Int J Comput Vis (2015) 113:163–175
DOI 10.1007/s11263-015-0818-9

Metric Regression Forests for Correspondence Estimation

Gerard Pons-Moll[1] · **Jonathan Taylor**[2] · **Jamie Shotton**[2] · **Aaron Hertzmann**[3] · **Andrew Fitzgibbon**[2]

Received: 27 May 2014 / Accepted: 17 March 2015 / Published online: 11 April 2015

Abstract We present a new method for inferring dense data to model correspondences, focusing on the application of human pose estimation from depth images. Recent work proposed the use of regression forests to quickly predict correspondences between depth pixels and points on a 3D human mesh model. That work, however, used a proxy forest training objective based on the classification of depth pixels to body parts. In contrast, we introduce Metric Space Information Gain (MSIG), a new decision forest training objective designed to directly minimize the entropy of distributions in a metric space. When applied to a model surface, viewed as a metric space defined by geodesic distances, MSIG aims to minimize image-to-model correspondence uncertainty. A naïve implementation of MSIG would scale quadratically with the number of training examples. As this is intractable for large datasets, we propose a method to compute MSIG in linear time. Our method is a principled generalization of the proxy classification objective, and does not require an extrinsic isometric embedding of the model surface in Euclidean space. Our experiments demonstrate that this leads to correspondences that are considerably more accurate than state of the art, using far fewer training images.

Keywords Human pose estimation · Model based pose estimation · Correspondence estimation · Depth images · Metric regression forests

Communicated by Tilo Burghardt, Majid Mirmehdi, Walterio Mayol-Cuevas, and Dima Damen.

✉ Gerard Pons-Moll
gerard.pons.moll@tue.mpg.de

Jonathan Taylor
jota@microsoft.com

Jamie Shotton
jamiesho@microsoft.com

Aaron Hertzmann
hertzman@adobe.com

Andrew Fitzgibbon
awf@microsoft.com

[1] Max Planck for Intelligent Systems, Tübingen, Germany

[2] Microsoft Research, Cambridge, UK

[3] Adobe Research, San Francisco, USA

1 Introduction

A key concern in a number of computer vision problems is how to establish correspondences between image features and points on a model. An effective method is to use a decision forest to discriminatively regress these correspondences (Girshick et al. 2011; Taylor et al. 2012; Shotton et al. 2013). So far, these approaches have ignored the correlation of model points during training, or have arbitrarily pooled the model points into large regions (parts) to allow the use of a classification training objective. The latter, however, can fail to recognize that a confusion between two nearby points that lie in different parts is not necessarily severe. Further, it can fail to recognize that confusion between two distant points, that belong to the same part can be severe. In this work, we propose the Metric Space Information Gain (MSIG) training objective for decision forests (Pons-Moll et al. 2013),[1] that, instead, naturally accounts for target dependencies during training and does not require the use of artificial parts. Our MSIG objective assumes that the model points lie in a space

[1] Note that this is an extended version of Pons-Moll et al. (2013). Some portions of Taylor et al. (2012) have been included for clarity.

in which a metric has been defined to encode correlation between target points. Among the larger class of problems where MSIG could apply, we focus on the challenging application of general activity human pose estimation from single depth images.

Human pose estimation has been a very active area of research for the last two decades. Algorithms can be classified into two main groups, namely generative (Pons-Moll and Rosenhahn 2011) and discriminative (Sminchisescu et al. 2011). Generative approaches model the likelihood of the observations given a pose estimate. The pose is typically inferred using local optimization (Bregler et al. 2004; Brubaker et al. 2010; Stoll et al. 2011; Pons-Moll et al. 2011; Ganapathi et al. 2012) or stochastic search (Deutscher and Reid 2005; Gall et al. 2010; Pons-Moll et al. 2011). Regardless of the optimization scheme used, such approaches are susceptible to local minima and thus require good initial pose estimates.

Discriminative approaches (Urtasun and Darrell 2008; Bo and Sminchisescu 2010; Lee and Elgammal 2010; Memisevic et al. 2012) learn a direct mapping from image features to pose space from training data. Unfortunately, these approaches can struggle to generalize to poses not present in the training data. The approaches in Shotton et al. (2011), Girshick et al. (2011) bypass some of these limitations by discriminatively making predictions at the pixel level. This makes it considerably easier to represent the possible variation in the training data, but yields a set of independent local pose cues that are unlikely to respect kinematic constraints.

To overcome this, recent work has fit a generative model to these cues (Ganapathi et al. 2010; Baak et al. 2011; Taylor et al. 2012). The most relevant example of such a hybrid system is that of Taylor et al. (2012) who robustly fit a mesh model to a set of image-to-model correspondences predicted by a decision forest.

Decision forests are a classic method for inductive inference that has recently regained popularity by yielding excellent results on a wide range of classification and regression tasks. The canonical example in pose estimation is Shotton et al. (2011) where a forest is used to segment the human body into parts. These parts are manually specified and the segmentation is used to define a per-pixel classification task. To train the forest, split functions are evaluated using a parts objective ('PARTS') based on discrete information gain. Specifically, the split is chosen to reduce the Shannon entropy of the resulting body part class distributions at the left and right child nodes. Motivated by the success of Hough forests (Gall et al. 2011) for object detection and localization, a follow-up paper (Girshick et al. 2011) directly regressed at each pixel an offset to several joint locations. They showed, surprisingly, that retrofitting a forest for this task that had been trained using the PARTS objective (Shotton et al. 2011) outperforms forests that had been trained using a standard regression objective based on variance minimization. The work of Taylor et al. (2012) followed suit in retrofitting a PARTS trained classification forest to predict model-image correspondences. Despite these successes, the somewhat arbitrary choice to bootstrap using a PARTS objective, clashes with the experience of several authors Buntine and Niblett (1992), Liu and White (1994), Nowozin (2012) who show that the objective function has a substantial influence on the generalization error of the forest.

We address this by showing that the image-to-model correspondences used in Taylor et al. (2012), can be predicted with substantially higher accuracy by training a forest using the 'correct' objective—an objective that chooses splits in order to minimize the uncertainty in the desired predictive distributions. When the target outputs lie in a metric space, minimizing the continuous entropy in that space is the natural training objective to reduce this uncertainty.

Our main contribution is showing how this continuous entropy can be computed efficiently at every split function considered in the training procedure, even when using millions of training examples. To this end, we estimate the split distributions using Kernel Density Estimation (KDE) (Parzen 1962) employing kernels that are functions of the underlying metric. To make this computationally tractable, we first finely discretize the output space and pre-compute a kernel matrix encoding each point's kernel contribution to each other point. This matrix can then be used to efficiently 'upgrade' any empirical distribution over this space to a KDE approximation of the true distribution. Although staple choices exist for the kernel function (e.g. Gaussian), its underlying metric (e.g. Euclidean distance) and discretization (e.g. uniform), they can also be chosen to reflect the application domain. In our domain of human pose estimation, the targets are points on a 3D mesh model surface. Interestingly, our MSIG objective can encode the body part classification objective (Shotton et al. 2011) by employing a non-uniform discretization. It is, however, much more natural to have a near uniform discretization over the manifold and to use the geodesic distance metric to encode target correlation on this manifold, see Fig. 1. As articulated shape deformations are $\epsilon-$isometric with respect to the geodesic distance, all computations in this space are independent of pose which removes the need to find an extrinsic isometric embedding in the Euclidean space as used in Taylor et al. (2012).

Our experiments on the task of human pose estimation show a substantial improvement in the quality of inferred correspondences from forests trained with our objective. Notably, this is achieved with no additional computational burden since the algorithm remains the same at test time. We further observe that with orders of magnitude less training

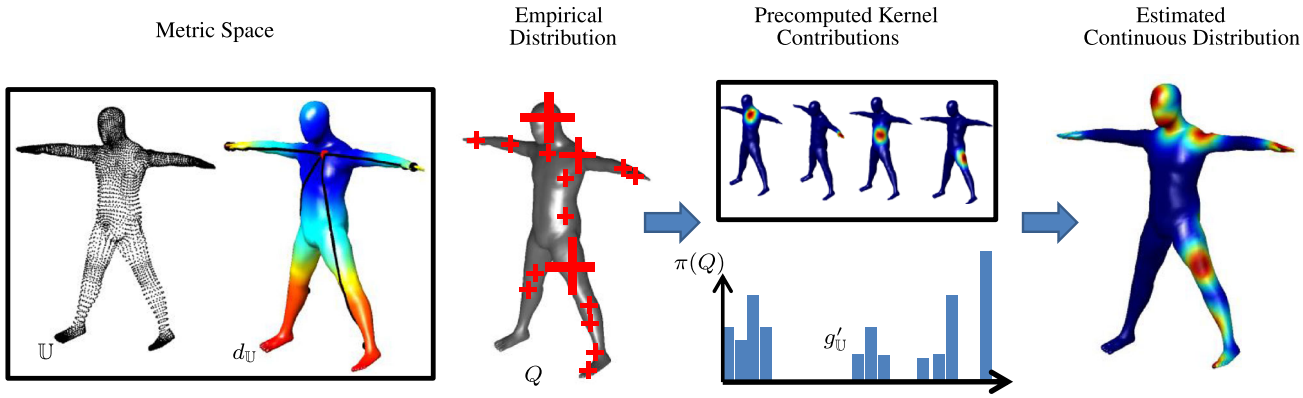


Fig. 1 We propose a method to quickly estimate the continuous distributions on the manifold or more generally the metric space induced by the surface model. This allows us to efficiently train a random forest to predict image to model correspondences using a continuous entropy objective. Notation is explained in Sect. 3

data, we can obtain state of the art human pose performance using the same fitting procedure as Taylor et al. (2012).

2 Forest Training

We employ the standard decision forest training algorithm and features. A forest is an ensemble of randomly trained decision trees. Each decision tree consists of split nodes and leaf nodes. Each split node stores a split function to be applied to incoming data. At test time, a new input will traverse the tree branching left or right according to the test function until a leaf node is reached. Each leaf stores a predictor, computed from the training data falling into that leaf. At training time, each split candidate partitions the set of training examples Q into left and right subsets. Each split function s is chosen among a pool $\mathcal{F}$ in order to reduce the average uncertainty of the predictions. This is achieved using a training objective $I(s)$ that assigns a high score if s reduces this uncertainty. Training proceeds greedily down the tree, locally optimizing I for each node, until some stopping criterion is met. In more detail, the forest is trained using the following algorithm (Breiman 1999)

1. At every node of the tree, generate a random set of split functions out of a pool $s_i \in \mathcal{F}$.
2. For every split function, split the training examples Q falling into that node into a left subset $Q_L(s_i)$ and a right subset $Q_R(s_i)$.
3. Choose the split function that maximizes some aproximate measure $\hat{I}(s;\, Q)$ of information gain I

$$s^* = \arg\max_{s_i} \hat{I}(s;\, Q) \tag{1}$$

$$\hat{I}(s;\, Q) = \hat{H}(Q) - \sum_{i\in\{L,R\}} \frac{|Q_i|}{|Q|}\hat{H}(Q_i), \tag{2}$$

where $\hat{H}$ is some approximation of the entropy computed from the empirical distribution Q.

4. Iterate until one of these conditions is satisfied (1) the tree depth is lower than the maximum allowed tree depth, (2) the information gain is bigger than a suer specified minimum, (3) the number of training examples in the node is lower than a chosen minimum.

In all of our experiments, we use the same binary split functions as Shotton et al. (2011) which consist of fast depth comparisons executed on a window centered at the input depth pixel $\mathbf{x}_i$ which are described in more detail in Sect. 4.2. For more details, we refer the reader to Criminisi and Shotton (2013). Notably, we are able to improve results significantly by changing only the measure of information gain I.

As our main contribution, we propose Metric Space Information Gain (MSIG) as the natural objective to learn to regress image-to-model correspondences where the target domain is a metric space. This objective aims to reduce the continuous entropy of the data on the metric space. In the case of a metric space induced by a reference 3D human mesh model with standard body proportions, this translates into the correspondence uncertainty over the model surface. To train a forest using MSIG we first need to define the metric for the target space which determines the correlation between the targets. Instead of assuming a uni-modal Gaussian distribution (e.g. Shotton et al. 2013) we use KDE to approximate the density where the kernels are functions of the metric chosen; see Fig. 2. Informally, distributions with probability mass at nearby locations will result in lower entropies than distributions with probability mass spread to distant locations. As we will show, MSIG outperforms the PARTS (Shotton et al. 2011; Taylor et al. 2012) and standard regression (Girshick et al. 2011) objectives, and can be computed efficiently in linear time.

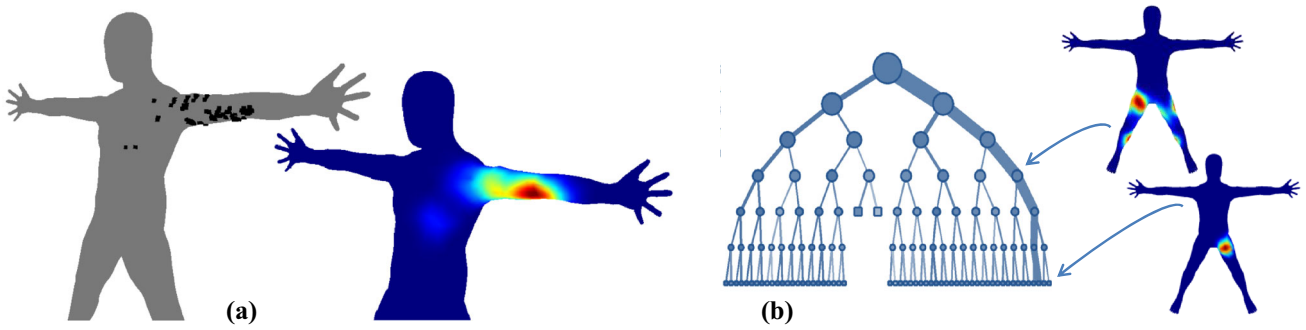


Fig. 2 **a** On the *left* we show an example of an empirical distribution and on the *right* our estimated continuous distribution. **b** Examples of the continuous distributions induced by KDE at different levels of the tree. The MSIG objective reduces the entropy of the distributions through each split resulting in increasingly uni-modal and lower entropy distributions deeper in the tree

3 Metric Space Information Gain

We use the surface of a canonical human body to define the metric space $(\mathbb{U}, d_{\mathbb{U}})$ of our targets. Here, $\mathbb{U}$ denotes the continuous space of locations on this model and $d_{\mathbb{U}}$ denotes the geodesic distance metric on the manifold induced by the surface model. Let U denote a random variable with probability density p_U whose support is a set $\mathbb{U}$ and let $B(s)$ be a random variable that depends on a split function s and takes the values L (left) or R (right). The natural objective function used to evaluate whether a split s reduces uncertainty in this space is the information gain,

$$I(s) = H(U) - \sum_{i \in \{L,R\}} P(B(s) = i) H(U|B(s) = i) \tag{3}$$

where $H(U)$ is the differential entropy of the random variable U. For a random variable U with distribution p_U this is defined as

$$H(U) = \mathbb{E}_{p_U(\mathbf{u})}\left[-\log p_U(\mathbf{u})\right] = -\int_{\mathbb{U}} p_U(\mathbf{u}) \log p_U(\mathbf{u}) d\mathbf{u}. \tag{4}$$

In practice the information gain can be approximated using an empirical distribution $Q = \{\mathbf{u}_i\}$ drawn from p_U as

$$I(s) \approx \hat{I}(s; Q) = \hat{H}(Q) - \sum_{i \in \{L,R\}} \frac{|Q_i|}{|Q|} \hat{H}(Q_i), \tag{5}$$

where $\hat{H}(Q)$ is some approximation to the differential entropy and $\|\cdot\|$ dennotes the cardinality of a set. One way to approach this is to use a Monte Carlo approximation of Eq. (4)

$$H(U) \approx -\frac{1}{N} \sum_{\mathbf{u}_i \in Q} \log p_U(\mathbf{u}_i)\ . \tag{6}$$

As the continuous distribution p_U is unknown, it must also be estimated from the empirical distribution Q. One way to approximate this density $p_U(\mathbf{u})$ is using KDE. Let $N = |Q|$ be the number of datapoints in the sample set. The approximated density $f_U(\mathbf{u})$ is then given by

$$p_U(\mathbf{u}) \simeq f_U(\mathbf{u}) = \frac{1}{N} \sum_{\mathbf{u}_j \in Q} k(\mathbf{u}; \mathbf{u}_j), \tag{7}$$

where $k(\mathbf{u}; \mathbf{u}_j)$ is a kernel function centered at $\mathbf{u}_j$. Plugging this approximation into Eq. (6), we arrive at the KDE estimate of entropy:

$$\hat{H}_{\text{KDE}}(Q) = -\frac{1}{N} \sum_{\mathbf{u}_i \in Q} \log \left(\frac{1}{N} \sum_{\mathbf{u}_j \in Q} k(\mathbf{u}_i; \mathbf{u}_j) \right). \tag{8}$$

That is, one evaluates the integral at the datapoint locations $\mathbf{u}_i \in Q$ in the empirical distribution, a calculation of complexity N^2. To train a tree, the entropy has to be evaluated at every node of the tree and for every split function $s \in \mathcal{F}$. Thus this calculation could be performed up to $2^L \times |\mathcal{F}|$ times, where L is the maximum depth of the tree. Clearly, for big training datasets one cannot afford to scale quadratically with the number of samples. For example, the tree structures used in this paper are trained from 5000 images with roughly 2000 foreground pixels per image, resulting in 10 million training examples. Therefore, as our main contribution, we next show how to train a random forest with a MSIG objective that scales linearly with the number of training examples.

To this end, we discretize the continuous space into V points $\mathbb{U}' = (\mathbf{u}'_1, \mathbf{u}'_2 \ldots, \mathbf{u}'_V) \subseteq \mathbb{U}$. This discretization simplifies the metric to a matrix of distances $D_{\mathbb{U}} = \left(d_{\mathbb{U}}(\mathbf{u}'_i, \mathbf{u}'_j)\right)$ that can be precomputed and cached. Even better, the kernel functions can be cached for all pairs of points $(\mathbf{u}'_i, \mathbf{u}'_j) \in \mathbb{U}'$. For our experiments, we choose the kernel function on this space to be an exponential

$$k(\mathbf{u}'_i;\mathbf{u}'_j)=\frac{1}{Z}\exp\left(-\frac{d_{\mathbb{U}}(\mathbf{u}'_i,\mathbf{u}'_j)^2}{2\sigma^2}\right) \tag{9}$$

where $d_{\mathbb{U}}(\mathbf{u}'_i,\mathbf{u}'_j)$ is the geodesic distance on the model and σ is the bandwidth of the kernel. The normalization constant Z ensures that the total amount of contribution coming from each point equals one and is thus invariant to the discretization. The geodesic distances are pre-computed on a high resolution triangulated mesh model using Dijkstra's algorithm (Dijkstra 1959). The discretization would ideally be uniformly distributed over the model surface, but we find that that simply using an appropriate sampling of the vertex locations of the original mesh sufficient to obtain good results.

In all the experiments shown in this paper we use $\sigma = 3$cm which roughly corresponds to the average nearest neighbor distance in the empirical distributions. A detailed discussion on kernel bandwidth selection can be found in Silverman (1986). Since the kernels fall off to zero, only a small subset of indices $\mathcal{N}_i \subseteq \{1.,...,V\}$ indicate neighboring points $\{\mathbf{u}'_j\}_{j\in\mathcal{N}_i}$ that contribute to $\mathbf{u}'_i$. Hence, for efficiency, we only store the significant kernel contributions for each discretized point $\mathbf{u}'_i$. For ease of explanation in the following, we assume here that each point has a constant number of neighbors $|\mathcal{N}_i| = M$ for all $i \in \{1,...,V\}$. Let $\mathcal{J}_{i,j}$ denote a look-up table that contains the node index of the j-th neighbor of the i-th node. This leads to the following kernel matrix that is pre-computed before training:

$$\mathbf{K}=\begin{bmatrix} k(\mathbf{u}'_1;\mathbf{u}'_{\mathcal{J}_{1,1}}) & k(\mathbf{u}'_1;\mathbf{u}'_{\mathcal{J}_{1,2}}) & \dots & k(\mathbf{u}'_1;\mathbf{u}'_{\mathcal{J}_{1,M}}) \\ k(\mathbf{u}'_2;\mathbf{u}'_{\mathcal{J}_{2,M}}) & k(\mathbf{u}'_2;\mathbf{u}'_{\mathcal{J}_{2,2}}) & \dots & k(\mathbf{u}'_2;\mathbf{u}'_{\mathcal{J}_{2,M}}) \\ \vdots & & \ddots & \vdots \\ k(\mathbf{u}'_V;\mathbf{u}'_{\mathcal{J}_{V,1}}) & k(\mathbf{u}'_V;\mathbf{u}'_{\mathcal{J}_{V,M}}) & \dots & k(\mathbf{u}'_V;\mathbf{u}'_{\mathcal{J}_{V,M}}) \end{bmatrix}. \tag{10}$$

Thus, given a discretization $\mathbb{U}'$ we can smooth the empirical distribution over this discretization using the kernel contributions as

$$g_{U'}(\mathbf{u}'_i;Q)\simeq\frac{1}{N}\sum_{j\in\mathcal{N}_i}\pi_j(Q)k(\mathbf{u}'_i;\mathbf{u}'_j) \tag{11}$$

where the weights $\pi_j(Q)$ are the number of data points in the set Q that are mapped to the bin center $\mathbf{u}'_j$. In other words, $\{\pi_j(Q)\}_{j=1}^V$ are the unnormalized histogram counts of the discretization given by $\mathbb{U}'$. In this way, we can use a simple histogram as our sufficient statistic to estimate the density, see Fig. 1. The expression in Eq. (11) can be efficiently computed using the precomputed kernel matrix $\mathbf{K}$ in Eq. (10)

$$g_{U'}(\mathbf{u}'_i;Q)=\frac{1}{N}\sum_{m=1}^{M}\pi_{\mathcal{J}_{i,m}}(Q)\mathbf{K}_{i,m}\,. \tag{12}$$

We can use this to further approximate the continuous KDE entropy estimate of the underlying density in Eq. (7) as

$$p_U(\mathbf{u})\simeq f_U(\mathbf{u};Q)\simeq g_{U'}(\alpha(\mathbf{u});Q) \tag{13}$$

where $\alpha(\mathbf{u})$ maps $\mathbf{u}$ to a point in our discretization. Using this, we approximate the differential entropy of $p_U(\mathbf{u})$ using the discrete entropy of $g_{U'}$ defined on our discretization. Hence, our MSIG estimate of the entropy on the metric space for an empirical sample Q is

$$\hat{H}_{\mathrm{MSIG}}(Q)=-\sum_{u_i\in\mathbb{U}'}g_{U'}(\mathbf{u}'_i;Q)\log g_{U'}(\mathbf{u}'_i;Q) \tag{14}$$

where the terms only need to be calculated when $g_{U'}(\mathbf{u}'_i;Q)\neq 0$.

Note that this is also equivalent to approximating the entropy defined in Eq. (4) by evaluating the integral only at the V points of the discretized space $\mathbb{U}'$. Note that in contrast to Eq. (6) we need to re-weight by $g_{U'}(\mathbf{u}'_i;Q)$ because we are sampling uniformly on a grid of points in the space as opposed to Eq. (6) where the samples are drawn from the empirical distribution Q. This is equivalent to importance sampling with a uniform proposal distribution.

The complexity of Eq. (14) is $V \times M$. When training a tree, each new split s requires a linear pass through the data to compute the left and right histograms. The total complexity of evaluating a split using Eq. (5) is thus $N + V \times M \ll N^2$ allowing trees to be trained efficiently. By using our approximation of the continuous entropy we can capture target correlations, as MSIG encourages distributions with mass localized in nearby locations which is crucial for obtaining good correspondences. This would be more difficult to achieve using a parts classification objective or a vertex histogram (see Fig. 3).

4 Pose Estimation

We now investigate the ability of MSIG trained forests to improve the accuracy of model based human-pose estimation. Hence, we follow the procedure of Taylor et al. (2012) as closely as possible. Our goal is to determine the pose parameters $\theta \in \mathbb{R}^d$ of a linearly skinned (Pons-Moll and Rosenhahn 2011; Balan et al. 2007) 3D mesh model so as to explain a set of image points $D = \{\mathbf{x}_i\}_{i=1}^n$.

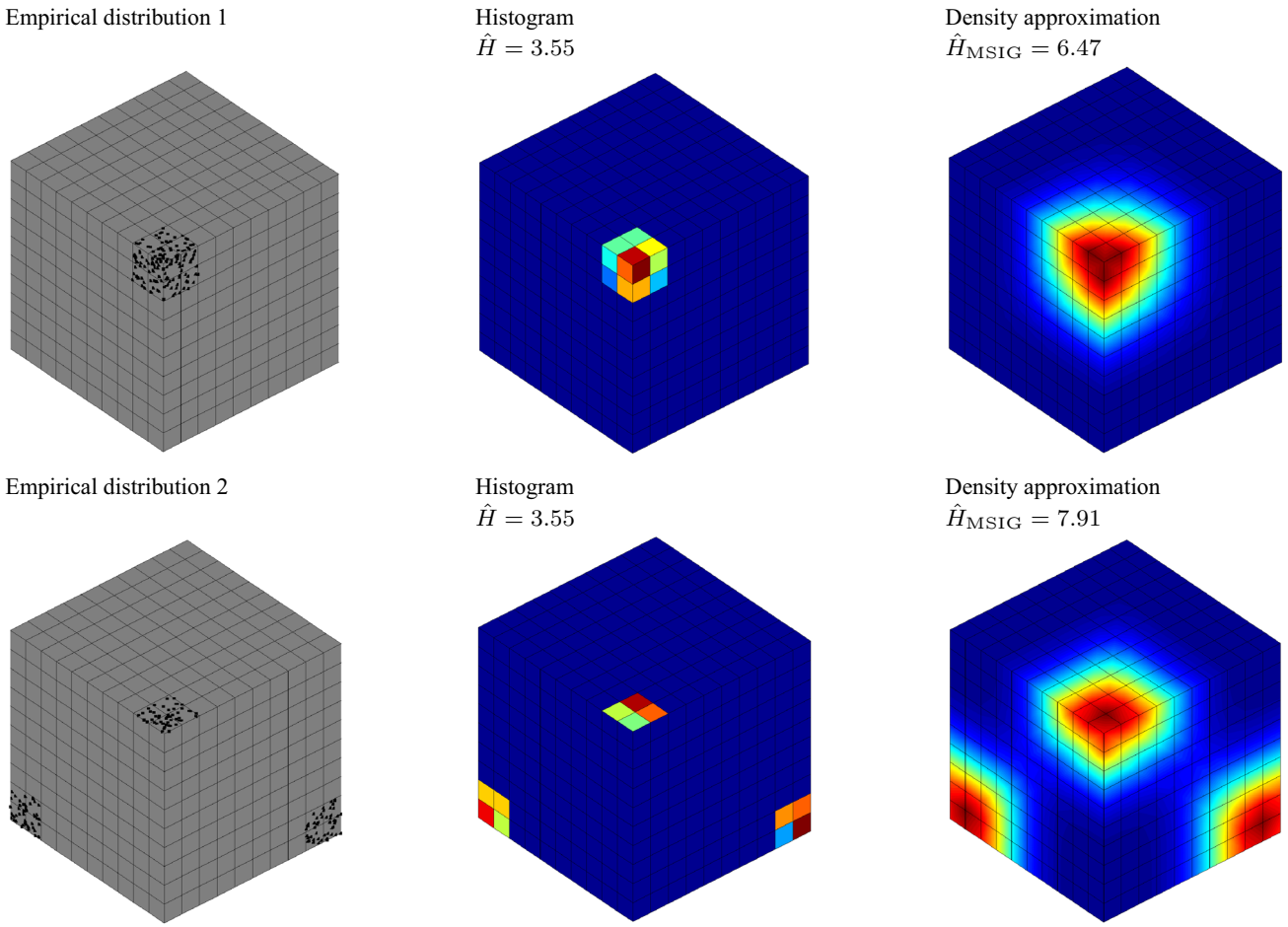


Fig. 3 We demonstrate here the result of using different approximations of the continuous entropy given an empirical distribution. On the *left*, show two empirical distributions. The *top* first distribution is highly concentrated in a single mode. In the second distribution, the mode has been split into three smaller modes. In the remaining columns, we show histograms representing the discretized empirical distribution before (*middle columns*) and after (*right column*) the kernel density approximation has been applied. What is important to note here is that the calculation of the Shannon entropy directly on the raw histogram (*middle column*), results in nearly the same entropy for both cases. By contrast, when the calculation is done on the smoothed distributions (*right column*), the resulting MSIG entropy is much higher for distribution 2 than 1. This is due to the fact that the kernel smooths the probability mass so that it accumulates in a localized point for the first distribution. Informally, distributions with points located at distant locations should result in higher entropies. As a result, distribution 2 should have a higher entropy than distribution 1. Therefore, our objective will favor splits that cluster points in nearby locations. It is also important to note that the absolute value of the entropy obtained using a given approximation is not important, what is important for training is that the relative entropies can be used to disambiguate peaked distributions (*top*) from uninformative distributions (*bottom*)

4.1 Human Body Model

The surface of our human body model, denoted as $\mathcal{S}(\theta)$ to indicate its dependence on θ, is a triangulated mesh supported by V vertices $\mathcal{V} = \{v_j\}_{j=1}^{V}$. The model is parameterized using a kinematic tree, or skeleton, consisting of L limbs. Each limb l has a rigid transformation $R_l(\theta)$ encoding the transformation from that limbs coordinate system to its parents. The rotational component of that transformation is parameterized by a 4D quaternion encoded in θ. In addition, a final global similarity transform $R_{\mathrm{glob}}(\theta)$ scales the model and places it in world space. This transformation is parameterized by an additional 4D quaternion, 3D translation and isotropic scaling encoded in θ. The transform $T_l(\theta)$ then encodes the transformation from limb l's coordinate system to the world and is defined by simply combining the transforms one encounters while walking up the tree to the root with $R_{\mathrm{glob}}(\theta)$.

Each vertex v_j in the mesh is defined as

$$v_j = \left(p_j, \{(\alpha_{jk}, l_{jk})\}_{k=1}^{K}\right), \tag{15}$$

where: *base vertex* p_j is the homogenous coordinates of the 3D vertex position in a canonical pose θ_0; 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 $K = 4$. The position of the vertex given a pose θ is then output by a global transform G which linearly combines the associated limb transformations:

$$G(v_j;\theta) = \Pi\left(\sum_{k=1}^{K} \alpha_{jk} T_{l_{jk}}(\theta) T_{l_{jk}}^{-1}(\theta_0) p_j\right) \tag{16}$$

where Π is the standard conversion from 4D homogeneous to 3D Euclidean coordinates. By applying this transformation to all the vertices in our mesh we obtain the human surface $\mathcal{S}(\theta)$ in a given pose θ.

4.2 Correspondence based Energy

For our main results we use image points that have a known 3D position, i.e., $\mathbf{x}_i \in \mathbb{R}^3$, obtained using a calibrated depth camera. Following standard practice, we assume reliable background subtraction. The goal, restated, is then to find the pose θ that induces a surface $\mathcal{S}(\theta)$ that best explains the observed depth image data. A standard way to approach this is to introduce a set of correspondences between image pixels and mesh points $\mathcal{C} = \{\mathbf{u}_i\}_{i=1}^n$, such that each correspondence $\mathbf{u}_i \in \mathcal{U}$. One then minimizes

$$E_{\text{data}}(\theta, \mathcal{C}) = \sum_{i=1}^{n} w_i \cdot d\big(\mathbf{x}_i, G(\mathbf{u}_i;\theta)\big) \tag{17}$$

where w_i weights data point i and $d(\cdot,\cdot)$, is some distance measure in $\mathbb{R}^3$. The energy defined in Eq. (17) 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 (Black and Rangarajan 1996) 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 via $z_i = [\,0\,0\,1\,]\,\mathbf{x}_i$ to compensate for proportionately fewer pixels and therefore contributions to the energy function as depth increases.

Unfortunately, deficiencies remain with (17), particularly with self-occlusion. In the following, we build up further terms to form our full energy in Eq. (22).

4.2.1 Visibility Term

For given parameters θ, the data term in Eq. (17) 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 Sect. 4.4, 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. We thus 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 the function $\hat{\mathbf{n}}(\mathbf{u};\theta)$ to return the surface normal of the model transformed into pose θ at $G(\mathbf{u};\theta)$. Then $\mathbf{u}$ is marked visible if the dot product between $\hat{\mathbf{n}}(\mathbf{u};\theta)$ and the camera's viewing axis A (typically $A = [0, 0, 1]$, the positive Z axis) is negative. One might then write

$$E_{\text{vis}} = \sum_{i=1}^{n} w_i \begin{cases} d(\mathbf{x}_i, G(\mathbf{u}_i;\theta)) & \hat{\mathbf{n}}(\mathbf{u}_i;\theta)^\top A < 0 \\ \tau & \text{otherwise} \end{cases} \tag{18}$$

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:

$$E'_{\text{vis}} = \sum_{i=1}^{n} w_i \big[V_i(\theta) \cdot d(\mathbf{x}_i, G(\mathbf{u}_i;\theta)) + (1 - V_i(\theta)) \cdot \tau\big] \tag{19}$$

where the visibility weight is set according to a logistic function $V_i(\theta) = \sigma_\beta(-\hat{\mathbf{n}}(\mathbf{u}_i;\theta)^\top A)$.

4.2.2 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:

$$E_{\text{prior}} = (\theta - \mu)^\top \Lambda (\theta - \mu) \tag{20}$$

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

4.2.3 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 $\Gamma = \{(p_s, r_s, l_s)\}_{s=1}^S$.[2] 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 $c_s(\theta) = \Pi\big(G(p_s, \theta)\big)$.

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

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

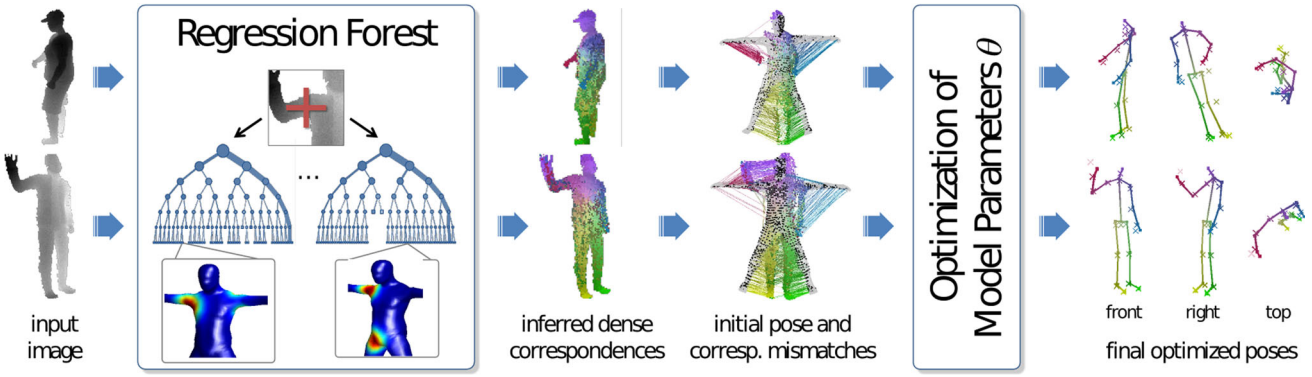


Fig. 4 Model based human pose estimation with correspondences inferred using a regression forest. From *left* to *right*: every pixel in the depth image is pushed through each tree in the forest. A series of split functions are applied to every pixel until a leaf node is reached. The correspondence distributions in different trees are aggregated and the correspondence for that pixel is taken as the top mode of the distributions. The inferred dense correspondences are then used to optimize the model parameters θ, i.e., the pose and scale of the person. We show at the right most image, the result obtained for the test images shown on the left

$$E_{\text{int}} = \sum_{(s,t)\in\mathcal{P}} \frac{\sigma_\gamma(K_{st} - \|c_s(\theta) - c_t(\theta)\|)}{\|c_s(\theta) - c_t(\theta)\|} \tag{21}$$

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.

4.2.4 Full Energy

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

$$\begin{aligned} E(\theta, \mathcal{C}) = {} & \lambda_{\text{vis}} E'_{\text{vis}}(\theta, \mathcal{C}) + \lambda_{\text{prior}} E_{\text{prior}}(\theta) \\ & + \lambda_{\text{int}} E_{\text{int}}(\theta) \end{aligned} \tag{22}$$

where the various weights $\lambda_\bullet$ along with any other parameters are set on a validation set. Further energy terms, such as silhouette overlap or motion priors, are straightforward to incorporate and remain as future work. An alternating minimization (or block coordinate descent) over θ and $\mathcal{C}$ would yield a standard articulated ICP algorithm (Besl and McKay 1992). Unfortunately, convergence is unlikely without a good initial estimate of either θ or $\mathcal{C}$. Therefore, we will use our proposed metric regression forest to estimate a set of image to model correspondences discriminatively. The key to the success of our pose estimation method is the use of a discriminative appearance model to estimate $\mathcal{C}$ directly instead of the more common approach of initializing θ.

4.3 Predicting Correspondences

We use a metric regression forest to predict a set of correspondences $\mathcal{C}$ to initialize the optimization of Eq. (22), see Fig. 4. To accomplish this, every foreground pixel $\mathbf{x}$ will be pushed down each tree in the forest in the following manner. When a non-terminal node is encountered, a binary *split function* will determine whether the left or right branch is taken. Let $x = (u, v)$ denote the image coordinates of the depth pixel $\mathbf{x}$. The value of the split function is then computed on an image window centered at image coordinates (u, v), for which we employ the fast depth comparison split functions of Shotton et al. (2011)

$$f_\phi = d_I\left(x + \frac{\mathbf{m}}{d_I(x)}\right) - d_I\left(x + \frac{\mathbf{n}}{d_I(x)}\right) \tag{23}$$

where $\phi = (\mathbf{m}, \mathbf{n})$ are a pair of 2D displacement vectors, see Fig. 5. A path is traversed from the root down to a leaf, branching left or right according to the evaluation of the split functions. In more detail, if $f_\phi < \tau$ the left branch will be taken and the right otherwise. Each leaf terminal leaf node contains a regression model.

At training time, we employ the MSIG objective and split functions defined above to construct the tree. The regression model stored in each terminal leaf node is built from the training data falling into the leaf in the following way. For further efficiency, we represent the leaf distributions as a small set of confidence-weighted modes $S = \{(\hat{\mathbf{u}}, \omega)\}$, where $\hat{\mathbf{u}} \in \mathbb{R}^3$ is the position of the mode in the embedding space, and ω is the scalar weighting. This set S 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 $\mathbf{u}_i \in \mathbb{U}$ from these aggregated distributions. To

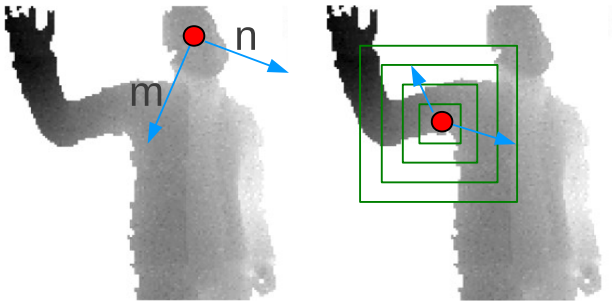


Fig. 5 Split functions: we use the depth offset features used in Shotton et al. (2011). The feature consists on comparing the depths of two pixels. If the difference is bigger than a chosen threshold the function takes value 1 and 0 otherwise. Every feature in itself is too simple to discriminate but many features combined together can be very descriptive: local appearance will be captured by small displacements whereas context will be captured by larger displacements. This is ilustrated in the right image with green squares

do this, we take the mode $\hat{\mathbf{u}}$ with largest confidence value ω. We also explored more sophisticated strategies such (i) minimizing expected loss with resepect the leaf distributions or (ii) predicting a set of confidence weighted correspondences for every image pixel or (iii) randomly sampling correspondences from the leave distributions. Unfortunately, these alternative strategies resulted in no improvement with respect to just retrieving the correspondence with highest confidence weight. For efficiency, 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.

4.4 Local Optimization Over θ

Although there are many terms, optimization of our energy function in Eq. (22) is relatively standard. For fixed correspondences $\mathcal{C}$ inferred by the forest, optimization of (22) 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.

4.4.1 Initialization

We initialize the optimization as follows. For the pose components of θ, we start at 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 (Kabsch 1976) to find the global rotation and translation that best rigidly aligns the model. Our experience has been that this initialization is helful to obtain faster convergence and improved accuracy. However, the accuracy of the initialization is not crucial in obtaining good results, i.e., the energy minimum found does not depend on initialization as long as the surface model is reasonably close to the observed data in the image.

4.4.2 Alternation Between θ and $\mathcal{C}$

In contrast to Taylor et al. (2012), we also *consider* a further ICP optimization to achieve additional gains. After optimizing θ, we hold θ fixed and update $\mathcal{C}$ by finding the closest visible model point to each depth pixel, instead of minimizing Eq. (22) keeping the $\mathcal{C}$ fixed to the forest predictions. This allows $\mathcal{C}$ to be updated efficiently using a k-D tree (Bentley 1975). To update θ, the non-linear optimizer is simply restarted with the new correspondences.

5 Experiments

We evaluate our approach using the same test set of 5000 synthetic depth images as used in Taylor et al. (2012). We examine both the accuracy of the inferred correspondences and their usefulness for single frame human pose estimation from depth images.

5.1 Setup

5.1.1 Forests

We use two forests in our experiments: MSIG and PARTS, indicating respectively that they were trained with our proposed MSIG objective or the standard PARTS based objective of Shotton et al. (2011), see Fig. 6.

Both forests contain three trees and were trained to depth 20. To learn the structure and split functions of each tree we use 5000 synthetic images per tree. The extra complexity in training a MSIG tree resulted in them taking roughly three times as long as the PARTS trees. This complexity does not exist at test time and thus speeds reported in Taylor et al. (2012) are obtainable using either type of tree.

To train the random forests we use the data from Shotton et al. (2011). 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 cor-

respondence image, where colors are used to represent the ground truth correspondences that we aim to predict using the forest. Examples of the training images are given in Fig. 7.

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. 7. Given this data, we can now train the trees MSIG and PARTS using the corresponding training objectives.

Fig. 6 Difference between PARTS and MSIG forest output domains. *Left* The outputs of a PARTS based forest is a body part label. The PARTS forest is trained using an objective that minimizes the Shannon entropy of a discrete distribution over the body part labels. This corresponds to a classification task, where a label has to be assigned to every depth pixel. *Right* The output of the MSIG forest are points on the manifold defined by the human surface model. The MSIG attempts to directly minimize the continuous entropy of the distribution of correspondences over the human surface model. This corresponds to a regression task, where every depth pixel is mapped to a point on the human surface model. The *left* image is courtesy of Shotton et al. (2011) and the *right* image of Taylor et al. (2012)

To populate the leaf distributions in both types of trees, we replicate the strategy of Taylor et al. (2012): we push the training data from 20000 (depth, correspondences) image pairs through the trees and find the mode of the distribution in the extrinsic isometric embedding of a human shape (the 'Vitruvian' pose) using mean-shift.

5.1.2 Pose Estimation

For human pose estimation we parametrize a model using a skeleton. We predict the following 19 body joints: head, neck, shoulders, elbows, wrists, hands, knees, ankles, feet, and hips (left, right, center).

5.1.3 Metrics

To evaluate the accuracy of the inferred correspondences, we use the *correspondence error* defined as the geodesic distance between the prediction and the ground truth model location. We use a model with standard proportions and thus a correspondence error of 25 cm is roughly the length of the lower arm. To measure pose accuracy we use the challenging *worst joint* error metric introduced in Taylor et al. (2012): the proportion of test scenes that have all predicted joints within a certain Euclidean distance from their ground truth locations (Figs. 9, 10).

5.2 Results

We evaluate the performance of our forest regressors to predict dense image to model correspondences. We quantify the proportion of predicted correspondences with an error less than a certain distance. We find that correspondences with an error of less than 15 cm tend to be useful for pose estimation whereas those with higher errors are usually treated as outliers. In Fig. 8 we show the correspondence accuracy for both the MSIG forest and PARTS forest at depths of 17, 18, 19 and 20. As it can be seen, the MSIG forest produces cor-

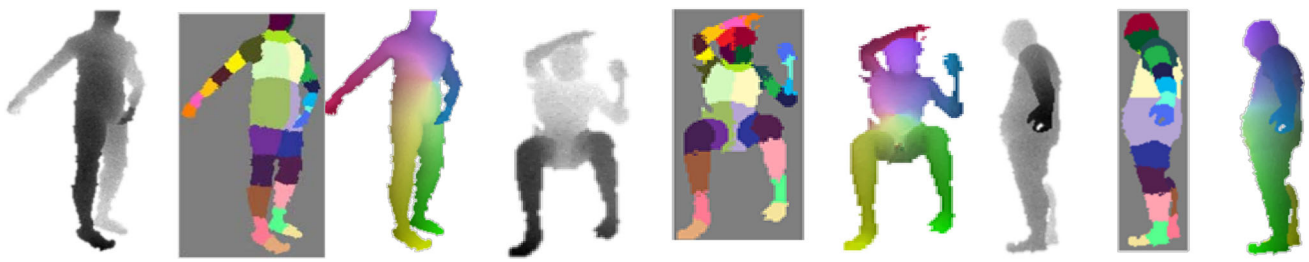


Fig. 7 Training data used to train the PARTS and MSIG forests. We show here three example training images in triplets. For every triplet, we show *left* to *right*: (1) the synthetic depth image, (2) the body PARTS output label and (3) the MSIG output. Because the synthetic images have been generated using the model, every pixel can be annotated with the ground truth correspondence on the the human surface model. Training images are randomly generated varying different factors: pose, shape and image cropping. The forest will have to learn invariances to all these factors

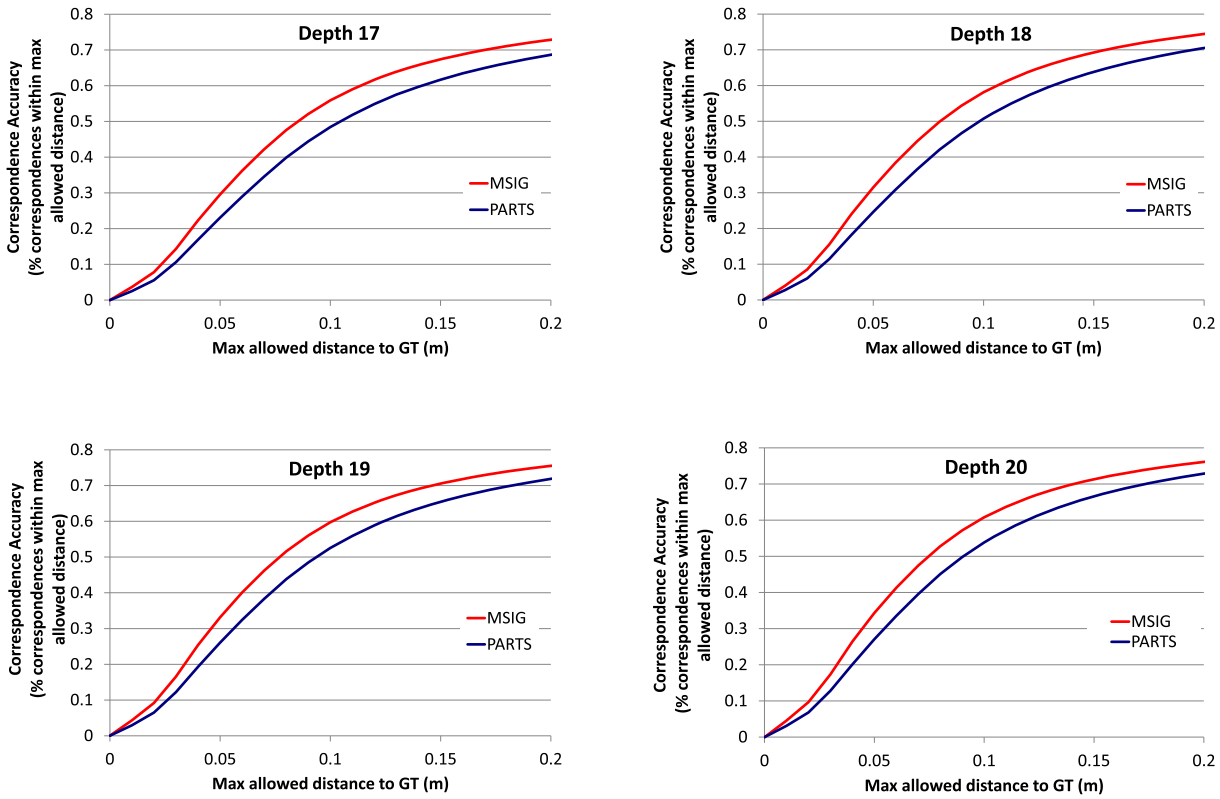


Fig. 8 Correspondence error comparison of PARTS forest with the proposed MSIG forest. We evaluate the accuracy for forests of depths 17, 18, 19, 20. It can be observed that our proposed method consistently produces considerably more accurate correspondences

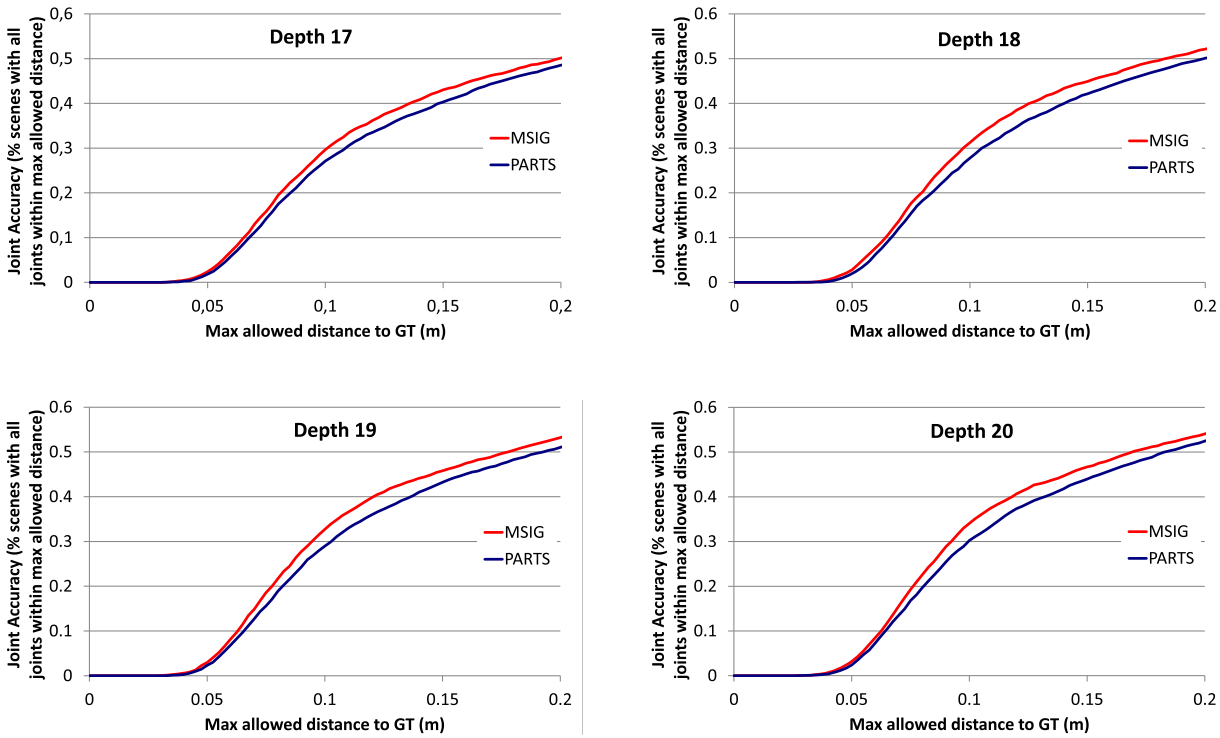
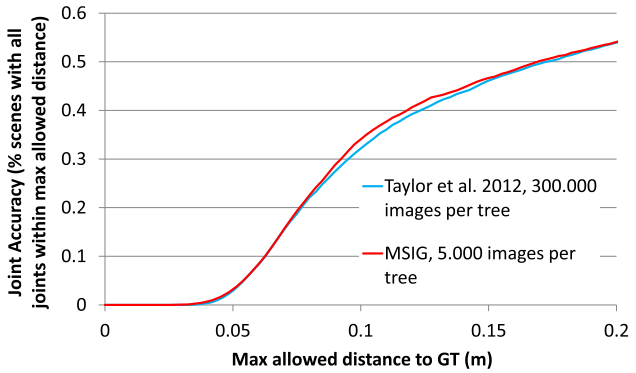


Fig. 9 Pose accuracy comparison using correspondences from both PARTS and proposed MSIG forests at depths 17, 18, 19 and 20. For both forests, we use the pose estimation algorithm of Taylor et al. (2012) as explained in Sect. 4.2 and evaluate using the worst joint error metric



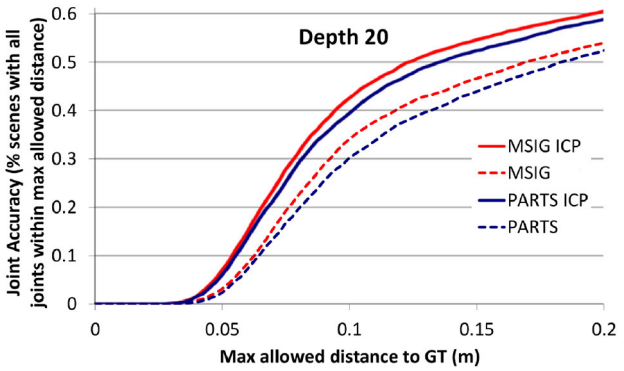


Fig. 10 *Left* Pose accuracy of our MSIG forest trained with 5000 images per tree compared to accuracy reported by Taylor et al. (2012) which used 300,000 training images. *Right* Pose accuracy for both PARTS and MSIG forests after 10 iterations of ICP. Note that the curve labelled MSIG in both the *left* (*solid red*) and *right* (*dashed red*) plots are the same

respondences that are consistently more accurate than those produced from the PARTS forest. This is very encouraging since forests trained using a PARTS objective had previously shown state of the art performance, far superior to those using other objectives such as the Hough-regression (Girshick et al. 2011). We attribute the better performance of our approach to the fact that MSIG favors distributions with mass concentrated (in the sense of the defined metric) in close locations.

Although the inferred dense correspondences can be used for a large number of tasks, we consider the task of single frame pose estimation as a motivational example. Therefore, we also show the impact in the pose accuracy again for forests of depth 17, 18, 19 and 20. As one would expect, better correspondences translate into more accurate pose estimates. As can be seen in Fig. 9, the MSIG forest produces a small but significant improvement w.r.t. to the PARTS forest. The smaller gains in pose accuracy are expected as the energy of Taylor et al. (2012) is designed to be robust to outliers from their forest. We also compare in Fig. 10 directly to the results provided by Taylor et al. (2012), which appears to be the state of the art for single frame pose estimation from depth images. Despite our MSIG forest using orders of magnitude less training images (300K images vs. 5K images per tree), we achieve equivalent performance.

We further demonstrate that our correspondences can be used to initialize classical registration methods such as articulated ICP as explained in Sect. 4.2. Contrary to what was alluded to in Taylor et al. (2012) we find that using just 10 such ICP alternations provides an additional performance gain of up to 10% with both PARTS and MSIG correspondences as demonstrated in Fig. 10. Furthermore, it can be seen that the gap between the MSIG and PARTS is not washed out by this downstream ICP processing. The resulting MSIG poses after ICP refinement, thus represent the state of the art on this dataset.

6 Conclusion

We have introduced MSIG, an objective function that evaluates a split function's ability to reduce the uncertainty over an arbitrary metric space using KDE. Using a discretization of this space, an efficient approximation to MSIG was developed as to facilitate its use in training random forests. Although the general framework can be tuned through the specification of an appropriate metric space, kernel function and discretization, natural choices exist making this approach widely applicable.

We employed MSIG in the context of human pose estimation to both simplify and enhance the inference of dense data to model correspondences by avoiding two arbitrary requisites of previous work: (i) our work does not require a segmentation of the human body into parts, and (ii) it does not require an extrinsic isometric embedding of the human shape. A number of experiments show that the more principled MSIG objective allows the inference of superior correspondences compared to those provided by standard training objectives. Additionally, these results translate into state of the art accuracy for single frame human pose estimation using far fewer training images.

References

Baak, A., Müller, M., Bharaj, G., Seidel, H., & Theobalt, C. (2011). *A data-driven approach for real-time full body pose reconstruction from a depth camera*. In: *IEEE international conference on computer vision* pp. 1092–1099.

Balan, A., Sigal, L., Black, M., Davis, J., & Haussecker, H. (2007). *Detailed human shape and pose from images*. In: *IEEE conference on computer vision and pattern recognition*.

Bentley, J. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM*, *18*(9), 509–517.

Besl, P., & McKay, N. (1992). A method for registration of 3d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *12*, 239–256.

Black, M., & Rangarajan, A. (1996). On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal on Computer Vision*, *19*(1), 57–91.

Bo, L., & Sminchisescu, C. (2010). Twin gaussian processes for structured prediction. *International Journal on Computer Vision*, *87*, 28–52.

Bregler, C., Malik, J., & Pullen, K. (2004). Twist based acquisition and tracking of animal and human kinematics. *International Journal on Computer Vision*, *56*(3), 179–194.

Breiman, L. (1999). *Random forests*. Berkeley: UC. (Technical Report TR567).

Brubaker, M., Fleet, D., & Hertzmann, A. (2010). *Physics-based person tracking using the anthropomorphic walker*. In: *International journal on computer vision*.

Buntine, W., & Niblett, T. (1992). A further comparison of splitting rules for decision-tree induction. *Machine Learning*, *8*(1), 75–85.

Criminisi, A., & Shotton, J. (2013). *Decision forests for computer vision and medical image analysis*. London: Springer.

Deutscher, J., & Reid, I. (2005). Articulated body motion capture by stochastic search. *International Journal on Computer Vision*, *61*(2), 185–205.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271.

Gall, J., Rosenhahn, B., Brox, T., & Seidel, H. P. (2010). Optimization and filtering for human motion capture. *International Journal on Computer Vision*, *87*, 75–92.

Gall, J., Yao, A., Razavi, N., Van Gool, L., & Lempitsky, V. (2011). Hough forests for object detection, tracking, and action recognition. *PAMI*, *33*(11), 2188–2202.

Ganapathi, V., Plagemann, C., Koller, D., & Thrun, S. (2012). *Real-time human pose tracking from range data*. In: *European conference on computer vision.*

Ganapathi, V., Plagemann, C., Thrun, S., & Koller, D. (2010). *Real time motion capture using a time-of-flight camera*. In: *Conference in computer vision and pattern recognition*.

Girshick, R., Shotton, J., Kohli, P., Criminisi, A., & Fitzgibbon, A. (2011). *Efficient regression of general-activity human poses from depth images*. In: *IEEE international conference on computer vision*, pp. 415–422.

Kabsch, W. (1976). A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica*, *32*(5), 922–923.

Lee, C., & Elgammal, A. (2010). Coupled visual and kinematic manifold models for tracking. *International Journal on Computer Vision*, *87*, 118–139.

Liu, W., & White, A. (1994). The importance of attribute selection measures in decision tree induction. *Machine Learning*, *15*(1), 25–41.

Memisevic, R., Sigal, L., & Fleet, D. J. (2012). Shared kernel information embedding for discriminative inference. *PAMI*, *34*(4), 778–790.

Nowozin, S. (2012). *Improved information gain estimates for decision tree induction*. In: *ICML*.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Aannals of Mathematical Statistics*, *33*(3), 1065–1076.

Pons-Moll, G., Baak, A., Gall, J., Leal-Taixe, L., Mueller, M., Seidel, H., & Rosenhahn, B. (2011). *Outdoor human motion capture using inverse kinematics and von mises-fisher sampling*. In: *International conference on computer vision*.

Pons-Moll, G., Leal-Taixé, L., Truong, T., & Rosenhahn, B. (2011). *Efficient and robust shape matching for model based human motion capture*. In: *DAGM*.

Pons-Moll, G.,& Rosenhahn, B. (2011). Model-based pose estimation. In *Visual analysis of humans* (pp. 139–170). London: Springer.

Pons-Moll, G., Taylor, J., Shotton, J., Hertzmann, A., & Fitzgibbon, A. (2013). *Metric regression forests for human pose estimation*. In: *British machine vision conference (BMVC)*.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011). *Real-time human pose recognition in parts from single depth images*. In: *IEEE conference in computer vision and pattern recognition*, pp. 1297–1304.

Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., & Fitzgibbon, A. (2013). *Scene coordinate regression forests for camera relocalization in RGB-D images*. In: *Conference in computer vision and pattern recognition*.

Silverman, B. (1986). *Density estimation for statistics and data analysis* (Vol. 26). London: CRC press.

Sminchisescu, C., Bo, L., Ionescu, C., & Kanaujia, A. (2011). Feature-based pose estimation. In *Visual analysis of humans* (pp. 225–251). London: Springer.

Stoll, C., Hasler, N., Gall, J., Seidel, H., & Theobalt, C. (2011) *Fast articulated motion tracking using a sums of gaussians body model*. In: *IEEE international conference on computer vision*, pp. 951–958.

Taylor, J., Shotton, J., Sharp, T., & Fitzgibbon, A. (2012). *The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation*. In: *Conference in computer vision and pattern recognition*.

Urtasun, R., & Darrell, T. (2008). *Sparse probabilistic regression for activity-independent human pose inference*. In: *IEEE conference in computer vision and pattern recognition*, pp. 1–8.