The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation

Jonathan Taylor[†⋆] Jamie Shotton[†] Toby Sharp[†] Andrew Fitzgibbon[†]

[†]Microsoft Research Cambridge [⋆]University of Toronto

Abstract

Fitting an articulated model to image data is often approached as an optimization over both model pose and model-to-image correspondence. For complex models such as humans, previous work has required a good initialization, or an alternating minimization between correspondence and pose. In this paper we investigate one-shot *pose estimation: can we directly infer correspondences using a regression function trained to be invariant to body size and shape, and then optimize the model pose just once? We evaluate on several challenging single-frame data sets containing a wide variety of body poses, shapes, torso rotations, and image cropping. Our experiments demonstrate that one-shot pose estimation achieves state of the art results and runs in real-time.*

1. Introduction

We address the problem of estimating the pose and shape of an articulated human model from static images. Human pose estimation has long been a core goal of computer vision, but despite the launch of commodity systems [19], there is considerable room for improvement in accuracy.

Following recent work, we combine generative and discriminative approaches. *Generative* approaches aim to explain the image data by optimizing an energy function defined over the parameters of a graphics-like model. Models of sufficient capacity can describe the data well, and *if a good minimum of the energy can be found*, provide excellent results. However, it is almost invariably the case that high-capacity models have many local minima, meaning that finding a good minimum requires either expensive global search [13, 14], depends on a good initial estimate, or is applicable only to a limited range of poses. *Discriminative* approaches [1, 24, 25] directly predict pose parameters from the image, e.g. by training a regression model on many examples. Recently, *hybrid* methods [22] combining discriminative and generative models have been shown to yield impressive results on real-world sequences. For example, Baak et al. [3] track a skinned surface model in depth image sequences by combining initial estimates from the previous frame with discriminative estimates obtained whenever five body extremities (hands, feet, and head) are detected by a data-driven process. They show impressive results on dynamic fast-moving sequences, but the system is restricted to near-frontal poses, and fast movements require all five extremities to be visible. Another recent system of note is that employed in the Kinect video game platform. Precise details of the end-to-end tracking algorithm are not public. However, it appears clear that the discriminative front-end reported in [23], which produces a set of hypotheses for each joint independently, is combined with a skeleton model to produce kinematically consistent pose estimates, again from depth sequences. When the input is not depth images, but multiple 2D silhouettes, hybrid methods again demonstrate excellent performance [21].

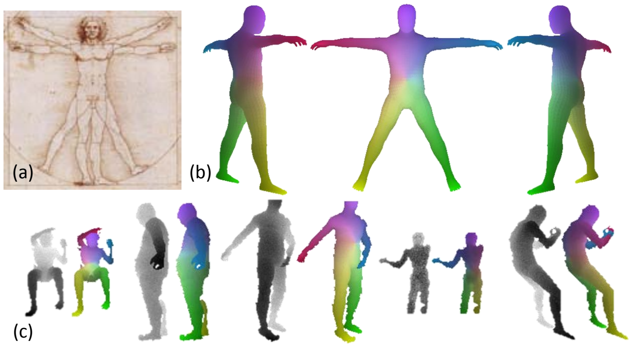

Figure 1. **(a)** Da Vinci's Vitruvian Man [11]. **(b)** The Vitruvian Manifold, as defined in Sec. 2. Viewed here from back-left, front, and back-right, using color to indicate position on the manifold. **(c)** Example (depth, correspondence) training image pairs. Note how the correspondence images adapt across body shape and pose, allowing us to learn to predict these correspondences in arbitrary test images.

It is common to express generative methods in terms of *correspondences* between features in the input images and points on the model surface. Given correct correspondences, as noted in [21], local optimization converges reliably even from distant poses. Previously, correspondences have been obtained from an initial estimate of model pose parameters: the model is rendered in the initial pose, and correspondences are found using some variant of iterated closest points (ICP), perhaps with compatibility functions based on shape contexts or related features [4, 6, 9, 21]. However, this means that the initial estimate of *pose* must be close enough that reasonable correspondences are found.

In this paper we propose an alternative approach (illustrated in Fig. 2) where we compute an estimate of *correspondences* from image to model independently of any initial pose. Specifically, we employ a regression forest [7, 10] to quickly predict at each pixel the distribution over likely correspondence to the body model surface. Unlike ICP methods, we do not need to iterate between optimization

978-1-4673-1228-8/12/$31.00 ©2012 IEEE

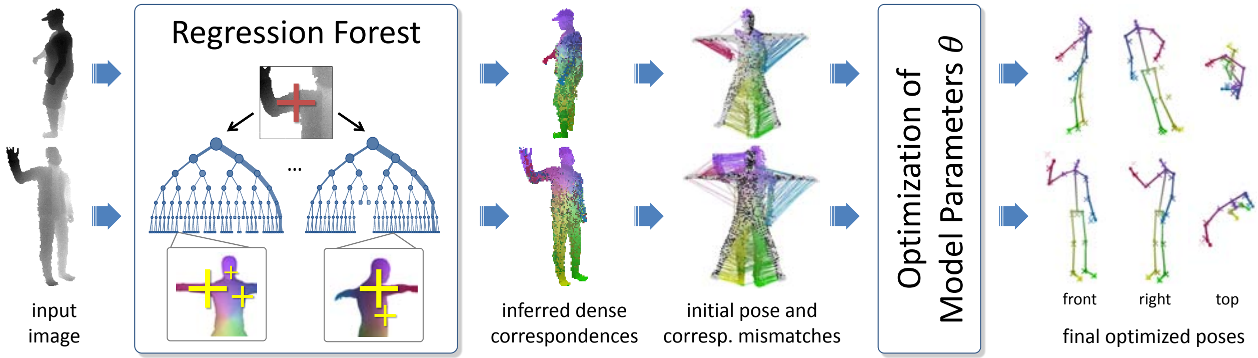

Figure 2. **Overview**. Our algorithm applies a regression forest to an image window centered around each pixel. Each leaf node in the forest contains a distribution over coordinates on the Vitruvian manifold; modes of these distributions are marked by the yellow crosses. The most confident mode across the forest is taken at each pixel as the correspondence to the model. This allows the use of standard continuous optimization over our energy function (Eq. 11). The result is one-shot pose estimation: a quick and reliable convergence to a good pose estimate, without separate initialization or alternating minimization of pose and correspondence.

and correspondence finding: our regression forest is able to directly estimate correspondences sufficiently reliably to enable a single 'one-shot' optimization to a robust result.

Our work builds on Pilet *et al.* [20]. They predict sparse correspondences to a deformable 2D mesh using a random forest trained on multiple views of a rigid exemplar. We extend this approach by inferring dense correspondences to the 3D surface of an articulated human mesh model, *invariant to pose and shape variations*. Illustrated in Fig. 1, this learned invariance requires training data containing varied articulation and shape, and yet known correspondences to a canonical model.

We address *real-world* pose estimation from depth and multi-view silhouette images. The challenges include arbitrary poses, a wide range of body shapes and sizes, unconstrained facing directions relative to the camera, and potentially cropped views of the person. We further focus on the particularly hard problem of *single frame* pose estimation where no temporal information is assumed. Recent work [23] has shown the value of this approach for scenarios such as video gaming where fast motion is common and the system must be robust over periods of hours.

In summary, our key contributions are the use of an efficient learned regression function to directly predict image to model correspondences for articulated classes of object, and the demonstration that this allows one-shot human pose estimation that considerably advances the state of the art. An additional contribution is a much more stringent and realistic test metric (number of fully correct frames) than the variants on average joint error which previous work has quoted. For example, in Fig. 3(a) the current state of the art achieves only 20% accuracy (our algorithm scores around 45%).

2. Preliminaries

Our goal is to determine the pose parameters $\theta \in \mathbb{R}^d$ of a linearly skinned [3, 4] 3D mesh model so as to explain a set of image points $D = \{x_i\}_{i=1}^n$. For our main results we use image points that have a known 3D position, *i.e.* $x_i \in \mathbb{R}^3$, obtained using a calibrated depth camera. Following standard practice, we assume a reliable background subtraction.

The 3D mesh model employs a standard hierarchical body joint skeleton comprising a set of $L = 13$ limbs. Each limb l has an attached local coordinate system related to the world coordinate system via the transform $T_l(\theta)$. This transform is defined hierarchically by the recurrence

$$T_l(\theta) = T_{\text{par}(l)}(\theta) R_l(\theta) \quad (1)$$

$$T_{\text{root}}(\theta) = R_{\text{glob}}(\theta) \quad (2)$$

where $\text{par}(l)$ indicates the parent of limb l in the hierarchy, and root indicates the root of the hierarchy. We use $R_l(\theta)$ to denote the relative transformation from the coordinate frame of limb l to that of its parent. This 4x4 matrix contains a fixed translation component and a parameterized rotation component formed from the relevant elements of θ. Finally, R_{glob} is a global transformation matrix that rotates, translates, and isotropically scales the model based on particular elements in θ. To allow the use of efficient off-the-shelf optimizers, we over-parameterize each rotation as the projection of an unconstrained 4D quaternion onto the unit sphere. This gives us a total of $4L + 4 + 3 + 1 = 60$ degrees of freedom in the parameter vector θ.

Using standard linear skinning [4, 3], the limbs allow us to define a mesh model surface. The mesh contains m *skinned vertices* written $\mathcal{V} = \{v_j\}_{j=1}^m$. Each vertex v_j is defined as

$$v_j = \left(p_j, \{(\alpha_{jk}, l_{jk})\}_{k=1}^K\right) , \quad (3)$$

where: *base vertex* p_j represents the 3D vertex position in a canonical pose θ_0 as a homogeneous vector; the α_{jk} are positive *limb weights* such that $\forall_j \sum_k \alpha_{jk} = 1$; and the $l_{jk} \in \{1, \ldots, L\}$ are *limb links*. In our model, the number of nonzero limb weights per vertex is at most 4, so $K = 4$. The position of the vertex given a pose θ is then output by a global transform M which linearly combines the associated limb transformations:

$$M(v_j;\theta) = \pi(\sum_{k=1}^{K} \alpha_{jk} T_{l_{jk}}(\theta) T_{l_{jk}}^{-1}(\theta_0) p_j) \qquad (4)$$

where π is the standard conversion from 4D homogeneous to 3D Euclidean coordinates.

The mesh further contains a set $\mathcal{T}$ of triangles as triplets of vertex indices. Our mesh is watertight and the transformed vertices thus define a closed continuous surface as the union of triangles

$$\mathcal{S}(\theta) = \bigcup_{(j_1,j_2,j_3)\in\mathcal{T}} \text{Triangle}(M(v_{j_1};\theta), M(v_{j_2};\theta), M(v_{j_3};\theta)) \,. \qquad (5)$$

Because the canonical pose θ_0 induces a surface $S(\theta_0)$ that resembles the Vitruvian Man [11], we call $S(\theta_0)$ the Vitruvian Manifold (see Fig. 1(b)).

The goal, restated, is then to find the pose θ, whose induced surface $\mathcal{S}(\theta)$ best explains the image data. A standard way to approach this is to introduce a set of correspondences $U = [u_1, ..., u_n]$, such that each correspondence $u_i \in \mathcal{V}$. One then minimizes

$$E_{data}(\theta, U) = \sum_{i=1}^{n} w_i \cdot d(x_i, M(u_i;\theta)) \qquad (6)$$

where w_i weights data point i and $d(\cdot,\cdot)$, is some distance measure in $\mathbb{R}^3$. An alternating minimization (or block coordinate descent) over θ and U would yield a standard articulated ICP algorithm. Unfortunately, convergence is unlikely without a good initial estimate of either θ or U. The key to the success of our method is the use of a discriminative appearance model to estimate U directly instead of the more common approach of initializing θ. Our experiments show that these correspondences further prove accurate enough to avoid the need to alternate optimization of θ and U.

3. Predicting Correspondences

Random forests [2, 7] have proven powerful tools for classification [18], regression [16], and more [10]. We employ a regression forest to predict the correspondences U by regressing from images to distributions over an embedding of our surface model, and thus to mesh vertices.

A regression forest is a set of binary trees. Each non-terminal node contains a binary *split function*. This is a decision function computed on an image window centered at pixel i, for which we employ the fast depth comparison split functions of [23]. Each terminal (leaf) node contains a *regression model*, to which we will come back shortly. At test time, each foreground pixel i is passed into each tree in the forest. A path is traversed from the root down to a leaf, branching left or right according to the evaluation of the split functions. Finally, we aggregate across trees the regression models at the leaves reached.

Ideally, our regression models would store a distribution over the model surface, but this is hard to represent efficiently. As a proxy to this, we use distributions defined over the 3D space in which the Vitruvian manifold $S(\theta_0)$ is implicitly embedded. So long as these distributions lie close to the manifold, they should be fairly accurate. For further efficiency, we represent the distributions as a small set of confidence-weighted modes $G = \{(\hat{u}, \omega)\}$, where $\hat{u} \in \mathbb{R}^3$ is the position of the mode in the embedding space, and ω is the scalar weighting. This set G can be seen as an approximation to a Gaussian mixture model. To aggregate the regression models across the different trees, we simply take the union of the various leaf node modes G.

We are left with the task of predicting pixel i's correspondence $u_i \in \mathcal{V}$ from these aggregated distributions. To do this, we take the mode $\hat{u}$ with largest confidence value ω, and perform a nearest-neighbor projection onto the manifold as $u = \text{argmin}_{v\in V} \|\hat{u} - M(v;\theta_0)\|_2$. This simple 'winner-takes-all' correspondence prediction approach has proven highly effective, partly due to our use of a robust distance measure $d(\cdot,\cdot)$; see below. Some qualitative examples of the correspondences achieved are illustrated in Figs. 2 and 4. Of course, there is potentially a rich source of information in the regression models that we are not currently using. Exploiting this effectively remains as future work. For an optimized implementation, one can thus store at each leaf only the single vertex index j and confidence weight ω resulting from projecting the mode with largest confidence in advance.

3.1. Learning the forest

To train the random forests we use the data from [23]. This is a set of synthetic images, each rendered using computer graphics, to produce a depth or silhouette image. The parameters of the renders (pose, body size and shape, cropping, clothing, *etc.*) are randomly chosen such that we can aim to learn invariance to those factors. Alongside each depth or silhouette image is rendered a correspondence image, where colors are used to represent the ground truth correspondences that we aim to predict using the forest. Examples are given in Fig. 1(c).

Crucially, the ground truth correspondences must align across different body shapes and sizes. For example, the correspondence for the tip of the right thumb should be the same, no matter the length of the arm. This was accomplished by deforming a base mesh model, by shrinking and stretching limbs, into a set of 15 models ranging from small child to tall adult. The vertices in these models therefore exactly correspond to those in the base model, as desired. This allows us to render the required correspondence image using a simple vertex lookup, no matter which body model is randomly chosen. This can also be seen in Fig. 1(c).

Given this data, we can now train the trees. Following [16] we use a forest where the tree structure is trained for the body part classification objective in [23]. This proxy to a regression objective was shown to work better on multimodal data than the standard variance minimization. We then 'retro-fit' the regression models at the leaves, as follows. We collect the set of training pixels reaching each leaf, and compute for each pixel i the embedding space position $\hat{u}_i = M(u_i; \theta_0) \in \mathbb{R}^3$ given the ground truth $u_i \in \mathcal{V}$. We then cluster the $\hat{u}$s using mean shift mode detection [8], adopting a Gaussian kernel with a fixed bandwidth parameter. The design of the Vitruvian pose allows the use of Euclidean distance in the embedding space to efficiently approximate geodesic distances locally on the manifold. The weight ω is set as the number of training pixels clustered to each mode.

4. Energy Function

The energy defined in Eq. 6 is quite standard, and because it sums over the data, it avoids some common pathologies such as an energy minimum when the model is scaled to zero size. To deal with mislabelled correspondences, it is sensible to specify $d(x, x') = \rho(\|x - x'\|)$ where $\rho(\cdot)$ is a robust error function. We use the Geman-McClure [5] function $\rho(e) = \frac{e^2}{e^2+\eta^2}$ due to its high tolerance to outliers. We choose $w_i = z_i^2$ as the pixel weighting, derived from the point's depth $z_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} x_i$ to compensate for proportionately fewer pixels and therefore contributions to the energy function as depth increases.

Unfortunately, deficiencies remain with Eq. 6, particularly with self-occlusion. In the following sections, we build up further terms to form our full energy Eq. 11.

Visibility term. For given parameters θ, the data term in Eq. 6 allows either visible or invisible model points to explain any observed image point. A more realistic model might include hidden-surface removal inside the energy, and allow correspondences only to visible model points. However, a key to our approach, described below in Sec. 4.1, is to use fast derivative-based local optimizers rather than expensive global optimizers, and thus an efficient energy function with well-behaved derivatives is required. Despite some excellent recent work in computing derivatives of mesh projections under visibility constraints [12], handling visibility remains quite difficult. One common strategy, holding visibility fixed during the optimization, greatly hinders the optimizer.

We adopt a useful approximation which is nevertheless effective over a very large part of the surface: we define visibility simply by marking back-facing surface normals. To do so, we define function $N(u; \theta)$ to return the surface normal of the model transformed into pose θ at $M(u; \theta)$. Then u is marked visible if the dot product between $N(u; \theta)$ and the camera's viewing axis A (typically $A = [0, 0, 1]$, the positive Z axis) is negative. One might then write

$$E_{\text{vis}} = \sum_{i=1}^{n} w_i \begin{cases} d(x_i, M(u_i; \theta)) & N(u; \theta)^\top A < 0 \\ \tau & \text{otherwise} \end{cases} \tag{7}$$

with τ a constant that must be paid by backfacing vertices. In practice, using a logistic function $\sigma_\beta(t) = \frac{1}{1+e^{-\beta t}}$ with 'sharpness' parameter β is preferable to a hard cutoff:

$$E'_{\text{vis}} = \sum_{i=1}^{n} w_i \big[V_i(\theta) \cdot d(x_i, M(u_i; \theta)) + (1 - V_i(\theta)) \cdot \tau\big] \tag{8}$$

where visibility weight $V_i(\theta) = \sigma_\beta(-N(u_i; \theta)^\top A)$.

Pose Prior. To further constrain the model, particularly in the presence of heavy occlusion, we use a conventional prior, the negative log of a Gaussian on the pose vector:

$$E_{\text{prior}} = (\theta - \mu)^\top \Lambda (\theta - \mu) \tag{9}$$

where μ and Λ, the mean and inverse covariance of the Gaussian, are learned from a set of training poses.

Intersection Penalty. Lastly, we add a term to discourage self intersection by building a coarse approximation to the interior volume of $\mathcal{S}(\theta)$ with a set of spheres $\Gamma = \{(p_s, r_s, l_s)\}_{s=1}^{S}$.[1] Each sphere s has radius r_s and homogeneous coordinates p_s in the canonical coordinate system of θ_0. The center of the sphere can be seen as a virtual vertex attached to exactly one limb, and thus transforms via $c_s(\theta) = \pi\big(T_{l_s}(\theta) T_{l_s}^{-1}(\theta_0) p_s\big)$. Intersection between spheres s and t occurs when $\|c_s(\theta) - c_t(\theta)\| < r_s + r_t = K_{st}$. We thus define a softened penalty as

$$E_{\text{int}} = \sum_{(s,t) \in \mathcal{P}} \frac{\sigma_\gamma(K_{st} - \|c_s(\theta) - c_t(\theta)\|)}{\|c_s(\theta) - c_t(\theta)\|} \tag{10}$$

where $\mathcal{P}$ is a set of pairs of spheres, and σ_γ is again a logistic function with constant 'sharpness' parameter γ.

The sphere parameters are chosen so that the centers $c_s(\theta_0)$ are distributed along the skeleton and the radii r_s are small enough so that the spheres lie within the interior of $S(\theta_0)$. In practice, only leg self-intersections have caused problems, and thus we place 15 spheres equally spaced along each leg, with $\mathcal{P}$ containing all pairs containing one sphere in each leg.

Full energy. Combining the above terms, we optimize an energy of the form

$$\begin{aligned} E(\theta, U) \;\; = \;\; & \lambda_{\text{vis}} E'_{\text{vis}}(\theta, U) + \lambda_{\text{prior}} E_{\text{prior}}(\theta) + \\ & \lambda_{\text{int}} E_{\text{int}}(\theta) \end{aligned} \tag{11}$$

where the various weights $\lambda_\bullet$ along with any other parameters are set on a validation set. Values for these parameters are provided in the supplementary material. Further energy terms, such as silhouette overlap or motion priors, are straightforward to incorporate and remain as future work.

4.1. Local optimization over θ

Although there are many terms, optimization of our energy function (Eq. 11) is relatively standard. For fixed correspondences U inferred by the forest, optimization of

[1] Distinct subscripts indicate whether p and l refer to vertices or spheres.

Eq. 11 over θ is a nonlinear optimization problem. Derivatives of θ are straightforward to efficiently compute using the chain rule. The parameterization means that E is somewhat poorly conditioned, so that a second order optimizer is required. However, a full Hessian computation has not appeared necessary in our tests, as we find that a Quasi-Newton method (L-BFGS) produces good results with relatively few function evaluations (considerably fewer than gradient descent). To maintain reasonable speed, in our experiments below we let the optimization run for a maximum of 300 iterations, which proved sufficient in most cases.

We initialize the optimization as follows. For the pose components of θ, we start at μ, the mean of the prior. For the global scale, we scale the model to the size of the observed point cloud. Finally we use the Kabsch algorithm [17] to find the global rotation and translation that best rigidly aligns the model. Our experience has been that a good initialization might be helpful in obtaining faster convergence but is not crucial in obtaining good results.

4.2. Multiple views and 2D images

The data term written above applies to a single 3D depth image, but can be easily extended to deal with combinations of depth images and 2D silhouettes. For this, we assume that Q images were captured by cameras with viewing matrices $\{P_q\}_{q=1}^{Q}$ yielding Q sets of observed image points $\{x_{qi}\}_{i=1}^{n_q}$ and corresponding weights $\{w_{qi}\}_{i=1}^{n_q}$. In the case of a depth image, camera q's observed points live in its own 3D coordinate system and P_q will transform our 3D model points into that system. For 2D silhouettes, the observed points are pixel locations in 2D, the weights are set to unity, and P_q suitably transforms and projects model points onto the 2D image plane. By defining A_q to be A rotated so it points along camera q's optical axis we can reformulate the visibility-aware data term as

$$\begin{aligned} E'_{\text{vis}} &= \sum_{q=1}^{Q}\sum_{i=1}^{n_q} w_{qi}\phi_{qi} \qquad (12) \\ \phi_{qi} &= V_{qi}(\theta)\cdot d(x_{qi}, P_q M(u_{qi};\theta)) + (1 - V_{qi}(\theta))\cdot\tau \end{aligned}$$

where now $V_{qi}(\theta) = \sigma_\beta(-N(u_{qi};\theta)^\top A_q)$. One can see that our original energy is just the special case where $Q = 1$ and P_q is the identity. To initialize the global translation, rotation, and scaling components of θ for multi-view data, we simply place the model near the intersection of the camera setup and perform a local optimization of these parameters.

5. Experiments

We now describe our evaluation and comparison with related work on several challenging datasets. Remember that in generating the results below, our algorithm does not use any temporal context: independently for each frame, it must infer both the pose and also the shape. Parameter settings are given in the supplementary material.

5.1. Setup

Skeleton. We parameterize and predict the following 19 body joints: head, neck, shoulders, elbows, wrists, hands, knees, ankles, feet, and hips (left, right, center).

Regression forest. We use a forest of 3 trees trained to depth 20. To learn the structure and split functions of the trees we use 300k synthetic images per tree. Given this trained tree structure, we pass a smaller training set of 20k (depth, correspondences) image pairs down each tree to learn the regression models at the leaves. In investigating these regression forests we found the final pose estimation accuracy to be largely insensitive to even substantially varied tree training parameters (*e.g.* number of images, clustering bandwidth). Training the forest structure took around a day on a large cluster, but training the regression models only took around an hour on a single workstation.

Test sets. As our main test set we employ 5000 synthetic images containing full 360° facing directions (0° means facing the camera), separate from the images used to train the forest. Synthetic depth data was shown in [23] to accurately validate pose estimation accuracy. Our images were rendered from 15 different body models (from thin to obese, with heights ranging $\sim 0.8 - 1.8$m), and contain simulated camera noise, cropping where the person is only partially in frame, and challenging poses from a large mo-cap corpus. For comparison with previous work we also evaluate on (i) the synthetic MSRC-5000 dataset from [23] which is limited to ±120° facing direction and thus somewhat easier than our test set; and (ii) on the Stanford set [15] of real depth images.

Metrics. We choose an extremely challenging error metric for our main results: the proportion of test images that have *all* predicted joints within a certain maximum Euclidean distance from their respective ground truth joint positions. We plot this proportion as a function of the distance threshold. This 'worst-case' error metric allows us to easily see how many of our images are essentially fully correct, for a range of definitions of correct. As further information, we also include graphs of the proportion of *joints* across all test images within the threshold ('joints average'). In our metrics we do not count any joints for which both the ground truth and the inferred joint positions are off-screen. However, we do still count inaccurate predictions for occluded joints as errors. In the graphs of worst-case accuracy below, we include a dotted line at D=0.2m, which, based on visual inspection, we believe to be a rough approximation to the maximum that might be useful for an interactive system.

5.2. Results

Accuracy of regression forest. The discriminative model of Sec. 3 clearly has a hard task: to predict u correspondences directly from image patches regardless of body pose and shape. We show here the proportion of pixels for which the Euclidean distance of the inferred correspondence to the

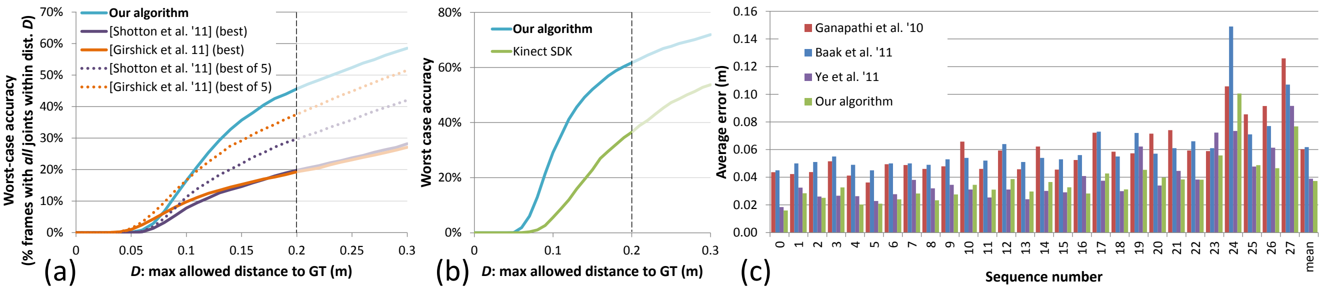

Figure 3. **(a)** Comparison with [23, 16]. Note the solid curves represent real algorithms, while the dotted curves are theoretical. **(b)** Comparison with the Kinect SDK on single frames. **(b)** Comparison on the Stanford dataset [15]. Best viewed digitally at high zoom.

ground truth correspondence is less than a threshold in Vitruvian units ('vit', where 1 vit = 1m for a 1.8m tall individual):

Threshold	0.1 vit	0.2 vit	0.3 vit
Proportion	41.5%	68.2%	77.3%

Comparison with [23] & [16]. We compare with [23, 16] on the MSRC-5000 test set on their 16 predicted joints: head, neck, shoulders, elbows, wrists, hands, knees, ankles, feet. A direct comparison is difficult: their approaches predict only individual joints (zero or more hypotheses for each), not whole skeletons. But, to obtain a reasonable comparison, we perform the following two experiments. First, we take their highest confidence prediction for each joint and treat these together as a skeleton. (For on-screen joints where no prediction was made, we create a virtual prediction as the mean of the other joint predictions to keep errors tolerably small). This first comparison is slightly overly critical as algorithms [23, 16] were designed with a subsequent model fitting stage in mind. As a second, fairer comparison, we allow an oracle to tell us which of their top 5 most confident predictions for each joint is closest to the ground truth, effectively simulating a perfect combinatorial optimization over their hypotheses for each joint.

As one can see in Fig. 3(a), beyond about $D = 0.1$m, our approach performs considerably better even than the best-of-5 oracle, indicating the potential gain from model fitting to our densely predicted correspondences instead of their sparse joint predictions. This improvement can be achieved at similar computational cost: an optimized implementation of our algorithm runs at around 120fps on a high-end desktop machine.

Comparison with the Kinect SDK. The Microsoft Kinect for Windows SDK is designed to track people from live video. As such, a direct comparison with our single frame approach is hard. However, we managed to obtain an *indicative* comparison, by repeatedly injecting a single test frame multiple times into the SDK. For many of our test frames, this resulted in a sensible skeleton being output. (The frames for which the SDK did not produce a skeleton output tended to be the more challenging ones). In the results in Fig. 3(b) we compare our algorithm with the SDK on only these successful test frames, using the ±120° test set since the SDK does not handle back-facing poses.[2] The results indicate a marked improvement over the SDK, though are indicative only for single frames. In visual inspection, the main improvements seem to be due to better scale estimation, handling of cropped frames, and handling of side-on poses.

Comparison on the Stanford dataset. We report results on the Stanford dataset [15] of real depth images in Fig. 3(c), using the mean joint error for comparison. This dataset contains depth images of a single person, and is considerably less varied (and thus less challenging) than our main synthetic test set (results below). Although these are real images, the pose variation is much smaller than the other test sets, and the current state-of-the-art algorithms make use of temporal information, which we eschew as discussed above. Furthermore, the marginal improvements provided over the current state of the art [25] are bolstered by our algorithm running at least two orders of magnitude faster. For this experiment we used a forest regressor trained on data limited to ±120° as almost all sequences contain only frontal depth images. The marker position predictions were generated from our model as virtual skinned vertices with bone offsets estimated by hand from 10 images with ground truth.

Main test set. We present a selection of results on our challenging test set in Fig. 4, with accuracy graphs in Fig. 5(a,b). On the worst-case metric, our algorithm (blue curve) gets around 55% of frames essentially correct (all joint within 0.2m). This is an extremely difficult test set and metric, and the comparisons above with related work suggest that this is actually a rather good score. The corresponding mean joint error is 0.092m. Note that the best achievable pose ('optimal θ'), obtained by minimizing the error in joint positions given the ground truth, does not give a perfect result (red curve). This is because the test set was rendered from models with more degrees of freedom than we fit (*e.g.* global scale poorly approximates the shape differences between a 4 year-old child and an overweight 2m tall adult). The result obtained given the ground truth correspondences (green curve) shows that there is considerable room for improve-

[2] For this experiment we did not include the hip joint predictions in the metrics as the definition of these joints differs considerably.

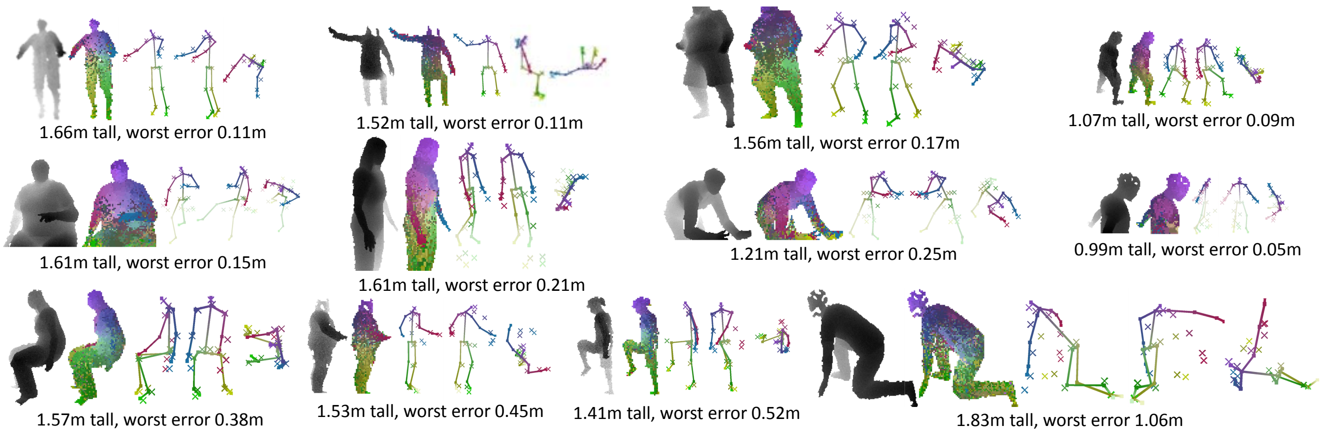
1.66m tall, worst error 0.11m
1.52m tall, worst error 0.11m
1.56m tall, worst error 0.17m
1.07m tall, worst error 0.09m
1.61m tall, worst error 0.15m
1.61m tall, worst error 0.21m
1.21m tall, worst error 0.25m
0.99m tall, worst error 0.05m
1.57m tall, worst error 0.38m
1.53m tall, worst error 0.45m
1.41m tall, worst error 0.52m
1.83m tall, worst error 1.06m

Figure 4. **Example inference results.** Each block shows the test depth image, a visualization of the inferred dense surface correspondences, and (front, right, top) views of the inferred skeleton after optimization (crosses are ground truth). Note high accuracy despite wide variety in pose, body shape and height, cropping, and rotations. The optimization of the model parameters can deal with large errors in the correspondences (*e.g.* the chest in row 1, column 3). Only single frames are used; there is no tracking. The third row shows some failure modes, including inaccurate predictions of occluded joints and confusion between legs.

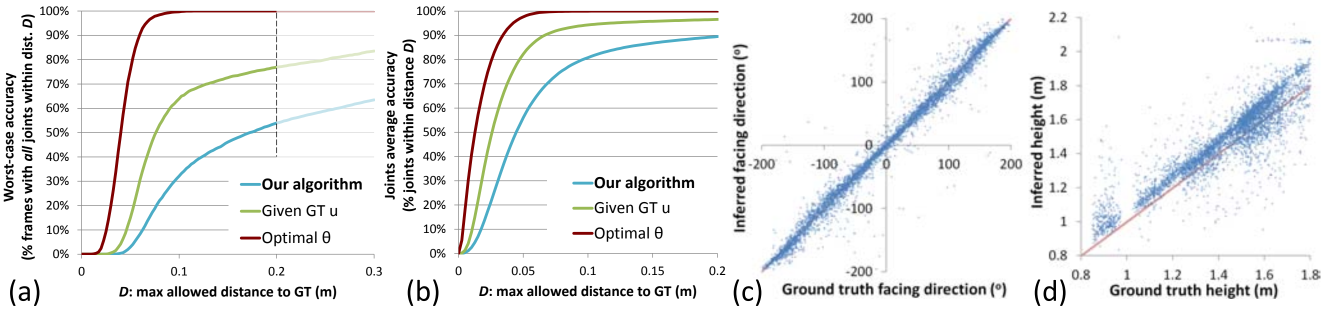

Figure 5. **(a, b)** We compare our algorithm's result to the result of optimizing our energy given the ground truth correspondences, and to the accuracy metrics computed at the optimal θ. See text for discussion. **(c, d)** Scatter plots of facing direction and height estimation accuracy.

ment of our algorithm by (i) estimating the correspondences more accurately, and (ii) improving the energy function or optimization. However, note that even with perfect correspondences, incorrectly predicted *occluded* joints are still penalized in our metric as errors. Fig. 6 further highlights the effect of noise in the correspondences.

To highlight the variability in our test set, we show in Fig. 5(c,d) scatter plots comparing the ground truth to inferred estimates of facing direction and person height (via the global scale parameter) across the 5000 frames in our test set. The large majority of frames have the facing direction accurately estimated by our algorithm across the full $360°$ range, though there are clearly some outliers which are flipped, probably because the forest sometimes fails to disambiguate the highly symmetric front and back sides of the body. Height estimation is also generally good, though with perhaps more spread. If temporal information were available one might expect these parameters to be more accurately inferred (height in particular is stationary over time), which should in turn improve the joint prediction accuracy.

Multi-view experiments. In Fig. 7 we show results on multi-view silhouette and depth data using the energy defined in Sec. 4.2; the silhouettes were generated by simply flattening the depth images, so a direct comparison is possible. Note that accurate 3D pose can be inferred even from 2D silhouettes. Having more views improves accuracy, and the much stronger cues from depth images result in superior pose estimation accuracy. For this experiment we fixed virtual cameras at $0°, \pm 30°, \pm 45°$ around a circle at 3m from the subject, and rendered a synthetic multi-view test set. During optimization we assume a known extrinsic calibration of the cameras. The reduction in occlusion given multiple views meant that the best results we obtained obtained were with $\lambda_{\text{prior}} = 0$. Due to the inherent ambiguity in facing direction given only silhouettes, we restrict the facing direction to $\pm 60°$ from the central camera. We also trained separate regression forests for each view; for example, the $-45°$ view was trained with with facing directions in the range $[-15°, 105°]$. Perhaps these restrictions might be relaxed given the facing direction from tracking.

6. Discussion

This paper has proposed the use of regression forests to directly predict dense correspondences between image pixels and the vertices of an articulated mesh model. These correspondences allow us to side-step the classic ICP problem of requiring a good initial pose estimate. Our exper-

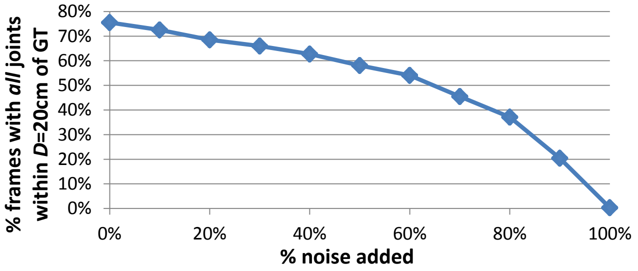

Figure 6. **Effect of noisy correspondences.** Starting at ground truth u, a given proportion of correspondences are randomly assigned a new mesh vertex. As more noise is added pose estimation accurate deteriorates.

iments on several challenging test sets have validated that these correspondences further allow accurate human pose estimation using one-shot optimization of the model parameters, for both depth and multi-view silhouette images. This efficient one-shot approach means that our algorithm can run at super real-time speeds.

The accuracy numbers we quote may in some cases appear low in absolute terms. We argue that tackling problems with such hard metrics and challenging test sets should be encouraged to drive progress. Indeed, our comparisons with other techniques should convince the reader that even these seemingly low scores are a significant improvement over the state of the art.

However, many aspects of the algorithm can still be improved: the inferred correspondences are quite noisy; the optimizer does not always find a good minimum; and several of the model hyper-parameters have not been properly optimized. The regression models in the tree leaves contain further information about correspondence that is not currently used. We believe our approach to be fairly general and as future work plan to investigate other models such as faces, as well as adding motion priors to allow smooth tracking of sequences.

Acknowledgements. The authors would like to acknowledge Mat Cook, Ross Girshick, and Christian Plagemann for their help, and the anonymous reviewers for their constructive feedback.

References

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

[2] Y. Amit and D. Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.

[3] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *Proc. ICCV*, 2011.

[4] L. Ballan and G. Cortelazzo. Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *In Proc. 3DPVT*, 2008.

[5] M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1), 1996.

[6] C. Bregler, J. Malik, and K. Pullen. Twist based acquisition and tracking of animal and human kinematics. *IJCV*, 56(3), 2004.

[7] L. Breiman. Random forests. *Tech. Rep. TR567, UC Berkeley*, 1999.

[8] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI*, 24(5):603–619, 2002.

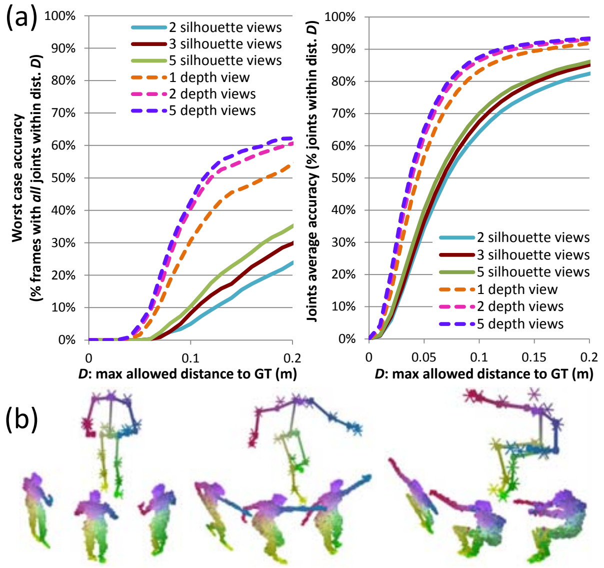

Figure 7. **Multi-view and silhouette results. (a)** Accuracy on the multi-view test set, comparing the number of viewpoints and depth *vs*. silhouettes. **(b)** Example results from 3 silhouette views.

[9] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno, and T. Andriacchi. Markerless motion capture through visual hull, articulated ICP and subject specific model generation. *IJCV*, 87(1), 2010.

[10] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework. *NOW Publishers, To Appear*, 2012.

[11] L. da Vinci. The Vitruvian Man. c. 1487.

[12] A. Delaunoy and E. Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3D reconstruction problems dealing with visibility. *IJCV*, 95(2):100–123, 2011.

[13] J. Deutscher and I. D. Reid. Articulated body motion capture by stochastic search. *IJCV*, 61(2):185–205, 2005.

[14] J. Gall, B. Rosenhahn, T. Brox, and H. Seidel. Optimization and filtering for human motion capture: A multi-layer framework. *IJCV*, 87, 2010.

[15] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real time motion capture using a single time-of-flight camera. In *Proc. CVPR*, pages 755–762. IEEE, 2010.

[16] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Proc. ICCV*, 2011.

[17] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, 1976.

[18] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *Proc. CVPR*, pages 2:775–781, 2005.

[19] Microsoft Corp. Redmond WA. Kinect for Xbox 360.

[20] J. Pilet, V. Lepetit, and P. Fua. Real-time non-rigid surface detection. In *Proc. CVPR*, 2005.

[21] G. Pons-Moll, L. Leal-Taixé, T. Truong, and B. Rosenhahn. Efficient and robust shape matching for model based human motion capture. In *Proc. DAGM*, 2011.

[22] M. Salzmann and R. Urtasun. Combining discriminative and generative methods for 3D deformable surface and articulated pose reconstruction. In *Proc. CVPR*, 2010.

[23] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *Proc. CVPR*, 2011.

[24] R. Urtasun and T. Darrell. Local probabilistic regression for activity-independent human pose inference. In *Proc. CVPR*, 2008.

[25] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys. Accurate 3D pose estimation from a single depth image. In *Proc. ICCV*, 2011.