Co-evolutionary Predictors for Kinematic Pose Inference from RGBD Images

Daniel L. Ly[1], Ashutosh Saxena[2] and Hod Lipson[1,3]
[1]Mechanical Engineering, [2]Computer Science, [3]Computing & Information Science
Cornell University
Ithaca, New York 14853
dll73@cornell.edu, asaxena@cs.cornell.edu, hod.lipson@cornell.edu

ABSTRACT

Markerless pose inference of arbitrary subjects is a primary problem for a variety of applications, including robot vision and teaching by demonstration. Unsupervised kinematic pose inference is an ideal method for these applications as it provides a robust, training-free approach with minimal reliance on prior information. However, these methods have been considered intractable for complex models. This paper presents a general framework for inferring poses from a single depth image given an arbitrary kinematic structure without prior training. A co-evolutionary algorithm, consisting of pose and predictor populations, is applied to overcome the traditional limitations in kinematic pose inference. Evaluated on test sets of 256 synthetic and 52 real images, our algorithm shows consistent pose inference for 34 and 78 degree of freedom models with point clouds containing over 40,000 points, even in cases of significant self-occlusion. Compared to various baselines, the co-evolutionary algorithm provides at least a 3.5-fold increase in pose accuracy and a two-fold reduction in computational effort for articulated models.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding

General Terms

Algorithms, Design, Performance

Keywords

Co-evolution predictor sampling, genetic algorithms, evolutionary computation, kinematic pose estimation

1. INTRODUCTION

A fundamental issue in a multitude of robotic and computer vision applications is the automated, three-dimensional pose inference of an articulated subject (Fig. 1). For example, teaching complex robotic movements via human demonstration relies on the ability to infer the teacher's pose [1, 2]. While recent advances have made capturing three-dimensional depth images convenient and affordable, extracting pose information from these images remains a challenge.

Ideally, pose inference operates by manipulating a kinematic skeleton of the subject to best explain the depth image, which provides a natural and robust approach. However, kinematic-based pose inference has often been considered an intractable problem due to a variety of reasons [3, 4], including the density of locally optimal solutions, the high dimensional problem space of articulated kinematic structures and the computational limits of dealing with point clouds, which of thousands of points from a single image.

As a result, state of the art methods in markerless pose inference revolve around two approaches: pose recognition [3, 5] and visual hull methods [6, 7]. While these approaches are fast and accurate, they rely on extensive, supervised training with vast data sets, and thus, they are constrained by the composition of the data set and the lengthy training time.

Nonetheless, a method to infer poses using only the kinematic structure is still profoundly desirable as it could operate in an unsupervised manner. A co-evolutionary frame-

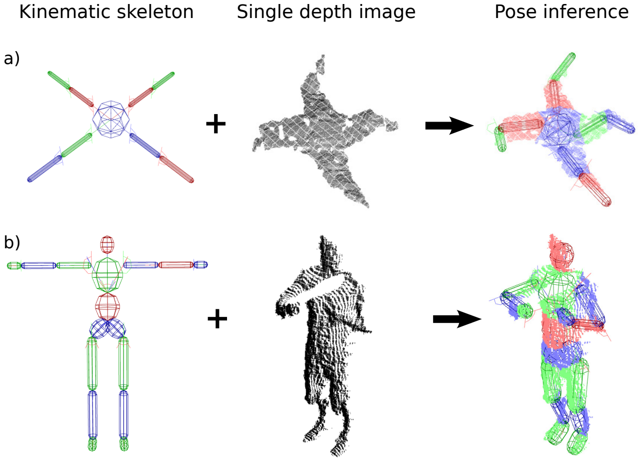


Figure 1: Inferring pose information from a single depth image and an arbitrary kinematic skeleton. Our framework is able to pose both (a) quadrupedal spider and (b) humanoid kinematic skeletons without any modifications or training.

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.
Copyright 2012 ACM 978-1-4503-1177-9/12/07 ...$10.00.

work is able to overcome the traditional limitations in kinematic pose estimation by leveraging competitive interactions between two populations to provide a tractable and reliable approach. Rather than evolving poses using the whole point cloud, a second population of subsampled points are simultaneously co-evolved to disambiguate the competing poses while also significantly reducing the computational load.

This paper presents a general approach to inferring poses of arbitrary kinematic skeletons from a single depth image without prior training[1]. The pose inference problem is defined as a model-based optimization and a learning algorithm based on co-evolutionary approaches is designed to efficiently search the vast parameter space. The primary contributions of the paper include: a volumetric parameterized description of kinematic skeletons, an effective fitness metric for pose estimation, and a co-evolutionary framework for the computationally intensive inference problem. The algorithm is applied to 34 and 78 degree of freedom models and reliably infers the model parameters for image reconstruction of point clouds with over 40,000 points. The inference algorithm is shown to be robust and can even accurately infer poses with non-trivial self-occlusions.

2. RELATED WORK

The vast majority of pose inference research focused on exclusively the human kinematic skeleton. Recent surveys [9, 10] describe two primary directions: pose assembly via probabilistic detection of body parts and example-based method. Pose assembly attempts to reconstruct the pose by first identifying body parts using pairwise constraints including aspect ratio, scale, appearance, orientation and connectivity. In contrast, example-based methods compare the observed point cloud with a database of samples. A primary limitation of these techniques stems from their supervised learning foundation: inference requires *labelled* training data and the generality of the inference algorithm depends on the content of the training data.

Body part classification has been successfully adapted to accurate, real-time implementations [3]. Shotton et al. described a particularly successful approach to human pose recognition that builds a probabilistic decision tree to first find an approximate pose of body parts, followed by a local optimization step [5]. While this technique is fast and reliable, it relies on significant training exclusive to the humanoid skeletal structure: 24000 core days of training on 1 million randomized poses. The algorithm learns a prior distribution of likely poses from the training set – consequently, the algorithm will do poorly for a test point cloud that is not within the learned prior distribution and thus lacks robustness with respect to arbitrary point clouds.

Due to the limitations of training-based algorithms, there are a variety of alternative approaches under investigation, including visual hull methods, interactive kinematic inference and particle-swarm optimization based methods.

Visual hull methods are approaches that do not depend on training data [6, 11]. In these approaches, an outer hull is mapped to the kinematic skeleton *a priori* using human experts, reducing the task to a hull-matching problem and rather utilizing the kinematic skeleton. For example, Gall et al. used laser-scanned models to find poses of complex models generated from animals and non-rigid garments in a markerless camera system [7]. This approach requires an accurate model per subject and cannot be readily adapted to generic or unknown subjects.

Katz et al. introduced an interactive method that infers relational representations of articulated objects by tracking visual features [12]. While this work does not focus on pose inference directly, it presents a framework to extract kinematic information from an unknown object using computer vision. However, it is limited to planar objects and requires a variety of interactions with the object.

A recent development in markerless pose estimation is the introduction of particle-swarm optimization, applied to kinematic skeletons of the human upper torso [13] and humanoid hands [14]. While these approaches approached real-time implementations on GPUs and provided direct searches on the pose parameter space, the largest demonstrated model contained 27 degrees of freedom with sparse point clouds containing around 1000 data points, a significantly simpler computational problem.

A primary application of unsupervised pose inference is teaching by demonstration, which has been shown to be an efficient and natural method to transfer knowledge to robots. Riley et al. used imitation to achieve human-like behaviour in highly-complex, humanoid robots [2] while Kober at al. explored how to use demonstrations to learn motor primitives and tackle complex dynamics problem via reinforcement learning [15]. Although, this illustrates potential uses for automated pose inference in robotics, current teaching by demonstration implementations rely on predefined transformations and there have been no attempts to generalize to arbitrary teachers. Better pose inference could also help in improving performance of human activity detection [16].

3. POSE INFERENCE ALGORITHM

Given a point cloud from the RGBD sensor, our algorithm poses a given kinematic skeleton to best explain the depth image. We first describe our volumetric parameterized representation of kinematic skeleton in Section 3.1, followed by the motivation and description of the fitness metric in Section 3.2. This section is concluded with a description of the evolutionary computation-based learning algorithm.

3.1 Kinematic models

Selecting a suitable representation of kinematic models is essential to inference, as an efficient encoding allows for generality as well as simplicity. We chose to represent the kinematic model as a collection of rigid links, organized in an acyclic graph structure (Fig. 2). The root node represents a frame of reference that describes the position and orientation of the model origin. The root parameters are unbounded.

Each child in the acyclic graph represents a rigid link that is modelled as a piecewise combination of cylinders and hemispheres. Although links are traditionally represented as line segments, a volumetric representation was chosen to match the 3D information of depth images. This parameterization defines a volume that is the locus of all points that are a constant radius away from a line segment.

Each link is described by three free parameters: link length, link radius and joint angle. The model parameters are defined using a parametric equation with two predefined bounds, and linear interpolation or SLERP [17] is used accordingly. The bounds allow for anatomically consistent definitions in the kinematic model. By defining the link radius with re-

[1] An earlier version of this work was presented at [8].

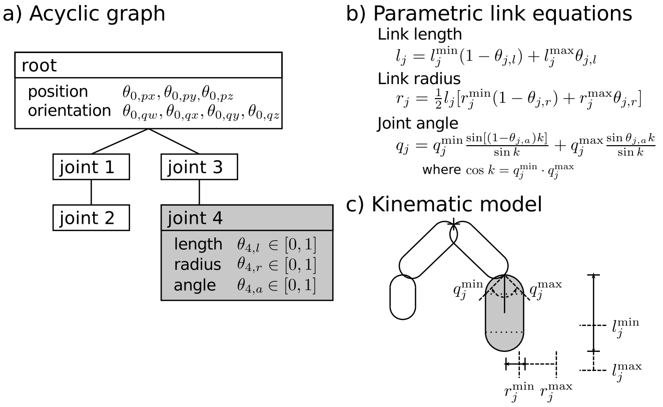

a) Acyclic graph
root
position $\theta_{0,px}, \theta_{0,py}, \theta_{0,pz}$
orientation $\theta_{0,qw}, \theta_{0,qx}, \theta_{0,qy}, \theta_{0,qz}$
joint 1
joint 3
joint 2
joint 4
length $\theta_{4,l} \in [0,1]$
radius $\theta_{4,r} \in [0,1]$
angle $\theta_{4,a} \in [0,1]$
b) Parametric link equations
Link length
$l_j = l_j^{\min}(1-\theta_{j,l}) + l_j^{\max}\theta_{j,l}$
Link radius
$r_j = \frac{1}{2} l_j [r_j^{\min}(1-\theta_{j,r}) + r_j^{\max}\theta_{j,r}]$
Joint angle
$q_j = q_j^{\min}\frac{\sin[(1-\theta_{j,a})k]}{\sin k} + q_j^{\max}\frac{\sin \theta_{j,a} k}{\sin k}$
where $\cos k = q_j^{\min} \cdot q_j^{\max}$
c) Kinematic model
$q_j^{\min}$
$q_j^{\max}$
$l_j^{\min}$
$l_j^{\max}$
$r_j^{\min}$
$r_j^{\max}$

Figure 2: a) The acyclic graph representation of the skeleton, b) the intermediate parametric equations and c) the corresponding visual depiction. Link 4 is highlighted for reference.

spect to the link length, the link maintains it length regardless of the radius and interpolates between a line segment to a complete sphere. This model allows for efficient geometric computations, such as finding distances of a point to the surface or collision detection (Eq. 3). Although individual links only provide a single degree of freedom, complex topologies, such as ball and sockets joints, can be obtained by cascading multiple zero-length links.

3.2 Fitness metric

As with all evolutionary algorithms, we must define a fitness metric that gives higher scores when the model better explains the observed point cloud. There are two criteria in this term: self-collisions must be avoided and the observed points must be well explained. Thus, we propose the following fitness metric:

$$F(\boldsymbol{\theta}) = -(1+\epsilon c)\left[\frac{1}{N}\sum_{n=0}^{N}\ln\left(1+\frac{||\vec{p}^{*}(\boldsymbol{\theta})-\vec{p}_n||}{\sigma}\right)\right] \tag{1}$$

where $\boldsymbol{\theta}$ are the collection of model parameters depicted in Fig. 2, ϵ is a small positive constant and c is the number of volumetric collisions between the links that are not adjacent in the graph structure or share the same predecessor node.

The summation term is a measure of the pose's ability to explain the point cloud. (Fig. 3). The term is based on the logarithmic error of the distance between an observed point, $\vec{p}_n$, and the nearest surface point of the candidate pose, $\vec{p}^*$, for all N points in the point cloud and σ is the standard deviation of the points in the point cloud.

The nearest surface point of the posed skeleton is defined as the minimum of the nearest surface point for each locally defined link, $\vec{p}_j$ for link j:

$$\vec{p}^{\,*}(\boldsymbol{\theta}) = \underset{\vec{p}_j(\theta_j)}{\operatorname{argmin}}\, ||\vec{p}_n - \vec{p}_j(\theta_j)|| \tag{2}$$

The links are composed of a combination of hemisphere and cylinder components (Section 3.1) and is volumetrically defined using a local representation aligned along the z-axis. The link is the locus of all points satisfying the following conditions:

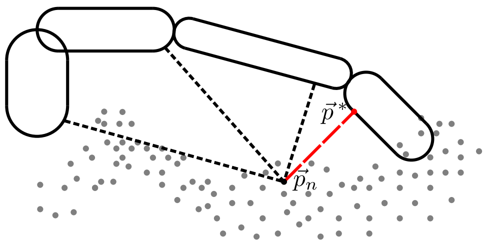


Figure 3: A visualization of the fitness metric evaluated for a single point. The distance between the point and the nearest surface is computed for each link, indicated by the dashed lines. Of these distances, the shortest length (highlighted) is used for the fitness calculation (Eq. 1).

$$T_0^j\vec{p}_j \Leftarrow \begin{cases} ||\vec{p}_j - \langle 0,0,r_j\rangle\,||^2 = r_j^2 & \text{, if } p_{j,z} < r_j \\ p_{j,x}^2 + p_{j,y}^2 = r_j^2 & \text{, if } r_j \leq p_{j,z} < l_j - r_j \\ ||\vec{p}_j - \langle 0,0,l_j - r_j\rangle\,||^2 = r_j^2 & \text{, if } p_{j,z} > l_j - r_j \end{cases} \tag{3}$$

where l_j and r_j are defined in Fig. 2.b) and T_0^j is the affine transformation from link j's frame to the origin.

The fitness metric, Eq 1, is related to a maximum likelihood with a heavy-tailed distribution. This distribution was chosen over popular exponential distributions for two reasons. First, the belief distribution from articulated kinematic structures is often multi-modal with isolated peaks. The heavy-tailed distribution allows more inclusive beliefs while the exponentially bounded distributions are more susceptible to exacerbating the effects of local optima by creating deeper valleys in the fitness landscape.

Second, a kinematic model often does not correspond directly to the visual hull of the depth image subject. Without prior information regarding the subject, the kinematic model only provides a rough approximation of the volumetric subject—details such as mass distribution, deformations at joints and clothing are not captured by kinematic models (compare the synthetic and real data in Fig. 7). Exponentially bounded distributions are not sufficiently robust to deal with this gap between the model and reality.

An essential feature of this fitness metric is its data-centric, as opposed to a model-centric, definition. The fitness function is defined strictly by the relationship of the data to the model, and not conversely. The primary benefit of this data-centric definition is its ability to deal with partial self-occlusion in an elegant manner. By avoiding a model-centric likelihood, there is no inherent penalty for positioning occluded links where no data exists. This approach can often lead to the good models by positioning and obstructing individual links such that the remainder of the link chain explains the observed data.

While this fitness metric has numerous advantages from a geometric and modelling perspective, it has many undesirable properties from a machine learning perspective. The fitness metric is not convex and, for articulated subjects, is densely populated with local optima. Furthermore, the parameter space can be extremely large for generalized models. As a result, we propose an evolutionary approach for this complex machine learning problem.

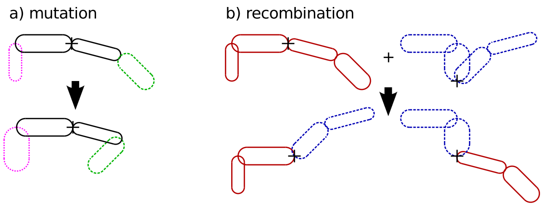


Figure 4: A visualization of the a) mutation and b) recombination operators. Mutation changed the parameters of the highlighted links. For recombination, the root was selected as the crossover point and the link chains were swapped to produce offspring.

3.3 Genetic algorithm

We propose using an genetic algorithm (GA) to determine the optimal kinematic parameters. GAs are stochastic, population-based, heuristic algorithms that iteratively selects and recombines solutions to produce increasingly better models. An evolutionary approach provides several benefits to the pose inference problem. First, GAs have been reliably applied to non-linear, non-convex optimization problems. Next, the population-based dynamics allow for an efficient search of large and high-dimensional parameter spaces. Finally, GAs are best suited for models with conditionally independent parameters, such as acyclic graphs, as recombination exploits locally optimized subrepresentations.

The population is initialized with randomly generated models: the root node position is initialized on a randomly selected point in the point cloud, the orientation is a quaternion sampled from a Gaussian distribution with a standard deviation of 1 followed by normalization, and the link parameters are interpolating values sampled from a uniform distribution between 0 and 1.

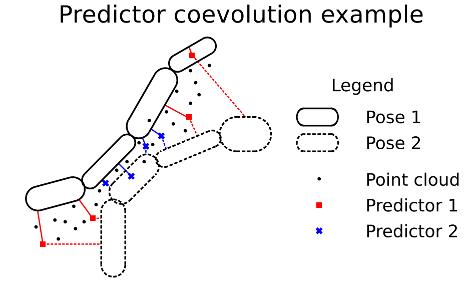
The inference algorithm then progresses via three processes: mutation, recombination and selection. Stochastic point mutations are applied to randomly parameters in a similar method to the initialization protocol, but localized to individual nodes (Fig. 4.a). For recombination, a random crossover point is selected for the existing parent pair, and the offspring are produced by swapping subgraphs (Fig. 4.b).

Finally, selection is the process of rejecting inferior models to maintain computational tractability – age-fitness Pareto selection was used [18]. Age-fitness Pareto selection is a selection algorithm that allows for the continuous addition of random individuals to avoid premature convergence. The number of generations that a model has existed, or the genotypic age, is logged and a multi-objective Pareto optimization is used to encourage promising individuals while simultaneously also protecting them from being dominated by more mature and optimized solutions. This selection method has been shown to increase the rate of convergence in high-dimensional evolutionary computation domains.

Although GAs are capable of efficiently finding general solutions, convergence to the local optima is often slow. Thus, a stochastic hill-climbing algorithm is applied in each iteration – a random vector is added to the model parameters and the changes are kept only if it results in a higher fitness.

3.3.1 *Co-evolutionary rank predictors*

A common criticism of evolutionary algorithms, and a prohibitive limitation in practice, stems from the computationally heavy demands of these algorithms. Often, the primary culprit in the computational requirements arises from fitness calculations. In pose inference, determining the fitness of a model requires repeatedly evaluating a local metric. A single point cloud can consist of tens of thousands of points and, since neighbouring points are similar, the computational resources required to calculate the exact fitness results in highly redundant and expensive computations.



Fitness ranking

Exact fitness

Fitness	Pose
-1.37	Pose 1
-1.78	Pose 2

Predictor 1

Fitness	Pose
-1.01	Pose 1
-1.99	Pose 2

Predictor 2

Fitness	Pose
-0.54	Pose 2
-0.63	Pose 1

Figure 5: An example of predictor co-evolution for pose inference. Two predictors, of four points each, are used to evaluate fitness (Eq. 1). Predictor 1 is superior as it obtains the same ranking as the fitness evaluated on the entire data set, while predictor 2 obtains an improper ranking.

Rather than using the entire point cloud, a lightweight approximation is substituted to alleviate the computational demands by co-evolving predictors. In this approach, the fitness is measured only on a dynamic subset of the data, which are co-evolved based on their ability to disambiguate the solution population [19], allowing for evolutionary progress through direct competition. Significant performance acceleration is achieved by a reduction of data in orders of magnitude using this dynamic sampling technique.

For point clouds, predictors are a small subset that references individual points in the point cloud. Fig. 5 illustrates a 2D example of predictor co-evolution for point clouds. The original point cloud consists of 32 points and there are two poses with two different predictors. Rather than evaluating the fitness of the poses on the complete point cloud, they are evolved on the predictor subset, which consists of only four points. Simultaneously, the predictors are evolved on their ability to obtain the same fitness ranking as one obtained by using the entire data set—in this example, predictor 1 provides a far superior fitness landscape over predictor 2. This direct competitive co-evolution, along with the reduced computation, greatly increases the solution convergence. For additional implementation details on rank predictor co-evolution, refer to [20, 21].

The complete evolutionary pose inference learning algorithm is summarized in Algorithm 1.

4. EXPERIMENTS

In this section, we describe the experiments performed to evaluate our method. We show both qualitative and quantitative results for two kinematic models and compare them against other approaches across a variety of metrics.

Algorithm 1 Evolutionary pose inference algorithm. Details of rank prediction and age-fitness Pareto selection are found in [20, 18].

```
for each model and predictor :
  initialize with random parameters
  model.age = 0

until termination condition :
  for each randomly selected pair of all models :
    recombine parents to produce offspring (Fig.4b)
    mutate both offspring (Fig.4a)
    add both offspring to model population
  for each model :
    calculate fitness using current predictor (Eq.1)
    hillclimb each model using current predictor (Eq.1)
    model.age = model.age + 1
  insert new random model into population with age = 0

  until model population is reduced to predefined size :
    for each randomly selected pair of all models :
      if a model has > age and < fitness than its pair :
        remove model from population

  for each predictor :
    for each randomly selected pair of all predictors :
      recombine parents to produce offspring
      mutate both offspring
      calculate fitness = ability to predict model ranking
      if offspring has >= fitness than parent :
        replace parent with offspring
  set current predictor as best predictor in population
```

4.1 Kinematic models

Two distinct models are used to evaluate the learning algorithm. The first is a *spider model*, based on a quadruped robot with 8 links (Fig. 1a). The model has 34 degrees of freedom but the links do not overlap workspaces. The second is a *humanoid model*, which consists of 17 links amounting to 78 degrees of freedom (Fig. 1b). In addition to the high dimensionality, the links' workspace have significant overlapping regions and there is no constraint on symmetry.

The parameter limits were chosen based on their real-world counterparts, but with an unusually wide range of variability. For example, the humanoid model with mean parameters was based on anatomical body proportions, but can deviate by 25%, which is far beyond the 95th percentile variation in human anatomy [22]. With such a variation in parameter limits, the algorithm is able to represent a wider range of poses than one would expect from real subjects.

4.2 Algorithms

As kinematic pose inference this of scope and complexity has been previously considered intractable, there are no related algorithms for direct comparison. Instead our algorithm is compared against baselines along three components:

1. **Sampling method: Co-evolved (C) vs Random (R)** – A comparison of the sampling method used to accelerate the computational performance. Co-evolution sampling is the method described in Section 3.3.1 and is implemented as 8 predictors, each as a subset of 64 points. There were 8 trainers, which were updated every 100 iterations. Random sampling is the baseline that consisted of a single predictor with 64 points selected each generation with a uniform distribution.

2. **Heuristic algorithm: Evolutionary (E) vs Hill-climbing (H)** – A comparison of the heuristic search algorithm. The evolutionary algorithm is the genetic algorithm described in Section 3.3. The evolutionary search parameters are: a population of 256 individuals with a mutation and recombination probability of 1% and 50%, respectively. In comparison, there is the hill-climbing alternative which was applied in parallel to 256 initially randomized models.

3. **Kinematic model: Volumetric (V) vs Linear (L)** – A comparison of the kinematic model. The volumetric model is the model described in Section 3.1, while the linear model is a variant that constrained the link radii to zero.

Table 1: Algorithm naming convention

	Sampling		Heuristic		Kinematic	
	Coev. (C)	Rand. (R)	Evol. (E)	Hill (H)	Vol. (V)	Lin. (L)
1. CEV	×		×		×	
2. REV		×	×		×	
3. CHV	×			×	×	
4. CEL	×		×			×

Rather than present every combination of the algorithmic variants, the co-evolved, evolutionary approach with volumetric models is used as a standard and three variants are presented where each component is reduced in a knock-out fashion. A summary of the four approaches; CEV, REV, CHV and CEL; is described in Table 1. Furthermore, the initial random population is provided as a baseline (Static) for comparing the effect of inference against random models.

All inference algorithms began with the same initial, random population. The learning algorithms were executed for 10^9 fitness evaluations, which approximately amounts to 10,000 iterations. On a single core 2.8GHz Intel processor, this required approximately 30 and 70 minutes per image for the spider and humanoid models, respectively.

4.3 Synthetic depth data

For a quantitative comparison, a synthetic data set of 128 randomly sampled poses was generated for each model based on the initialization protocol described in Section 3.3. A noiseless point cloud was generated via a ray tracing algorithm using 640×480 rays on a field of view of $57^\circ \times 48^\circ$. The model parameters were sampled uniformly, resulting in a varied data set. The baseline algorithms were compared using four metrics:

1. **Fitness metric (Fit.)** The fitness metric which was used for parameter optimization (Eq 1).

2. **Mean point distance error (Abs.)** The mean distance between the cloud points and the closest point on the model surface: $E = \frac{1}{N}\sum_{n=0}^{N} ||\vec{p}^* - \vec{p}_n||$

3. **Root mean squared point error (RMS)** The root mean squared of the distance between the cloud points and the closest surface point: $E = \sqrt{\frac{1}{N}\sum_{n=0}^{N} ||\vec{p}^* - \vec{p}_n||^2}$

4. **Mean joint distance error (Joint)** The inferred joint locations with those from the ground truth model: $E = \frac{1}{J}\sum_{j=0}^{J} ||\vec{l}_{j,m} - \vec{l}_{j,i}||$ where $\vec{l}_{j,m}$ and $\vec{l}_{j,i}$ are the jth joint positions for the ground truth model and inferred model, respectively.

Table 2: Performance on synthetic images

	Spider model			
	Fit. [$\times10^{-2}$]	Abs. [$\times10^{-3}$]	RMS [$\times10^{-3}$]	Joint [$\times10^{-2}$]
CEV	$\mathbf{-1.07 \pm .03}$	$\mathbf{1.3 \pm .4}$	$\mathbf{2.0 \pm .6}$	$\mathbf{8 \pm .7}$
REV	$-1.36 \pm .03$	$1.8 \pm .7$	$2.8 \pm .8$	$9 \pm .8$
CHV	$-1.62 \pm .05$	$2.1 \pm .7$	3 ± 1	10 ± 1
CEL	$-8.1 \pm .3$	$8.0 \pm .4$	11 ± 3	10 ± 1
Static	$-22.6 \pm .3$	34 ± 6	53 ± 8	29 ± 2

	Humanoid model			
	Fit. [$\times10^{-2}$]	Abs. [$\times10^{-2}$]	RMS [$\times10^{-2}$]	Joint [$\times10^{-1}$]
CEV	$\mathbf{-2.51 \pm .03}$	$\mathbf{1.2 \pm .3}$	$\mathbf{1.8 \pm .5}$	$\mathbf{1.6 \pm .9}$
REV	$-2.85 \pm .06$	$3.1 \pm .8$	$4.8 \pm .7$	$5.6 \pm .8$
CHV	$-6.0 \pm .1$	$3.4 \pm .9$	$5.3 \pm .8$	6 ± 1
CEL	$-8.5 \pm .1$	$4.8 \pm .3$	$6.7 \pm .6$	$8.6 \pm .7$
Static	$-21.8 \pm .3$	56 ± 3	58 ± 6	31 ± 6

These metrics provide an important basis of comparison as solely relying on the fitness metric presents a skewed perspective. The fitness metric was specifically designed to solve the inference problem. However, due to the logarithmic nature of Eq. 1, relative difference in scores are often misleading. Absolute error and RMS provide a more intuitive measure of performance. However, the best metric is the joint error which leverages information from the ground truth to provide an objective measure of performance.

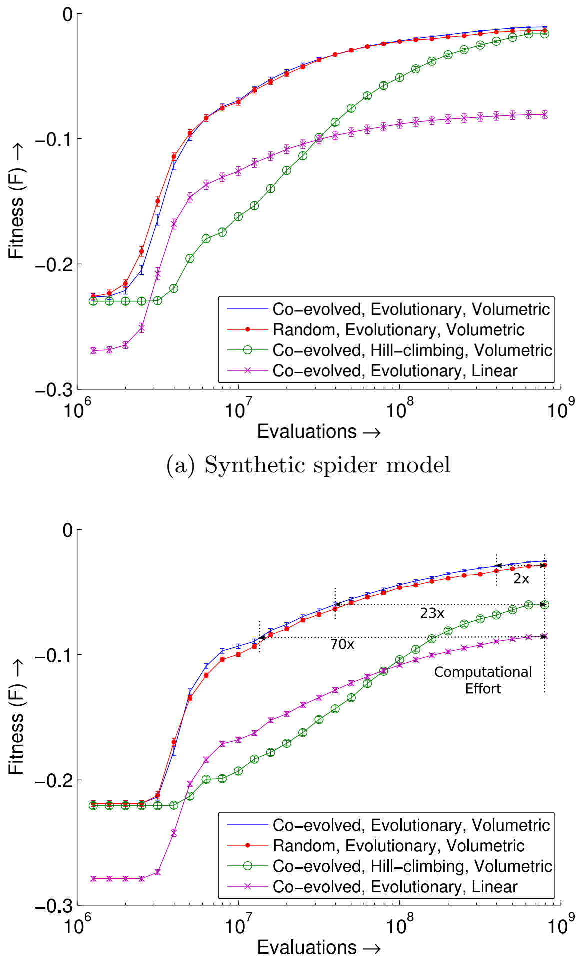
In Table 2, we compare the approaches across the various metrics of the best individual after 10^9 fitness evaluations. A single run of each algorithm was performed for each image and the results are averaged over 128 images with standard error reported. By comparing the joint error metric, it is clear that the algorithmic variations are not critical to performance for low-dimensional problems – the problem space is sufficiently small that all four approaches comprehensively cover the search space within the allotted computational effort. Nonetheless, CEV's performance indicates that it is able to fine tune the parameters in order to achieve the best results over the entire range of metrics.

However, the algorithmic variations play vital role in inferring the higher dimensional humanoid model. First, CEV is able to achieve the best metric scores, with at least a 3.5-fold improvement in the joint error metric. While REV was able to achieve a similar fitness score to CEV, it is clear that is more susceptible to local optima as it is significantly worse in the other metrics – in fact, REV is only marginally better than CHV. CHV and CEL produced increasingly inferior models, respectively, across the metrics.

In Fig. 6, we compare the fitness to computational effort for both models. Although relative fitness values are misleading and should not be compared directly, fitnesses are still meaningful as benchmarks to measure how fast an algorithm finds equivalently performing models.

For the low-dimensional spider model, CEV and REV achieve similar learning rates, and drastically outperform CHV and CEL. REV is able to provide superior early optimization over CEV, but is overtaken around 10^7 evaluations. This slow start suggests that the competitive co-evolution in CEV requires more overhead to initially build up good individuals in both the solutions and predictor populations, but is able to further optimize the populations when compared to the random subsampling approach.

In the high-dimensional humanoid model, the trends are similar but the discrepancy between the approaches is further amplified. CEV performs significantly better than the other approaches. In fact, for the same final fitness at 10^9 evaluations, CEV provides a 2-, 23- and 70-fold reduction in computation effort over REV, CHV and CEL, respectively.



(a) Synthetic spider model

(b) Synthetic humanoid model

Figure 6: Fitness of the best individual vs. computational effort averaged over 128 images. Error bars indicate standard error.

4.4 Real depth data

We compared the algorithms using depth images captured via a Kinect camera [23]. The Kinect platform was ideal as the consumer hardware is a popular platform for robotic applications with limited accuracy and resolution.

For the spider model, a robot with eight limbs with fourteen degrees of freedom was arranged in four distinct poses, and five images ranging in inclination angles were taken per pose. The spider model used the same kinematic structure but had 34 degrees of freedom to account of unknown limb lengths and thickness. The variation in inclination angles provided numerous examples of self-occlusion. For the humanoid model, eight images were taken of four human subjects, totalling to 32 images. The images in both data sets were pre-processed with manual background subtraction so only the pose of interest remained.

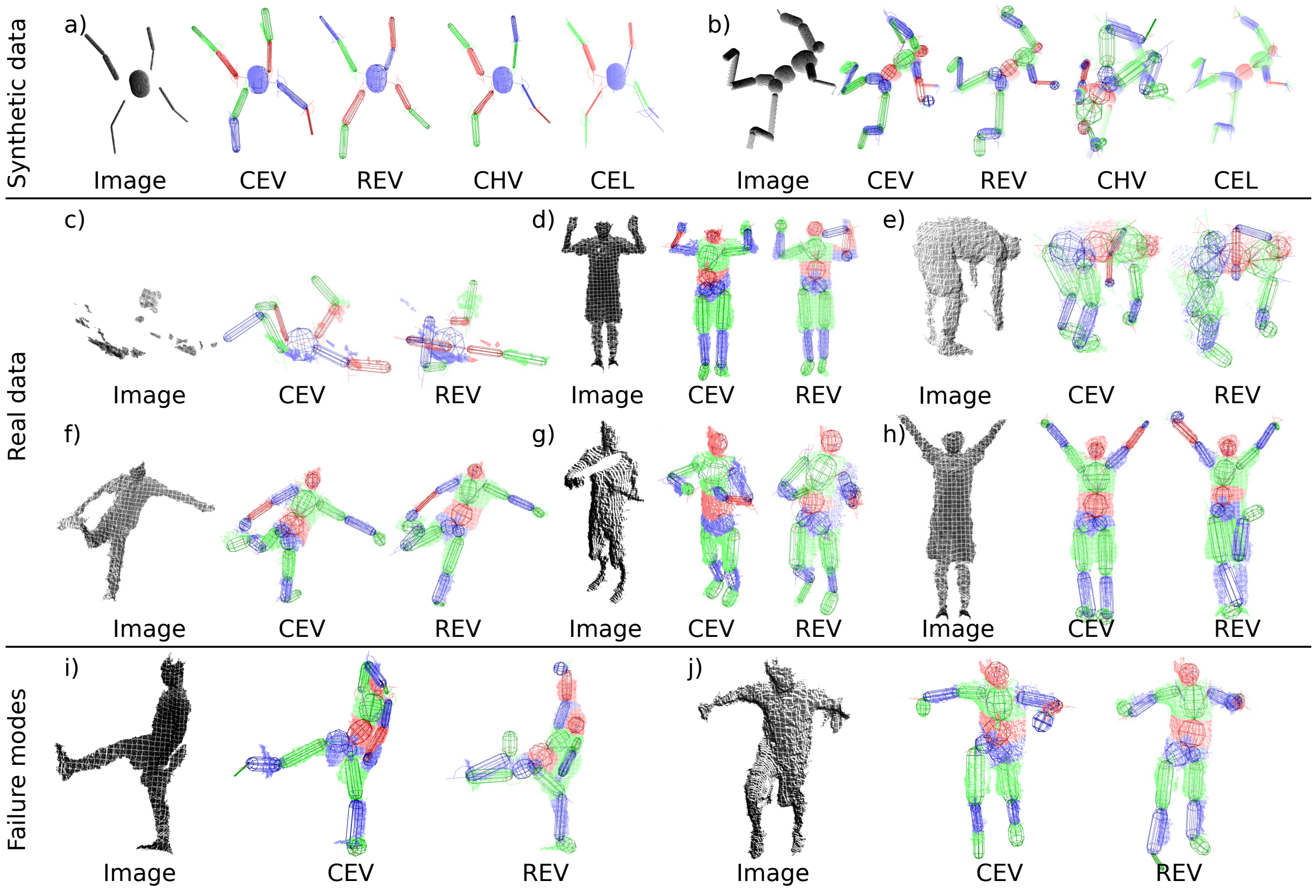


Figure 7: Pose inference examples on synthetic and real world data. Note the point clouds are not pre-segmented and the colourized links are the result of post-processing for the ease of interpretability.

Table 3: Performance on real images

	Spider model		Humanoid model	
	Score [1-5]	LEP [%]	Score [1-5]	LEP [%]
CEV	**4.9 ± .1**	**1 ± 1**	**4.1 ± .9**	**16 ± 4**
CHV	4.2 ± .2	12 ± 2	3.2 ± .9	45 ± 5

For the real depth images, only CEV and REV models were applied as they were the superior approaches from the synthetic data experiments. A single run of each algorithm was performed for each image. Unfortunately, quantitative metrics were unavailable due to the lack of ground truth data. Instead, the resulting models were rated by four volunteers on a scale of 1-5, with 5 as a perfect inference. Furthermore, the number of incorrectly positioned links was reported, and used to calculate the probability of misplacing a link (the Link Error Probability or LEP).

The results are summarized in Table 3 with standard error reported. For the low-dimensional spider model, the algorithms are comparable with a slight advantage to CEV in both metrics. However, comparing the high-dimensional humanoid model provides a sharp contrast – REV had difficulties inferring the original pose and often misplaced limbs.

The difference in the algorithms' performance is evident in the inference examples shown in Fig. 7. CEV is able to consistently infer a reasonable approximation to the ground truth, while REV is often caught in spurious local optima that, when rendered, have little in common with the ground truth. The inference algorithm was successful even in cases of significant self-occlusion (Fig. 7c,e,g). Although large portions of a limb or the torso were missing, CEV was able to place links in the position of occluded points and infer the correct pose, while REV contorted the kinematic skeleton to find a locally optimal pose.

The inferred models were still reasonable even in CEV's failure modes (an average score lower than 4), especially compared to its REV counterpart. The failure modes were a result of the inference algorithm settling on a local optima within the allotted computational effort. Superior poses might be found with more computational effort, but there is no guarantee of convergence.

Finally, additional analysis indicates that predictor coevolution plays a critical role. By logging which points were referenced, a histogram displaying the frequency that a point was used in the predictor was generated (Fig. 8). As the predictor selected 64 points simultaneously, a 375-fold speed-up over using the whole point cloud was obtained in this example. Since points are selected to best disambiguate competing models, a point with higher selection frequency is more useful than its peers for fitness evaluation. The histogram clearly shows a non-uniform distribution, indicating that specific data are more relevant than others.

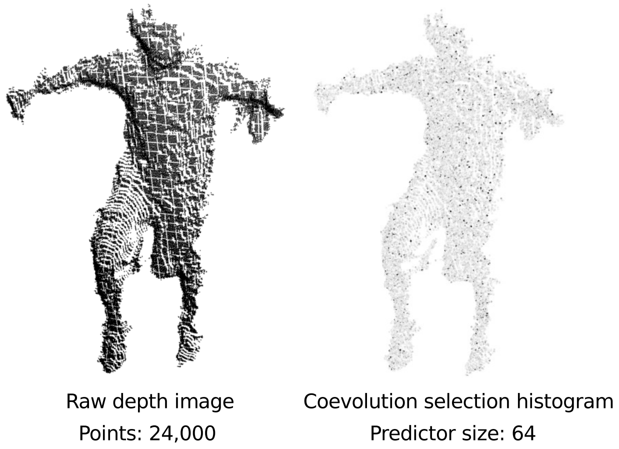


Figure 8: A histogram indicating the frequency, proportional to colour intensity, that a point was selected to be used in a predictor.

5. CONCLUSION AND FUTURE WORK

The proposed framework of using volumetric kinematic representations and searching for pose parameters using coevolutionary algorithms based on a heavy-tailed distribution was validated. The poses for 34 degree of freedom spider model and 78 degree of freedom humanoid models were reliably inferred for the synthetic and real RGBD images, even in cases of self-occlusion. The co-evolutionary algorithm achieves a 3.5-fold increase in pose accuracy and a two-fold reduction in computational effort over the baselines.

Although our algorithm is slower than state of the art methods, it is not dependent on extensive training sets. Rather, this work successfully shows that articulated kinematic structures can indeed be posed in an unsupervised manner, a problem previously considered intractable. This is a initial step towards fast, unsupervised methods that are more robust than their trained counterparts.

While it is unlikely that kinematic pose inference will be quicker than trained approaches, fast or real-time implementations may be possible. Evolutionary algorithms are naturally parallel and there is room for further optimization. Other areas of interest include using inferred poses to extract kinematic transformations via non-isomorphic structures.

6. ACKNOWLEDGMENTS

This work was supported in part by NIH NIDA grant RC2 DA028981, NSF CDI Grant ECCS 0941561, DTRA grant HDTRA 1-09-1-0013 and NSERC's PGS program.

7. REFERENCES

[1] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, pp. 109–116, 2004.

[2] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling real-time full-body imitation: a natural way of transferring human movement to humanoids," in *ICRA*, 2003.

[3] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real time motion capture using a single time-of-flight camera," in *CVPR*, 2010.

[4] L. Sigal, S. Bhatia, S. Roth, M. Black, and M. Isard, "Tracking loose-limbed people," in *CVPR*, 2004.

[5] J. Shotton, A. Fitzgabbon, M. Cook, T. Sharp, M. Finnochio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from a single depth image." in *CVPR*, 2011.

[6] S. Corazza, L. Mundermann, A. M. Chaudhari, T. Demattio, C. Cobelli, and T. Andriacchi, "A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach." *Annals of Biomedical Engineering*, vol. 34, no. 6, pp. 1019–1029, 2006.

[7] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenbahn, and H. Seidel, "Motion capture using joint skeleton tracking and surface estimation." in *CVPR*, 2009.

[8] D. L. Ly, A. Saxena, and H. Lipson, "Pose Estimation from a Single Depth Image for Arbitrary Kinematic Skeletons," in *RGB-D Workshop at RSS*, 2011.

[9] T. Moeslund, A. Hilton, and V. Kruger, "Survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[10] R. Poppe, "Vision-based human motion analysis: an overview," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 4–18, 2007.

[11] S. Knoop, S. Vacek, and R. Hillmann, "Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model," in *ICRA*, 2006.

[12] D. Katz, Y. Pyuro, and O. Brock, "Learning to manupulate articulated objects in unstructured environments using a grounded relational representation," in *RSS*, 2008.

[13] C. Robertson and E. Trucco, "Human Body Posture via Hierarchial Evolutionay Optimization," in *BMCV*, 2006.

[14] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient Model-based 3D tracking of Hand Articulations using Kinect," in *BMCV*, 2009.

[15] J. Kober and J. Peters, "Learning motor primitives for robotics." in *ICRA*, 2009.

[16] J. Sung, C. Ponce, B. Selman, and A. Saxena, "Unstructed Human Activity Detection from RGBD Images," in *ICRA*, 2012.

[17] K. Shoemake, "Animating rotation with quaternion curves," in *SIGGRAPH*, vol. 19, no. 3, 1985.

[18] M. D. Schmidt and H. Lipson, "Age-Fitness Pareto Optimization," *Genetic Programming Theory and Practice*, vol. 8, pp. 129–146, 2010.

[19] A. Bucci, "Emergent geometric organization and informative dimensions in coevolutionary algorithms," Ph.D. dissertation, Brandeis University, 2007.

[20] M. D. Schmidt and H. Lipson, "Predicting Solution Rank to Improve Performance," in *GECCO*, 2010.

[21] D. L. Ly and H. Lipson, "Trainer Selection Strategies for Coevolving Rank Predictors," in *CEC*, 2011.

[22] C. C. Gordon, "US Army Anthropometric Survey Database: Downsizing, Demographic Change, and Validity of the 1988 Data in 1996," US Army Natick Research Labs, Tech. Rep. TR-07/003, 1998.

[23] Microsoft Corp., Kinect for Xbox 360.