Human Pose Estimation from a Single View Point, Real-Time Range Sensor

Matheen Siddiqui and Gérard Medioni
Institute for Robotics and Intelligence Systems
University of Southern California, Los Angeles, CA, 90089
{mmsiddiq,medioni}@iris.usc.edu

Abstract

We estimate and track articulated human poses in sequences from a single view, real-time range sensor. We use a data driven MCMC approach to find an optimal pose based on a likelihood that compares synthesized depth images to the observed depth image. To speed up convergence of this search, we make use of bottom up detectors that generate candidate head, hand and forearm locations. Our Markov chain dynamics explore solutions about these parts and thus combine bottom up and top down processing. The current performance is 10 frames per second. We provide quantitative performance evaluation using hand annotated data. We demonstrate significant improvement over a baseline ICP approach.

This algorithm is then adapted to estimate the specific shape parameters of subjects for use in tracking. In particular, limb dimensions are included in the human pose parametrization and are automatically estimated for each subject in short training sequences. Tracking performance is quantitatively evaluated using these person specific trained models.

1. Introduction

Human pose estimation is an important task in computer vision. A successful tracking system can find applications in motion capture, human computer interaction, and activity recognition. The estimation of a human body pose from a single view, however, continues to remain a difficult problem, as the process of extraction is complicated by a high dimensional search-space riddled with local minima and ambiguous configurations.

Many of these difficulties can be mitigated with the use of multi-view camera systems, in that 3D information provides measurements that can disambiguate difficult poses. Such systems, however, have more complexities than their single view counterparts due to the spatial positioning of multiple camera nodes.

Depth sensors bridge the gap between single and multi-view systems by providing 3D measurements from a single viewpoint. A key enabling factor is the recent development of affordable real-time depth-sensing cameras. These sensors produce images where each pixel has an associated depth value. Depth input enables the use of 3D information directly, which eliminates many of the ambiguities and inversion difficulties plaguing 2D image-based methods.

In this work, we present a human pose estimation and tracking system that makes use of depth sensor images. In particular, we are using the SR3000 from MESA, which produces images of size 176×144 at 25fps. Our system combines both bottom-up and top-down processing in a data driven Markov Chain Monte Carlo (MCMC) framework. The rest of this paper is organized as follows: in section 2, we describe related works. In section 3 we present our representation, which consists of a skeleton model with cylinders attached. In section 4 we describe our data driven MCMC system. Pose estimation results and analysis are presented in section 5. Following this we adapt our pose tracking framework to find person specific shape parameters (i.e. the limb dimensions) in section 6. In section section 7, we present some final remarks.

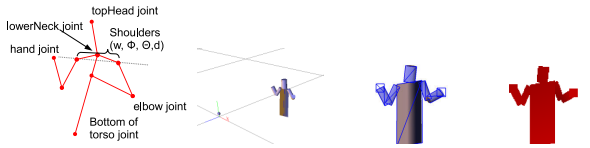
2. Related Work

There exist many approaches to address pose estimation [1][2][3][4] [5][6]. These approaches differ in how bodies are encoded, visual saliency metrics, and the machinery that performs the alignment with an underlying image.

In [1][7][8][9], fully articulated and detailed 3D models are employed. These methods search a solution space defined by complex top-down metrics. In [8] this is accomplished using a gradient search. In [7] this is done via randomized search. Our work is motivated by [1] in which a data driven MCMC approach is used to explore a high-dimensional solution space in a single image and bottom up limb detectors are used to enhance this search. In [9], both gradient and randomized search techniques are employed.

An alternative to searching directly for a 3D parameterization of the human body, is to search for its projection.

978-1-4244-7030-3/10/$26.00 ©2010 IEEE



(a) 3D Pose Skeleton (b) Synthesized Pose(3D) (c) Projected Pose (2D) (d) Rasterized Pose(2D)

Figure 1. Representation of Poses

In [3] this idea is formalized using the Scaled Prismatic Model (SPM), which is used for tracking poses via registering frames. In [10] a 2D articulated model is aligned with an image by matching a shape context descriptor. In [11] articulated poses are estimated directly using a large database of exemplars and an appropriate hashing function.

Other relevant approaches model a human as a collection of separate but elastically connected limb sections, each associated with its own detector [2][12][13][14]. In [2] this collection of limbs is arranged in a tree structure. It is aligned with an image by first searching for each individual part over translations, rotations, and scales. Following this, the optimal pose is found by combining the detection results of each individual part efficiently. In [15] limbs are arranged in a graph structure and belief propagation is used to find the most likely configuration.

The use of depth information has also been explored. In [16], a coarse labeling of depth pixels is followed by a more precise joint estimation to estimate poses. In [17], control theory is used to maintain a correspondence between model based feature points and depth points. In [18], Iterative Closest Point (ICP) is used to track a pose initialized using a hashing method. In [19], information in depth images and silhouettes is used in a learning framework to infer poses from a database.

In our work, we find poses by optimizing a generative likelihood that accounts for an observed depth image directly. We solve this problem by combining both top down and bottom up processing in a data driven MCMC framework to find an optimal pose using only depth imagery. We do not need a database of poses nor a large training step. We also do not rely on precise segmentation of the depth streams, which is often problematic in sequences with significant sensor noise or motion blur. Thus, our system is able to track effectively and is robust to discontinuous motion and tracking failures.

3. Representation

We model the body as a skeleton with fixed width cylinders attached as shown in Fig. 1(a). The joints for the hands, elbows, top of head, lower neck, and bottom of torso are parameterized by their 3D positions. The position of the head and bottom of the torso are defined relative to the lower neck position. The shoulders' joints are defined in 3D relative to the lowerNeck joint. They are parameterized by: w, the distance between them, θ/ϕ, the orientation of the line passing between them, and d, the position of this line along the line between the lower neck, and bottom of torso joints.

This model is defined in the coordinate system aligned with the camera. Thus, depth is along the optical axis and depth information is separated from the image positions. Because depth measurements are more noisy than image plane measurements, parameterizing in this way, allows the solution space to be explored more effectively.

The limbs, head, and torso are fixed width cylinders attached to this skeleton. The cylinder lengths, however are scaled to allow their ends to coincide with their corresponding joints. For each subject, the skeleton and cylinder dimensions are measured from a simple initial training pose.

The main step in evaluating pose fit quality with an underlying depth image consists of estimating a depth image from the given model. This is done by efficiently rendering the model defined above into a depth buffer. We first attach a cylinder to each limb on the model. We then find the occluding boundaries of each corresponding cylinder. Given a pinhole camera, these boundaries correspond to a pair of line segments. From this pair of line segments we can render a pair of triangles into the depth buffer, and interpolate depths at the endpoints of the line segments.

This approximation works well for cylinders parallel to the image plane. However, forearms can become orthogonal to the image plane (pointing gestures). To account for this case, we render hand positions as squares at a single depth location. The size of this square is determined by the projected width of the corresponding cylinder. This process is illustrated in Fig. 1.

4. Formulation

We denote the skeleton as $\mathbf{X}$, the depth image as $\mathbf{I}$, and the depth image rendered from the view point of the camera as $\tilde{\mathbf{I}}$. In order to find a human pose $\mathbf{X}$ in a range image $\mathbf{I}$, at time t we seek to find the pose that optimizes the likelihood:

$$\mathbf{X}_t = \arg\max p(\mathbf{X}|\mathbf{I}_t) = \arg\max p_{obs}(\mathbf{I}_t|\mathbf{X}) p_{prior}(\mathbf{X}) \quad (1)$$

where p_{obs} and p_{prior} are the observation likelihood and prior respectively. The observation likelihood is based on estimating an expected range image from $\mathbf{X}$ and comparing the rendered and observed depth images. The prior is used to give impossible poses zero probability.

This optimization is accomplished using a data driven MCMC framework[20]. The Metropolis Hastings (MH) Algorithm is used to generate samples that represent $p(\mathbf{X}|\mathbf{I})$ from a proposal distribution, $q(\mathbf{X}^i|\mathbf{X}^{i-1})$. From these samples we keep track of the optimum. This is an iterative process in which we perturb the current sample $\mathbf{X}^{i-1}$, evaluate, and then accept the sample based on a computed likelihood.

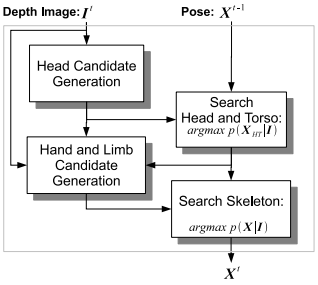


Figure 2. Estimation of a Human Pose in a Depth Image

To speed up convergence of this iterative process, we design effective proposal mechanisms detailed in section 4.4 and we search over sets of parameters separately. In particular, our proposal mechanisms make use of candidate parts, depth points, and the found pose in the previous frame. By generating proposals through combining information from these sources, we are able to effectively search the solution space.

The overall approach is shown in Fig. 2. We first update the head, torso and shoulder parameters while preserving the parameters associated with the arms. In this step we make use of candidate head positions as described in section 4.3.1. This estimate for the head and torso is used in the forearm candidate detection of section 4.3.2. Following this we can search for the skeleton using the dynamics described in section 4.4.

The optimal pose found using this approach, $\mathbf{X}_{max}$, is used to initialize the process in the next frame. By tracking a pose this way, we are able to maintain the optimum under the likelihood. Since the proposal distribution makes use of part candidates found throughout the image, we are able to combine both bottom-up and top-down processing and remain robust to discontinuous motion and tracking failures.

4.1. Observation Likelihood

The observation likelihood of a pose defined in section 3 is based on rendering a synthesized depth image into a buffer and computing its difference with the observed depth image. In particular, our observation likelihood is:

$$p_{obs}(\mathbf{I}|\mathbf{X}) \sim \exp(-(\quad \lambda_1\phi_s(\tilde{\mathbf{I}},\mathbf{I}) + \lambda_2\phi_d(\tilde{\mathbf{I}},\mathbf{I}) + \lambda_3\phi_{dt}(\tilde{\mathbf{I}},\mathbf{I}))) \tag{2}$$

In this equation $\mathbf{X}$ represents our body model, $\mathbf{I}$, the observed depth image, and $\tilde{\mathbf{I}}$ the rendered/expected depth image from $\mathbf{X}$. In this metric, we separate the overall saliency into terms that depend on the foreground silhouettes, ϕ_s, and depth information, ϕ_d and ϕ_{dt}.

The term ϕ_s counts the number of pixels which are different between foreground silhouettes. The term ϕ_d computes the sum of the squared differences (thresholded to D^2_{max})between the observed and estimated depth images.

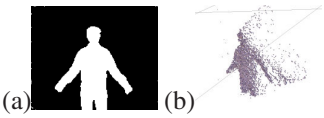


Figure 3. Silhouette(a) and Depth Data (b)

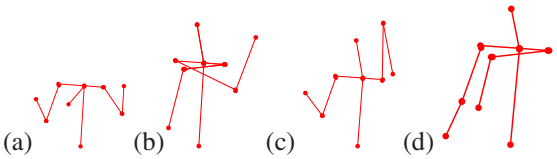


Figure 4. Classes of Impossible Poses: (a) Top of head falls below lower neck, (b) Upper arms crossing, (c) Elbows pointing up, (d) Arms crossing the torso and bending down

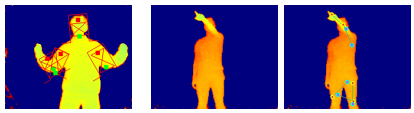
The term ϕ_{dt} counts the number of pixels missed in depth. In particular, it computes the number of pixels in a pair of depth images whose difference is greater than a predetermined threshold, d_{thresh}.

The term ϕ_{dt} allows us to explicitly penalize pixels missed in depth, whereas ϕ_d ensures smoothness of the observation likelihood in depth. In our work $d_{thresh} = .1$.

Separating depth (3D) and silhouette (2D) terms this way is important because we can assign weights based on their reliability. A pixel is considered part of the foreground silhouette if its depth is closer than a threshold, D_{max}. In Fig. 3, we see the absolute depth value is susceptible to measurement errors as well as motion blur, whereas the silhouette is very stable in the image plane. Also, foreground silhouette information is important for body configurations that do not vary significantly in depth, but in which the limbs are still visible.

4.2. Prior

The prior, p_{prior}, in equation (1) assumes that limb lengths are Gaussian distributed about average lengths, but assigns zero probability to poses that are highly unlikely to correspond to actual poses. This can be enforced by determining if $\mathbf{X}$ violates physical constraints imposed on the joints of a human. As a first approximation, we consider impossible poses to be those whose projections are exemplified by the classes shown in Fig. 4. This includes those poses whose projection causes the top of the head to fall below the lower neck, as shown in Fig. 4(a) or in which the upper arms cross, as shown in Fig. 4(b). We also include poses where the arms bend in unlikely ways. In particular, the class shown in Fig. 4(c) includes poses where the elbow is above its corresponding shoulder and hand. The class in Fig. 4(d) includes poses which cause the upper arm to cross the torso and form an angle less than 135° with its corresponding forearm.

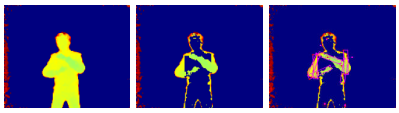


(a) Head Candidates (b) Pointing Hand Candidates

Profile Hand candidates:

(c) Original (d) 2D Image Skeleton (e) Candidates

Rectangle Detection:



(f) Original (g) Head and Torso Pixels Removed (h) Candidates

Figure 5. Part Candidate Generation

4.3. Part Detection

Our estimation algorithm finds body poses that optimize equation (1). To aid the search we make use of bottom up detectors to find candidate head and forearm positions. Bottom up processing occurs at each frame and speeds up convergence of the MCMC algorithm presented in section 4.4.

4.3.1 Head Detection

To find candidate head positions, we search for the outline of a head shape in the Canny edges of the depth image. At each position and orientation in the image we determine the outline of a fronto-parallel head, situated at a distance given by the underlying depth value. This is illustrated in Fig. 5(a). If the head shape sufficiently overlaps foreground pixels, we grade it according to the average of distances from each point on the outline to the nearest Canny edge.

We locate the first candidate head, h_1, by finding the position and orientation with the best score. The second candidate, is the best head pose that also differs from h_1 by at least 20 pixels and 10 degrees in orientation. Subsequence poses are found in a similar manner. This is illustrated in Fig. 5(a). We keep all such candidates.

4.3.2 Forearm Candidate Detection

To find hand candidates, we find the endpoints of the foreground image skeleton. As shown in Fig. 5, this works well for profile arms outside the body trunk[21].

We also find other hand candidate points by finding depth points that are relatively close to the camera. This is done by maintaining a list of hand positions that are close to the camera but are also at least 20 pixels apart. This heuristic works when the arm points to the camera and the user faces the camera, which is common.

From the hand candidates, we estimate the elbow by first finding the orientation, θ, of the forearm in the image plane. The orientation is taken from a rectangle with one end anchored at the hand tip position and also minimizes the average distance of edge points to its boundary. Using depth information and the known length of the forearm we can determine the 3D location of the hand tip and the elbow.

We also find forearms by segmenting the head and torso pixels. Given an estimate of the head and torso we can label all depth points that are within a threshold to the head or torso cylinders as torso pixels. When these pixels are removed we can better find candidate forearms by considering all positions and orientations in the range image. Each location is graded by how well a local cylinder fits the underlying depth. Multiple candidates are found by repeatedly finding the locations that minimize this score and also differs from previous candidates by 20 pixels and 30 degrees.

4.4. Markov Chain Dynamics

A critical step in generating samples using data driven MCMC is how the samples are perturbed. In this work, a new sample, $\mathbf{X}^i$, is generated from a previous sample, $\mathbf{X}^{i-1}$, by selecting one of the following methods:

Random Walk Here we generate $\mathbf{X}^i$ by perturbing the parameters of $\mathbf{X}^{i-1}$ under a Gaussian distribution.

Snap to Previous Pose In this move a limb is assigned its position in the previous frame. After this alignment, the updated parameters are perturbed by Gaussian noise.

Snap to Head Here we align the head with one of the candidate head positions selected with equal probability by aligning the top head joint. The lower neck joint is aligned at random. We also randomly adjust the torso to be directly under the head. This is illustrated in Fig. 6(a), where the head is aligned with a head candidate. After this alignment, the updated positions are perturbed by Gaussian noise.

Snap to Forearm In this proposal we first randomly select a candidate limb or a pair of candidate limbs. We then assign the pose to its 2D hand positions. The depth assigned is either that of a nearby pixel, the average depth in a window about the hand, or its previous depth. The corresponding elbow is either assigned its position from $\mathbf{X}^{i-1}$ or the estimated elbow position. This is illustrated in Fig. 6(b), where a forearm is aligned with a forearm candidate. We also, at random swap the hand and elbow position, place the elbow directly behind the hand the length of the forearm, or place the elbow at the midpoint of the hand and corresponding shoulder. After this alignment, the arm is perturbed by Gaussian noise.

Snap to Depth Here, we select at random either a hand, elbow, lower neck, top head, or bottom torso joint. We then

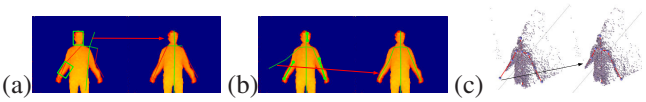


Figure 6. Markov Chain Dynamics: (a)Snap to Head (b) Snap to Forearm (c) Snap to Depth

adjust its depth by either: assigning it the depth of a nearby depth point, or computing the average depth in a window about the point with equal likelihood. In the case of the top head joint, we also can assign it the depth of the lower neck. This is illustrated in Fig. 6(c), where the hand joint is snapped to a depth point under it.

4.5. Optimizing using Data Driven MCMC

To obtain better convergence properties, we do not optimize over all parameters simultaneously. Given the high dimensionality of the solution space, a full search would require a prohibitively large number of iterations. Instead, we optimize over groups of parameters.

We first update the head, torso and shoulder parameters while preserving the parameters associated with the arms using 600 iterations of the MH algorithm. After this we perform 600 iterations on both arms, then 200 iterations only on the parameters associated with the right arm, and then 200 iterations on the parameters associated with the left arm. We then, in two iterations, again search for head/torso, followed by the left arm, followed by the right arm. This gives us a total of 3600 iterations. This, together with 600 MCMC iterations devoted to estimate the head and torso prior to hand and forearm candidate generation (see Fig. 2) gives us 4200 total iterations.

5. Pose Estimation Evaluation

To evaluate our system, we make use of annotated test sets to compute performance bounds as well as compare it to standard approaches used to track poses in range data. In these experiments we assume the model parameters are known and fixed. Estimation of the model parameters for a specific subject is addressed in section 6.

In our comparative analysis we evaluate our system on annotated test sets. Our dataset consists of four general categories of motions. The first category is Single Arm Profile motions (SAPr) and consists of one arm moving such that the arm is mostly lateral to the image plane. This includes motions such as waving. We consider a single arm pointing toward the camera in Single Arm Pointing (SAPt). We consider both arms moving in Two Arm Motion (TAM) and motion dominated by the body in Body Arm Motion (BAM). Our data sets include four different subjects with different limb widths and lengths. Example images of each category along with the skeletons found by our system are shown in Fig. 7.

These data sets have been manually annotated using specially designed software tools that allow us to visualize the depth data in 3D and position joints accordingly. We designed these tools using OpenGL/glut.

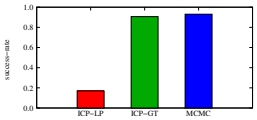


Figure 9. Success rates for tracking systems

Using this data, we can evaluate our performance quantitatively as shown in Fig. 8. As a comparison, we show results for the ICP-based algorithm described in [22]. In this ICP implementation we use a cylinder based pose model parameterized by a skeleton with fixed limb lengths and widths. As ICP is primarily a tracking algorithm, we consider two re-detection schemes: In the first scheme, ICP-LP, if the error that ICP minimizes is greater than a threshold, we use assume the tracker failed and use the last recovered pose. In the second scheme, ICP-GT, we manually re-initialize the model to the ground truth whenever the maximum difference between their corresponding joints exceeds 15 pixels. This re-initialization occurs at the start of each frame. Because we are using ground truth data, ICP-GT represents the performance of ICP using the best pose re-detection possible.

In computing Fig. 8 we consider a frame in which the maximum joint error between the pose and the ground truth in the image plane is less than 15 pixels to be a success. The statistics shown are for frames that are successful in each motion category, as well as for all frames combined (ALL). In Fig. 8 the top row shows the results for our system, while the bottom row show the results for ICP-GT. The left column shows image errors, the middle column shows depth errors, and the right column shows the success rates

From these results, we see that our system works quite well for frames in which poses were recovered. For SAPt, we note that the elbow positions are often occluded by the hands. While this affects their estimation accuracy, in this class the hand tips, which are localized well, are of greater significance. Our system does lose track, as is indicated by success rates less than 100%. This usually happens when the arms are very close to the body, completely occluded, or under fast motion. In these cases, the system can lose track of one or both arms, however, it is able to recover.

The overall success rates for each system is shown in Fig. 9. Our approach had a success rate of 0.930 whereas the rate of ICP-LP was only 0.169. Here we see ICP alone, without addressing tracking failures has little chance of per-

Figure 7. Examples:(a)Single Arm Profile (SAPr)(b)Single Arm Pointing (SAPt)(c)Two Arm Motion(TAM)(d)Body Arm Motion(BAM)

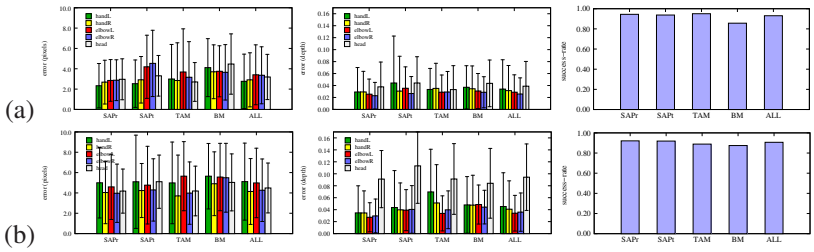


Figure 8. Quantitative Evaluation of Motion Types: (a) Data Driven MCMC, (b) Iterative Closest Point w Ground Truth Re-Initialization.

forming well. The success rate was 0.907 for ICP-GT. Recall that ICP-GT manually reinitializes to the correct pose when its estimate is too different from the ground truth, and thus represents the best possible re-detection with an ICP-based tracker. Even in this case, our recovery rates are slightly higher. This demonstrates the effectiveness of our tracking and bottom up processing in automatically recovering from tracking failures.

Our system is also able to track at higher levels of accuracy for recovered poses. The overall average error for our system is 2.56 pixels in the image and 0.036 in depth. In contrast ICP-GT with manual re-initialization had errors of 3.13 pixels in the image and 0.050 in depth. These results are significant with a p-value of less than 0.1. This demonstrates the modeling accuracy of our likelihood measures.

5.1. Limitations

From these results, we see that our system works quite well for frames in which poses were recovered. However, our system has difficulties, when the arms are fully occluded. Our system will try to explain the current image assuming the arms are visible. If an arm is fully occluded, it will try to align the arm with another part of the body. Other difficulties occur when the low level detectors completely miss their targets. This can happen if there is too much motion blur in the frame, or the size of the corresponding parts are too small. Since we search about the pose in the previous frame, we generally do not need the detectors to work all the time when motion is slow. When motion is fast, however, and the detectors fail, our system is unable to find the optimal configuration within the number of iterations available.

6. Model Parameter Estimation

Heretofore, we assumed the lengths and widths of the human were known and fixed. Given these model parameters, we were able to find the pose of the user in each frame of a sequence. While the performance of this tracking system is robust, it does depend on the the model parameters used. This is especially true on subjects whose dimensions vary significantly from the model parameters employed, as in the case of tracking a slender subject on a model appropriate for a sumo wrestler.

Using a Bayesian framework, these parameters are estimated by alternating between estimating the pose and model parameters in each frame in the sequence independently and then estimating the overall model parameters. While the recovered model parameters are useful in measurement related tasks, we show that tracking performance is also affected. Having good estimates of these model parameters results in improved tracking performance.

To estimate pose and model parameters we introduce $\hat{\mathbf{X}}_t$ which encodes the pose as well as shape parameters (i.e lengths and widths of each limb). We also denote the fixed prior of these model parameters as $\mathbf{M}$. These can be found as the optimum of a likelihood:

$$\{\hat{\mathbf{X}}_\mathbf{t}\}_{t=1}^{N}, \mathbf{M} = \arg\max \prod_t p(\hat{\mathbf{X}}_\mathbf{t}, \mathbf{M}|\mathbf{I}_t) \quad (3)$$

Here, we denote the shape component of $\hat{\mathbf{X}}_t$ as $\mathbf{M}_t$. Using Bayes' rule:

$$\prod_t p(\hat{\mathbf{X}}_\mathbf{t}, \mathbf{M}|\mathbf{I}_t) = \prod_t p(\mathbf{I}_t|\hat{\mathbf{X}}_\mathbf{t}, \mathbf{M}) p(\hat{\mathbf{X}}_\mathbf{t}, \mathbf{M})$$
$$= \prod_t p(\mathbf{I}_t|\hat{\mathbf{X}}_\mathbf{t}) p(\hat{\mathbf{X}}_\mathbf{t}|\mathbf{M}) p(\mathbf{M}) \sim \prod_t p(\mathbf{I}_t|\hat{\mathbf{X}}_\mathbf{t}) p(\hat{\mathbf{X}}_\mathbf{t}|\mathbf{M})$$
$$= \prod_t p(\mathbf{I}_t|\hat{\mathbf{X}}_\mathbf{t}) p(\hat{\mathbf{M}}_\mathbf{t}|\mathbf{M}) \quad (4)$$

Success Rate

sequence \ model	A	B	C	D	E
A	0.89	0.89	0.84	0.27	0.79
B	0.98	0.99	0.98	0.37	0.96
C	0.91	0.93	0.90	0.56	0.82
D	0.76	0.71	0.67	0.83	0.33
E	0.46	0.46	0.39	0.16	0.72

Image Errors

sequence \ model	A	B	C	D	E
A	3.58	3.81	4.53	6.88	4.52
B	2.63	2.40	3.04	4.55	3.92
C	3.73	3.25	3.72	7.86	4.48
D	4.70	5.26	4.95	3.80	6.52
E	4.90	5.86	5.96	8.90	4.50

Figure 10. Performance over sequences of Specific Users

Here we assume the image, $\hat{\mathbf{I}}$ only depends on $\hat{\mathbf{X}}_t$ and that all shapes are equally likely (i.e $p(\mathbf{M})$ is constant). Furthermore we model the distribution $p(\mathbf{M}_t|\mathbf{M})$ as independent Gaussian in each component (i.e. in each model length and width dimension).

To find the optimum of the likelihood in equation 4 we alternate between computing $\hat{\mathbf{X}}_t$ in each frame while fixing $\mathbf{M}$ and then fixing $\hat{\mathbf{X}}_t$ while updating $\mathbf{M}$. Computing $\hat{\mathbf{X}}_t$ consists of alternating between finding the pose as before, and then updating the model parameters $\mathbf{M_t}$ (using both the image likelihood and the distribution, $p(\mathbf{M}_t|\mathbf{M})$) in each frame using sampling methods. Updating $\mathbf{M}$ is a simple average of the model parameters in each frame.

6.1. Model Parameter Estimation Evaluation

To evaluate the performance of our model parameter estimation framework we make use of annotated sequences of 5 test subjects. These sequences include a training sequence followed by a longer test sequence consisting of the motions used in section 5.

From the training sequences we estimate the model parameters for 5 test subjects. Each of the resulting models are used to track the annotated test sequences of each subject. The the performance is shown in the matrices in Figure 10.

From this analysis we see that overall performance is dependent on the model parameters used in tracking. Lower errors and higher success rates on the diagonals of the matrix in Figure 10 show that having the correct model parameters results in better performance. This is most evident in the subjects "D" and "E". These subjects had shape significantly different from the others. As a result their performance was most impacted by using the model parameters trained on the other subjects.

7. Discussion

In this work, we developed a robust pose tracking system that combines bottom-up part candidates and an efficient

top-down likelihood using a data driven MCMC framework on range images. Currently our system runs at approximately 10fps in a single threaded framework running on 32-bit 3GHz-Xeon processor with 8GB of Ram. Most of the processing is devoted to rasterizing and computing differences between depth buffers.

We have also extended this framework to estimate model parameters from a sequence. The estimation of these model parameters takes a few seconds approximately per frame. While this processing is not at interactive rates, the parameters are not expected to change for a given user. We can thus use this method in a calibration phase on the initial frames before processing using our pose estimation system.

We plan on using of General Purpose Graphics Processing Units for further improvements in speed and modeling accuracy. In particular, we can process multiple parallel Markov chains and rasterize more complex limb models.

References

[1] M. W. Lee and I. Cohen, "A model-based approach for estimating human 3D poses in static images," *PAMI*, vol. 29, no. 6, pp. 905–916, 2006. 1

[2] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, vol. 61, no. 1, pp. 55–79, 2005. 1, 2

[3] J. M. Rehg, D. D. Morris, and T. Kanade, "Ambiguities in visual tracking of articulated objects using two- and three-dimensional models," *I. J. Robotic Res.*, vol. 22, no. 6, pp. 393–418, 2003. 1

[4] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, "Tracking loose-limbed people," in *CVPR*, Washington, DC, 2004, pp. I: 421–428. 1

[5] A. Agarwal and B. Triggs, "3D human pose from silhouettes by relevance vector regression," in *CVPR*, July 2004, pp. II:882–888. 1

[6] G. Mori, X. Ren, A. A. Efros, and J. Malik, "Recovering human body configurations: combining segmentation and recognition." in *CVPR*, Washington, DC, 2004, pp. II:326–333. 1

[7] J. Deutscher and I. D. Reid, "Articulated body motion capture by stochastic search," *IJCV*, vol. 61, no. 2, pp. 185–205, Feb 2005. 1

[8] C. Bregler and J. Malik, "Tracking people with twists and exponential maps," in *CVPR*, Santa Barbara, CA, June 1998, pp. 8–15. 1

[9] C. Sminchisescu and B. Triggs, "Estimating articulated human motion with covariance scaled sampling," *I. J. Robotic Res.*, vol. 22, no. 6, pp. 371–392, 2003. 1

[10] G. Mori and J. Malik, "Recovering 3D human body configurations using shape contexts," *PAMI*, vol. 28, no. 7, pp. 1052–1062, 2006. 1

[11] G. Shakhnarovich, P. A. Viola, and T. J. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *ICCV*, 2003, pp. 750–757. 2

[12] D. Ramanan and D. A. Forsyth, "Finding and tracking people from the bottom up," in *CVPR*, Madison, WI, 2003, pp. II: 467–474. 2

[13] S. X. Ju, M. J. Black, and Y. Yacoob, "Cardboard people: A parameterized model of articulated image motion," in *Proc. of the 2nd Int. Conf. on Automatic Face and Gesture Recognition*, 1996, pp. 38–44. 2

[14] M. Siddiqui and G. Medioni, "Efficient upper body pose estimation from a single image or a sequence," in *Workshop on Human Motion*, 2007, pp. 74–87. 2

[15] L. Sigal, M. Isard, B. H. Sigelman, and M. J. Black, "Attractive people: Assembling loose-limbed models using non-parametric belief propagation," in *NIPS*, 2003, pp. 1539–1546. 2

[16] Y. Zhu and K. Fujimura, "Constrained optimization for human pose estimation from depth sequences," in *ACCV (1)*, 2007, pp. 408–418. 2

[17] K. F. Youding Zhu, Behzad Dariush, "Controlled human pose estimation from depth image streams," in *CVPRW*, 2008, pp. 1–8. 2

[18] D. Demirdjian, T. Ko, and T. Darrell, "Untethered gesture acquisition and recognition for virtual world manipulation," *Virtual Reality*, vol. 8, no. 4, pp. 222–230, 2005. 2

[19] H.-D. Yang and S.-W. Lee, "Reconstruction of 3D human body pose from stereo image sequences based on top-down learning," *Pattern Recognition*, vol. 40, no. 11, pp. 3120–3131, 2007. 2

[20] Z. Tu and S. C. Zhu, "Image segmentation by data-driven markov chain monte carlo," *PAMI*, vol. 24, no. 5, pp. 657–673, 2002. 2

[21] P. Correa, F. Marqués, X. Marichal, and B. M. Macq, "3d posture estimation using geodesic distance maps," *Multimedia Tools Appl.*, vol. 38, no. 3, pp. 365–384, 2008. 4

[22] D. Grest, J. Woetzel, and R. Koch, "Nonlinear body pose estimation from depth images," in *DAGM*, 2005, pp. 285–292. 5