Computer Science and Artificial Intelligence Laboratory
Technical Report

MIT-CSAIL-TR-2005-079
AIM-2005-034

December 1, 2005

Conditional Random People: Tracking Humans with CRFs and Grid Filters

Leonid Taycher, Gregory Shakhnarovich,
David Demirdjian, Trevor Darrell

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Conditional Random People: Tracking Humans with CRFs and Grid Filters

Leonid Taycher[†], Gregory Shakhnarovich[‡], David Demirdjian[†], and Trevor Darrell[†]

[†] Computer Science and Artificial Intelligence Laboratory,
Massachusetts Institute of Technology,
Cambridge, MA 02139
{lodrion, demirdji, trevor}@csail.mit.edu

[‡] Department of Computer Science,
Brown University,
Providence, RI 02912
gregory@cs.brown.edu

Abstract

We describe a state-space tracking approach based on a Conditional Random Field (CRF) model, where the observation potentials are learned *from data. We find functions that embed both state and observation into a space where similarity corresponds to L_1 distance, and define an observation potential based on distance in this space. This potential is extremely fast to compute and in conjunction with a grid-filtering framework can be used to reduce a continuous state estimation problem to a discrete one. We show how a state temporal prior in the grid-filter can be computed in a manner similar to a sparse HMM, resulting in real-time system performance. The resulting system is used for human pose tracking in video sequences.*

1 Introduction

Tracking articulated objects (such as humans) is an example of state estimation in a high-dimensional space with a non-linear observation model that has been a focus of considerable research attention. The combination of frequent self-occlusion and unobservable degrees of freedom with the large volume of the pose space make probabilistic methods appealing. The vast majority of probabilistic articulated tracking methods are based on a generative model formulation.

Current state-of-the-art generative tracking algorithms use non-parametric density estimators, such as particle filters, due to their ability to model arbitrary multimodal distributions [18, 10]. Unfortunately, several properties conspire to make particle filtering extremely computationally intensive. On one hand, a large number of particles is needed in order to faithfully model the distributions in question. On the other hand, a complex likelihood function needs to be evaluated for each particle at every iteration of the algorithm. A further drawback of generative-model based algorithms is that the likelihood function is too complicated to be learned from data and is usually specified in an ad-hoc fashion. Recently, the use of directed discriminative models with parameters learned directly from data have been proposed [1, 27, 22].

In this work we pose state estimation as inference in an *undirected* Conditional Random Field model (CRF) [17]. This allows us to replace the likelihood function with a more general observation potential (compatibility) function that can be automatically learned from training data. These functions might be expensive to evaluate in general, but can be made efficient at run-time if all state (pose) values at which they can be evaluated are known in advance. In this case much of the computation can be performed off-line, thus greatly reducing run-time complexity.

This algorithm naturally operates on a discrete set of samples, and we will show how we can estimate the posterior probability in a *continuous* state space using grid filtering methods. The idea underlying these methods is that if the state-space can be partitioned into regions that are small then the posterior can be well approximated by a constant function within each region.

The direct application of grid filtering would, of course, result in the need to evaluate the potential function in each region in the partition, which is impossible to do in reasonable time even with fast implementation. Fortunately this is not necessary, since at a particular time-step, the *prior state probability* would be negligible in the vast majority of the regions, allowing us to concentrate only on locations with a significant prior.

Our algorithm operates in a standard predict-update framework: at every step we first estimate the temporal prior probability of the state being in each of the regions in the partition. We then evaluate the observation potential only for regions with non-negligible prior. When the set of cells is fixed, we can precompute the transition probabilities between cells, and thus reduce the temporal prior com-

putation to a *single sparse matrix/vector multiplication*, in a manner similar to HMMs [20], thus avoiding a sampling step altogether.

After reviewing related prior work, we first describe the CRF-based tracking formulation and describe a way to learn a particular observation potential function based on image embedding (Section 3). We then discuss a grid-filter-based inference method which can be realized with a sparse HMM computation (Section 4). The results of our method are demonstrated and compared against competing algorithms in Section 5.

2 Prior Work

Probabilistic articulated pose estimation is often approached using state-space methods. The majority of the approaches have been based on a generative model formulation, with varying assumptions about the forms of the pose distribution and transition probabilities. Early methods [14, 21] assumed that both were Gaussian and used Kalman filtering. Extended and Unscented [28] Kalman filters enabled modeling of non-linear transitions, but still constrained pose distribution to be Gaussian. These methods required a relatively small number of evaluations of the likelihood function, but lost track due to restrictive distribution models.

The need to relax the unimodality assumption led first to use of mixture models [11, 5], and then to Monte-Carlo methods that represent distributions with sets of discrete samples (particles) [18, 10, 25, 26]. While theoretically sound, particle filtering methods are not very successful in high dimensions [15] – they require large numbers of particles to faithfully represent the distribution, which entails large computational costs of likelihood evaluation. Furthermore, the emission probability model used in likelihood evaluation is very expensive to train, and is often hand-designed in an ad-hoc fashion.

Several discriminative methods have been proposed for visual pose tracking. These algorithms apply various regression techniques while leveraging large number of annotated image sequences. For example, one [1], or a mixture [27] of simple experts were trained to predict current pose based on the past pose estimates and the current observation. Robust regression combined with fast nearest neighbor search was used for single frame pose estimation in [23].

In this paper we dispense with directed models altogether and opt for a Conditional Random Field (CRF) [17] model. The main advantage of this model over generative models is that CRFs do not require specification (and evaluation) of the emission probability, but only similarity between state and observation(s). CRFs are also a more flexible model than the previously proposed regression methods. They allow for modeling the relationship between the state and an arbitrary subset of observations. They are also better able to adjust to sequences not appearing in the training data. For example the MEMM model (similar to one used in [27]) has been shown to be subject to label bias problem [17].

While in the present work we use a simple chain-structured CRF (Figure 1(b)), which directly models the dependency between concurrent state and observation, it can be extended by introducing more general relationships between state and observations.

We learn the observation potential function for our model using the parameter sensitive embedding introduced in [23]. This algorithm allows us to learn a transformation of images of humans into a space where the distance between embeddings of two images is likely to be small if the poses are similar and large otherwise. The observation potential of a particular pose is then determined by the distance between embeddings of the rendering of the figure in this pose and the observed image.

If for every pose at which we would like to evaluate the potential we had to render the corresponding image, our method would be extremely slow. By discretizing the continuous pose space, we are able to precompute the embeddings of all discrete poses off-line, thus drastically reducing run-time complexity. Fixing the set of poses at which observation potential can be computed would seem to be an unreasonable restriction, since we are operating in a continuous pose space, but we overcome this problem by using a variant of the grid-filtering technique [6, 2].

The main idea underlying grid filtering is that sufficiently discretized random variable is indistinguishable from a continuous one. That is, if the distribution can approximated by a piece-wise constant function, then it is sufficient to evaluate it only at one point in every "constancy region" (cell) [6]. This reduces a continuous estimation problem to a discrete one (albeit with very large number of discrete points). We show that in the case where both observation potential and the temporal prior are constant in each cell, tracking can be formulated as state estimation in an HMM framework, allowing us to use existing inference algorithms; further, we have found in practice that a manageable number of cells suffices for realistic tracking tasks.

3 Tracking with Conditional Random Fields

Figure 1(a) shows the dynamic generative model that is commonly used in tracking applications. The state (pose[1]) at time t is denoted as θ^t, and the observed images as I^t. The full model is specified by the initial distribution $p(\theta^0)$,

[1] In this work we consider only first order Markov models of motion.

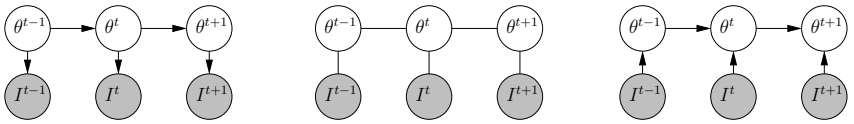

Figure 1: Chain-structured generative (a), CRF (b), and MEMM (c) tracking models. In all models the state of the object (pose) at time t is specified by θ^t, and the observed image by I^t. The generative model is described by transition probability $p(\theta^t|\theta^{t-1})$ and the emission probability $p(I^t|\theta^t)$. The CRF model is described by motion compatibility (potential) function $\phi(\theta^t, \theta^{t-1})$ and the image compatibility function $\phi_o^t(\theta^t) = \phi(I^t, \theta^t)$. Note the contrast with the MEMM model [27], specified by the conditional distribution $p(\theta^t|\theta^{t-1}, I^t)$ as shown in (c).

the transition probability model $p(\theta^t|\theta^{t-1})$, and the emission distribution $p(I^t|\theta^t)$. This model describes the *joint* probability of the state(s) and observation(s)

$$p(\theta^{0..T}, I^{1..T}) = p(\theta^0) \prod_{t=1}^{T} [p(\theta^t|\theta^{t-1}) p(I^t|\theta^t)],$$

from which appropriate conditional distributions of the pose parameters can be derived.

While reasonable approximations can be constructed for the transition probability, $p(\theta^t|\theta^{t-1})$, the problem for generative models lies in specifying the emission model $p(I^t|\theta^t)$. In practice, to evaluate the likelihood function at a particular pose, a figure in this pose is first rendered as an image, and this image is then compared with the observation using a certain metric[13]. Evaluating the likelihood thus becomes computationally expensive.

The major difference between generative-model based approaches and ours is that we formulate pose estimation as inference in a Conditional Random Field (CRF) model, and are able to *learn* a compact and efficient observation and transition potentials from data.

A chain version of a CRF is shown in Figure 1(b). While, apart from the lack of arrows, it is quite similar to the generative model, the underlying computations are quite different. This model is specified by the motion potential $\phi(\theta^t, \theta^{t-1})$ and the observation potential $\phi_o^t(\theta^t) = \phi(I^t, \theta^t)$. The observation potential function is the measure of compatibility between the latent state and the observation. Of course, one choice for it might be the generative model's emission probability $p(I^t|\theta^t)$, but this does not have to be the case. It can be modeled by any function that is large when the latent pose corresponds to the one observed in the image and small otherwise.

Rather than modeling the joint distribution of poses and observations, the CRF directly models the distribution of poses conditioned on observation,

$$p(\theta^{0..T}|I^{1..T}) = \frac{1}{Z} p(\theta^0) \prod_{t=1}^{T} [\phi(\theta^t, \theta^{t-1}) \phi_o^t(\theta^t)],$$

where Z is a normalization constant.

Once the observation potential is defined, a chain-structured CRF[2] can be used to perform on-line tracking

$$p(\theta^t|I^{1..T}) \propto \phi_o^t(\theta^t) \int \phi(\theta^t, \theta^{t-1}) p(\theta^{t-1}|I^{1..t-1}) d\theta^{t-1}. \quad (1)$$

The main advantage of this model from our standpoint is that the observation potential $\phi_o^t(\theta^t)$ may be significantly simpler to learn and faster to evaluate than the emission probability $p(I^t|\theta^t)$. Below we describe an model of such potential based on similarity between images.

Suppose that we can measure the similarity $\mathcal{S}$ such that, given two images I_a and I_b with underlying poses θ_a and θ_b, respectively, $\mathcal{S}(I_a, I_b)$ is with high probability small if $d_\theta(\theta_a, \theta_b)$ is small, and large otherwise.[3] Suppose now that we are interested in evaluating the potential $\phi(I^t, \theta)$, and that we have access to an image I^θ that corresponds to the pose θ (for instance, we can render it using computer graphics). Then, we can define the observation potential based on distance in the image embedding space:

$$\phi(I^t, \theta) \; = \; N(\mathcal{S}(I^t, I^\theta); 0, \sigma^2). \quad (2)$$

In this work, we follow the approach in [23] for learning a binary *embedding* $H(I)$ of images such that the L_1 distance in the H space serves as a proxy for such a pose-sensitive similarity $\mathcal{S}$. Briefly, the learning algorithm is based on formulating a classification problem on image pairs (similar/dissimilar), and constructing an embedding based on a labeled training set of such pairs.

Once the desired M-dimensional embedding $H = [h_1(I), \ldots, h_M(I)]$ has been learned, the induced similarity is the Hamming distance in H: $\mathcal{S}(I_a, I_b) = \sum_{m=1}^{M} |h_m(I_a) - h_m(I_b)|$.

This potential could conceivably be used in a continuous domain, for example by using Monte Carlo methods in the CRF framework, as it captures features relevant to pose estimation better than generic image similarity. Unfortunately it would not reduce computational cost since it would require rendering the image I^θ at runtime for every pose θ which we would like to evaluate.

[2]While both transition and observation potentials in a CRF are often trained jointly, it is possible to train them separately, as we do in this case.

[3]Here d_θ stands for the appropriate distance in pose space

This approach becomes particularly efficient when we have a finite (albeit large) set of possible pose hypotheses $\theta_1, \ldots, \theta_N$. In such a case we can render an image I_i for each pose in the set, and compute its embedding $H(I_i)$. The only calculation required at runtime is computing the embedding $H(I^t)$ and calculating the Hamming distances between the bit vectors. We capitalize on this efficiency in the grid-filtering framework described in the next section.

4 Grid Filtering

In the previous section we have proposed a CRF tracking framework where the observation potential is computed as the distance between embeddings of state and observation described in the previous section. Computing this potential for an arbitrary pose and image is relatively slow since it would involve rendering an image of a person in this pose and then computing the embedding. This is part of the problem with generative-model-based tracking which we wanted to avoid.

Fortunately, if all of the poses where the observation potential is to be evaluated are known in advance, then we can precompute the appropriate embedding off-line, drastically reducing runtime evaluation cost. We would then compute a single embedding for the observed image, which would be amortized when the potential is evaluated at multiple poses.

While fixing the poses in advance seems too restrictive for continuous space inference, grid-based techniques pioneered by [4, 16] show that this can be a profitable approximation. The main idea underlying these methods is that many functions of interest can be approximated by piecewise constant functions, if the region of support for each constant "piece" is small enough. As metioned above, we follow the convention and denote such region of support as a "cell".

In our case, the function we are interested in is the posterior probability of the pose conditioned on all previously seen observations (including the current one). The posterior is proportional to the product of the temporal prior (the pose probability based on the estimate at the previous timestep and the motion model) and the observation potential. We would like to define the cells such that both of the components are almost constant. The observation potential is often sharply peaked, so the cells should be small in the regions of pose space where we expect large appearance variations, but large in other regions. On the other hand the motion models are usually (and our work is no exception) very approximate and compensate for it by inflated dynamic noise. Thus the temporal prior is broad and should also be approximately constant on cells small enough for observation potential constancy. We derive the grid filter based on the assumption that the partition of the pose space into cells with the properties described above is available.

Let the space of all valid poses Θ be split into N disjoint (and not necessarily regular) cells $\mathcal{C}_i$, $\Theta = \cup_{i=1}^{N} \mathcal{C}_i$, $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset, i \neq j$, such that both likelihood and prior can be approximated as constant within each cell. Furthermore, let us have a sample $\theta_i \in \mathcal{C}_i$ in every cell. The set of sample points $\{\theta_i\}_1^N$ is referred to as "grid" in the grid-filtering framework.

By virtue of our assumptions, the temporal prior can be expressed as

$$\begin{aligned} p(\theta^t \in \mathcal{C}_i | \theta^{t-1} \in \mathcal{C}_j) &= \int_{\mathcal{C}_i} \int_{\mathcal{C}_j} \phi(\theta^t, \theta^{t-1}) d\theta^{t-1} d\theta^t \\ &\approx \phi(\theta_i, \theta_j) |\mathcal{C}_i| |\mathcal{C}_j|, \end{aligned} \quad (3)$$

where $|\mathcal{C}_i|$ is the volume of the ith cell, with the approximation valid when the noise covariance in the transition is much wider than the volume of the cell. So the (time independent) transition probability from jth to ith cell is

$$T_{ij} = \frac{\phi(\theta_i, \theta_j)|\mathcal{C}_i|}{\sum_{k=1}^{N} \phi(\theta_k, \theta_j)|\mathcal{C}_k|}. \quad (4)$$

The compatibility between observation and the pose belonging to a particular cell can be written as

$$\phi_o^t(\mathcal{C}_i) = \int_{\mathcal{C}_i} \phi_o^t(\theta) d\theta \approx \phi_o^t(\theta_i)|\mathcal{C}_i|. \quad (5)$$

Combining eqs 1, 3, and 5, the posterior probability of pose being in the ith cell is

$$\begin{aligned} p(\theta^t \in \mathcal{C}_i | I^{1..t}) &\approx \frac{1}{Z} \phi_o^t(\theta_i) |\mathcal{C}_i| \sum_{j=1}^{N} T_{ij} p(\theta^{t-1} \in \mathcal{C}_j | I^{1..t-1}) \\ &= \frac{1}{Z} \phi_o^t(\theta_i) \sum_{j=1}^{N} S_{ij} p(\theta^{t-1} \in \mathcal{C}_j | I^{1..t-1}), \end{aligned} \quad (6)$$

where $S_{ij} = |\mathcal{C}_i| T_{ij}$ is time independent and can be computed offline.

If we denote

$$\pi^t = \begin{pmatrix} p(\theta^t \in \mathcal{C}_1 | I^{1..t}) \\ p(\theta^t \in \mathcal{C}_2 | I^{1..t}) \\ \vdots \\ p(\theta^t \in \mathcal{C}_N | I^{1..t}) \end{pmatrix} \quad \text{and} \quad l^t = \begin{pmatrix} \phi_o^t(\theta_1) \\ \phi_o^t(\theta_2) \\ \vdots \\ \phi_o^t(\theta_N) \end{pmatrix},$$

then the posterior can be written in vector form

$$\pi^t = \frac{1}{W} S \pi^{t-1} .* l^t, \quad (7)$$

where $.*$ is the element-wise product, and the scaling factor $W = \sum_{i=1}^{N} (S\pi^{t-1} .* l^t)_i$ is necessary for probabilities to sum up to unity.

Algorithm	CRP	CRPS	kNN	ICP	CND	ELMO
Seconds	0.05	0.07	0.5	0.1	120	8

Table 1: Average times required for algorithms tested to process a single frame.

The final equation has striking resemblance to the standard HMM update equations. It defines our online CONDITIONAL RANDOM PERSON tracking algorithm (CRP). We can also use standard HMM inference methods [20] to define a batch version of CRP: CRP SMOOTHED (CRPS) uses a forward-backward algorithm to find the pose distribution at every time step conditioned on all observed images. In addition, the most likely pose sequence can be found by using Viterbi decoding and we call the resulting method CRP VITERBI (CRPV).

5 Implementation and Evaluation

We have implemented CRP and CRPS as described in the previous sections. We have used the database of 300,000 pose exemplars generated from a large set of motion capture data in order to cover a range of valid poses. The images are synthetic, and were rendered, along with the foreground segmentations masks, in Poser [7] for a fixed viewpoint. The motion-capture sequences are available from [12] and include large body rotations, complex motions, and self-occlusions. The transition matrix was computed by locating 1000 nearest neighbors in joint position space for each exemplar, and setting the probability of transitioning to each neighbor to the be Gaussian with $\sigma = 0.25$. The volume of each cell was approximated as that of a ball with radius equal to the median distance to 50 nearest neighbors.

We used the multiscale edge direction histogram (EDH) [23] as the basic representation of images. The binary embedding H is obtained by thresholding individual bins in the EDH. It was learned using a training set of 200,000 image pairs with similar underlying poses (we followed an approach outlined in [29] for estimating false negative rate of a paired classifier without explicitly sampling dissimilar pairs). This resulted in 2,575 binary dimensions.

The tracking algorithms are initialized by searching for 50 exemplars in the database closest to the first frame in the sequence in the embedding space.

Due to the sizes of the database and the transition matrix, both algorithms require large amounts of memory, so we performed our tests on a computer with 3.4GHz Pentium 4 processor and 2GB of RAM. The algorithms were implemented in C++, and were able to achieve real-time performance with average speeds of 20 frames per second for CRP and 14 frames per second for CRPS.

5.1 Experiments with synthetic data

We have quantitatively evaluated the performance of our tracking method on a set of motion sequences. These sequences were obtained in the similar way as the sequence used for training our algorithm but were not included in the training set.

We compared our online algorithm, CRP, and its batch version CRPS (CRP SMOOTHED), to four state-of-the-art pose estimation algorithms. The first baseline was a stateless k-Nearest Neighbors (kNN) algorithm that at every frame searches the whole database for 50 closest poses based on the embedding distance. The remaining baseline methods were incremental tracking algorithms: deterministic gradient descent method using the Iterative Closest Point (ICP) algorithm [8], CONDENSATION [25], and ELMO [9]. The ICP algorithm directly maximizes the likelihood function at every frame, whereas CONDENSATION and ELMO evaluated the full posterior distribution. In our experiments, the posterior distribution in CONDENSATION was modeled using 2000 particles. In ELMO, the posterior distribution was modeled using a mixture of 5 Gaussians. The likelihood function defined in ICP, CONDENSATION and ELMO was based on the Euclidean distance between the articulated model and the 3D (reconstructed) points of the scene obtained from a real-time stereo system. In contrast, both CRP and kNN algorithms require only single view intensity images and foreground segmentation.

We have chosen to use the mean distance between estimated and true joint positions as an error metric [3]. In Figure 2 we show the performance of 6 algorithms described above on four synthetic sequences.[4] As can be seen, both CRP and CRPS consistently outperform kNN, and CONDENSATION[5], and compare favorably to ICP. While CRP produces somewhat worse results than ELMO, it does not use stereo data, and is hundred and sixty times faster. The timing information for all compared algorithms is presented in Table 1.[6]

5.2 Statistical analysis of results

In order to evaluate the statistical significance of these results we used the following methodology. We use the mean joint position error as a measure of accuracy of pose prediction on a given frame. Suppose that algorithms A and B are both tested on the total of N frames, producing on the frame i errors e_i^A and e_i^B, respectively. The quantity of interest is the error difference $d_i^{A-B} = e_i^A - e_i^B$. Figure 3 shows

[4] These results reflect correction of an error in earlier CRPS implementation.

[5] Increasing the number of particles used for CONDENSATION should improve performance, but the computational costs would become prohibitive.

[6] We have used more iterations of gradient descent than the implementation described in [9].

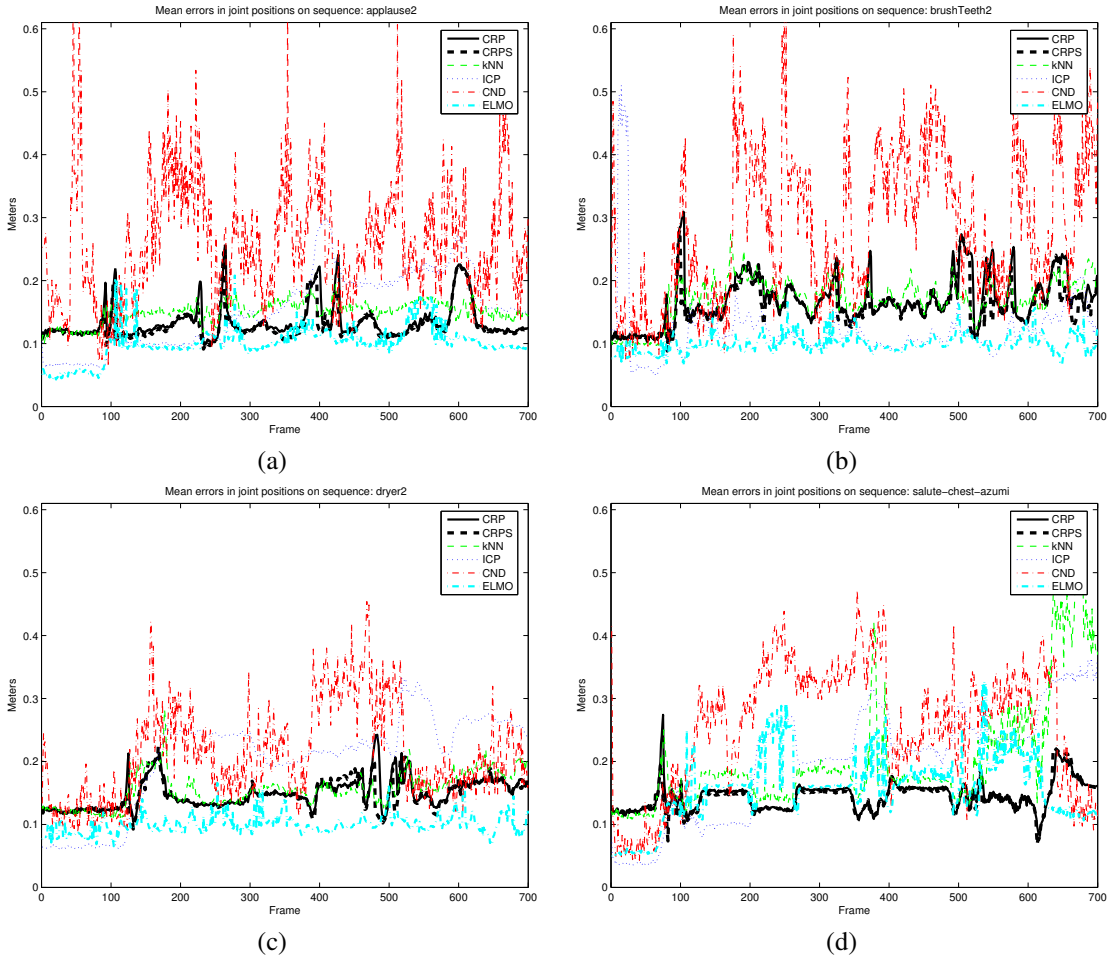

Figure 2: Comparing algorithm performance on four synthetic sequences: "applause" (a), "brush teeth"(b), "dryer" (c), and "salute" (d). The error is measured as an average distance between true and estimated joint positions. The graphs are best viewed in color.

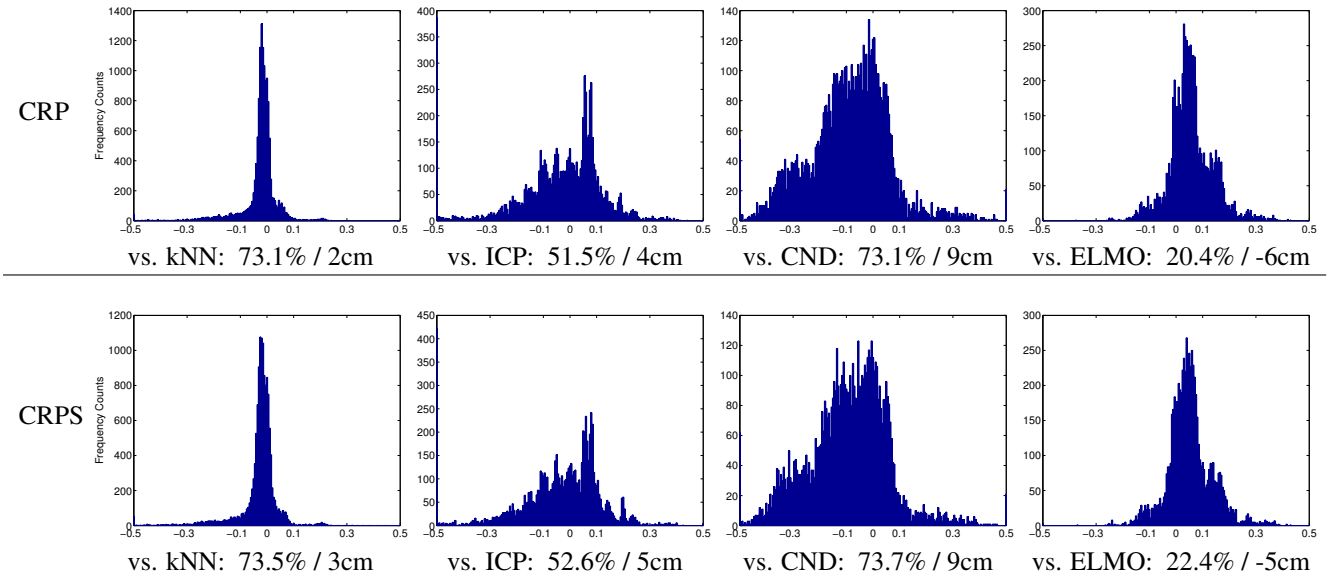

Figure 3: Distributions of improvements in joint position estimates of CRP (first row) and CRPS (second row) vs. kNN (first column), ICP (second), CONDENSATION (third), and ELMO (fourth). Negative values along the x-axis mean lower error for the proposed algorithm. Given for each comparison are the proportion of frames in which CRP/CRPS were better than the alternative, and the average improvement in error. See text for results on statistical significance.

the distribution of these differences between our algorithms (CRP and CPRS) and competing algorithms computed over a large number of synthetic sequences. For example, the top right plot shows the distribution of $d^{CRP-ELMO}$. Negative value of d_i^{A-B} means that on frame i the algorithm A was better than the algorithm B. The lack of a parametric model for the distribution of d^{A-B} makes it difficult to apply thorough statistical testing to hypotheses involving the mean of that distribution. Therefore, the anlysis below focuses on the median, which lends itslef more easily to non-parametric tests.

One question we can ask is whether the results support the conclusion that A is expected to be better more than half the time. We answer this question using the binomial sign test [24]. Intuitively, it is equivalent to modeling the outcome of each comparison (on one frame) by a coin flip in which "tails" means that the sign of d^{A-B} is negative. The null hypothesis we wish to reject is that the coin is fair. We applied this test to the data histogrammed in Figure 3, using p-value[7] of $p = 0.001$. At this significance level, CRP was better than kNN and CONDENSATION and worse than ELMO. CRPS was better than kNN, CONDENSATION and ICP and worse than ELMO; we could not establish significant differences in error of CRP vs. ICP.

A more refined statistical evaluation of the difference in performance between two estimation algorithms is based on establishing a confidence interval on the *median improvement*. Given the desired confidence value p we seek a value D such that the probability of the median difference of the errors being above D is less than p.

We apply the following procedure to perform this test. Suppose that D is the q-upper quantile of the observed distribution of d^{A-B}, i.e. qN values are above D. Under the assumption that the true median of the distribution lies below D, we have $P_D = Pr(d^{A-B} \geq D) < 1/2$. Now, we define a random variable Z_D that is the count of observed values of d^{A-B} that exceed D. Its distribution is binomial with parameters P_D and N. Consequently,

$$Pr(Z_D \geq qN) = \text{Bino}(Z_D; P_D, N) < \text{Bino}(Z_D; 1/2, N), \quad (8)$$

where $\text{Binomial}(x;\ p, n) = \binom{n}{x} p^x (1-p)^{n-x}$. Using De Moivre-Laplace approximation [19],

$$\text{Bino}(Z_D;\ 1/2, N)\ \approx\ N(Z_D, N/2, \sqrt{N}/2), \quad (9)$$

where $N(x;\ \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-(x-\mu)^2/2\sigma^2)$. Combining (8) and (9), and solving for the desired signifcance p, we get

$$q = \mathbb{G}^{-}1(1-p;\ N/2, \sqrt{N}/2)/N,$$

[7]The p-value is the probability of obtaining the observed data under the null hypothesis; that hypothesis is rejected if the p-value falls below a specified threshold, which determines the significance of the test.

Method	kNN	ICP	CND	ELMO
CRP vs.	-1.5cm	0.04cm	-7.13cm	4.57cm
CRPS vs.	-1.75cm	-0.28cm	-7.47cm	4.27cm

Table 2: Confidence intervals for median error reduction, with $p = 0.001$: with probability $1 - p$, the true median of d^{A-B} falls below the value for row A and column B. Negative values indicate cases where we are confident with respect to the improvement (error reduction) achieved by CRP/CRPS over competing methods.

where $\mathbb{G}$ is the inverse of the normal (gaussian) cumulutive distribution function. In other words, if we choose the value of D corresponding to such q, the probability of the true median being *lower* than D is at least $1 - p$. Results of this test for $p = 0.001$ are given in Table 2: there is a robust advantage to both CRP methods over kNN and Condensation, but not over ELMO.

A relatively large difference between estimated mean (Figure 3) and median (Table 2) improvements of CRP variants over ICP can be explained by the fact that ICP is more likely to completely loose track (thus producing large errors) than CRP.

5.3 Experiments with real data

For the real data, segmentation masks were computed using color background subtraction. Sample frames from a complicated real image motion sequence are shown in the Figure 4 (the video of the original sequence and results of our algorithm are available as supplementary material) and Figure 5. The top right pane in the supplementary video was obtained by smoothing CRPV output and rendering the resulting pose sequence.

6 Conclusions and Discussion

We have presented CRP, an algorithm for tracking articulated human motion in real-time. The main contributions of this work are the discriminative CRF formulation of the tracking problem; use of similarity preserving embedding for modeling observation potential function; and the grid-filter inference algorithm that transforms the continuous density estimation problem into a discrete one. The resulting algorithm is capable of accurately tracking complicated motions in real-time (20fps in our experiments for both synthetic and real data).

As future work we are interested in using extra domain knowledge to further improve the performance of the algorithm in two ways. First, when the set of poses that need to be tracked is restricted, then the size of the sample database

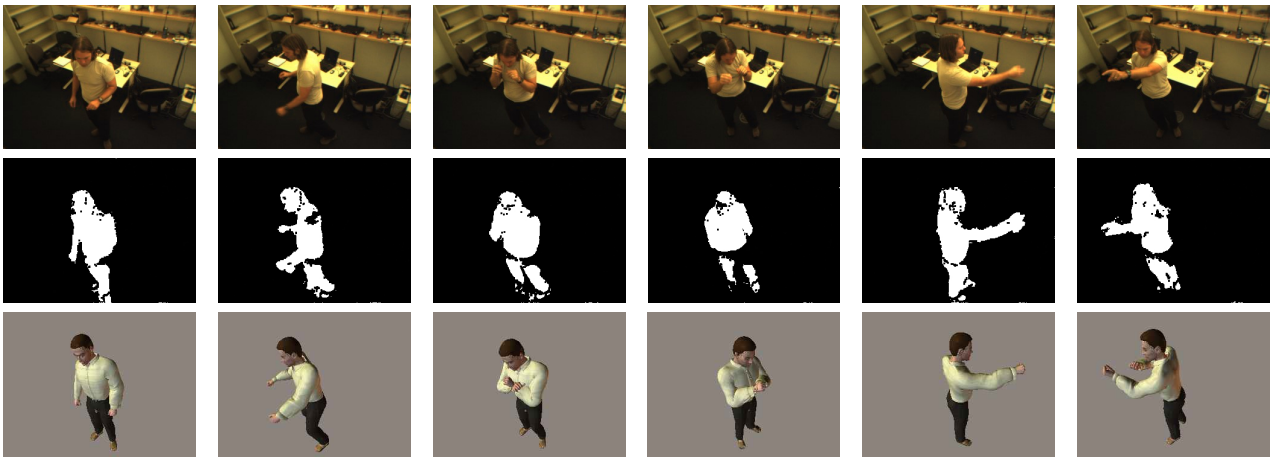

Figure 4: Sample frames from a gesture sequence (first row), segmentation masks (second row) and the corresponding frames from a most likely sequence computed by CRPV algorithm (third row). See supplimentary material for the full video.

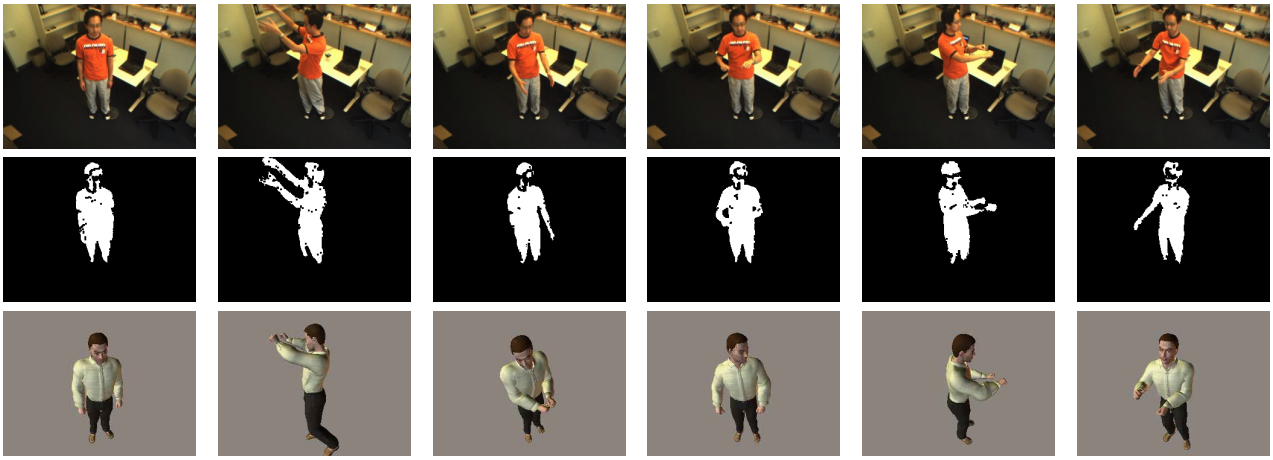

Figure 5: Sample frames from a gesture sequence (first row), segmentation masks (second row) and the corresponding frames from a most likely sequence conputed by CRPV algorithm (third row)

can be decreased by removing all of the unnecessary poses. Second, when the motion patterns are constrained, for example in a dance, tracking can be made more robust by using a specialized transition matrix (resulting, in the limit, in tracking on a motion graph).

Acknowledgments

We would like to thank Nati Srebro for help with statistical analysis of the results. Part of this work was funded by the Office of Naval Research and by DARPA.

References

[1] Ankur Agarwal and Bill Triggs. Learning to track 3d human motion from silhouettes. In *Proceedings of the 21 st International Conference on Machine Learning*, 2004.

[2] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE trans. on Signal Processing*, 50, Feb 2002.

[3] A. O. Balan, L. Sigal, and M. J. Black. A quantitative evaluation of video-based 3d person tracking. In *PETS 2005*, 2005.

[4] R. S. Bucy and K. D. Senne. Digital synthesis of nonlinear filters. *Automatica*, pages 287–298, 1971.

[5] Tat-Jen Cham and James M. Regh. A mutiple hypothesis approach to figure tracking. Technical report, Compaq Cambridge Research Laboratory, 1998.

[6] Zhe Chen. Bayesian filtering: From Kalman filters to particle filters, and beyond. Technical report, Adaptive Systems Lab, McMaster University, 2003.

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

[8] D. Demirdjian, T. Ko, and T. Darrell. Constraining human body tracking. In *IEEE International Conference on Computer Vision*, pages 1071–1078, 2003.

[9] David Demirdjian, Leonid Taycher, Gregory Shakhnarovich, Kristen Grauman, and Trevor Darrell. Avoiding the streetlight effect: Tracking by exploring likelihood modes. In *Proceedings of the International Conference on Computer Vision*, volume I, pages 357–364, 2005.

[10] Jonathan Deutscher, Andrew Davison, and Ian Reid. Automatic partitioning of high diminsional search spaces associated with articulated body motion capture. In *Computer Vision and Pattern Recognition*, Dec 2001.

[11] David E. DiFranco, Tat-Jen Cham, and James M. Regh. Reconstruction of 3-d figure motion from 2-d correspondences. In *Computer Vision and Pattern Recognition*, 2001.

[12] {Eyes, JAPAN}. Motion capture sequences database. www.mocapdata.com, 2005.

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

[14] David C. Hogg. Model-based vision: A program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[15] O. King and David A. Forsyth. How does CONDENSATION behave with a finite number of samples? In *ECCV (1)*, pages 695–709, 2000.

[16] S. C. Kramer and H. W. Sorenson. Recursive bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6):789–801, 1988.

[17] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA, 2001.

[18] John MacCormick and Michael Isard. Partitioned sampling, articulated objects, and interface-quality hand tracking. In *ECCV (2)*, pages 3–19, 2000.

[19] A. Papoulis. *Probability, random variables, and stochastic processes*. McGrow Hill, New York, 3rd edition, 1991.

[20] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[21] K. Rohr. Towards models-based recognition of human movements in image sequences. *CVGIP*, 59(1):94–115, Jan 1994.

[22] Romer Rosales and Stan Sclaroff. Inferring body pose without tracking body parts. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, June 2000.

[23] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proc. 9th Intl. Conf. on Computer Vision*, 2003.

[24] David J. Sheskin. *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall/CRC, 3rd edition edition, 2004.

[25] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Proc. European Conference on Computer Vision*, 2000.

[26] Christian Sminchiesescu and Bill Triggs. Estimating articulated human motion with covariance scaled sampling. *Int. J. Robotics Research*, 22:371–391, June 2003.

[27] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Discriminative density propagation for 3d human motion estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[28] B. Stenger, P. R. S. Mendonca, and R. Cipolla. Model-based hand tracking using an unscented kalman filter. *Proc. British Machine Vision Conference*, 2001.

[29] Y. Ke, D. Hoiem, and R. Sukthankar. Computer Vision for Music Identification. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.