Object Pose Detection in Range Scan Data

Jim Rodgers, Dragomir Anguelov, Hoi-Cheung Pang, Daphne Koller
Computer Science Department
Stanford University, CA 94305
{jimkr, drago, hcpang, koller}@cs.stanford.edu

Abstract

We address the problem of detecting complex articulated objects and their pose in 3D range scan data. This task is very difficult when the orientation of the object is unknown, and occlusion and clutter are present in the scene. To address the problem, we design an efficient probabilistic framework, based on the articulated model of an object, which combines multiple information sources. Our framework enforces that the surfaces and edge discontinuities of model parts are matched well in the scene while respecting the rules of occlusion, that joint constraints and angles are maintained, and that object parts don't intersect. Our approach starts by using low-level detectors to suggest part placement hypotheses. In a hypothesis enrichment phase, these original hypotheses are used to generate likely placement suggestions for their neighboring parts. The probabilities over the possible part placement configurations are computed using efficient OpenGL rendering. Loopy belief propagation is used to optimize the resulting Markov network to obtain the most likely object configuration, which is additionally refined using an Iterative Closest Point algorithm adapted for articulated models. Our model is tested on several datasets, where we demonstrate successful pose detection for models consisting of 15 parts or more, even when the object is seen from different viewpoints, and various occluding objects and clutter are present in the scene.

1. Introduction

The detection of articulated objects and their pose is a difficult problem with multiple practical applications, including human pose detection, intelligent interpretation of surveillance and other video data, advanced user interfaces, model-based coding of gestures, and more (see, for example, [10] for a survey). Not surprisingly, the bulk of the work on this topic has focused on this problem in the context of 2D images. However, with the increasing availability of 3D sensors (including laser range finders and stereo-based sensors), a treatment of this problem in 3D scenes is becoming more and more relevant. As illustrated in Fig. 1, our goal in this paper is to identify the pose of an articulated 3D model within a range scan of a complex, cluttered scene, even when significant portions of the model are occluded from the field of view.

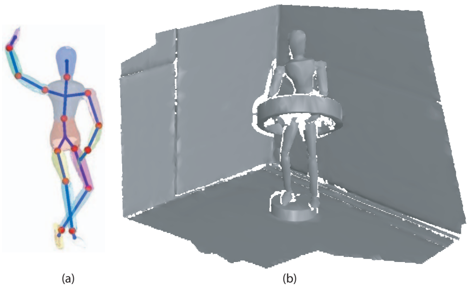


Figure 1. (a) An articulated object model consists of rigid parts, whose motion is constrained by joints. Our goal is to detect the model pose in (b) range scans containing clutter and occlusion.

Our algorithm takes as input an articulated 3D object model, made up of rigid parts whose motion is constrained by joints, as shown in Fig. 1(a). Given a single range scan of a scene (Fig. 1(b)), our goal is to find a consistent placement of the model within that scene. This involves obtaining a set of rigid transformations that place the model parts in the scene, and respect the joints.

Following Ioffe and Forsyth [11], our method uses detectors to provide initial hypotheses for the placement of each object part. However, due to occlusion and the lack of strong identifying features, it is surprisingly difficult to identify, with any accuracy, individual object parts within the scene. Some parts may have a very large number of candidate placements, whereas others may not be detected even by the best available methods. Therefore, the set of conceivable placements of the articulated object within the scene is combinatorially large.

We address this issue by using a probabilistic Markov network model, which defines a joint distribution over the placement of the individual parts. This model incorporates a range of (soft) constraints: that the surfaces and edge discontinuities of model parts are matched well in the scene and part placements respect the rules of occlusion, that joint constraints and angles are maintained, and that object parts do not intersect. We can then use loopy belief propagation [18, 16] to find the most likely object configuration in this Markov network. Importantly, the elements in the distribution associated with possible part placement configurations can be computed efficiently using OpenGL rendering, greatly speeding up the inference.

However, this solution by itself is insufficient; as we discussed, for some parts, the correct placement may not be in the initial set of hypotheses at all. We therefore utilize a novel hypothesis generation procedure, which uses the articulated model structure to produce new hypotheses for a part, given the current hypotheses about its neighbors.

We test our method on two challenging data sets, containing multiple poses of a puppet and a human, whose articulated models consist of 15 parts or more. The scans are obtained from different views, and contain various types of clutter and occluding objects. Despite the difficulty and variety in this data set, we show that our method is capable of recovering the object pose correctly in many cases. Using a third, synthetic data set, we show that we can successfully place parts with a low error relative to the ground truth.

2. Related Work

Many researchers avoid the complexity of dealing with explicit part-based models, and learn the mapping between appearance and pose directly [25, 15, 24]. Such approaches deal with the combinatorial explosion of possible poses, object scale and scene clutter by requiring vast amounts of training examples. The poses of new examples can be found by nearest-neighbor search in high dimensions [20, 4], or by training discriminative regressors [1, 22]. The drawback with these approaches is that tens of thousands to millions of examples are needed in order to obtain good predictions.

The bulk of the work on pose detection has focused on detecting 2D models in 2D image data, typically modeling the body as an assembly of 2D parts [8, 11, 26]. Most frequently, these approaches use a tree-shaped graphical model to capture the geometric constraints between the parts, which allows efficient and exact dynamic programming methods to be applied. The hypotheses provided to these methods are usually obtained from low-level detectors [26, 11], or by enumerating all possible configurations for pairs of adjacent parts, which is feasible in 2D, as shown by Felzenszwalb and Huttenlocher [8]. More recent work [6, 19] relaxes the tree-shaped constraint graph assumption, which allows the model to discourage cases when non-adjacent parts are overlapping. The resulting optimization problem can be solved efficiently with approximate methods such as loopy belief propagation [14, 12] and constrained integer programming [19]. A key problem with 2D object models is that they are strongly view and scale-dependent, and are prone to fail if the viewpoint relative to the object changes.

In 3D, the problem becomes more difficult, since the dimensionality of the search space increases significantly. Because of this, current work on 3D articulated pose detection has focused mainly on tracking, where temporal information constrains the space of reasonable pose hypotheses. For example, Sigal *et al*. [21] use a sampling-based version of belief propagation for tracking the human body in video, and a similar work by Sudderth *et al*. [23] is applied to the problem of hand tracking. Their techniques can in principle be applied to the problem of pose detection, but are less efficient than our approach, and in the absence of good initial hypotheses for all body parts, the sampling process may take a very long time or fail to converge altogether.

There has been less work on object and pose detection in 3D range data, which has become readily available only recently. Most of the existing methods for range data focus on devising efficient descriptors for detecting rigid objects [13, 9]. Algorithms for dealing with more complex or deforming objects usually make simplifying assumptions. For example, the method of Anguelov *et al*. [3] makes the registration of deformable 3D articulated objects tractable by assuming that there is no clutter in the scene. This is an overly strict assumption in most practical cases.

3. Probabilistic Model

Our goal is to detect the placement of all articulated model parts in range scans containing that model. We define a joint probability distribution over the possible configurations of the articulated object using a *Markov network* [18]. In this network, we have a variable for every part, whose domain consists of the possible configurations for that part; these hypotheses are obtained by a mechanism described in Section 5. The network contains *local part potentials*, which evaluate the match between the part placements and the observed data — they favor part placements where the surface and edges are consistent with the observed scene. It also contains *part interaction potentials*, which reflect the constraints between the placements of pairs of parts in the articulated object; these favor part placements where parts do not intersect, distances at the joints are respected, and angles around joints are appropriate.

3.1. Local Part Potentials

Each variable in our Markov network encodes the possible location hypotheses of a particular object part. For part p, we consider the location hypotheses $h_1^p, \dots, h_K^p$, where each one corresponds to a transformation placing the part somewhere in the scene. The *local part potential* represents

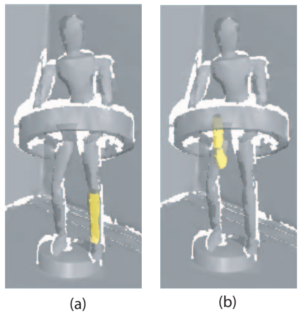


Figure 2. Area explanation/occlusion score prefers placement of parts that are (a) well aligned with surface as opposed to (b) those that occlude surface.

the likelihood of a part placement. For each hypothesis k of part p, this potential consists of two scores:

$$\phi(h_k^p) \quad = \quad \phi_{\textit{surface-match}}(h_k^p) \cdot \phi_{\textit{edge-match}}(h_k^p), \qquad (1)$$

where $\phi_{\textit{surface-match}}$ and $\phi_{\textit{edge-match}}$ quantify how well the model surfaces and edges match those in the scene. These scores are described below.

3.1.1 Surface match score

In defining this score, we assume that the range scanner provides surface connectivity information in addition to the depth readings. This is not a restrictive assumption, as such information is provided by many scanners [7], or can be obtained by post-processing methods [17].

The surface match score favors placements of parts for which the *expected part surface* — the surface we would expect to see given the part placement — aligns with the observed surface. In addition, it penalizes parts that protrude in front of the observed surface, thereby occluding portions of the scanned scene. Fig. 2 shows an example of a good and a bad scoring placement of a lower leg part. The example in Fig. 2(b) represents a not-so-likely lower leg match, as the expected surface aligns poorly with the range scan surface, and even occludes some of it.

To define the surface match score, we discretize the part surface. For each point i on that surface (transformed according to the hypothesis h_i^k), we focus on the line connecting it to the scanner viewpoint. If this line intersects the observed surface, we examine the distance d_i between the expected surface and the observed surface points on the line. If d_i is small, we allow observed surface points to be explained away using a 0-mean Gaussian distribution with variance σ_s^2. If the observed surface is far in front of the expected surface, we assign a moderate uniform probability α, as there is a reasonable chance that the part is simply behind some other object in the scene. Thus, for observed points in front of expected points, we define $\delta_i = \max(\alpha, \exp(\frac{-{d_i}^2}{2{\sigma_s}^2}))$. When the observed surface is behind the expected surface, we define $\delta_i = \max(\beta, \exp(\frac{-{d_i}^2}{2{\sigma_s}^2}))$. Here the low uniform probability $\beta < \alpha$ accounts for the unlikely case that we observe some surface behind the part, despite our expectation that it should be occluded by the part. Finally, we have to deal with cases where the expected surface falls in a region where no surface was observed in the scan. We cannot assume there is actually no surface there, as the lack of readings could be due to physical limitations of the scanner. Thus, we assign a moderate uniform probability $\delta_i = \gamma$ to part points that fall within such regions, representing the reasonable likelihood of observing no surface where a part is placed in the scene.

The total score of a part then is the product of the point scores weighted by the expected area of the points:

$$\phi_{\textit{surface-match}}(h^p) = \prod_{i \in p} {\delta_i}^{area(i)}. \qquad (2)$$

This formulation is an approximation, which assumes different points within the same part are independent, and ignores the possible correlations arising from the fact that neighboring points tend to be occluded by the same object.

The surface match score is implemented using OpenGL, which allows accurate calculation of surface occlusion that can not be obtained from 3D point information alone. Furthermore, we are able to leverage hardware accelerated 3D rendering in the graphics card to score several hundred part hypotheses per second. We do this by first rendering the range scan in OpenGL, as seen from the scanner viewpoint. Then each object part that we wish to score is rendered separately. The graphics engine provides an automatic discretization of the scene into pixels. We can compare the Z-buffer values between the part and the range scan to determine where the two surfaces lie relative to each other. This is sufficient for estimating the surface match score.

3.1.2 Edge match score

In our formulation, we use the term edges to refer to depth discontinuities in the scene. The edge match score reflects the idea that scene and part edges should align. The edges in the scene can be computed efficiently by rendering it using OpenGL and applying the well-known Canny edge detection algorithm on the OpenGL Z-buffer image. We can identify the depth discontinuities of the object parts by rendering the expected part surfaces in the same manner.

For each point along the edge of the expected part surface, we find the closest edge point in the observed surface. We prefer that the distance e_i between these points is small, which is enforced using a 0-mean Gaussian distribution over e_i with variance ${\sigma_e}^2$. Because of noise and missing surface in the range scans, we assign a uniform probability γ_e in then cases when e_i is large, resulting in the following score for an individual edge point i:

$$\epsilon_i = \max(\gamma_e, \exp(\frac{-{e_i}^2}{2{\sigma_e}^2})). \quad (3)$$

As before, to obtain the score for the entire part, we weight the score of each point by its size (here, its segment length):

$$\phi_{\textit{edge-match}}(h^p) = \prod_i {\epsilon_i}^{\textit{length}(i)}. \quad (4)$$

3.2. Articulated Object Prior

Our probabilistic model contains additional potentials between pairs of articulated model parts, which are responsible for enforcing the articulated model constraints. For two parts p and q, we create a potential $\psi(h_i^p, h_j^q)$ for a joint configuration where part p is transformed by hypothesis i and part q by hypothesis j:

$$\psi(h_i^p, h_j^q) \quad = \quad \psi_{\textit{joint}}(h_i^p, h_j^q) \cdot \psi_{\textit{intersection}}(h_i^p, h_j^q). \quad (5)$$

The joint consistency score $\psi_{\textit{joint}}$ and the part intersection score $\psi_{\textit{intersection}}$ are described in more detail below.

3.2.1 Joint consistency score

In a connected articulated model, two neighboring parts should meet at their shared joint. For the hypothesis h_i^p, we define $v_i^{(p\to q)}$ to be location of the (p, q) joint point according to h_i^p; we define $v_j^{(q\to p)}$ correspondingly for h_j^q. The distance between these two points is simply:

$$d_{i,j}^{(p,q)} = \|v_i^{(p\to q)} - v_j^{(q\to p)}\|_2. \quad (6)$$

We then introduce a Gaussian distribution over $d_{i,j}^{(p,q)}$, with mean 0 and variance σ_j^2.

Similarly, we introduce a prior over the angle between two adjacent parts, to bias against configurations where neighboring parts are twisted in ways that are inconsistent with the restrictions of the joint. The configuration of each of the two parts determines a 3-dimensional vector describing one part's rotation in the scene, relative to the position of the joint with the neighboring part. We compute the angle $\theta_{i,j}^{(p,q)}$ between the two parts using a dot-product operation. We then compare that angle to the angle $\tilde{\theta}^{(p,q)}$ defined by the model. To avoid bias for normal motion around the joint, we use a uniform potential for cases where $|\theta_{i,j}^{(p,q)} - \tilde{\theta}^{(p,q)}|$ is less than some threshold t; outside that range, we use a Gaussian distribution with mean 0 and variance ${\sigma_l}^2$.

3.2.2 Part intersection score

Finally, our model enforces that in the detected pose, different parts do not overlap. In practice, we encounter cases where both the upper left and right leg are explaining the same observed surface. To prevent this, we need to introduce potentials between *all pairs* of parts, which favor object poses in which there is little overlap between the parts.

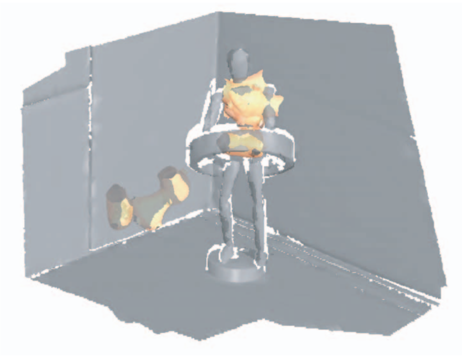


Figure 3. Top scoring placement hypotheses for upper torso obtained by using spin-image detectors.

Our part intersection score is based on the amount of overlapping volume between two parts. We discretize the part volumes using a 3D grid, and for each pair of parts compute the total number of grid points that lie in the intersection of their volumes. We score the amount of intersected volume using a Gaussian distribution with mean 0 and variance ${\sigma_v}^2$.

4. Probabilistic Inference

The probabilistic model in the previous section defines a joint distribution over the possible placements of the articulated object in the scene. Our task is to find the joint assignment whose probability is highest. In other words, we want to find the most likely (MAP) assignment in the pairwise Markov network defined above. Several algorithms exist for solving the MAP inference problem. We chose to use loopy belief propagation (LBP) [18], which has been shown to work effectively in a broad range of applications.

As defined, the pairwise Markov network includes a potential over all pairs of parts in the articulated model, representing the constraint that no two parts can overlap. Thus, we have a densely connected graphical model, with a number of edges that grows quadratically with the number of parts. This makes the inference process quite costly, and can also lead to non-convergent behavior of LBP.

Fortunately, the intersection potentials are not necessary between most non-adjacent parts, as it is rare that a configuration where they intersect has a high score. Thus, we define an incremental inference process (also proposed in [3]), where we begin by including in the model only the intersection potentials for adjacent parts. We then run inference, and examine the resulting MAP assignment. We introduce intersection potentials only for pairs of parts that are overlapping in the MAP assignment, and then re-run inference. This process is repeated until no parts intersect in the MAP assignment. Note that, in some cases, inference may fail to converge once an intersection potential is added. In this case, we consider each of the two conflicting parts in turn, fix its value to the one in the MAP assignment, and run inference. This process gives rise to two configurations; we select the one that has the higher likelihood in our model.

5. The Pose Detection Pipeline

Before building the probabilistic model, we must initialize variable domains with placement hypotheses for the parts. Exhaustive search of the high-dimensional, continuous space of possible part placements is not feasible. We use low-level detectors to suggest part placement hypotheses. In particular, we use very efficient spin-image features [13] to find similar surfaces in the model and the scene. In our data sets, the spin-image radius is chosen to be approximately the size of the (puppet or human) head. Following the work of Johnson and Hebert [13], we cluster the spin-image matches to obtain coarse part placement hypotheses. These hypotheses are refined further using a standard method for rigid surface alignment [5]. This process frequently produces good hypotheses, but also generates a substantial number of incorrect ones (Fig. 3).

The spin-image suggestions can be used to define the Markov network model, described in Sec. 3. Search in the space of possible object poses is performed using the loopy belief propagation algorithm, as described in Sec. 4. The result of this algorithm is an object pose which has the highest or one of the highest scores in our model.

The pose obtained from the probabilistic inference can be further refined using an algorithm called Articulated ICP [2]. The original hypotheses provided to the probabilistic inference are generated by detectors and aligned separately to the range scan surface. Articulated ICP optimizes the alignment of all parts simultaneously, and respects the model joints. Specifically it computes the rigid transformations for all parts that minimize the distance between neighboring parts and between each part and the nearby surface. The result of this process is a more accurate placement of the articulated model in the range scan.

6. Dealing with Missing Part Hypotheses

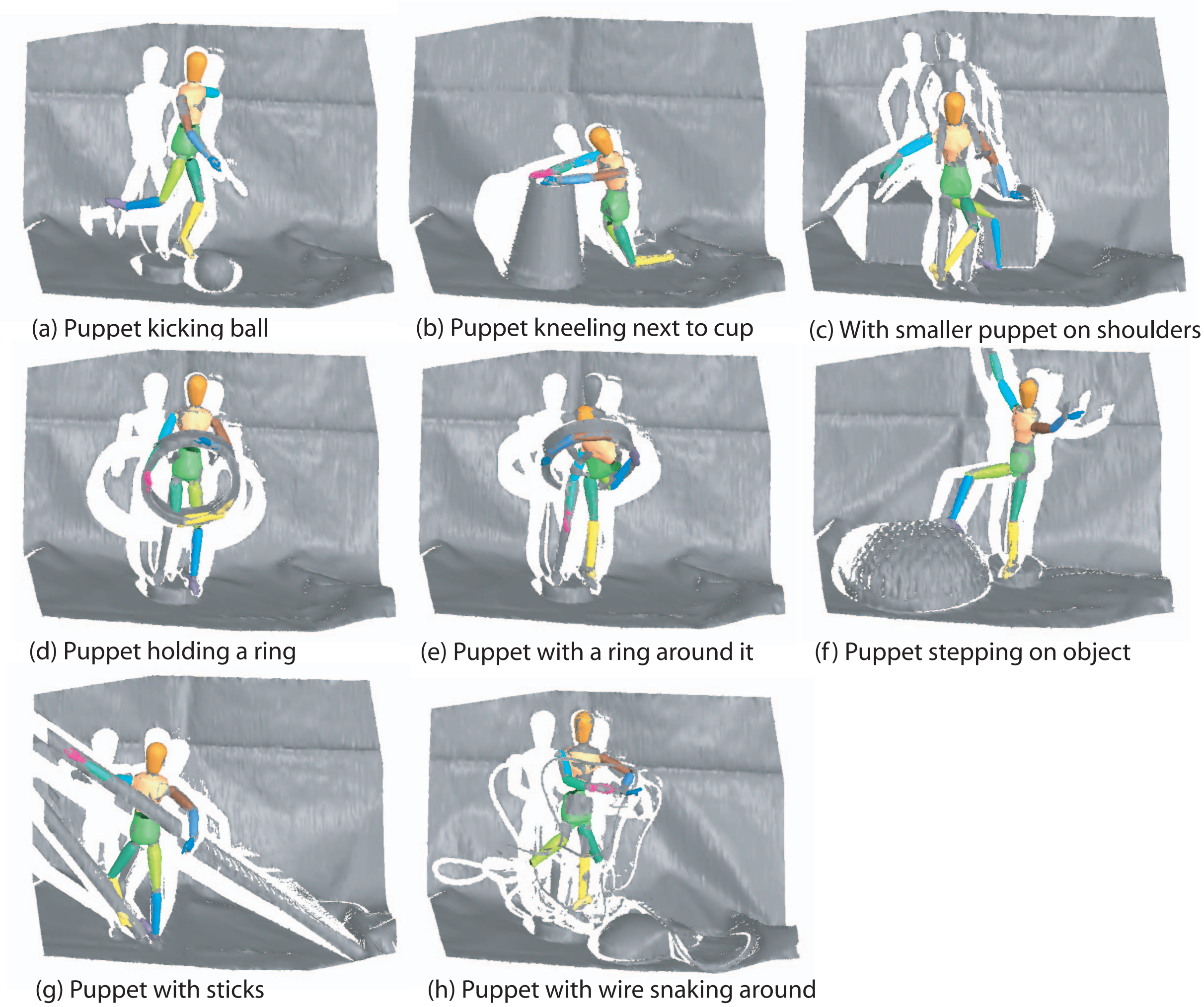
In the previous section, we avoided discussing a fundamental challenge. For many object parts, the detectors can fail to get good placement hypotheses altogether, either because of the failure of the detector itself, or because the part was not observed in the range scan. Because of this, joint constraints can be severely violated, causing the inference process to fail. One of the main contributions of this paper is to provide a practical solution for this problem.

Our first strategy is to try to generate the missing hypotheses. Before running inference, we introduce a *domain enrichment* phase, in which object parts propose placement hypotheses for their neighbors. We do this in two different ways. First, the hypotheses for each part are used to generate hypotheses for the part's immediate neighbors. For each neighbor part we consider transformations which preserve the joint with the original part, and align the part surface to similar surface in the scene (this similarity is quantified using a spin-image score). In this way, a correctly-placed arm can suggest a placement for a heavily occluded torso. Second, we use hypotheses for two parts on either side of a given part to suggest placements for it. For example, a correct torso and lower arm placement can suggest the placement of a missing upper arm. In practice, the combination of these two strategies is very effective in finding good hypotheses which have been missed by the detector.

However, this not sufficient to completely deal with the problem. Our probabilistic model, described so far, requires that all of the object parts are placed somewhere in the scene. In scenes with significant occlusion, this constraint is not necessarily desirable. When some parts are completely hidden, there is no data from which to obtain placement hypotheses consistent with the rest of the puppet parts. Thus, we may not have any good placement hypotheses in the domains of those parts and they can only be placed in locations inconsistent with other parts. Therefore, we relax the constraint that all parts must be placed in the scene, choosing instead to obtain a high-scoring connected component in the tree structure corresponding to the articulated object.

More precisely, we select a central root part r, which is required to be present after the domain enrichment phase. Other parts can be missing from the scene. Once we declare a part p to be missing, all parts in the tree structure that are linked to the root via a path involving p are also declared to be missing. This follows the intuition that, if we cannot place p in the scene, our placements for neighboring parts are also likely to be wrong. In our experiments, the upper torso appears to work well as a root part. In general, other root parts can be tried in addition to it.

We augment our probabilistic model to allow missing parts by including a part-missing hypothesis h^p_{null} in the domain for each part p. The local part potential associated with this hypothesis is designed so as not to score higher than any hypothesis that explains an area or edge, but to score higher than a part that occludes surface it should not. The pairwise score $\psi(h^p_{null}, h^q_j)$ between a missing part p and its upstream neighbor q is a neutral uniform potential, which does not impose any joint-consistency or part-intersection penalties. The potential between a pair of part-missing hypotheses $\psi(h^p_{null}, h^q_{null})$ is simply 1. Finally, the potential $\psi(h^p_{null}, h^q_j)$ between a missing part p and a non-missing *downstream* neighbor q is 0. With this model a part



(a) Puppet kicking ball

(b) Puppet kneeling next to cup

(c) With smaller puppet on shoulders

(d) Puppet holding a ring

(e) Puppet with a ring around it

(f) Puppet stepping on object

(g) Puppet with sticks

(h) Puppet with wire snaking around

Figure 4. Sample of scenes and the resulting puppet embeddings

will always be added to the network as long as its addition (or the addition of it and parts beneath it) have a positive effect on the likelihood of the resulting configuration.

7. Experimental Results

We present results of running our algorithm on two data sets: the model puppet described before and a set of human scans. We then present analytical results on a third, synthetic data set based on the puppet.

We tested our algorithm on several scenes involving a 15 part puppet model, viewed in different poses and from various directions. The scenes have various cluttering and occluding objects. The dataset was obtained using a temporal stereo scanner [7]. A sample of the scenes and the resulting puppet embeddings are shown in Fig. 4.

We are able to correctly identify the torso and head of the puppet in almost all cases, even in situations with substantial occlusion. In most cases, we are able to place the limbs correctly. This is possible even in scenes with substantial occlusion. For example, in the scene of the puppet kicking the ball (Fig. 4(a)), its own arm occludes much of it from view. In the scene with the puppet holding two sticks (Fig. 4(g)) much of the puppet is not visible, but the limbs are placed generally correctly.

Even in situations where limbs are not placed correctly, they are placed in a configuration consistent with the scene data. For example, in the scene of the puppet holding the ring (Fig. 4(d)), the leg is turned up and placed along the surface of the ring. This is consistent with our model, where we wish to place our object in such a way that it explains the observed data. In the scene in which we obtain the worst results (Fig. 4(e)), the puppet is twisted in an unusual manner in an attempt to fit the observed surface. This, in fact, high-

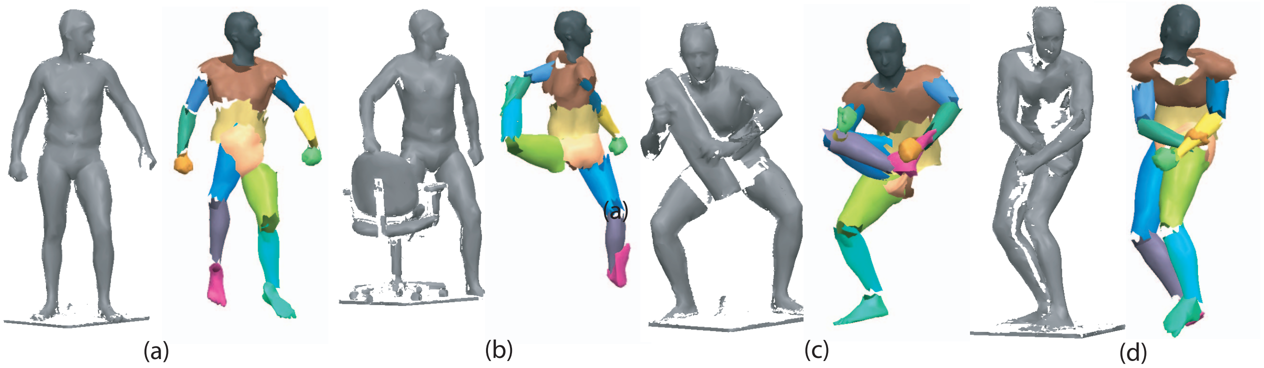


Figure 5. Sample of 4 scenes and the resulting human poses. For each pair, the image on the left is the original scene and the image on the right is the recovered human pose.

lights the power of our hypothesis generation mechanism and the difficulty of distinguishing an object from similarly-shaped nearby surfaces without additional knowledge of the space of likely poses.

We also test our algorithm on a dataset of partial views of a human. We matched a 16-part articulated model to data produced using a Cyberware WRX scanner. This dataset introduces an additional challenge because the individual parts are now deformable since human bodies are non-rigid. Fig. 5 shows 4 scenes and the corresponding pose recovered for each one. These results were obtained with the same model settings as those used for the puppet. The only exception was the strength of the joint angle prior, which differs for humans and puppets.

In the human scenes, we find correct placements of the heads and torsos, as well as most of the limbs. For example, in Fig. 5(d), we are able to correctly reconstruct the entire skeleton, despite the fact that most of the torso is missing, due to occlusion from the arms and missing data caused by shadows. In the cases where the limbs are not placed in the correct location, they are consistent with the observed data, placed either behind an occluding object or in the unknown area surrounding the person (this area occurs because there are no readings in some parts of the scan). Overall, we demonstrate a fairly robust performance in a variety of difficult settings including changes of the field of view, occlusion and clutter.

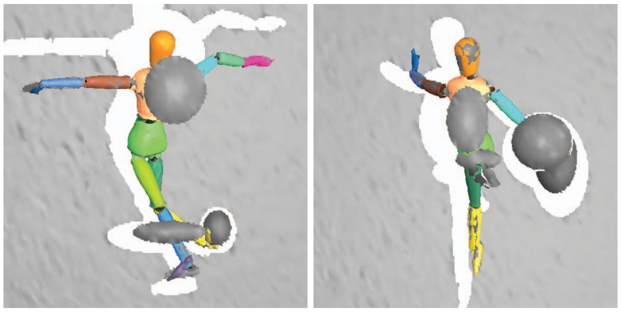


Figure 6. Synthetic scene examples. The algorithm achieves low error relative to ground truth.

Finally, to provide a quantitative analysis of our algorithm, we created synthetic puppet scenes and tested our detector. We generated 25 scenes with the puppet in random non self-intersecting poses. Between 2 and 5 ellipsoids of different sizes were randomly placed between the puppet and the scanner viewpoint.The scene was created by introducing Gaussian noise with standard deviation between 0.1 and 0.35, and keeping only the surfaces visible from the scanner viewpoint. We then ran our detector on the scenes, and compared the results to the ground truth poses. The part placement error was measured by comparing the displacements of its centroid and its adjacent joints in our result. Recognizing the symmetry of the puppet parts, we allowed the upper torso and arms to flipped without penalty, and did the same with the lower torso.On average, we found placements for 13.84 of the 15 parts. For the placed parts, the average displacement was 3.56% of the puppet height.Fig. 6 shows the resulting placements in sample scenes. Finally, it is worth noting that this experiment poses an easier problem than real-world object detection, as (a) occluding spheres may look reasonably different from the puppet parts in many cases and (b) an actual range scanner may fail to capture some of the surfaces due to shadows and dark colors. Nonetheless, the results demonstrate the effectiveness of our algorithm, as we were successful in finding the pose even when large object portions were occluded.

8. Conclusions and Future Directions

This paper focuses on the challenging problem of detecting the pose of an articulated 3D object in range scans. Our algorithm is based on a probabilistic framework that combines a local model of occlusion and scene generation with a global model enforcing consistency in the assignments to different object parts. Our method utilizes spin-image detectors to generate individual part-placement hypotheses, and then performs inference in the probabilistic model to determine a coherent articulated object configuration. A key contribution of our work is its treatment of elusive parts — those for which the low-level detector provides no reasonable hypotheses. These cases are surprisingly common in our data sets, due to the presence of occlusion and clutter. We describe a systematic approach for generating reasonable hypotheses for these missing parts by leveraging the articulated model structure, as well as a graceful method for backing off when this hypothesis generation fails. We demonstrate that our method can successfully identify the pose in data sets, consisting of complex, cluttered scenes with significant occlusion.

There are several important directions for future work. One obvious improvement is to introduce a more informed prior over the space of likely object poses; such an extension would allow us to prune more configurations that are consistent with the data but place the object in highly unlikely poses. Given a richer data set, we could also incorporate additional appearance cues, such as color or texture. More broadly, we would like to define a richer probabilistic model of articulated object pose and its appearance in real images, and learn (certain aspects of) this model from data.

Acknowledgements

We gratefully acknowledge the help of James Davis from UC Santa Cruz and Stefano Corazza and Lars Mündermann from the Stanford Biomechanics Lab in acquiring the range scan data. This work was supported by DARPA under the Transfer Learning program.

References

[1] A. Agarwal and B. Triggs. 3d human pose from silhouettes by relevance vector regression. In *Proc. CVPR*, 2004.

[2] D. Anguelov, L. Mündermann, and S. Corazza. An iterative closest point algorithm for tracking articulated models in 3d range scans. In *ASME/Summer Bioengineering Conference*, Vail, Colorado, 2005.

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, H.-C. Pang, and J. Davis. The correlated correspondence algorithm for unsupervised surface registration. In *Proc. NIPS*, 2004.

[4] A. Athistos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *Proc. ICCV*, 2003.

[5] P. Besl and N. McKay. A method for registration of 3d shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[6] J. Coughlan and S. Ferreira. Finding deformable shapes using loopy belief propagation. In *Proc. ECCV*, 2002.

[7] James Davis, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime stereo : A unifying framework for depth from triangulation. In *CVPR*, 2003.

[8] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *Proc. CVPR*, pages 66–73, 2000.

[9] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. ECCV*, 2004.

[10] D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[11] S. Ioffe and D. Forsyth. Probabilistic methods for finding people. *Int. Journal of Computer Vision*, 43(1):45–68, 2001.

[12] Michael Isard. Pampas: Real-valued graphical models for computer vision. *Proc. CVPR*, 1:613–620, 2003.

[13] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *Proc. IEEE PAMI*, 21:433–449, 1999.

[14] M. P. Kumar, P. Torr, and A. Zisserman. Obj cut. In *Proc. CVPR*, 2005.

[15] G. Mori and J. Malik. Estimating human body configurations using shape context matching. *Proc. ECCV*, 3:666–680, 2002.

[16] K. Murphy and Y. Weiss. Loopy belief propagation for approximate inference: An empirical study. In *Proc. Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI '99)*, pages 467–475, 1999.

[17] M. Pauly, N. Mitra, and L. Guibas. Uncertainty and variability in point cloud surface data. In *Proc. of Symposium on Point-Based Graphics*, pages 77–84, 2004.

[18] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, 1988.

[19] Xiaofeng Ren, Alexander C. Berg, and Jitendra Malik. Recovering human body configurations using pairwise constraints between parts. In *Proc. ICCV*, 2005.

[20] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. ICCV*, 2003.

[21] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and Michael Isard. Tracking loose-limbed people. *Proc. CVPR*, 1:421–428, 2004.

[22] C. Sminchisescu, A. Kanaujia, Zhiguo Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *Proc. CVPR*, 2005.

[23] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Distributed occlusion reasoning for tracking with nonparametric belief propagation. In *Proc. NIPS*, 2004.

[24] J. Sullivan and S. Carlsson. Recognizing and tracking human action. *Proc. ECCV*, 1:629–644, 2002.

[25] K. Toyama and A. Blake. Probabilistic exemplar-based tracking in a metric space. *Proc. ICCV*, 2:50–57, 2001.

[26] S. Yu, R. Gross, and J. Shi. Concurrent object recognition and segmentation with graph partitioning. In *Proc. NIPS*, 2002.