Key Developments in Human Pose Estimation for Kinect

Pushmeet Kohli and Jamie Shotton

Abstract The last few years have seen a surge in the development of natural user interfaces. These interfaces do not require devices such as keyboards and mice that have been the dominant modes of interaction over the last few decades. An important milestone in the progress of natural user interfaces was the recent launch of Kinect with its unique ability to reliably estimate the pose of the human user in real time. Human pose estimation has been the subject of much research in Computer Vision, but only recently with the introduction of depth cameras and algorithmic advances has pose estimation made it out of the lab and into the living room. In this chapter we briefly summarize the work on human pose estimation for Kinect that has been undertaken at Microsoft Research Cambridge, and discuss some of the remaining open challenges. Due to the summary nature of this article, we limit our description to the key insights and refer the reader to the original publications for the technical details.

1 Introduction: The Challenge

In the summer of 2008, computer vision researchers at Microsoft Research Cambridge received a call from an Xbox team who were working on a top-secret project code-named Project Natal[1]. The Xbox team, headed by Alex Kipman, told researchers that they were building a system for human pose estimation that would work using the output of a depth sensor. The team demonstrated a tracking-based system for pose estimation which, once initialized to the correct pose, could track

Pushmeet Kohli
Microsoft Research, Cambridge, UK, e-mail: pkohli@microsoft.com

Jamie Shotton
Microsoft Research, Cambridge, UK, e-mail: jamiesho@microsoft.com

[1] This project would eventually be launched as Kinect.

the pose of the human from one frame to the next. However, the system suffered from two critical problems: (i) it required the user to adopt an initialization pose, and (ii) it would typically lose track after a few frames. The Xbox team wanted the researchers to help build a system that avoided the initialization pose, that looked at a single frame at a time to avoid possible loss-of-track, and that was super efficient – it had to use just a fraction of the computational power of the Xbox 360.[2]

In this article, we summarize the publications that have resulted from working on this challenge of efficient pose estimation from single depth images. We start in Section 2 by describing the key ideas and intuitions that led to the development of the body part classification system as described fully in Shotton *et al.* [14]. This system works by first estimating which body part each pixel in the depth image belongs to, and then using this information to reason about the location of different body joints. We then move in Section 3 to discuss the offset vote regression approach [6] where instead of predicting their own body part labels, pixels vote for where they think the different body joints are located in 3D. In Section 4 we discuss [15], which shows how pose estimates can be improved by using a conditional random forest model that uses a latent variable to incorporate dependencies between joint positions. This latent variable encodes some global property of the image, such as the person's height or the direction they are facing. Section 5 gives an overview of the recently proposed Vitruvian Manifold model [16] that predicts at each pixel an estimate of the correspondence to an articulated mesh model. An energy function can then be optimized to efficiently fit the model to the observed data. Finally in Section 6 we briefly discuss some of the remaining open challenges.

2 Body Part Classification - The Natural Markers Approach

Human pose estimation is a well studied problem in the computer vision community (see [8, 10] for a survey of the literature). Certain variants of the problem, for instance, estimation of the pose of a human from a single RGB image remain unsolved. Early commercial systems for human pose estimation worked by tracking easily localizable markers that were pre-attached on the body of the human subject. Marker based systems for human pose estimation are usually quite reliable and highly accurate, but suffer from the limitation that markers need to be worn. Further, the approach also requires a calibration step where the relationship between the position of the markers and that of the body parts needs to be defined.

A natural approach motivated by the success of the marker based pose estimation systems is to use classifiers that are be able to identify and thus localize different parts of the body. Variants of this approach have been explored in a number of research studies [1, 17]. For instance, a prominent approach for recognition and pose estimation is the Pictorial Structures model [4] that tries to estimate the location of different human body parts while maintaining certain spatial relationships between

[2] For more detail on the story behind Kinect, please see the Foreword.

them. In light of these observations, Shotton *et al.* [14] decided to formulate the pose estimation problem as a body part labeling problem where the human body is divided into 31 body parts that were naturally associated with certain skeletal joint positions that needed to be estimated.

2.1 Generating the Training Data

The datasets used for training machine learning systems need to cover the variations the system would observe when it is deployed. Creating such a dataset is an expensive and time consuming process. Researchers have used computer graphics to overcome this problem [11] but this approach has its own set of problems. Synthetic body pose renderers use, out of necessity, real motion capture (mocap) data. Although techniques exist to simulate human motion they do not yet produce a full range of volitional poses or motions of a human subject that the system may encounter in the real world. The team at Microsoft overcame this problem by collecting a very large and diverse set of motion capture [14]. Rendering realistic intensity images is also hampered by the huge color and texture variability induced by clothing, hair, and skin. However, as the Kinect skeletal tracking system works with depth images, which are invariant to factors such as colour or texture, this issue does not create a problem. Other factors which do affect the depth image, such as body shape, were varied as much as possible when creating the dataset. The result was a dataset of a million synthetic pairs of images of people of varied shapes in varied poses. Each image pair contained the depth image expected from the camera, and the body part label image that we were to train the system to recognize.

2.2 Randomized Forests for Classification

The body part classification problem is similar to many image labeling problems encountered in computer vision. These problems are generally formulated using Markov random field (MRF) models that have produced impressive results for various problems [3]. However, MRFs are currently too computationally expensive for real time human pose estimation. Shotton *et al.* [13] had proposed a decision forest based method to overcome this problem which avoided the need for sophisticated and computationally expensive inference algorithms. This decision forest framework is not only simple and efficient but also allows for parallelization and could be implemented on a GPU [12]. These properties made decision forests a natural choice for solving the body part classification problem.

Shotton *et al.* [14] used a family of features that involved computing the differences between just a few depth image pixels. These were thus very inexpensive to compute. The decision trees were learned by employing the standard entropy minimization based objective and greedy tree learning schedule. The final processing

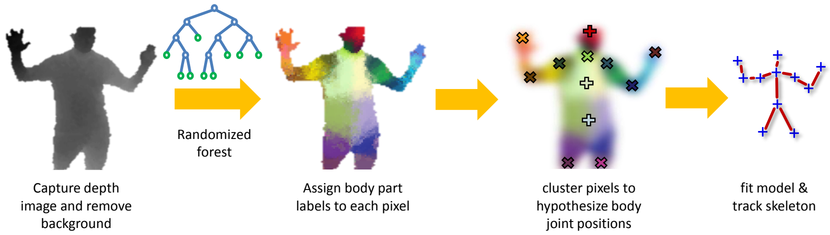
Capture depth image and remove background
Randomized forest
Assign body part labels to each pixel
cluster pixels to hypothesize body joint positions
fit model & track skeleton

Fig. 1 The basic pipeline of the Kinect skeletal tracking system.

pipeline involved computing features on every pixel and depending on the response traversing down the left or right side of the decision tree. This process is repeated until a leaf node is reached which contained a learned histogram. This histogram represents the posterior distribution over the body part label that the pixel should be assigned. These per-pixel body part distributions could then be clustered together to produce reliable hypotheses about the positions of the various joints in the body.

Figure 1 illustrates the full skeleton tracking pipeline as used for Kinect. This pipeline takes the depth image, removes the background, applies the body part recognition and clustering algorithm described above, and finally applies a model fitting stage which exploits kinematic and temporal constraints to output a full skeleton.

3 Random Forest Regression - The Voting Approach

The body part classification algorithm allowed us to solve the hard challenges we needed to ship Kinect. However, it of course did not work perfectly in all scenarios, and so we set out to fix some of its limitations. Because body part classification works by labeling pixels, it cannot estimate the location of joints whose surrounding body parts are not visible in the image due to occlusion or field of view of the sensor. Furthermore, its two-step procedure comprising of pixel labeling followed by clustering may introduce errors. To overcome these problems we decided to investigate an offset regression approach [6] where pixels vote directly for the position of the different body joints, without going through the intermediate body part representation.

Similar voting approaches have been used in the literature for solving object localization problems. For example, in the implicit shape model (ISM) [7], visual words are used to learn voting offsets to predict 2D object centers. In an extension, Müller *et al.* [9] apply ISM to body tracking by learning separate offsets for each body joint.

In [6], we use a random regression forest to learn to predict how pixels vote for 3D locations of the various joints in the body. The regression forest shares the

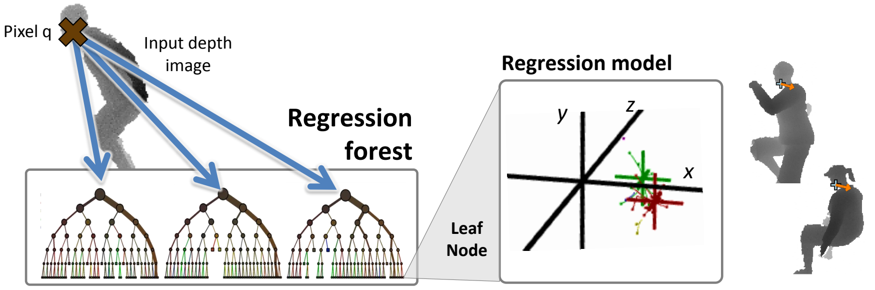

Fig. 2 The regression forest model for body joint position estimation.

same structure and features as the body part classification approach [14]. However, as illustrated in Figure 2, at each leaf node of a regression tree we instead store a distribution over the relative 3D offset from the re-projected 3D pixel coordinate to each body joint of interest. Representing these leaf node distributions efficiently is very important given the large size of our decision trees. Approximating the distribution over offsets as a Gaussian would be inappropriate, because even for fairly deep trees, we have observed highly multi-modal empirical offset distributions at the leaves. One alternative, Hough forests [5], is to represent the distribution non-parametrically as the set of all offsets seen at training time. However, Hough forests trained on our large training sets would require vast amounts of memory and be prohibitively slow for a realtime system. Therefore, we instead represent the distribution using a compact set of 3D relative vote vectors learned by clustering the training offsets. The result is a system that can, in super real time, cast votes from each pixel to potentially all joints in the body, whether they are visible or not. Furthermore, these votes can directly predict interior body joint positions rather than the positions on the body surface predicted by [14]. Overall this was seen to improve body joint prediction accuracy significantly over [14].

4 Context-Sensitive Pose Estimation - Conditional Regression Forests

Even with the improvements in [6], there are some remaining limitations. The model does not encode dependency relationships between positions of different joint positions explicitly; the predictions for each body joint are made independently. Further, the model is not able to exploit prior knowledge that might be available during prediction in certain application scenarios. For instance, while estimating the pose of a person playing a golf game, information about the player's height or torso orientation might be available and potentially useful. Similarly, in a surveillance application, we might know the walking directions of pedestrians.

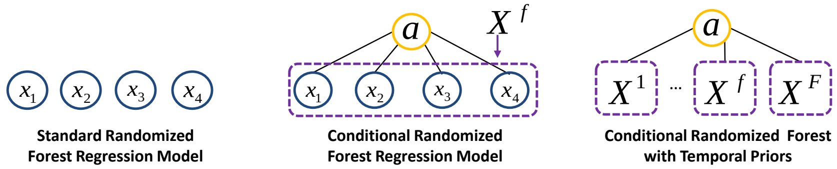
x_1
x_2
x_3
x_4
Standard Randomized Forest Regression Model
a
X^f
x_1
x_2
x_3
x_4
Conditional Randomized Forest Regression Model
a
X^1
...
X^f
X^F
Conditional Randomized Forest with Temporal Priors

Fig. 3 The figure shows the relationships encoded by different regression forest models. The basic model (left) makes predictions for each body joint independently. The conditional model (center) encodes dependency relationships between the body joint positions and a global variable. The temporal conditional regression forest model (right) incorporates the prior that the value of the global variables associated with multiple image frames should be consistent. For instance, the height of a human subject remains constant through the video sequence.

In [15], we show how both these limitations can be simultaneously overcome by incorporating a latent variable in the regression forest prediction model that encodes some global property of the image. In particular, we show that by conditioning on the players height or torso orientation, we can outperform [6] in terms of joint prediction accuracy, and as a by-product, make predictions about the conditioned-upon properties. The relationships encoded by different models are depicted in Figure 3.

5 One-Shot Model Fitting: The Vitruvian Manifold

All the work presented above estimates zero, one, or more hypotheses for the positions of each body joint. However, the work above cannot enforce kinematic constraints such as limb lengths, and is not able to disambiguate which hypotheses to stitch together into a coherent skeleton. In the Vitruvian Manifold paper [16] we attempt to address these concerns by fitting an articulated skeleton model to the observed data. A standard way to represent such an articulated skeleton is a global transformation (rotation, translation, scale) and then a hierarchical kinematic tree of relative transformations. In these transformations, the translation relative to the parent might be fixed (representing fixed limb lengths) but the rotation is parameterized (representing bending joints). Given the kinematic hierarchy of transformations, one can use, for example, linear blend skinning to generate a surface mesh of the body.

A standard way to fit the parameters of such a mesh model to the data is called Iterated Closest Point (ICP) [2]. Starting from an initialization, ICP alternates between finding the closest corresponding point on the model for each observed data point, and optimizing the parameters of the model (e.g. joint rotations) to minimize the sum of squared distances between the corresponding model and observed points. Unfortunately, ICP requires a good initialization, and can take many iterations to converge. In the Vitruvian Manifold paper [16] we decided to address 'One-Shot' pose estimation: could we achieve a good model fit by inferring these correspondences directly from the test image, and then performing only a single optimization of the model parameters. To investigate this, we took our body part classification

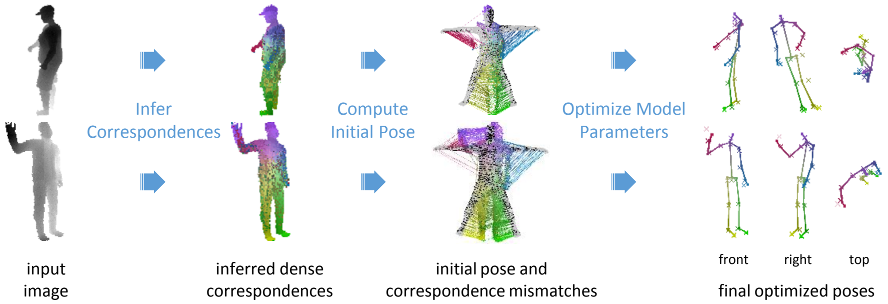

Fig. 4 The pose estimation pipeline in the Vitruvian Manifold algorithm.

forests from [14] and extended them to predict at each pixel the corresponding vertex on the surface of the mesh model in a canonical pose (the so-called Vitruvian Manifold). The forests effectively become regression forests over this manifold, and allow a dense estimate of correspondence across the test image, without any initialization. Taking these correspondences and optimizing the model parameters resulted in most cases in a very accurate fit of the articulated skeleton to the observed data at low computational cost. An illustration of the algorithm is given in Figure 4.

6 Remaining Challenges

The contributions summarized above represent a considerable advance in the state of the art in single image human pose estimation. But there remain open questions. How can we fix the remaining inaccuracies to achieve a reliable pose estimation for everyone, all the time, no matter what pose they adopt and no matter their body shape? How can we make such a system work from standard RGB cameras as well as depth cameras? How can we reliably map out the fine detail of the body, face, clothing, hair, etc.? How can we achieve a level of detail that means that instrumenting the body for motion capture becomes redundant? We believe that these and many other questions will mean that human pose estimation remains an active area of research for years to come.

Acknowledgements

This chapter is a summary of existing published work, and we would like to highlight the contributions of all the original authors.

References

1. D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, and A. Ng. Discriminative learning of Markov Random Fields for segmentation of 3D scan data. In *Proc. CVPR*, 2005.
2. P. Besl and N. McKay. A method for registration of 3-D shapes. *IEEE Trans. PAMI*, 1992.
3. A. Blake and P. Kohli. Introduction to Markov Random Fields. In *Markov Random Fields for Vision and Image Processing*. MIT Press, 2011.
4. P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, January 2005.
5. J. Gall and V. Lempitsky. Class-specific Hough forests for object detection. In *Proc. CVPR*, 2009.
6. 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.
7. B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, 2008.
8. T.B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *CVIU*, 2006.
9. J. Müller and M. Arens. Human pose estimation with implicit shape models. In *ARTEMIS*, 2010.
10. R. Poppe. Vision-based human motion analysis: An overview. *CVIU*, 108, 2007.
11. G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *Proc. ICCV*, 2003.
12. T. Sharp. Implementing decision trees and forests on a GPU. In *Proc. ECCV*, 2008.
13. J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. CVPR*, 2008.
14. Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.
15. M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *Proc. CVPR*, 2012.
16. J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The Vitruvian Manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proc. CVPR*, 2012.
17. Z. Tu. Auto-context and its application to high-level vision tasks. In *Proc. CVPR*, 2008.