massachusetts institute of technology — computer science and artificial intelligence laboratory

Virtual Visual Hulls: Example-Based 3D Shape Estimation from a Single Silhouette

Kristen Grauman, Gregory Shakhnarovich and Trevor Darrell

AI Memo 2004-003 January 2004

Abstract

Recovering a volumetric model of a person, car, or other object of interest from a single snapshot would be useful for many computer graphics applications. 3D model estimation in general is hard, and currently requires active sensors, multiple views, or integration over time. For a known object class, however, 3D shape can be successfully inferred from a single snapshot. We present a method for generating a "virtual visual hull"– an estimate of the 3D shape of an object from a known class, given a single silhouette observed from an unknown viewpoint. For a given class, a large database of multi-view silhouette examples from calibrated, though possibly varied, camera rigs are collected. To infer a novel single view input silhouette's virtual visual hull, we search for 3D shapes in the database which are most consistent with the observed contour. The input is matched to component single views of the multi-view training examples. A set of viewpoint-aligned virtual views are generated from the visual hulls corresponding to these examples. The 3D shape estimate for the input is then found by interpolating between the contours of these aligned views. When the underlying shape is ambiguous given a single view silhouette, we produce multiple visual hull hypotheses; if a sequence of input images is available, a dynamic programming approach is applied to find the maximum likelihood path through the feasible hypotheses over time. We show results of our algorithm on real and synthetic images of people.

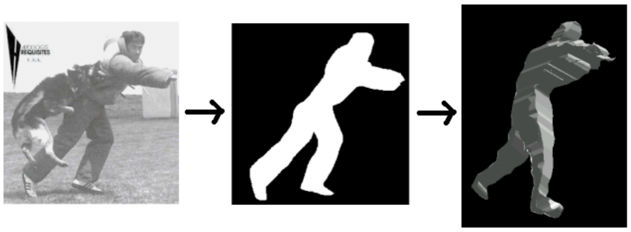


Figure 1: The goal of this work: given only a monocular silhouette of an object from a known class, estimate its 3D shape.

1 Introduction

Estimating the 3D shape of an object is an important computer graphics problem, but current techniques are still expensive or restrictive. Active sensing techniques can build accurate models quickly, but require scanning a physical object. Structure from motion or from multiple views is non-invasive, but requires a set of comprehensive views of an object. Often designers would like to populate 3D worlds with objects for which a set of simultaneous views is not available, e.g., to recover a shape based on a single archival photograph. Current techniques cannot construct a 3D model with just a single view.

In contrast, human observers can easily infer an approximate 3D model from even a single silhouette, especially if they know the class of objects to which the silhouette belongs (e.g., people, cars, books, etc). Clearly, knowledge of a particular object class can be exploited to perform shape inference. In this paper we show how shape inference can proceed from a single silhouette of a known object class. This is achieved by an example-based algorithm that retrieves those 3D shape examples in a large database which contain 2D views similar to the given input silhouette, and interpolates between these examples to obtain a shape estimate most consistent with that silhouette.

A silhouette-based approach to shape modeling – visual hull construction – approximates the 3D surface of an object by intersecting the viewing cones formed by the rays passing through the optical centers of calibrated (or weakly calibrated) cameras and their corresponding image silhouettes. Typically a relatively small number of input views (4-8) is sufficient to produce a compelling 3D model. Our method estimates the “virtual visual hull” for an object of a known class given only a single silhouette observed from an unknown viewpoint, with the object at an unknown orientation (and unknown articulated pose, in the case of non-rigid objects). We presume silhouettes have either been extracted manually or obtained by common background removal techniques.

We construct a model of the 3D shape of an object class using a database of many multi-view silhouette examples. The camera parameters corresponding to each multi-view training instance are known, but they are possibly different across instances. To infer a single novel silhouette's visual hull, we first search the database for examples in which one of the views is similar to the input, where similarity between silhouette contours is measured with the Hausdorff distance. An efficient parallel implementation allows us to search over 140,000 examples in modest time. Having retrieved such examples, we align and interpolate the corresponding visual hulls so as to optimize the correspondence between a projection of the resulting shape and the input.

To enable shape refinement within a local neighborhood of database examples, we introduce a new virtual view paradigm for interpolating between neighboring visual hull examples. Examples are re-rendered using a canonical set of virtual cameras; intermediate 3D shapes are then linear combinations in this multi-view contour space. This technique allows combinations of visual hulls for which the source cameras vary in number and calibration parameters.

The information about the shape contained in a single silhouette is, of course, ambiguous; to provide all possibilities to a user or designer would be overwhelming. To deal with this ambiguity in a single frame we develop a method for clustering the possible 3D interpretations, so that a relatively small number of candidate interpretations can be presented to the user. We also develop a dynamic programming method for the case when sequential data is available, so that some of the ambiguities inherent in silhouettes may be eliminated by incorporating information revealed by how the object (or camera) moves.

Our approach enables 3D surface approximation for a given object class with only a single silhouette view and requires no knowledge about either the object's orientation (or articulation), or the camera position. Our method can use sequential data to resolve ambiguities, or alternatively it can simply return a set of confidence-rated VH hypotheses for a single frame. We base our shape model on the concise 3D descriptions that visual hulls provide: we can match the multi-view model in one viewpoint and then generate on demand the necessary virtual silhouette views from the training example's visual hull. Our method's ability to use multi-view examples from different camera rigs allows training data to be collected in a variety of real and synthetic environments. With our method approximate 3D models can be quickly obtained from silhouettes extracted from a single snapshot. This capability has a variety of applications – an animator can render a character from an arbitrary viewpoint based on a single simple sketch, or populate interactive 3D environments with avatars or other objects.

2 Related Work

Visual hull algorithms have the advantage that they can be very fast to compute and re-render, and they are also much less expensive in terms of storage requirements than volumetric approaches such as space carving or voxel coloring [16, 23]. The input representation required by visual hull construction algorithms - 2D silhouettes - are generally easy to obtain with a background subtraction module. The output polyhedral surfaces are directly appropriate for rendering, or rendering can be done directly from source images [20]. Since the visual hull is by definition an upper bound on the true object shape, it is a useful representation for applications in interactive graphics or manipulative robotics such as obstacle avoidance and visibility analysis.

Algorithms for computing the visual hull of an object have been developed based on the explicit geometric intersection of generalized cones [17]. Recent advances in visual hull construction techniques have included ways to reduce their computational complexity [20, 19, 18, 6, 2], or to allow for weakly calibrated cameras [18]. In [12] it is shown how to incorporate priors that reconcile errors in the initial silhouette segmentation with the known object shape. A method combining visual hulls and stereo is given in [5, 4] for the purpose of refining an object's visual hull by aligning its hulls from multiple frames over time. We rely on the efficient construction algorithm of [19] to calculate polygonal mesh visual hulls.

In the absence of calibrated cameras, Structure-From-Motion (SFM) techniques may be used with a sequence of data to estimate both the observed object's shape as well as the motion of the camera observing it. Most such algorithms rely on establishing point or line correspondences between images and frames, yet smooth surfaces without a prominent texture and wide-baseline cameras make correspondences difficult and unreliable to determine. Moreover, in the case of SFS, the occluding contours of the object are the only feature available to register the images. Current techniques for 3D reconstruction from silhouettes with an uncalibrated camera are constrained to the cases where the camera motion is of a known type (e.g., [26, 7, 24]).

When sequences of images are available, an alternative to geometry- or point correspondence- based approaches like SFM is to utilize knowledge about the dynamics, or motion behavior, of the object moving in the video. For instance, when the object class is people, knowledge about the types of motions the person is likely to perform may be exploited in order to infer the person's pose or shape. In the work of [3], a hidden Markov model is used to model the dynamics and 3D pose configurations of a human figure in order to infer pose from a sequence of silhouettes by solving for the optimal path through the model via entropy minimization. Our method for integrating temporal information is related to this approach. Our handling of sequential data uses dynamic programming to find the maximum likelihood path through a sequence of hypothesis virtual visual hulls, where the probability of choosing a certain hypothesis is measured by both the smoothness in the shape transitions over time, as well as the similarity between the observed silhouette and the projected views of the hypothesis virtual visual hull. Our temporal integration step differs from that of [3], however, in that it does not rely on a strong model of dynamic behavior (learned from mocap data), it processes different features (contours instead of central moments), and our method seeks to estimate a 3D surface that fits the actual measurements of our input instead of rendering a "tinman" based on configural pose estimates.

A popular way to represent the variable shape of an object has been to employ a parametric distribution that captures the variation in the object shape. Such a model is often used for tracking, pose inference, or recognition. The use of linear manifolds estimated by PCA to represent an object class's shape, for instance, has been developed by several authors [15, 8, 1]. An implicitly 3D probabilistic shape model was introduced in [12], where a multi-view contour-based model using probabilistic PCA was given for the purpose of visual hull regularization [12]. A method for estimating unknown 3D structure parameters with this model was later given in [13]. However while [12, 13] require input views to be taken from cameras at the same relative angles as the training set, our method requires a single view with no calibration information at all.

Example-based and non-parametric density models of object shape have also been explored previously. In such models the object class is represented by a set of prototypical examples (or kernel functions centered on those examples), using either raw images or features extracted from them. For instance, the authors of [25] use 2D exemplars to track people and mouths in video sequences, a template hierarchy is developed for faster detection of pedestrian-shaped contours in [11], and in [21] a database of single view images with annotated body pose parameters is searched for a match to a test body shape based on its edges. In [22], a special mapping is learned from multi-view silhouettes to body pose. We employ a class-specific database comprised of many examples of multi-view silhouettes taken from different calibrated camera rigs.

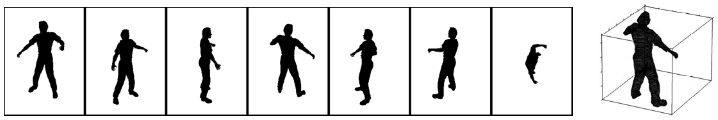


Figure 2: Examples in the model are composed of some number of silhouette views, plus their camera calibration parameters.

3 Searching the Shape-Silhouette Database

The shape-silhouette database is the key data structure in our approach. A multi-view example $\mathbf{X}^{(i)}$ in the database consists of m_i silhouettes $\{\mathbf{I}_j^{(i)}, \ldots, \mathbf{I}_{m_i}^{(i)}\}$ and the associated camera calibration parameters $\mathbf{p}_j^{(i)} \in \mathbb{R}^4$, such that $\mathbf{I}_j^{(i)}$ is obtained as the projection of the shape $S^{(i)}$ on $\mathbf{p}_j^{(i)}$:

$$\mathbf{I}_j^{(i)} = \mathrm{Proj}\left(S^{(i)}, \mathbf{p}_j^{(i)}\right).$$

(See Figure 2.) The exact shape that produced the example in general cannot be reconstructed faithfully, but an approximation of $S^{(i)}$ can be obtained from the m_i silhouettes by constructing the visual hull (VH). In this work we use the algorithm that computes a polyhedral mesh approximation of the VH [19]; in the remainder of the paper we refer to this approximation as the shape $S^{(i)}$.

Note that we do not require that the camera parameters and even the number of cameras be identical across the database. This creates a flexible paradigm of data generation: various camera rigs, at different locations, on different days, etc. may be employed to generate the examples. Moreover, this paradigm allows for generating additional silhouettes for $S^{(i)}$ by rendering its projection for a desired virtual viewpoint, thus increasing m_i as desired.

The first step of our approach consists of finding in the database shapes that match the input silhouette $\mathbf{I}$. We define the distance between $\mathbf{I}$ and a multi-view example $\mathbf{X}^{(i)}$ as the minimal distance between $\mathbf{I}$ and any view of $\mathbf{X}^{(i)}$. In principle, computing this requires a search over all possible silhouettes of $S^{(i)}$:

$$D\left(\mathbf{I}, \mathbf{X}^{(i)}\right) = \min_{\mathbf{p}} D_S\left(\mathbf{I}, \mathrm{Proj}\left(S^{(i)}, \mathbf{p}\right)\right), \tag{1}$$

where $D_S(\cdot,\cdot)$ is the distance function defined over the space of 2D silhouettes. The choice of D_S and of the representation of $\mathbf{I}$ will affect the semantics of our approach. We represent a silhouette as a set of points sampled uniformly from its contour, and use the Hausdorff distance, which for two sets of points $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$ is defined as:

$$||A - B||_H = \max(\max_{a \in A} D(a, B), \max_{b \in B} D(b, A)), \tag{2}$$

where $D(p, Q)$ is the shortest Euclidean distance from point p to any point in set Q. The Hausdorff distance has been proven to be an effective shape matching measure for such tasks as object recognition [14].

Generating the infinite set of all views of $S^{(i)}$ is impossible, of course. Therefore, we rewrite Eq. (1) as

$$D\left(\mathbf{I}, \mathbf{X}^{(i)}\right) = \min_j D_S\left(\mathbf{I}, \mathbf{I}_j^{(i)})\right), \tag{3}$$

that is, the distance is determined by the *best matching database view* of $\mathbf{X}^{(i)}$. One objection to this approach is that we may miss an example that is very close to the input, because its silhouettes in the database happen not to include the “right” one. However, this is not an impediment to our approach: we need to find *some* similar examples, not necessarily all of them and not necessarily the best ones.

A crucial parameter of the search algorithm is the similarity criterion which dictates when two silhouettes are considered sufficiently similar. We use K-nearest neighbors to retrieve a fixed number of top matches from the database. For the rare cases where the K neighbors contain an example whose 3D shape is much different than the others, our clustering mechanism (described in Section 4.2) will effectively handle the outliers.

To summarize: for the purpose of the similarity search, we treat the database as a collection of individual silhouettes. Given a silhouette, we find the K silhouettes $\mathbf{I}_{j_1}^{(i_1)}, \ldots, \mathbf{I}_{j_K}^{(i_K)}$ in the model database with the smallest Hausdorff distance from the input. The visual hulls corresponding to these silhouettes, $S^{(i_1)}, \ldots, S^{(i_K)}$, are then the best matching *shapes*. In the following section we describe a means of combining or interpolating between these best shape matches to improve the shape estimate.

4 Shape Interpolation

4.1 View alignment

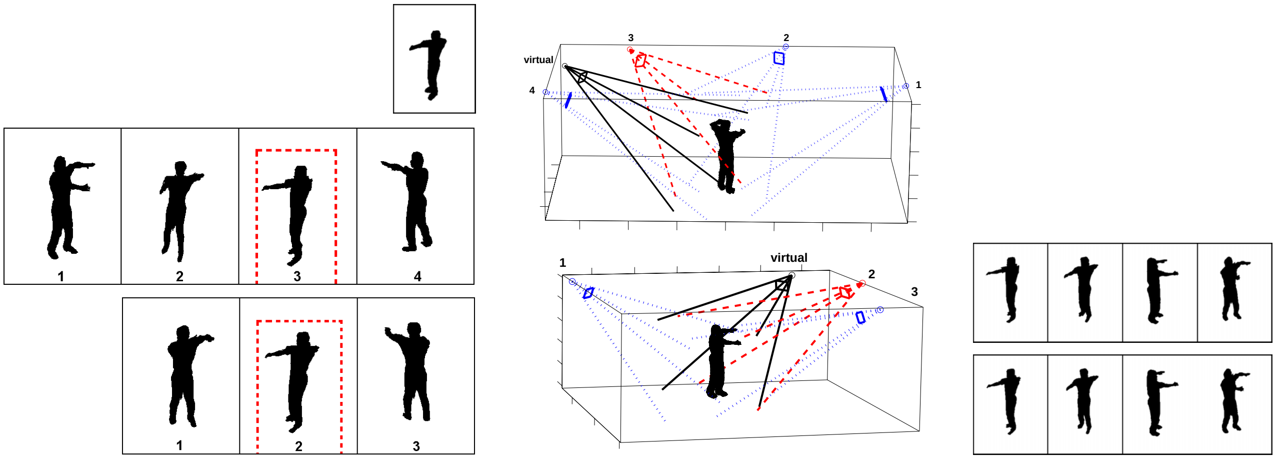
A key problem is how to combine multiple 3D shape examples to form a single shape estimate; naive interpolation of unaligned triangle meshes will not yield meaningful shapes. Instead, we propose a method to interpolate between visual hulls using weighted combinations of multi-view contours rendered from a set of canonical viewpoints, as described in this section.

In order to allow for meaningful interpolation, the shapes must be represented in the same *canonical* coordinate system with respect to object's inherent coordinate frame. Recall that each example $\mathbf{X}^{(i_k)}$, $k = 1, \ldots, K$ is selected due to the high similarity of a particular view $\mathbf{I}_{j_k}^{(i_k)}$ to the input $\mathbf{I}$. Under the hypothesis that they come from the same shape, the matching views are likely to correspond to the same viewpoint in terms of the intrinsic coordinate frame of the object, even if the extrinsic coordinates of these viewpoints may differ. Thus the first canonical viewpoint $\mathbf{P}_1^{(i_k)}$ of an example $\mathbf{X}^{(i_k)}$ is the database viewpoint $\mathbf{P}_{j_k}^{(i_k)}$.

The m-th canonical viewpoint of $\mathbf{X}^{(i_k)}$ can now be obtained by applying a fixed affine transformation T_m (same for all k) to $\mathbf{P}_1^{(i_k)}$. While this transformation will have a different interpretation in the extrinsic coordinates system of each example, in all the examples it will result in the same viewpoint in the object coordinate system. For every $\mathbf{P}_l^{(i_k)}$ we render the corresponding canonical silhouette

$$\mathbf{C}_l^{(i_k)} = \text{Proj}\left(S^{(i_k)}, \mathbf{P}_l^{(i_k)}\right).$$

To clarify with an example: suppose two multi-view neighbors for a novel input contain four and three views with camera parameters $\{\mathbf{P}_1^1, \ldots, \mathbf{P}_4^1\}$ and $\{\mathbf{P}_1^2, \ldots, \mathbf{P}_3^2\}$, respectively (see Figure 3). Suppose the third view in the first example and the second view in the second example matched the input. The first canonical views for the two examples are thus these matching silhouettes. Then the second canonical view for the first example is taken from the projection of its visual hull onto the image plane corresponding to the virtual camera found by rotating $\mathbf{P}_3^1$ by θ degrees about the visual hull's vertical axis. Similarly, the second canonical view for the second example is taken from a camera placed θ degrees from $\mathbf{P}_2^2$. Subsequent canonical virtual views for each example are taken at equal intervals relative to these initial virtual cameras. After taking a weighted



(a) Input and actual camera views of neighbors; matching views are outlined

(b) Camera rigs of neighbors

(c) Virtual views generated from neighbors' VHs

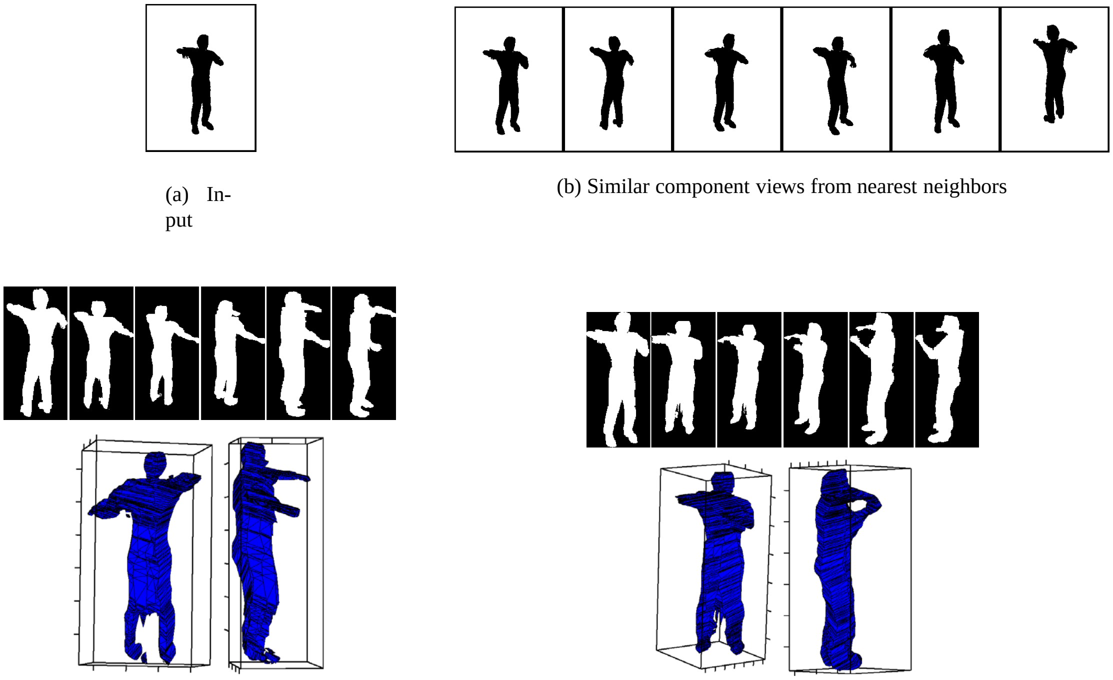
Figure 3: An example of rendering canonical views from the K-nearest neighbors of the input. Views that matched the input are marked with dotted boxes (a), and their corresponding cameras in the two examples' rigs are also shown with dotted lines (b). See text for details.

combination of the contours from these aligned-viewpoint virtual views, we will construct the output visual hull using the camera parameters of the nearest neighbor's similar view and its virtual cameras; since it was the best match for the input in a single view, it is believed to contain a viewpoint relative to the object that is most like the true unknown input camera's, up to a scale factor.

4.2 Clustering the neighbors

The 3D shape of a single view silhouette is inherently ambiguous: self-occlusions make it impossible to determine the full shape from a single frame, and the global orientation of the object may be uncertain if there is symmetry in the shape (e.g., a silhouette frontal view of a person standing with their legs side by side will be very similar to the view from behind). Thus we can expect the visual hulls corresponding to the "neighbor" single view silhouettes to manifest these different possible 3D interpretations. To combine widely varying contours from the neighbors' very different 3D shapes would produce a meaningless result (see Figure 4).

Instead, we provide multiple hypotheses for a single input silhouette corresponding to the multiple possible 3D interpretations for the 2D shape. The nearest neighbors' aligned (canonical viewpoint) multi-view virtual silhouettes are clustered into



(a) Input

(b) Similar component views from nearest neighbors

(c) Hypothesis 1

(d) Hypothesis 2

Figure 4: An example where the input shape (a) has multiple interpretations. Its six nearest neighbors (b) contain views from examples originating from two general types of shapes: one front-facing body with the right elbow extended more, the other back-facing with the left elbow extended more. Aligned-viewpoint virtual views are projected from each of the six neighbors' visual hulls, and our algorithm finds two separate shape hypotheses (c,d) based on how these multi-view images cluster. Each hypothesis is shown from a frontal view and right side view here. The mean multi-view shape for each hypothesis's virtual views are shown in the middle row with black backgrounds.

enough groups such that the distance between two multi-view examples in one cluster is less than a threshold. In order to perform the clustering, a set of multi-view silhouettes is encoded as a vector by simply translating the centroid of each of its silhouette views to a equally spaced origins in image coordinates, and forming a vector by concatenating the columns of each view image. In this way, a single vector contains multiple silhouette views, and each view within it is represented as spatially aligned with the other examples' silhouettes from the same canonical viewpoint. Each cluster of examples yields one hypothesis visual hull. The confidence for a given hypothesis is inversely proportional to the mean Hausdorff distance between the matching views from the examples that belong to the given cluster and the input silhouette.

For the single frame case, the hypotheses are returned to a user or designer, together with their confidences. When multiple, sequential frames are the input, then the hypotheses from each time step together undergo an additional processing stage, whereby the most likely hypothesis at each frame is selected for the output (see Section 5).

4.3 Shape estimation by interpolation

After the canonical contours are produced and clustered as described above, they are normalized in location (translated) and length (resampled), in order to align the contour points in the same view. For each cluster, we obtain the contours $(\mathbf{Y}_1, \ldots, \mathbf{Y}_v)$ that will form a hypothesis virtual visual hull by taking a linear combination of the aligned contours belonging to that cluster:

$$\mathbf{Y}_l = \sum_{j=1}^{n} w_j pts(\mathbf{C}_l^{(i_j)}), \tag{4}$$

where there are n members in the particular cluster, v total canonical viewpoints, and $pts(\cdot)$ refers to the process of resampling and translating the contour points along a silhouette boundary. The weight w_j assigned to the j^{th} multi-view contour in a given cluster is inversely proportional to the initial Hausdorff distance that was computed between the input silhouette and the example's matching view, and weights for a particular cluster are normalized so that they sum to one.

Since the vectors of contour points have been resampled to a common input length, the set of contour points produced in $\mathbf{Y}_l$ will not necessarily form a connected closed contour. Thus in order to produce the closed contour output required to form a silhouette, we fit a spline to the image points of $\mathbf{Y}_l$, and then perform a flood fill from the centroid. Note that this shape is, generally, no longer in the database, and may provide a better match for the input than any single training shape.

5 Integrating over time

When a sequence of monocular silhouettes is available, we apply a dynamic programming approach to find the maximum likelihood path through the feasible visual hulls at each time step, given our observations over time and the probabilities of transitioning from one shape to another. We construct a directed graph where each node is a time-indexed visual hull hypothesis (see Figure 5). Each directed path through the graph represents a legal sequence of states. Probabilities are assigned to each node and arc; node probabilities P_n^t are conditional likelihoods and arc probabilities P_a^t are transition probabilities:

$$P_n^t = P(\mathbf{I}^t | \mathbf{S}^t = \mathbf{S}_i) \tag{5}$$

$$P_a^t = P(\mathbf{S}^t = \mathbf{S}_j | \mathbf{S}^{t-1} = \mathbf{S}_i). \tag{6}$$

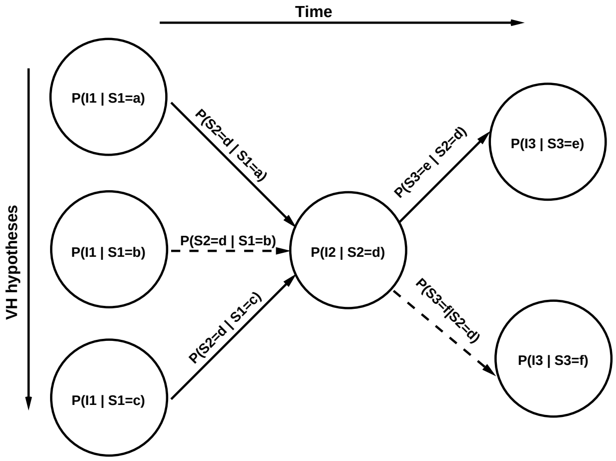


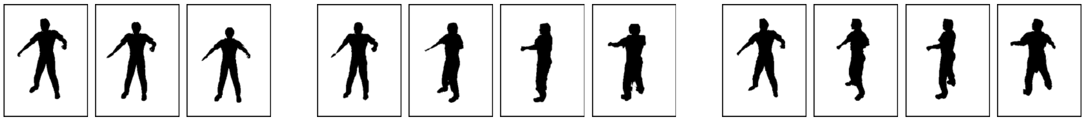
Figure 5: Illustration of a directed graph corresponding to three consecutive frames. Nodes are visual hull hypotheses, edges are shape transition probabilities. The dotted line indicates ML path found with dynamic programming; here, the most likely sequence of visual hull hypotheses consist of the second hypothesis at time 1 (b), the only hypothesis at time 2 (d), and the second hypothesis at time 3 (f).

P_n^t is thus an estimate of the probability of observing silhouette $\mathbf{I}$ at time t given that the 3D shape of the object at time t is best approximated by the i^{th} cluster hypothesis's visual hull, $\mathbf{S}_i{}^t$. We measure P_n^t in terms of the average Hausdorff distance between the input and the nearest-neighbor matching views belonging to the examples that were placed in the i^{th} cluster. P_a^t is the probability of the object having a certain 3D shape given its shape at the previous time step – a measure of the similarity between two hypotheses at consecutive time steps, which we estimate in our experiments in terms of the sum of the Hausdorff distances between a set of canonical views rendered from the two respective hypotheses visual hulls.

The maximum likelihood sequence of hypotheses in the graph is found using dynamic programming [10]. In this way an optimal path through the hypotheses is chosen, in that it constrains shapes to vary smoothly and favors hypotheses that were most likely given the observation. This process may be performed over windows of the data across time for sequences of significant length.

6 Experiments

We chose to build the model for human bodies from synthetic silhouettes using Poser, a computer graphics animation package [9]. The 3D person model is rendered from various viewpoints in randomized poses, and its silhouettes and camera parameters are recorded. An efficient parallel implementation allowed us to search over 140,000 examples in modest time (a



(a) Input view (left) and two nearest neighbors

(b) Views from inferred visual hull

(c) Same views from actual visual hull

Figure 6: Example of ground truth comparison for test on synthetic input with $E = 73$. See text for details.

few seconds per input).

We tested the inference method on both real and synthetic images. For the synthetic image tests, we generated a separate set of multi-view silhouettes using Poser, and then withheld each test example's visual hull information for ground truth comparisons. One view from each synthetic test example was input to our algorithm, and we measured the error E of the set of output visual hull hypotheses H for one test example as $E = \min_{h \in H} \left(\sum_{i=1}^{4} ||\hat{\mathbf{s}}^{\mathbf{P}_i} - \mathbf{s}^{\mathbf{P}_i}||_H \right)$, where $\hat{\mathbf{s}}^{\mathbf{P}_i}$ is the virtual view seen by a camera at pose $\mathbf{P}_i$ as inferred by our algorithm, and $\mathbf{s}^{\mathbf{P}_i}$ is the actual view seen by a camera at that pose for the withheld ground truth visual hull. For a synthetic set of 20 examples, the mean sum of Hausdorff distances over four views was 80. The Hausdorff distance errors are in the units of the image coordinates, thus a mean error of 80 summed over four views means that on average, per view, the farthest a contour point was from the ground truth contour (and vice versa) was 20 pixels. The synthetic test images are 240 x 320, with the contours having on average 700 points, and the silhouettes covering about 8,000 pixels in area. A typical example comparison between the virtual views generated by our virtual VH and the ground truth VH is shown in Figure 6.

We also inferred virtual visual hulls for real images. Some example real image results on are shown in Figures 8, 9, 10, 11, 12, 13, and 14. In these figures, each row corresponds to a different virtual visual hull hypothesis; three different viewpoints are rendered for each hypothesis. Beside each hypothesis is the rendering of a 3D stick figure, which is an estimate of the underlying pose (joint positions) of that virtual visual hull hypothesis. These 3D pose estimates were obtained in the same manner as the interpolated canonical virtual views, by taking a weighted combination of the known 3D poses recorded for the nearest-neighbor examples within a given cluster. An example from a short sequence of real images is given in Figure 7.

The real input images were simply pulled off the Web, and we performed manual segmentation on them to obtain silhouettes. Translation and scale invariant representations of the real image contours were obtained by subtracting the silhouette's 2D

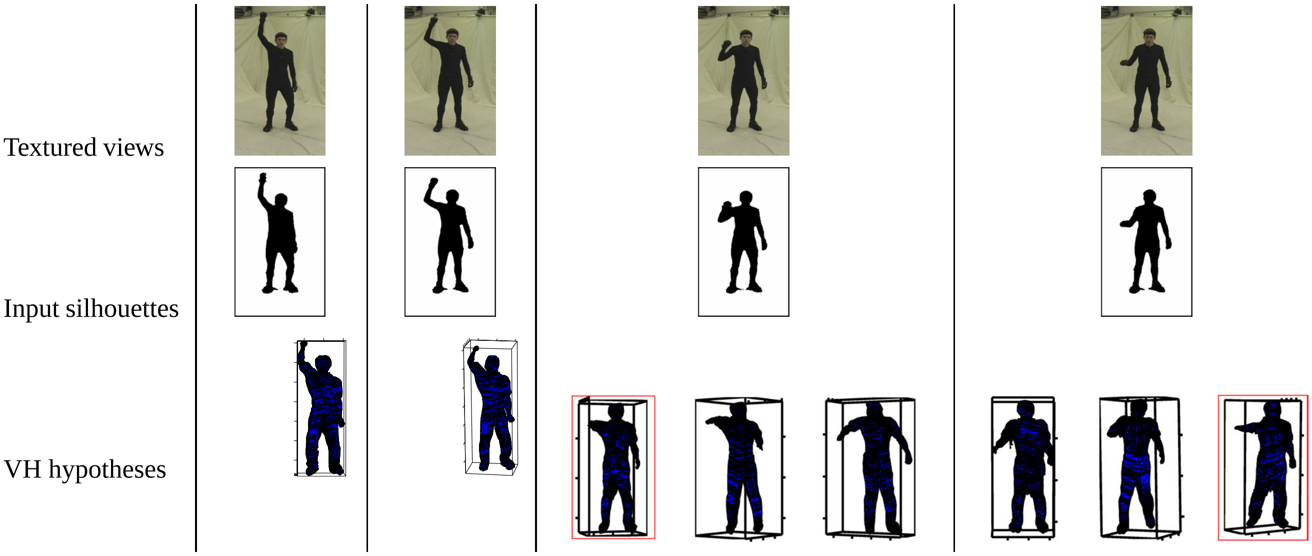


Figure 7: Example result on four sequential frames. The visual hull hypotheses are shown directly below the corresponding input at each time step. The hypotheses on the ML path appear in boxes. For the first two frames there is only one hypothesis.

center of mass from each contour point's image coordinate and normalizing by its approximate size so that the scale would be on par with the scale of the synthetic training images. The inferred visual hull hypotheses are typically in agreement with the input, and provide a good, fast approximation of the 3D shape.

7 Conclusions and Future Work

We showed how the 3D shape of objects in a known class could be inferred from single input silhouettes extracted from archival photographs. A multi-view silhouette database containing examples from calibrated, though possibly varied, camera rigs are used to match input silhouettes to the best 3D shape interpretations. Visual hull inference is performed by finding the shape hypotheses most consistent with the observed 2D contour. For single frame inputs, clustering is performed to reduce the number of shape hypotheses a user must select from; a set of likely but qualitatively different visual hulls is returned. When a sequence of frames is available, we use a dynamic programming technique to find the most consistent trajectory of hypothesis shapes over time.

Interpolation between neighboring examples allows our method to return shape estimates that are not literally in the set of examples used to define the prior model. We developed a new technique for 3D shape interpolation, using a set of viewpoint-aligned virtual views which are generated from the visual hulls corresponding to nearby examples. Interpolation between the contours of the aligned views produces a new set of silhouettes that are used to form the output visual hull approximating the 3D shape of the novel input.

We demonstrated our algorithm on a variety of real and synthetic images using a large database of synthetic images of people. The accuracy of shape inference was evaluated quantitatively with held-out synthetic test images, and qualitatively with real images. We expect our method to be useful anytime a fast approximate 3D model must be acquired for a known object class, yet active sensors, calibrated cameras or even multiple cameras are not available. Specifically, we see useful applications of our technique in computing virtual models of objects and people for interactive virtual reality applications. In the future we intend to investigate how we might incorporate a model of dynamics in order to apply our method on sequences of images, and we would like to explore different distance metrics for shape matching and apply our technique to additional object classes.

References

[1] A. Baumberg and D. Hogg. An adaptive eigenshape model. In *British Machine Vision Conference*, pages 87–96, Birmingham, Sept 1995.

[2] E. Boyer and J.-S. Franco. A hybrid approach for computing visual hulls of complex objects. In *Computer Vision and Pattern Recognition*, Madison, WI, June 2003.

[3] M. Brand. Shadow puppetry. In *International Conference on Computer Vision*, pages 1237–1244, 1999.

[4] G. K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Computer Vision and Pattern Recognition*, Madison, WI, June 2003.

[5] G. K. M. Cheung, S. Baker, and T. Kanade. Visual hull alignment and refinement across time: A 3d reconstruction algorithm combining shape-from-silhouette with stereo. In *Computer Vision and Pattern Recognition*, Madison, WI, June 2003.

[6] R. Cipolla, K. Astrom, and P. J. Giblin. Motion from the frontier of curved surfaces. In *International Conference on Computer Vision*, 1995.

[7] R. Cipolla and A. Blake. Surface shape from the deformation of apparent contours. *International Journal of Computer Vision*, 9(2):83–112, 1992.

[8] T. Cootes and C. Taylor. A mixture model for representing shape variation. In *British Machine Vision Conference*, pages 110–119, 1997.

[9] Curious Labs, Inc., Santa Cruz, CA. *Poser 5 - Reference Manual*, 2002.

[10] D. Forsyth and J. Ponce. Computer vision: A modern approach. pages 552–554, 2003.

[11] D. Gavrila and V. Philomin. Real-time object detection for “smart“ vehicles. In *International Conference on Computer Vision*, pages 87–93, Vancouver, BC, Sept 1999.

[12] K. Grauman, G. Shakhnarovich, and T. Darrell. A bayesian approach to image-based visuall hull reconstruction. In *Computer Vision and Pattern Recognition*, Madison, WI, 2003.

[13] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *International Conference on Computer Vision*, Nice, France, Oct 2003.

[14] Huttenlocher, Klanderman, and Rucklidge. Comparing images using the hausdorff distance. *PAMI*, 15, 1993.

[15] M. Jones and T. Poggio. Multidimensional morphable models. In *International Conference on Computer Vision*, Jan 1998.

[16] K. Kutulakos and S. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[17] A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 16(2):150–162, Feb 1994.

[18] S. Lazebnik, E. Boyer, and J. Ponce. On computing exact visual hulls of solids bounded by smooth surfaces. In *Computer Vision and Pattern Recognition*, pages 156–161, Lihue, HI, Dec 2001.

[19] Wojciech Matusik, Chris Buehler, and Leonard McMillan. Polyhedral visual hulls for real-time rendering. In *Proceedings of EGWR-2001*, pages 115–125, London, England, June 2001.

[20] Wojciech Matusik, Chris Buehler, Ramesh Raskar, Steven J. Gortler, and Leonard McMillan. Image-based visual hulls. In *Siggraph 2000, Computer Graphics Proceedings*, pages 369–374. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[21] G. Mori and J. Malik. Estimating Human Body Configurations using Shape Context Matching. In *ECCV*, 2002.

[22] R. Rosales and S. Sclaroff. Specialized mappings and the estimation of body pose from a single image. In *IEEE Human Motion Workshop*, pages 19–24, Austin, TX, 2000.

[23] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *International Journal of Computer Vision*, 35(2):151–173, 1999.

[24] R. Szeliski and R. Weiss. Robust shape recovery from occluding contours using a linear smoother. *International Journal of Computer Vision*, 28(1):27–44, 1998.

[25] K. Toyama and A. Blake. Probabilistic tracking in a metric space. In *International Conference on Computer Vision*, pages 50–59, Vancouver, BC, July 2001. IEEE Computer Society.

[26] K-Y. Wong and R. Cipolla. Structure and motion from silhouettes. In *International Conference on Computer Vision*, 2001.

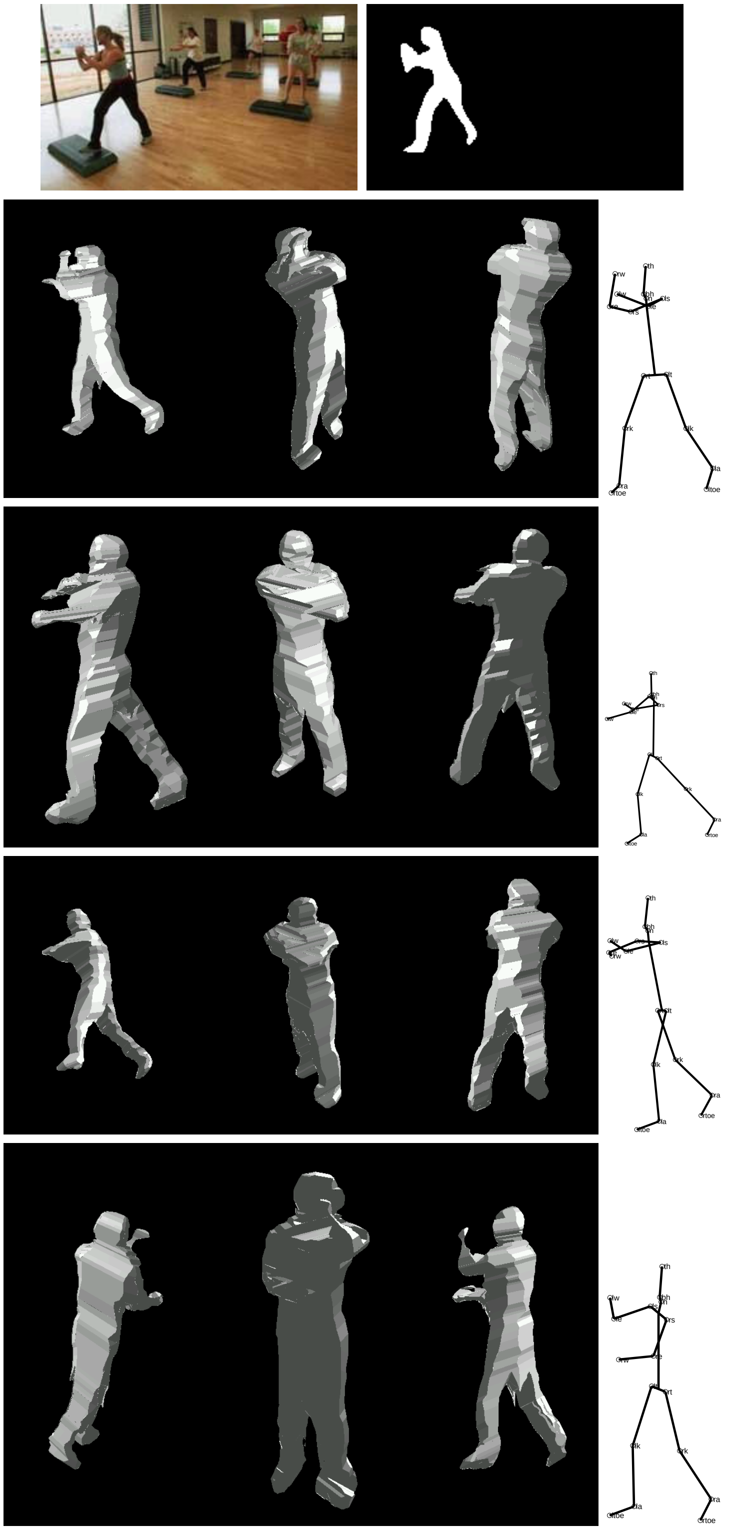


Figure 8: An example of four virtual visual hull hypotheses found by our system for a single input silhouette (top). Each row corresponds to a different hypothesis; three different viewpoints are rendered for each hypothesis. Stick figures beside each row give that VH's underlying 3D pose, retrieved by interpolating the poses of the examples which built the VHs. See text for details.

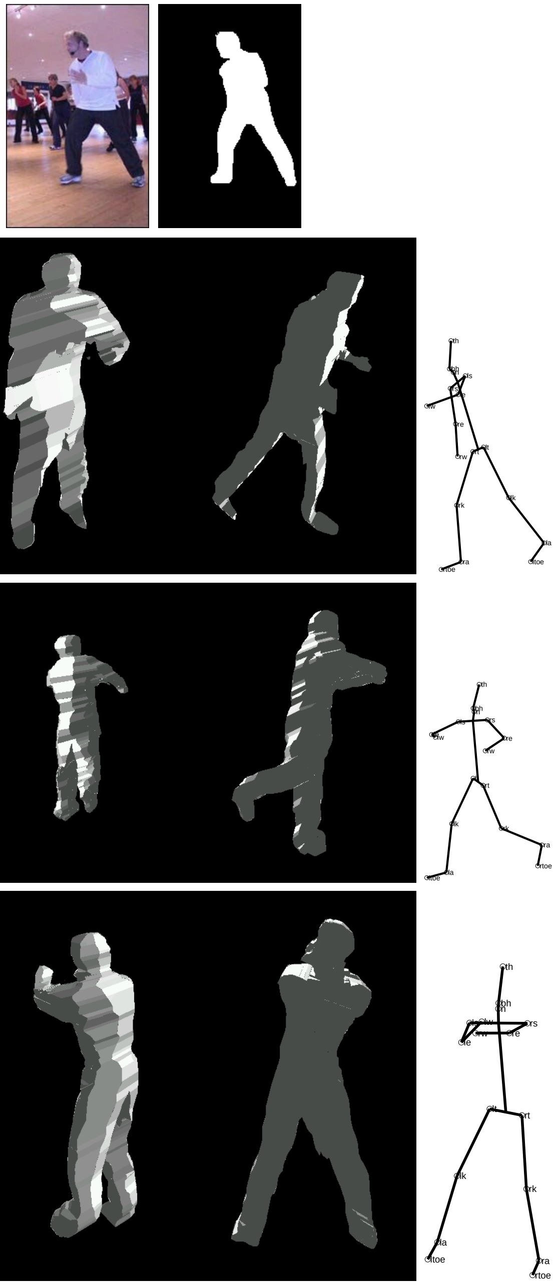


Figure 9: Example of inferred virtual visual hull hypotheses for a real image; plotted as in previous figure.

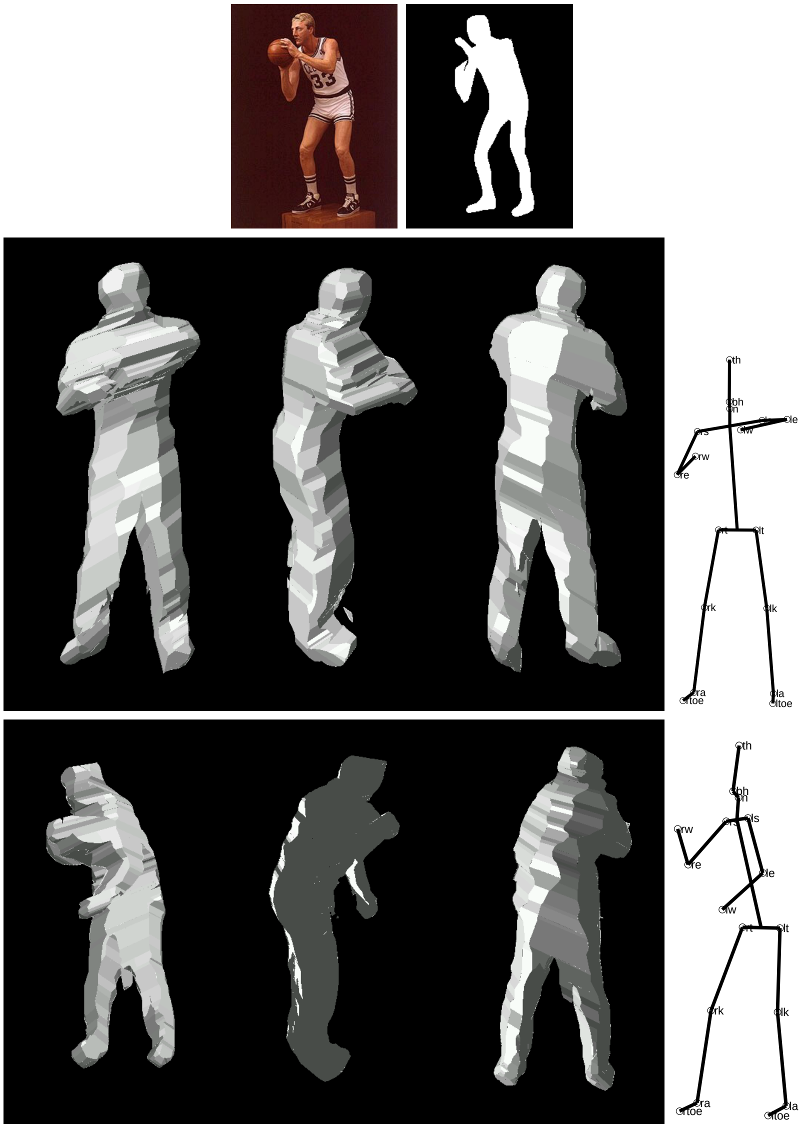


Figure 10: Example of inferred virtual visual hull hypotheses for a real image; plotted as in previous figure.

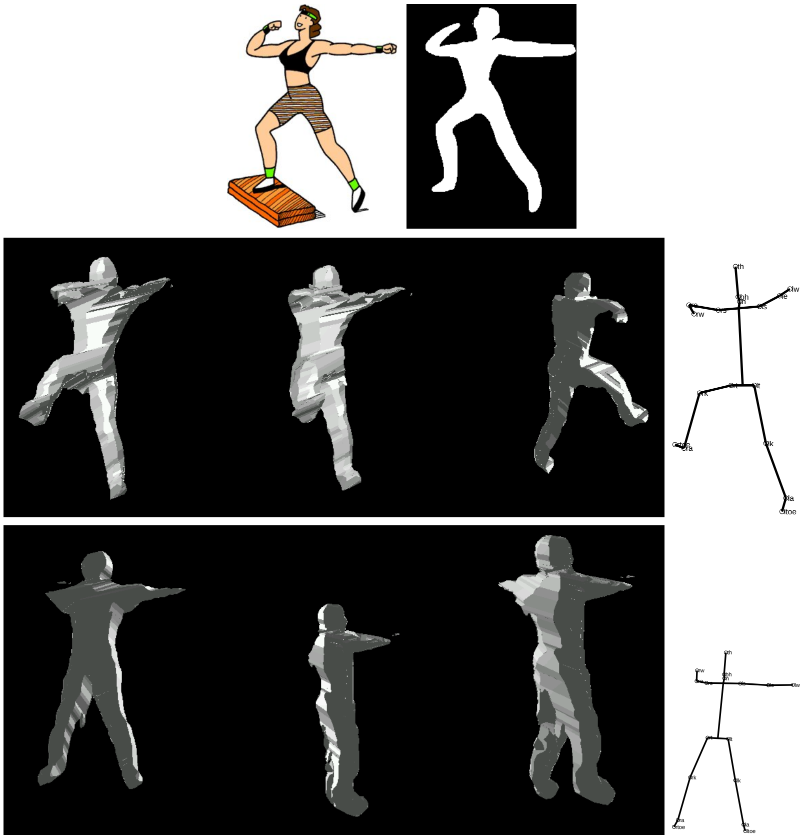


Figure 11: Example of inferred virtual visual hull hypotheses for a real image; plotted as in previous figure.

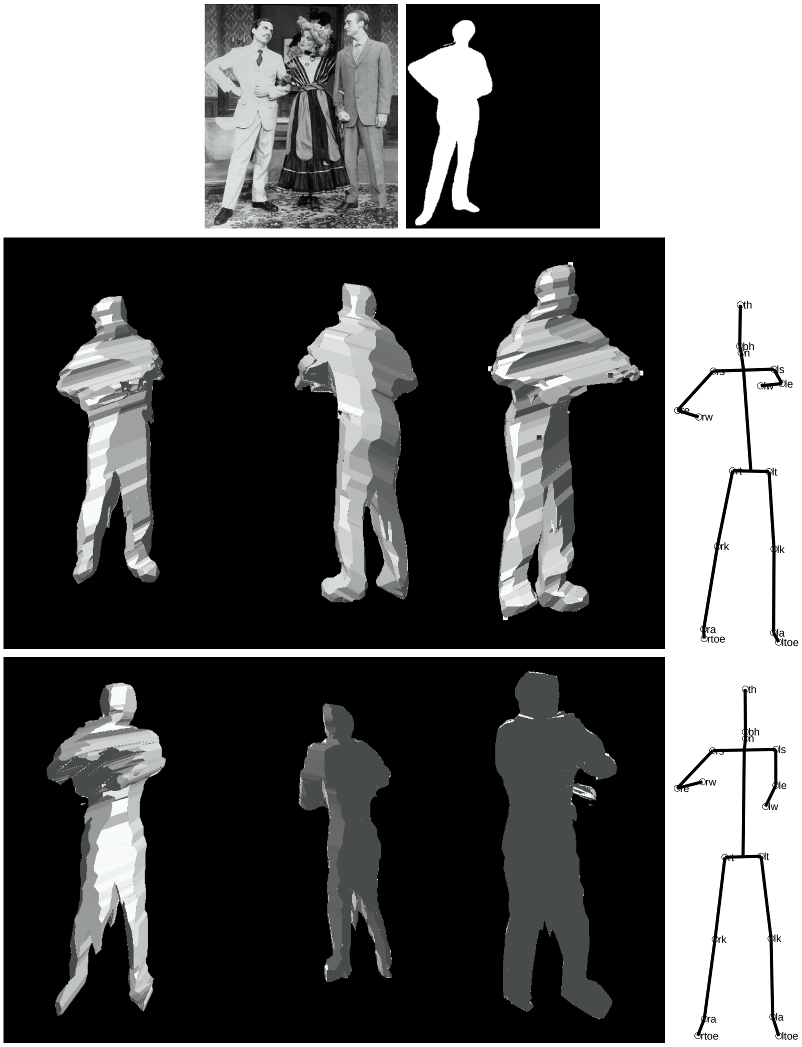


Figure 12: Example of inferred virtual visual hull hypotheses for a real image; plotted as in previous figure.

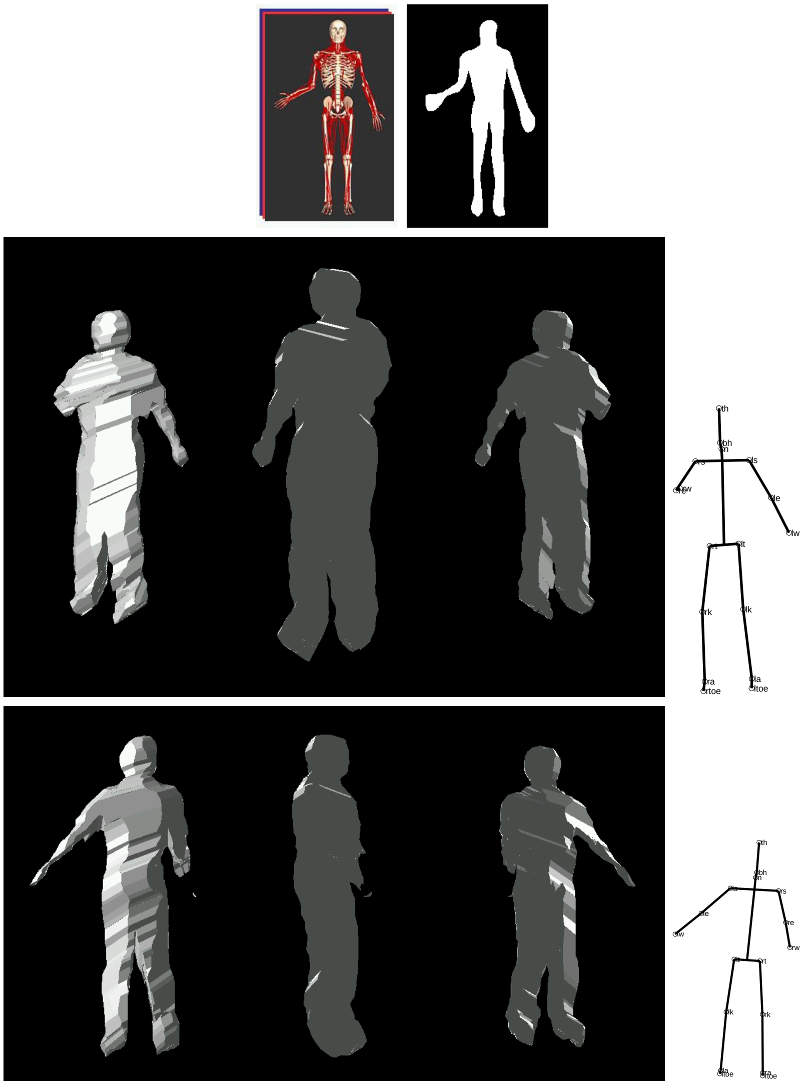


Figure 13: Example of inferred virtual visual hull hypotheses for a real image; plotted as in previous figure.

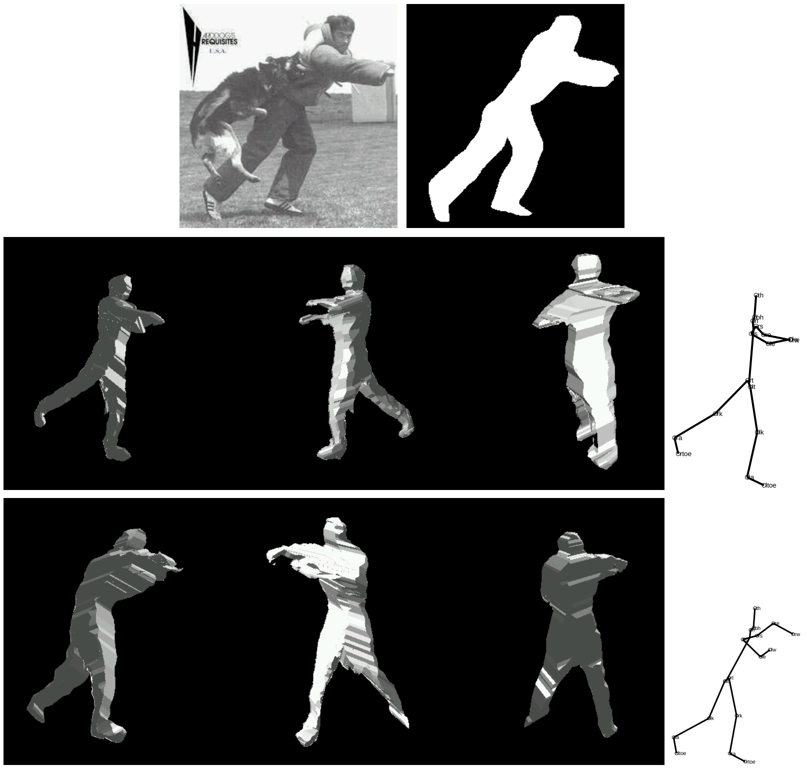


Figure 14: Example of inferred virtual visual hull hypotheses for a real image; plotted as in previous figure.