

Quantifying Ambiguities in Inferring Vector-Based 3D Models

Eng-Jon Ong and Shaogang Gong
Machine Vision Laboratory,
Department of Computer Science,
Queen Mary and Westfield College,
London, E1 4NS, UK
{ongej, sgg}@dcs.qmw.ac.uk

Abstract

This paper presents a framework for directly addressing issues arising from self-occlusions and ambiguities due to the lack of depth information in vector-based representations. Visual data directly observed from an image are used to indirectly recover the parameters of an underlying dynamic model of an articulated object. The proposed framework allows us to learn the ambiguities of a representation from training examples. The resulting model is then used to measure the ambiguities of each estimated underlying model parameter given the available visual information. This provides an indication of how much we can “trust” the visual data for estimating certain parts of the model. We then provide a working example of multi-view data fusion for tracking 3D skeletons of articulated objects in a multi-camera environment.

1 Introduction

Ambiguities are a constant cause of many problems in computer vision. It is especially true in tracking 3D articulated objects. In this work, we are interested in recovering the parameters of an underlying 3D articulated object from measurable 2D features in visual images. Examples includes location and orientation for the different parts of a 3D object. In general, 3D model parameters cannot be obtained directly from input images. However, we know that a 3D object does generate certain visual features which *can* be measured directly from an input image (e.g. its shape information given by edges). Therefore, the task is to recover the underlying 3D object’s data using measurable visual features.

However, the lack of depth information in visual images can cause serious problems. One problem is that of self occlusion where parts of an object are obscured by other parts, causing important visual features to be lost. Another problem lies in the inadequacy of 2D projections to uniquely represent 3D objects at certain poses, whereby the underlying 3D model at different poses generates very similar visual features. The aim of this paper is to address these problems by proposing a method to learn quantitatively when certain parts of the underlying 3D model cannot be accurately recovered from the visual information available images.

A common method for handling ambiguous information and self-occlusion is to use multiple cameras. This approach is popular in tracking articulated 3D objects such as

human bodies [5, 1]. The freedom for various parts of the human body to assume a great many poses increases the occurrence of self-occlusions and ambiguous poses. However, ambiguities caused by lack of depth information are generally not computed explicitly but indirectly handled by camera calibration along with 3D model fitting [10, 4, 8, 3]. The calibration process provides a mechanism for transforming a model consistently between different camera views. The ambiguities in model estimation are minimised by the fitting process since it considers information from all views. An alternative approach is given in [7] where a 3D human model’s projection is used to calculate the visibility of a body part. The view from which the body part is most visible is used assuming it contains the least ambiguous visual information.

In the rest of this paper, Section 2 introduces a hybrid-vector approach to recover the underlying 3D model’s parameters and provides a definition of what constitutes an ambiguous hybrid vector which can lead to unreliable estimation of 3D model parameters. Section 3 then details a framework for learning the ambiguities of each inferred component (3D skeleton joint angles in this context) given a set of measurable data. We also present a method for estimating the ambiguities of a novel hybrid-vectors using the ambiguity model in Section 3.3. A working example of tracking 3D skeletons models of articulated objects (e.g. human bodies or hands) using multiple camera views is given in Section 4. The learnt ambiguities are then used to aid the fusion of 3D skeleton estimations from individual views, yielding a more accurate overall 3D skeleton estimation. Preliminary results on the experiments are also given. Finally we conclude in Section 5.

2 Measuring Ambiguities in Vector-Based Representations

We would like to infer indirectly the parameters of an underlying 3D model using only visual features extracted from an image. One approach to achieve this is to combine both the visual feature components and its underlying 3D model parameters into a high dimensional *hybrid-vector* representation [2, 9].

Formally, we define (A) different types of visual features ($\mathbf{v}_1, \dots, \mathbf{v}_A$) as *measurable* data because it can be directly extracted from an input image. The vector, \mathbf{v}_i , with u_i number of components contains information on the visual feature it represents: $\mathbf{v}_i = \{v_{i,1}, \dots, v_{i,u_i}\}$. For example if \mathbf{v}_i represents a point distribution model (PDM) of a contour, its components would consists of the (x, y) coordinates of its points. We then concatenate all the visual vectors’ contents into a *measurement-data* vector, $\mathbf{w} = \{v_{1,1}, \dots, v_{1,u_1}, \dots, v_{A,1}, \dots, v_{A,u_A}\}$.

Next we define the *hidden-data* vector (\mathbf{m}) for storing the (B) underlying 3D model parameters: $\mathbf{m} = \{m_1, \dots, m_B\}$. Finally, we define the hybrid vector (\mathbf{y}) as the concatenation of the measurements-data along with its corresponding hidden-data: $\mathbf{y} = (\mathbf{w}, \mathbf{x})$.

A constraint model (volume or surface) can be constructed to capture valid instances of the measurable data (visual features) and its corresponding underlying 3D model components. Recovering the missing 3D model data given only input visual features can be achieved by finding the point on the constraint model whose visual feature components are closest to the given input data. This yields a vector which contains visual features closest to those recovered from the input image while containing the corresponding “hidden” 3D model’s parameters.

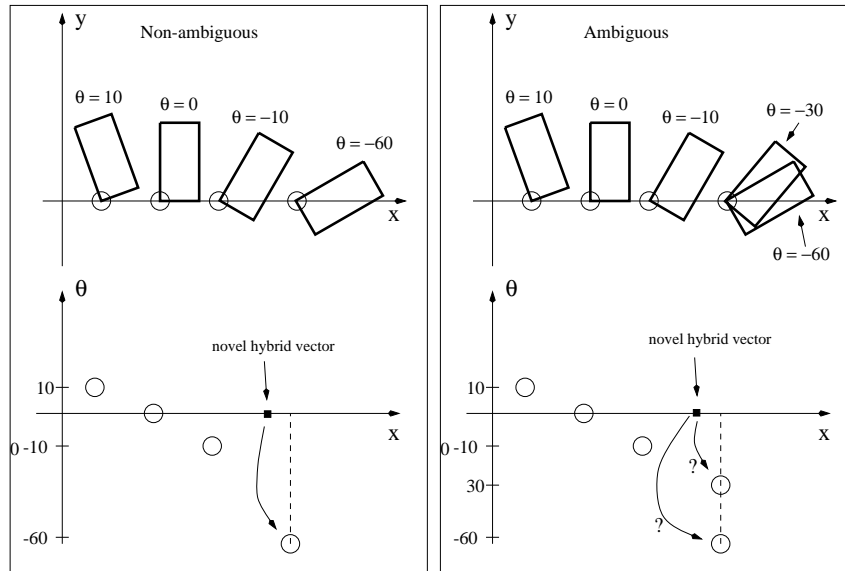


Figure 1: A simple illustration of model ambiguity situation. Here, we wish to recover the “hidden” rotation angle (θ) of the box. However, the only measurable component on the box is the corner with the circle. Moreover, the box can only be at 4 discrete x positions. This is shown on the top part of the figure. We can represent this system with a 2D hybrid vector (x, θ). The bottom part shows the constraint model: a set of points representing valid instances of the box parameters. In the unambiguous case on the left, given a novel “invalid” hybrid vector as indicated by the black square, we move it to the point on the constraint model with the x value closest to the novel vector’s x value. However, when it is possible for the box to assume two different poses at the same position, as shown on the right, ambiguities occur. We no longer know to which point on the constraint surface should the novel hybrid vector be moved to.

Ambiguous and self-occluded visual features can cause multiple points on the constraint surface and have measurement-data equally similar to those extracted from the input image, but each with significantly different corresponding hidden-data components [9]. As a result, it is not possible to decide which 3D model parameters can be selected for the given visual features. A simple example illustrating this can be seen in Fig. 1. In other words, a hybrid vector has ambiguous measurable components when there exists many hybrid vectors with similar measurable components but dis-similar inferred components. In the next section, we describe a method to quantify this problem through learning.

3 Learning the Ambiguities

Having defined the characteristics of a hybrid-vector example with ambiguous measurements, we now introduce a two step method for learning the ambiguities of this hybrid representation: (1) Extracting the ambiguities of the visual feature components of the training data (Section 3.1). (2) Modelling the ambiguity values (Section 3.2).

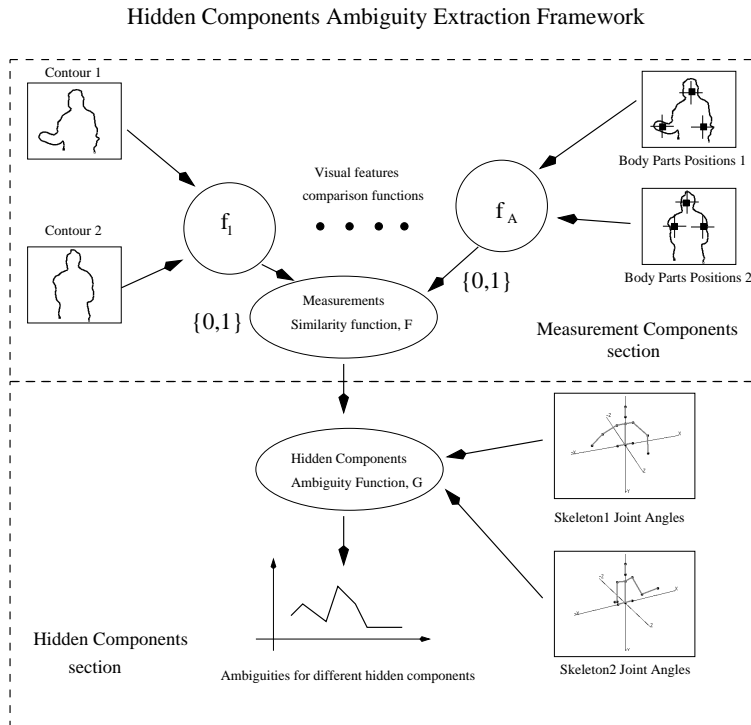


Figure 2: An overview diagram of the ambiguities extraction method described in Section 3.1

3.1 Extracting the Ambiguities from Training Data

Here we describe a method for extracting the ambiguity values for each hidden components of the representation using its corresponding measurement components. As illustrated in the diagram given in Fig. 2, the method consists of two components; the *measurements similarity function* and the *hidden components ambiguity function*.

We first describe the measurements similarity functions. As defined in Section 2, the measurement vector consists of a number (A) of visual feature vectors ($\mathbf{v}_1, \dots, \mathbf{v}_A$). In order to compare two sets of visual features, we introduce a set of functions (f_1, \dots, f_A) for measuring the similarities between two instances of a set of (A) visual features. The i^{th} visual feature similarity function ($f_i(\mathbf{v}_i^1, \mathbf{v}_i^2)$) is a mapping $f : \mathbf{R}^{u_i} \times \mathbf{R}^{u_i} \rightarrow \{0, 1\}$, where u_i is the number of components in \mathbf{v}_i . This mapping is responsible for comparing two visual feature instances ($\mathbf{v}_i^1, \mathbf{v}_i^2$). This similarity function depends on the visual features being compared. Each function returns 1 when the visual feature instances ($\mathbf{v}_i^1, \mathbf{v}_i^2$) being compared are deemed similar and 0 if not similar enough. An example can be seen in Section 4 for comparing the similarities of an articulated object's silhouette contours.

These individual visual feature similarity functions together define a *measurement-similarity function*,

$$\mathcal{F}(\mathbf{w}^1, \mathbf{w}^2) = \prod_{i=1}^A f(v_i^1, v_i^2) \quad (1)$$

where the a^{th} measurement vector is defined as $\mathbf{w}^a = (\mathbf{v}_i^a, \dots, \mathbf{v}_i^a)$ and $a \in \{1, 2\}$. \mathcal{F} returns 1 if all visual features in two instances of the measurement vector are similar enough.

We now describe the second function responsible for measuring the ambiguities of the hidden components given the measurements. In the case where the measurements are similar, as determined by, \mathcal{F} , we check for significant differences between its corresponding hidden components $(\mathbf{x}^1, \mathbf{x}^2)$. To do this, we introduce a function $\mathcal{G}(\mathbf{x}^1, \mathbf{x}^2)$ where $\mathcal{G} : \mathbf{R}^B \times \mathbf{R}^B \rightarrow \mathbf{R}^B$. This function provides a vector which indicates to what degree each of the hidden components differ from one another, given the corresponding measurements $(\mathbf{w}^1, \mathbf{w}^2)$. Function \mathcal{G} , depends on what the hidden components represent. An example of this function is given in Section 4 for comparing two instances of 3D skeleton joint angles and determining to what degree they differ and thus to what degree they are ambiguous.

Now, we introduce an algorithm for extracting the hidden components' ambiguity values from the examples in a set of N training hybrid-vectors $(\{\mathbf{y}^1, \dots, \mathbf{y}^N\})$.

Initialisation Step

- Make N number of B -dimensional vectors $(\{\mathbf{c}_1, \dots, \mathbf{c}_N\})$ for storing the ambiguity values for the hidden-data components for each training example.
- Initialise all the components of \mathbf{c}_c to 0, where $c \in \{0, \dots, N\}$.

Ambiguities Extraction Loop

- For each training example $\mathbf{y}^a = (\mathbf{w}^a, \mathbf{x}^a)$, where $a \in \{1, \dots, N\}$,
 - For each of the other training examples, $\mathbf{y}^b = (w^b, x^b)$, where $b \in \{1, \dots, b-1, b+1, \dots, N\}$,
 - if($\mathcal{F}(\mathbf{w}^a, \mathbf{w}^b) == 1$),
 - Evaluate ambiguity values (y) between hidden components, \mathbf{x}^a and \mathbf{x}^b ,

$$y = \mathcal{G}(\mathbf{x}^a, \mathbf{x}^b)$$
 - Update the ambiguity values for example a ,

$$c_{a,j} = y_j, \text{ if } y_j > c_{a,j}, \text{ where } j \in \{1, \dots, B\}$$

This extraction process results in a set of vectors containing the ambiguity measures for the hidden components in each example. Having associated all the training data with their appropriate ambiguity measures, we next describe a method for modelling such ambiguities and labelling novel visual measurements.

3.2 Modelling the Ambiguities

Having extracted the ambiguities from the training data as described above, let us now construct a new ‘‘ambiguity training set’’, $\{\mathbf{d}_1, \dots, \mathbf{d}_N\}$. We replace the hidden components (\mathbf{x}_i) in each training example (\mathbf{y}_i) with its corresponding ambiguity vector (\mathbf{c}_i) ; $\mathbf{d}_i = (\mathbf{w}_i, \mathbf{c}_i)$. The resulting training example is defined as an *measurement-ambiguity vector*. We then model the space taken up by the measurement-ambiguity training examples using Hierarchical Principal Components Analysis (HPCA) [6]. Spatially, this results in a hierarchical structure containing a set of global eigenvectors spanning the subspace of the training data. We define the subspace modelled by the global eigenvectors as the *global eigenspace*. Localised clusters are then used to account for potential

non-linear structures in the training data within the global eigenspace. Conceptually, the HPCA structure captures known valid examples while allowing for in-between-example generalisations to take place. Additionally, it inherently models the correlations between visual measurements and the ambiguity values of its corresponding hidden components.

3.3 Estimating Ambiguities of Novel Hybrid Vectors

We can now use the HPCA model to estimate the ambiguities of a novel hybrid vector captured from visual input similar to [6, 2, 9]. We extract the visual measurements before forming a potentially “invalid” measurement-ambiguity vector defined above, by concatenating either a zero vector or a previously estimated ambiguity vector to the end of the measurement vector.

We project this new ambiguity vector into the global eigenspace, effectively reducing its dimensionality. In the global eigenspace, we determine whether this dimensionality-reduced ambiguity vector falls inside one of the clusters by projecting it onto the principal components of the clusters. It is contained within a cluster if all the projections are less than the eigenvalues of the principal components’. If true, do nothing.

However, if the projection lies outside all the clusters, we first locate its nearest cluster using Euclidean distance. The projected ambiguity vector is moved into the closest cluster by projecting onto the cluster’s principal components. The magnitudes of all projections are limited to fall within the eigenvector’s corresponding eigenvalue. Secondly, we obtain the newly moved projected ambiguity vector by taking a linear combination of the cluster’s principal components, with the eigenvalue-limited projections being the coefficients. This will provide us with a vector which has valid ambiguity values for corresponding measurements which are closest to that of the novel hybrid vector.

4 Application: Tracking 3D Skeleton Models using Multiple Views

We adopt a hybrid representation consisting of a combination of measurable 2D image features together with the underlying 3D model parameters for tracking 3D skeletons [2, 9]. Here, the 2D image features used are PDMs for representing the object’s contour. Following the terminology introduced in Section 2, we have a set of visual feature vectors, \mathbf{v}_1 (i.e. $A = 1$). These vectors represent the visual measurements which can be directly extracted from the image.

We define the PDM of the object’s silhouette contour to be a vector (\mathbf{v}_1) containing the coordinates of a number (u_1) of evenly distributed 2D points; $\mathbf{v}_1 = (x_1, y_1, \dots, x_{u_1}, y_{u_1})$.

We define the hidden components vector (\mathbf{x}) to contain $2u_2$ joint angles for a 3D skeleton with u_2 number of joints; $\mathbf{x} = (\theta_1, \phi_1, \dots, \theta_{u_2}, \phi_{u_2})$. Each joint contains two angles, θ and ϕ , which represents the angles of the joint off its local x and z axes respectively (see Fig 3).

However, self occlusions together with the lack of depth information can give rise to situations where very similar contours can correspond to significantly different 3D skeletons. Therefore, to increase the tracking robustness, the ambiguity model defined in Section 3.2 is used as a mechanism for a multiple camera setup. The ambiguity model can allow us to measure the potential accuracy of each component of the estimated 3D

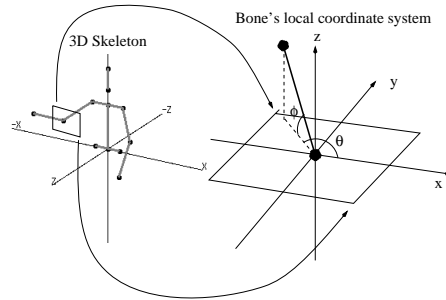


Figure 3: An illustration of the 3D skeleton joint angles, θ and ϕ in the local (x, y, z) coordinate system of a joint.

skeleton in each view. Consequently, a more accurate 3D skeleton can be found by using the least ambiguous estimation for each of its components.

In order to use the framework for learning the ambiguities described in the previous section, let us now define the similarity function for the measurable data (2D components) of the hybrid vector (Section 4.1) and the ambiguity function for the hidden components, i.e. the underlying 3D skeleton's joint angles (Section 4.2). Finally, we introduce a method for using the ambiguity model to estimate the least ambiguous 3D skeleton, where each view contributes the least ambiguous estimations (Section 4.3).

4.1 Similarity Functions for 2D Measurements

We now define the similarity functions for the hybrid representation's two measurable-data sub-group: the body contour. Given two PDMs representing the body contours, \mathbf{v}_1^1 and \mathbf{v}_1^2 , both vectors having $2u_1$ number of components (i.e. u_1 number of 2D points), we define the similarity function as a sum of distances between all the corresponding points throughout the entire contour:

$$f_1(\mathbf{v}_1^1, \mathbf{v}_1^2) = \begin{cases} 1 & \text{if } d_1(\mathbf{v}_1^1, \mathbf{v}_1^2) \leq t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$d_1(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{u_1} \sqrt{(a_{2i} - b_{2i})^2 + (a_{2i+1} - b_{2i+1})^2} \quad (3)$$

where $\mathbf{a} = (a_1, \dots, a_{2u_1})$, $\mathbf{b} = (b_1, \dots, b_{2u_1})$ and the preset value (t) represents how close all the corresponding points on two contours must be before they are considered similar.

4.2 Ambiguity Function for the Skeleton Joint Angles

Given two joint angles of an articulated object's 3D skeleton, we define them to be similar if they are both within a preset range (γ) of each other. This preset range determines the coarseness of the 3D skeleton's joint angles estimation. Formally, the similarity function for comparing two corresponding 3D skeleton joint angles sets, \mathbf{x}^1 and \mathbf{x}^2 , is given as

$$\mathcal{G}(\mathbf{x}_1, \mathbf{x}_2) = (d_3(x_1^1, x_1^2), \dots, d_3(x_B^1, x_B^2)) \quad (4)$$

$$d_3(\theta_1, \theta_2) = \begin{cases} |\theta_1 - \theta_2| & \text{if } \theta_1 + \gamma > \theta_2 > \theta_1 - \gamma \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where, $\mathbf{x}^1 = (x_1^1, \dots, x_B^1)$, $\mathbf{x}^2 = (x_1^2, \dots, x_B^2)$ and B is the number of joint angles of the two skeletons compared.

4.3 Tracking 3D Skeletons in Multiple Views

We perform tracking of the 3D skeletons using the framework described in [9]. Each view provides a viewpoint invariant estimation of the current 3D skeleton. Additionally, we estimate the associated ambiguities for each 3D skeleton component given its associated 2D measurements for each view. To do this, we form a vector consisting of the 2D measurements and the ambiguity vector of the previous time-step for that view. We then use the ambiguity model described in Section 3.2 to recover the correct associated ambiguities, as described in Section 3.3. Consequently, for each view, we have both the 3D skeleton's estimation along with the ambiguity values for each of its estimated components. To form the least ambiguous (and therefore most reliable) skeleton estimation, for each of its components, we use the estimation from the view which has the smallest ambiguity value.

4.4 Experiments

For our experiments, we have chosen the hand as the object of interest. A training set of 450 hand images at different poses was acquired. A PDM with 50 points was chosen to represent the hand contour. The corresponding underlying 3D skeleton of the hand was then obtained manually. Next, the joint angles of each skeleton were recovered. The algorithm described in Section 3.1 was then used to recover the ambiguities of the joint angles of each training example. Preliminary results on labelling the 3D skeletons with the ambiguity measurements are shown in Fig.4. Here, we show hand poses at three levels of ambiguities; unambiguous poses, partially ambiguous poses and highly ambiguous poses. The first type consists of hand poses whose corresponding contour is uniquely associated with only one skeleton configuration. Next, the partially ambiguous hand pose is one whose corresponding contour can bring about ambiguous angle estimations for certain joints while unambiguous angle estimations for other joints. Finally, the highly ambiguous hand pose is one whose contour can be associated with many skeleton configurations with very different joint angle configurations across all the different finger joints of a hand. This makes the joint angle estimation highly ambiguous.

5 Conclusion

In this paper, we formally addressed quantitatively the general problem of ambiguities in using 2D visual data for recovering underlying 3D model parameters. To this end, we proposed a framework for learning the ambiguities of hybrid-vector representations, whereby measurable data and "hidden-data" (i.e. model parameters) are combined together. This is achieved by extracting ambiguities of the hybrid-vector's hidden components given their corresponding visual measurements. The computation is based on a definition that model ambiguities are measured by the degree of which similar measurable components (e.g.

visual features) give rise to significantly dis-similar underlying model parameters. We then model the ambiguity values of the underlying model parameters using HPCA, allowing a novel hybrid-vector to have its underlying parameter's ambiguity values estimated quantitatively.

We have described an example of applying this framework to learning the ambiguity model for 3D hand skeletons inferred using 2D contours. This can then be used in a multi-camera setup to fuse estimated 3D skeletons from different views. Here, the 3D skeletons are estimated using each view's 2D visual measurements. The ambiguities of the resulting 3D skeleton estimations are computed. A more robust 3D skeleton is found, whereby each of its components is the result of the least ambiguous estimation.

References

- [1] J.K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding*, 73(3):428–440, March 1999.
- [2] R. Bowden, T. Mitchell, and M. Sarhadi. Reconstructing 3d pose and motion from a single camera view. In *BMVC*, pages 904–913, Southampton, 1998.
- [3] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *Proc. of the IEEE International Conference on Computer Vision*, pages 716–721, September 1999.
- [4] D. Gavrilu and L. Davis. 3d model based tracking of humans in action: a multi-view approach. In *Proc. of IEEE CVPR*, San Francisco, 1996.
- [5] D.M. Gavrilu. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, January 1999.
- [6] T. Heap and D. Hogg. Improving specificity in pdms using a hierarchical approach. In *BMVC*, pages 80–89, Essex, UK, September 1997.
- [7] I. Kakadiaris and D. Metaxas. Model-based estimation of 3d human motion with occlusion based on active multi-viewpoint selection. In *CVPR*, San Francisco, June 1996.
- [8] A. Sato M. Yamamoto and S. Kawada. Incremental tracking of human actions from multiple views. In *CVPR*, 1998.
- [9] E. Ong and S. Gong. A dynamic 3d human model from multiple views. In *British Machine Vision Conference*, pages 33–42. BMVA, September 1999.
- [10] J.M. Regh. *Visual Analysis of High DOF Articulated Objects with Application to Hand Tracking*. PhD thesis, Carnegie Mellon University, April 1995.

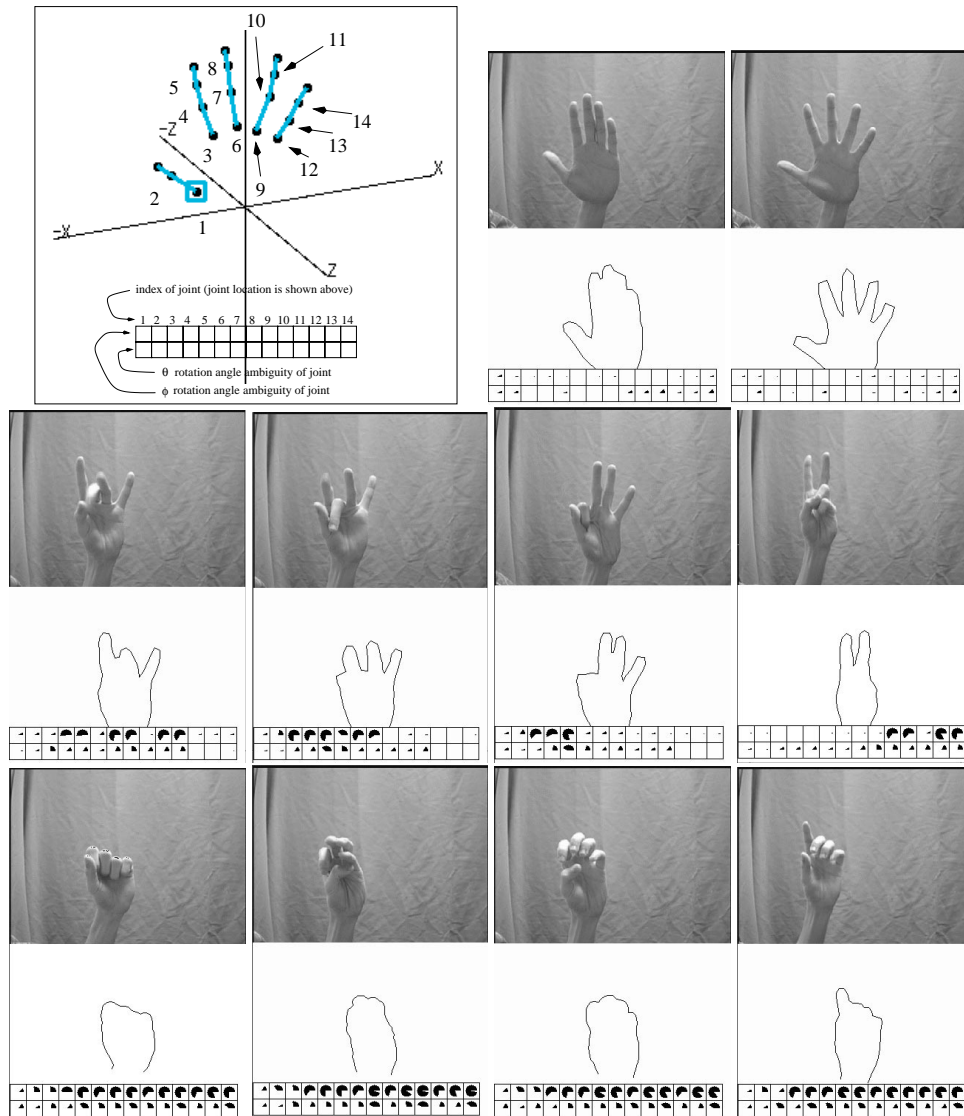


Figure 4: This figure shows different levels of ambiguous poses of a hand. The top left image shows the index for the joints of the skeleton along with the location of a joint's ambiguity measurements display box. For each hand pose, the input image is shown on the top, the PDM contour on the middle and the joint angle ambiguity measure is shown in the two rows of boxes below the contour. The top and bottom row of boxes indicate the ambiguities for the joint angles (ϕ) and (θ) respectively. In each box, the angle ambiguity magnitude is shown as a filled arc (e.g. a quarter circle indicates the existence of other hand poses with similar contours but a different corresponding joint angle, where the magnitude of the angle variation is 90 degrees). The images on the top right shows hand poses which are unambiguous. The middle row shows partially ambiguous hand poses. The bottom row shows hand poses which will provide highly ambiguous and unreliable contours.