produce synthetic frontal (middle row of Figure 2) and profile (bottom row) views. (Note that there are some texture rendering artifacts present in the profile sequence–these are visually distracting but do not cause problems for our silhouette based gait algorithm.)

The assumption of fronto-parallel motion implicit in our trajectory analysis can be relaxed by combining the motion-based orientation estimate with one based on face-detection, as described in the next section.

3.2 Detection-based view estimation

Fig. 4. View-normalized gait and face recognition features based on trajectory in Figure 3.

A pattern detection approach can be applied to a set of rendered virtual views to find those that are most "canonical" relative to a desired class. For faces, we use a real-time face-detection method [20] to detect the frontal view condition. This implementation, which uses small number of highly-relevant features, can process images of 400x300 pixels in roughly .07 seconds. However, we need to apply it to much smaller images. Given the VH of a person, and assuming roughly upright body pose, we need to consider only the top part of the VH. In our experiments we chose to look at the top 1.5 feet. We place the virtual camera at the distance that would produce the desired resolution of the image (in the described setup, 60x60 pixels).

If no trajectory information is available, we can search a circle of views around the 3-D location of a users head (Figure 5). If trajectory information is available, the head area is then rendered for a small range of spatial angles around the currently estimated face orientation.

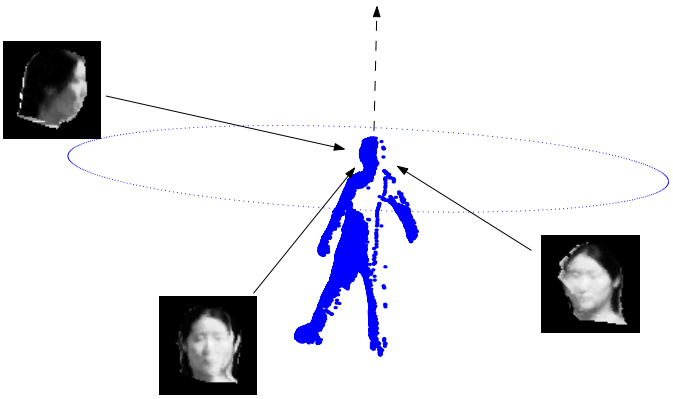


Fig. 5. Virtual views can also be generated based on the position of the users head and a ground plane constraint.

A set of 25 such images has the same total size as one 300x300 image, and takes similar time for a face detector to process.

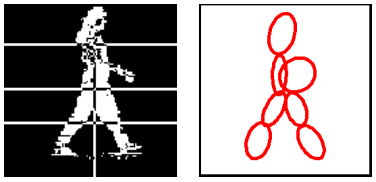
We also reduce the scale space, since the virtual camera is placed at a known distance from the VH, thus leading only a small range of possible sizes of the face.

4 Recognition on virtual sequences

We take the virtual sequences rendered from canonical viewpoints and input them to view-dependent face and gait recognition algorithms. Typically these methods are based on 2-D or 2.5-D (XY+T) analysis.

4.1 Gait recognition

Human gait can serve as a discriminative feature for visual recognition, as suggested by theoretical biometric ([10]) and empirical ([7, 16, 18]) results. Here we applied a simple gait recognition scheme based on silhouette extent analysis, which was developed separately from our work. The basic method is reported in [13] and was successfully demonstrated on sequences where the direction of motion was explicitly parallel to the camera plane.



(a) Partition of a silhouette

(b) Fitting an ellipse to each region

Fig. 6. Computing the feature vector for gait recognition. From [13].

The gait dynamics feature vector consists of smoothed versions of moment features in image regions containing the walking person. For each silhouette of a gait video sequence, we find the centroid of the whole silhouette and divide it into 7 regions using the centroid. For each of the regions, we fit an ellipse to describe the centroid, the aspect ratio and the orientation of the portion of foreground object visible in that region(Figure 6(b)). These silhouette–based features are computed for each frame of a video sequence. These time-varying signals from a video sequence are compressed across time using the mean and standard deviation of the centroid, aspect ratio, and orientation of each region. The time-compressed features from all 7 regions together form a gait feature vector. A diagonal covariance Gaussian model is used for each of these features, and a nearest neighbor classifier is used to decide which person has walking dynamics closest to the query feature vector. This method is surprisingly simple, but works in a range of realistic conditions [13]. More complex models, including those that recover kinematic biometrics and/or periodic features, could also be easily integrated into our framework.

The features used in this gait recognition algorithm are clearly view-dependent, and it is generally impractical to collect data for each person across all possible views. Recognition using a sequence rendered from a virtual viewpoint in canonical position is an appealing alternative. For each sequence of multiframes $\mathbf{x}$, two silhouette sequences are produced - a synthetic view from the left and from the right can be created for each frame, relative to the estimated motion vector. We denote those by $\mathbf{s}_L$ and $\mathbf{s}_R$. Figure 4(top) shows an example view-normalized silhouette input to the recognition method.

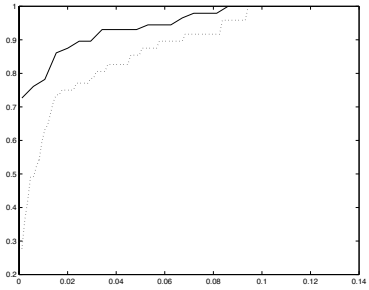


Fig. 7. A rank-threshold plot for gait recognition using view-normalization (solid line) versus using only the raw input silhouettes (dotted line).

We maintain an ID-tagged database of silhouette sequences, obtained from the VH of the previously observed people. To recognize a new sequence, we compute the distances between the feature vector of $\mathbf{s}_L$ and $\mathbf{s}_R$ and those of all the silhouette sequences $\mathbf{s}$ in the database. We exclude $\mathbf{s_L}$ and $\mathbf{s_R}$ themselves, and choose the minimum between the two values as the distance between $\mathbf{x}$ and the other silhouettes. Then, we normalize the vector

$$\mathbf{p}_g(\mathbf{x}) = \left[1/\min_{label(\mathbf{s})=1} dist(\mathbf{s},\mathbf{x}), \ldots, 1/\min_{label(\mathbf{s})=K} dist(\mathbf{s},\mathbf{x})\right] \quad (2)$$

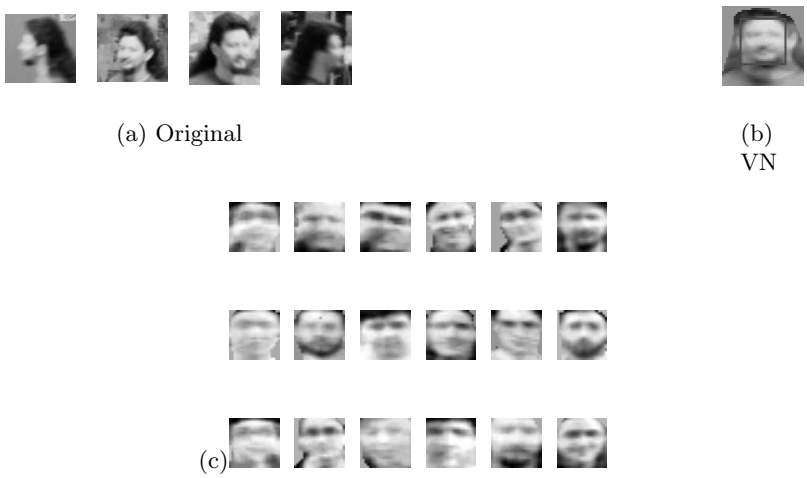
The estimated confidence that $\mathbf{x}$ is actually from person k is denoted $\mathbf{p}_{g_k}(\mathbf{x})$. Choosing k which maximizes this confidence gives our classification decision.

4.2 Face recognition

When a scene is viewed by a small number of far-placed cameras, often there is no view close enough to frontal to allow face recognition, and even detection. For example, on all of the original textures in Figure 8(a) face detection fails. However, faces are easily detected in the frontal virtual views, such as that shown in Figure 4(bottom) and Figure 8(b). Figure 8(c) shows a sample of view-normalized model faces.

We consider face recognition algorithms that are trained on a database with certain amount of view-dependence. Typically such a database includes frontal views of faces. So far, we tested our approach with eigenfaces.

For each multiframe x_t, we render synthetic views of the top part of VH for a small range of spatial angles around the estimated motion



(a) Original (b) VN

(c)

Fig. 8. Face detection typically fails on the input views due to varying pose (a), but succeeds on the visual hull-based view-normalized image (b). Pairs of view-normalized faces from the same individual are shown in (c). Conventional view-dependent face recognition methods can match (b) to the appropriate individual in (c) (top row, right or bottom row, second right).

vector. These images are processed by a face detector, and the ones where a face was detected are included in a set of $Faces_t(\mathbf{x})$. After having seen n frames, the set $Faces(\mathbf{x}) = \bigcup_{i=1}^{n} Faces_i(\mathbf{x})$. If $Faces(\mathbf{x})$ is non-empty, we can use all the face images in it for recognition. Let $m = |Faces(\mathbf{x})|$. Let D be an $m \times K$ matrix of distances between each $I_i \in Faces(\mathbf{x})$ and each one of the K eigenspaces represented in the database:

$$D_{ij} = |I_i - \mathbf{S}_j I_i|, \tag{3}$$

Then we compute for each image I_i a weight vector

$w_i = [1/D_{i1}, \ldots, 1/D_{iK}]$, which is further normalized to produce a confidence vector. This vector describes the estimated confidence that I_i belongs to the Kth person. We have m images, so for the whole sequence $\mathbf{x}$ we compute the confidence vector

$$\mathbf{p}_f(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{w}_i. \tag{4}$$

Our classification is then done by selecting

$$\mathbf{x} = \underset{j}{\operatorname{argmax}}\, p_{f\,j}(\mathbf{x}).$$

4.3 Multi-modal recognition

Finally, we combine the face and gait recognition results in order to establish a higher confidence level. Since empirically the success rates of face and gait classifiers were similar (c.f. Table 1(d)), we assigned an equal weight of .5 when combining confidence vectors. Given $\mathbf{p}_f(\mathbf{x})$ and $\mathbf{p}_g(\mathbf{x})$ for the observed sequence of multi-views $\mathbf{x}$, we compute the multi-modal confidence vector

$$\mathbf{p}_c(\mathbf{x}) = \begin{cases} \mathbf{p}_g(\mathbf{x}), & \text{if } \mathit{Faces}(\mathbf{x}) = \emptyset \\ \left(\mathbf{p}_g(\mathbf{x}) + \mathbf{p}_f(\mathbf{x})\right)/2, & \text{otherwise.} \end{cases} \tag{5}$$

(a)

5	0	0	0	0	0	0	0	0	**1**	0	0
0	**4**	0	0	0	0	0	0	**1**	0	0	**1**
0	0	**3**	0	0	0	0	**1**	0	0	0	0
0	0	0	**4**	0	0	0	0	0	0	0	0
0	0	0	0	**4**	0	0	0	0	**1**	0	0
0	0	0	0	0	**3**	0	0	0	0	**1**	0
0	0	0	0	0	0	**6**	0	0	0	0	0
0	0	0	**1**	0	0	0	**4**	0	0	**1**	**1**
0	0	0	0	0	0	0	0	**2**	0	0	0
0	0	0	0	0	0	0	0	0	**6**	0	0
0	0	0	0	0	0	0	0	0	**1**	**2**	0
0	0	0	0	0	0	0	0	0	0	0	**4**

(b)

5	0	**1**	0	0	0	0	0	0	0	0	0	0
0	**4**	0	0	0	0	0	0	0	0	0	0	0
0	0	**2**	0	0	0	**2**	0	0	0	0	0	0
0	0	0	**4**	0	0	0	0	0	0	0	0	0
0	0	0	0	**5**	0	0	0	0	0	0	0	0
0	0	0	0	**3**	0	0	0	0	0	0	**1**	
0	0	**1**	0	0	0	**5**	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	**1**	0	
0	0	0	0	0	0	0	0	0	**6**	0	0	
0	0	0	0	0	0	0	0	0	0	**3**	0	
0	0	0	0	0	0	0	0	0	0	0	**4**	

(c)

6	0	0	0	0	0	0	0	0	0	0	0	0
0	**4**	0	0	0	0	0	0	0	0	0	0	0
1	0	**3**	0	0	0	0	0	0	0	0	0	0
0	0	0	**4**	0	0	0	0	0	0	0	0	0
0	0	0	0	**4**	0	0	0	0	**1**	0	0	
0	0	0	0	0	**3**	0	0	0	0	**1**	0	
0	0	0	0	0	0	**6**	0	0	0	0	0	0
0	0	0	0	0	0	0	**4**	0	0	**1**	**1**	
0	0	0	0	0	0	0	0	**2**	0	0	0	
0	0	0	0	0	0	0	0	0	**6**	0	0	
0	0	0	0	0	0	0	0	0	0	**3**	0	
0	0	0	0	0	0	0	0	0	0	0	**4**	

(d)

Modality	(chance)	No VH-face	No VH-gait	No VH-gait and face	VH-face only	VH-gait only	VH-gait and face
Recognition rate	.08	.31	.52	.44	.8	.87	.91

Table 1. Confusion matrices for (a) gait-only, (b) face-only, and (c) integrated recognition using VH. Note that there was no face data obtained for subject 8, who was wearing a hat during the experiments. (d) Summary of the recognition results.

5 Results

We tested our methods using an installation with four monocular cameras. Each were located at roughly the same height, approximately 45 degrees apart, yielding set of images like that in Figure 1. The intersection of their fields of view defines the working space of our system. The cameras were calibrated off-line and temporally synchronized in hardware.

Silhouettes were computed using a simple color background model. For each pixel, the mean and variance of its values are computed over a large number of frames when the scene is known to contain no object. Segmentation is performed with three steps. First, each pixel in the data image is labelled 'background' if its value is within two standard deviations from the mean, and 'foreground' otherwise. Second, a normalized correlation analysis is then computed for a small window around each foreground pixel, and it is reset to background if the correlation score is sufficiently high. Finally, a morphological close operation is performed. The last two steps reduce the impact of shadows.

For 12 subjects we collected between 2 and 6 VH sequences as they walked in an arbitrary direction through the visual hull workspace, which was approximately 3m in diameter. The accuracy of gait classification was estimated using leave-one-out cross-validation. Figure 7 compares gait recognition performance using normalized vs. unnormalized views. Accuracy vs. rank threshold is plotted for the each approach, indicating the percentage of trials where the correct label was within the top n predicted labels (where n is the rank-threshold value). As can be seen, recognition with the unnormalized sequences was substantially worse than with our view-normalization approach. A confusion matrix for $n = 1$ is shown in Table 1(a)

View-normalized face recognition was also performed on these data, using the method described above. Table 1(b) shows the results of classification using only the face observations. Finally, Table 1(c) shows the confusion matrix for integrated recognition. Table 1(d) summarizes the overall recognition rates for face-only, gait-only, and integrated recognition. Integrated recognition reduced the rank-threshold=1 recognition error rate from 13% to 9%.

Note the significantly inferior performance of the recognition in both modalities with the same data, but when no view-normalization is applied (Table 1 (d)). In this experiment, we used the images from all the four cameras, where segmented silhouettes were fed to the gait classifier, and face detection was used to extract faces from the textured camera inputs (with silhouettes defining the search regions). Face recognition performed especially poorly. In many sequences not a single face was detected, which is not surprising after looking at Figure 8. In addition, some false detections further decrease the performance.

6 Conclusions and future work

We have described a view-normalization approach for integrated tracking and recognition of people. Our system combines face and gait recog-

nition methods, and information from multiple views. An image-based visual hull is used for shape modeling and for trajectory tracking. Results were shown using view-dependent face and gait recognition modules, and were better than the unnormalized or single modality results. Each component of the system runs at real-time speeds.

Currently the implementation uses monocular silhouettes based on color segmentation with static backgrounds, but could be extended to accommodate more sophisticated segmentation algorithms. Our system works within the strict intersection of the field of view of all cameras, but we expect this to be relaxed as a more general visual hull algorithm is developed. Finally, our confidence integration method is clearly primitive in present form, and should be extended to an explicit probabilistic framework.

7 Acknowledgments

We are grateful to M. Jones and P. Viola who provided us with the implementation of the face detector, and to W. Matusik, C. Buehler and L. McMillan who developed the Visual Hull system, on which our experiments were based. This generous financial support by the DARPA HumanID program and MIT Project Oxygen is gratefully acknowledged.

References

1. Sumit Basu, Irfan Essa, and Alex Pentland. Motion regularization for model-based head tracking. In *Proceedings, 13th International Conference on Pattern Recognition*, Vienna, Austria, August 1996. IEEE Computer Society Press.
2. D. Beymer and T. Poggio. Face recognition from one example view. In *Proccedings of the International Conference on Computer Vision*, pages 500–507, 1995.
3. M. J. Black and Y. Yacoob. Recognizing facial expressions in image sequences using local parameterized models of image motion. In *Proceedings of the International Conference on Computer Vision*, 1995.
4. V. Blanz and T. Vetter. A morphable model for the synthesis of 3–d faces. In *Computer Graphics, SIGGRAPH Proceedings*, pages 71–78, Los Angeles, CA, 1999.
5. F. G. Callari and F. P. Ferrie. Active recognition: Using uncertainty to reduce ambiguity. In *In Proceedings of the 13th International Conference on Pattern Recognition*, Vienna, Austria, Aug.25-30 1996.

6. M. La Cascia, J. Isidoro, , and S. Sclaroff. Head tracking via robust registration in texture map images. In *Proceedings Computer Vision and Pattern Recognition (CVPR'98)*, pages 508–514, Santa Barbara, CA, 1998.
7. J.E. Cutting and L.T. Kozlowski. Recognizing friends by their walk: gait perception without familiarity cues. *Bull. Psychonometric Soc.*, (9):353–356, 1977.
8. G. J. Edwards, T. F. Cootes, and Christopher J. Taylor. Face recognition using active appearance models. In *ECCV (2)*, pages 581–595. Springer, 1998.
9. G. D. Hager and P. N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(10):1025–1039, 1998.
10. G Johansson. Visual motion perception. *Scientific American*, (232):76–88, 1975.
11. K. Kutulakos and S. Seitz. A theory of shape by space carving. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV-99)*, volume I, pages 307–314, Los Alamitos, CA, September 20–27 1999. IEEE.
12. A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, February 1994.
13. L Lee. Gait dynamics for recognition and classification. Technical Report AIM-2001-019, MIT AI Lab Memo, Sept. 2001.
14. Wojciech Matusik, Chris Buehler, and Leonard McMillan. Polyhedral visual hulls for real-time rendering. to appear in *Proceedings of EGWR-2001*, 2001.
15. Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, Annual Conference Series, pages 369–374. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.
16. S.A. Niyogi and E.H. Adelson. Analyzing and recognizing walking figures in xyt. In Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.469-474, 1994.
17. A. Pentland, B. Moghaddam, T. Starner, O.Oliyide, and M. Turk. View-based and modular eigenspaces for face recognition. Technical Report 245, MIT Media Lab Vismod, 1993.
18. R. Polana and R. Nelson. Low level recognition of human motion. In IEEE Workshop on Motion of Non-Rigid and Articulated Objects, pages 77– 82, Austin, 1994.
19. S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Proceedings Computer Vision and Pattern Recognition (CVPR'97)*, pages 1067–1073, 1997.
20. Paul A. Viola and Michael J. Jones. Robust real-time object detection. Technical report, COMPAQ Cambridge research Laboratory, Cambridge, MA, February 2001.

21. L. Wiskott, J.M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):775-779, 1997.
22. A.L. Yuille. Deformable templates for face recognition. Journal of Cognitive Neuroscience, 3(1):59-70., 1991.