KTH Computer Science
and Communication

Correspondence Estimation in Human Face and Posture Images

VAHID KAZEMI

Doctoral Thesis
Stockholm, Sweden 2014

TRITA-CSC-A 2014:14
ISSN 1653-5723 KTH School of Computer Science and Communication
ISRN KTH/CSC/A--14/14--SE SE-100 44 Stockholm
ISBN 978-91-7595-261-1 SWEDEN

Tryck: Eprint AB 2014

Abstract

Many computer vision tasks such as object detection, pose estimation, and alignment are directly related to the estimation of correspondences over instances of an object class. Other tasks such as image classification and verification if not completely solved can largely benefit from correspondence estimation. This thesis presents practical approaches for tackling the correspondence estimation problem with an emphasis on deformable objects.

Different methods presented in this thesis greatly vary in details but they all use a combination of generative and discriminative modeling to estimate the correspondences from input images in an efficient manner. While the methods described in this work are generic and can be applied to any object, two classes of objects of high importance namely human body and faces are the subjects of our experimentations.

When dealing with human body, we are mostly interested in estimating a sparse set of landmarks – specifically we are interested in locating the body joints. We use pictorial structures to model the articulation of the body parts generatively and learn efficient discriminative models to localize the parts in the image. This is a common approach explored by many previous works. We further extend this hybrid approach by introducing higher order terms to deal with the double-counting problem and provide an algorithm for solving the resulting non-convex problem efficiently. In another work we explore the area of multi-view pose estimation where we have multiple calibrated cameras and we are interested in determining the pose of a person in 3D by aggregating 2D information. This is done efficiently by discretizing the 3D search space and use the 3D pictorial structures model to perform the inference.

In contrast to the human body, faces have a much more rigid structure and it is relatively easy to detect the major parts of the face such as eyes, nose and mouth, but performing dense correspondence estimation on faces under various poses and lighting conditions is still challenging. In a first work we deal with this variation by partitioning the face into multiple parts and learning separate regressors for each part. In another work we take a fully discriminative approach and learn a global regressor from image to landmarks but to deal with insufficiency of training data we augment it by a large number of synthetic images. While we have shown great performance on the standard face datasets for performing correspondence estimation, in many scenarios the RGB signal gets distorted as a result of poor lighting conditions and becomes almost unusable. This problem is addressed in another work where we explore use of depth signal for dense correspondence estimation. Here again a hybrid generative/discriminative approach is used to perform accurate correspondence estimation in real-time.

Acknowledgments

Many people have contributed directly or indirectly to this thesis. First and foremost I would like to thank my supervisors Josephine Sullivan and Stefan Carlsson. Josephine helped me in many different aspects before and during my PhD studies and without her this thesis would not have happened. Thanks to Stefan who helped broaden my perspective over the whole field of computer vision.

During my visit to Microsoft Research I had the chance to work with some of the greatest researchers in the field of computer vision and machine learning whom I would like to thank. Thanks to Pushmeet Kohli, Jonathan Taylor, Cem Keskin and Shahram Izadi for their contribution.

Thanks to my co-authors at KTH, Hossein Azizpour and Magnus Burenius with whom I enjoyed working with. Many more people have indirectly contributed to this work, I specially want to thank Hedvig Kjellström, Atsuto Maki, Carl Henrik Ek and Arne Leijon.

I also want to thank my current and past colleagues at CVAP who made the hardship of scientific research much easier. I would like to thank Omid, Oscar, Miro, Ali, Heydar, Cheng, Martin, Yuquan, Alessandro, Akshaya, Virgile, Püren, Yasemin, Florian, John, Mårten, Christian, Petter, Niklas, Jeannette, Javier, Alper, Marianna, Andrzej, Dan, Alireza, Babak, Nils, Ioannis, Rasmus, Kaiyu, Johannes, Emil, Alejandro, Francisco, Johan, Erik, Marina, Rares Andrei, Fredrik, Sergio, Zhan, Michele, and David.

Special thanks to Patric Jensfelt who accepted me to his master program and continued his support until my graduation, and thanks to Danica Kragic for doing such a wonderful job single handedly managing CVAP.

Lastly I would like to thank my family who always encouraged me to pursue science. I hope you are proud of me.

List of Papers

The thesis is based on the following papers:

[A] Vahid Kazemi and Josephine Sullivan. Face Alignment with Part-Based Modeling. In *Proceedings of the 2011 IEEE British Machine Vision Conference (BMVC '11)*, Dundee, Scotland, Sept 2011.

[B] Vahid Kazemi and Josephine Sullivan. Using Richer Models for Articulated Pose Estimation of Footballers. In *Proceedings of the 2012 IEEE British Machine Vision Conference (BMVC '12)*, Guildford, England, Sept 2011.

[C] Vahid Kazemi, Hossein Azizpour, Magnus Burenius, and Josephine Sullivan. Multi-view Pose Estimation of Human Body. Submitted to *International Journal of Computer Vision (IJCV)*, Springer, 2014.

The paper is an extended version of the following award winning paper:

Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view Body Part Recognition with Random Forests. In *Proceedings of the 2013 IEEE British Machine Vision Conference (BMVC '13)*, Bristol, England, Sept 2013. [1]

[D] Vahid Kazemi and Josephine Sullivan. One Millisecond Face Alignment with an Ensemble of Regression Trees. In *Proceedings of the 2014 IEEE Computer Vision and Pattern Recognition Conference (CVPR '14)*, Columbus, OH, USA, June 2014.

[E] Vahid Kazemi, Cem Keskin, Johanatan Taylor, Pushmeet Kohli, and Shahram Izadi. Real-time Face Reconstruction from a Single Depth Image. In *Proceedings of International Conference on 3D Vision (3DV '14)*, Tokyo, Japan, 2014. [2]

[1] Magnus Burenius and I have contributed equally to this paper. I focused on the 2D aspects of the problem while Magnus focused on the 3D aspects. The paper won the award for Best Industry Paper at BMVC '13.

[2] The paper is the result of my internship at Microsoft Research in Cambridge.

Contents

Part I

Introduction

Chapter 1

Introduction

Vision, like many other human abilities, is very familiar to us, as we use it all the time in our daily life, yet we know very little about the underlying process that makes us see and understand things. Computer vision is an emerging branch of computational science that aims to construct algorithms that replicate this ability on software platforms, enabling computers to see what we see and understand the world around us as we do.

At the time of writing this thesis, we are already seeing applications of computer vision in our daily life. Almost every camera comes with an automatic face detector and a feature tracker which helps the camera to keep the focus on the target. Computer vision has been successfully used in the movie industry to create realistic facial animations in movies such as James Cameron's *Avatar*. With the advent of *Kinect*, Microsoft brought a state-of-the-art body tracking system to consumers, enabling controller free gaming. Using computer vision technology, Google has been able to produce driver-less cars that can autonomously traverse through city traffic.

These are just a few examples of what we have achieved so far, but computer vision has much greater potential than what we have seen. As the field is moving forward, we are developing algorithms that are faster and more accurate. The consensus is that, someday, computer vision will reach and surpass human performance, and that is when we will see another technological revolution perhaps with as big as an impact or greater than the invention of computers.

1 Problem Statement

This thesis, which is based on a collection of papers [22, 23, 20, 24, 19, 21], aims to present practical solutions for solving computer vision problems. We are specifically interested in the problem of correspondence estimation over instances of an object class (See figure 1). Many of the applications we mentioned earlier are in someway related to the correspondence estimation problem. We define the problem

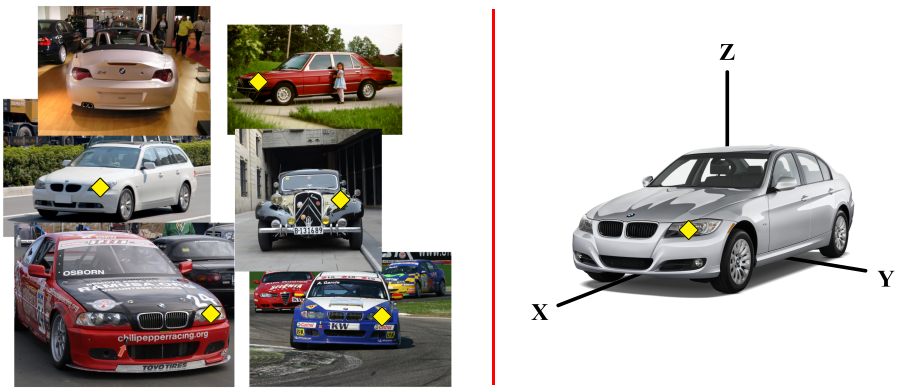


Figure 1: Given an input image of an object, like a car, the correspondence estimation problem deals with locating points on the surface of a generic object model that correspond to object's pixels on the image. Variations in pose, lighting, color and texture make this a challenging problem in computer vision.

of correspondence estimation as follows:

> *Given an input image of an object, for each object's pixel, find the corresponding location on the surface of a generic object model.*

Note that the above definition is commonly referred to as dense correspondence estimation. Sometimes we are interested in finding the correspondences for a subset of these pixels. For example when dealing with humans, we are usually interested in finding the location of a small number of landmarks corresponding to body joints. For face applications we are often interested in locating landmarks corresponding to the location of eyes, mouth corners, and etc. (See figure 2). In these cases we fix the location of landmarks on the model, and the task is to find the corresponding points on the image. This is an alternative way of presenting the problem, but it is essentially the same problem.

Correspondence estimation is a challenging problem in computer vision. This is primarily because points on the surface of objects greatly vary in appearance in 2D images with small variations in pose, camera parameters, and lighting. The collection of papers in this work focus on the case of deformable objects (See figure 2). Correspondence estimation for deformable objects is even more challenging, because deformation can also change the appearance of surface points. [1]

There are lots of applications in computer vision that in someway involve estimation of correspondences. An example, which was mentioned earlier, is human body tracking. This is the core technology behind Microsoft's Kinect gaming

[1] Throughout this thesis we use the terms landmarks or surface points interchangeably to refer to certain points on the object that we are interested in putting in correspondence.

Figure 2: The focus of this thesis is on the problem of estimating correspondences across images of deformable objects (particularly faces and the human body). Correspondence estimation of deformable objects is more challenging compared to rigid objects, since the appearance of surface points on the object greatly vary with deformation.

platform[34], where players are able to control games by body gestures and without the need for an external controller. We also mentioned the use of computer vision techniques in the movie industry for creating realistic animations through facial performance capture. This is another direct application of the correspondence estimation problem. There are many more applications in medical imaging for reconstruction and tracking of bones and tissues through analyzing x-ray images. Furthermore estimating correspondences is an essential part of other computer vision methods such as face recognition and action recognition, which have important applications in human-computer interaction and security.

A generalization of the correspondence estimation problem which is beyond the scope of this thesis treats all different classes of objects as a single deformable object class and aims to find inter-class as well as intra-class correspondences over images. While we did not have enough time to explore further on this idea, we find it fascinating and worth putting some thought into.

2 Discriminative Modeling

A common approach for finding correspondences across images of an object is to use discriminatively trained classifiers that can distinguish certain landmarks from the background. This for example can be achieved by extracting HOG (histogram of oriented gradients) [9] features from the image patches and learning SVM (support vector machine) [38] filters that can best separate image patches belonging to a certain landmark from anything else. At test time we then exhaustively evaluate all the possible patches and pick the ones with the highest response. This is a standard approach for building object detectors[9], but it can also be used for landmark localization. This approach though suffers from a number of problems.

One problem is that a feature descriptor such as HOG discards some spatial

information. This is done to gain robustness to moderate amounts of deformation, which comes at the cost of lower discriminability. An alternative approach is to directly use the RGB (or depth in case we have access to a range sensor) signal as the input feature. In fact this is the approach that we take in paper D and E. One should note though that the amount of variation in the RGB signal between different instances of a single surface point is much higher than that of HOG. That means that one needs to use a model with a much higher capacity to recognize the landmarks over multiple images.

Another problem with this approach is caused by the fact that the location of surface points are highly correlated and can not be treated as independent random variables. If we try to locate these landmarks independently we risk producing inconsistent estimations. For example in a scenario where we are interested in locating facial landmarks, using independent classifiers for each landmark might lead to confusing the left and right eye. One solution to this problem is to use a global regressor that can jointly estimate the location of landmarks (paper D). An alternative solution is to explicitly model the relations between landmarks using a generative model (paper A).

Discriminative modeling is a powerful approach that is widely used in computer vision literature [12, 34, 35, 6, 11, 10, 37] and also in this thesis, but it comes with a major shortcoming. The problem is related to the generalization ability of discriminative models. To achieve good performance on the test set, a discriminative model needs to be trained with lots of labeled examples. Insufficient number of training examples is the root of all evil when learning discriminative models. It leads to the common problem of overfitting, which occurs when the model learns relations that hold for training examples but do not generalize to test examples. There are however ways to overcome this problem. Regularization techniques which we will briefly talk about in chapter 2 can reduce the chance of overfitting. This is when we utilize additional prior information to constrain model parameters. Another solution that we exploit in papers D and E is to extend the training data by generating synthetic images using computer graphics techniques. Although, in some cases synthesizing realistic examples is too complex and we need to use real images. The labeling process often requires human supervision and thus is time consuming and undesirable. A way to address this problem is to use generative models which we describe now.

3 Generative Modeling

In a generative approach we model the distribution of the observed data. The standard approach to estimate the parameters of such a model is maximizing the likelihood of observed examples – this is commonly referred to in short as *maximum likelihood*[2]. One of the reasons for why generative modeling is attractive is because this approach allows for the use of unlabeled examples. This is a desirable property since most of the time in computer vision applications unlabeled examples are cheap

and abundant. However, the generative modeling approach also comes with a set of drawbacks.

Firstly, discriminative models have been shown to outperform their generative counterparts given enough training examples [31, 27]. This observation has motivated use of discriminatively trained generative models [15]. The problem though here is that these approaches can no longer take advantage of unlabeled examples.

Another drawback with the generative modeling approach is that inferring the unknown parameters of the model is often non-trivial, expensive, and sometimes intractable. While in some certain cases efficient inference is possible, it may require over-simplification of the model [14, 33, 15, 39]. This is an issue that we will visit in paper B. A solution to this problem is combining discriminative models with generative models. We provide some examples for the use of this approach in the next sections.

4 A Hybrid Approach

The vast majority of papers in this thesis as well as some previous works [7, 1, 36] use a combination of generative and discriminative modeling, this is because most of the time neither a purely generative nor discriminative model leads to satisfying results. In the following we briefly describe how a hybrid approach is used to solve correspondence estimation problem in different papers included in this thesis.

- Paper A uses a generative approach to model the configuration of facial parts namely the eyes, nose, and the mouth. A discriminative model is then used to regress the location of facial landmarks (e.g. corners of eyes) from the patches extracted from corresponding parts.
- Paper B uses a generative model to produce multiple hypotheses for the pose of a human and then uses a discriminative model to select the best configuration.
- In paper C, a discriminative model is used to estimate the likelihood of body joints across multiple views. A generative model is then used to infer a single consistent configuration from the likelihood maps.
- Finally in E, a discriminative model is used to estimate an initial value for hidden variables of our generative model. In an iterative procedure, unobserved parameters are then further optimized to estimate the final correspondences.

Often there is not a right or wrong way of combining generative and discriminative models, but the determining factor is the type of data.

Chapter 2

Background

The purpose of this chapter is to provide some background of the key concepts discussed in this thesis. Each section of this chapter is independent of the others and can be read or skipped if the concepts are familiar to the reader.

1 Feature Descriptors

In computer vision, a feature descriptor is referred to a transformation of the input image that is used as the input to the computer vision model. An ideal feature descriptor should be compact, specific, invariant to noise, and it should disentangle physical information, but these qualities are often contradictory and compromises have to be made when designing features. In the following, we discuss these properties in more detail.

An ideal feature descriptor should be compact. If the same information can be represented in fewer number of dimensions, often we prefer the low dimensional representation. A high dimensional feature space is not desirable because it usually adds to the computational time of our algorithms, and more importantly can lead to the problem of curse of dimensionality [18].

Feature descriptors also should be specific while invariant to noise. This is a rather subjective trade-off. We want the feature to be specific so that it can distinguish between different attributes that are relevant to the recognition task, yet we do not want the descriptor to be sensitive to noise and irrelevant attributes. For example in a human pose estimation application, we want the feature to be sensitive to the pose, but invariant to the clothing of the person. Obviously, such a property is task specific, a more generic desirable property is disentanglement of physical information. In other words, an ideal feature descriptor should separate physical properties of the object(s) in the image.

The next obvious question is, how do we design such features? The traditional approach is to hand engineer the features, i.e. use our prior knowledge about the images to design the features. An alternative approach which is beyond the scope

of this thesis is to try to learn these features with tons of examples.

In this section we suffice to describe a well known handcrafted generic feature that has been empirically shown to perform well for a variety of visual recognition tasks.

1.1 Histogram of Oriented Gradients

The HOG feature was introduced by Dalal and Triggs [9], and has been amongst the best performing generic feature descriptors. As its name implies, the HOG descriptor consists of a histogram of gradient orientations. The procedure for calculating this descriptor is as follows.

Starting from an input image, the image is divided to an array of equally sized regions called *cells*. For each of these cells, a histogram of image gradients is calculated where each bin of the histogram corresponds to a certain gradient orientation. Furthermore contiguous cells are grouped together to form *blocks*. The histogram corresponding to each cell is then normalized with respect to all the nearby cells within its block to gain some invariance to global lighting.

The HOG feature has been shown to work very well particularly in conjunction with linear SVM classifiers [9], and until recently the state-of-the-art object detection methods[15] used HOG as the input feature. In this thesis we also extensively use HOG features as the input to classifiers for detecting face and body parts (paper A, B, and C).

2 Linear Models

This section will introduce two popular linear models that are widely used in computer vision and other pattern recognition applications. We start off by introducing the ridge regression method, which is a regularized least-square model, and is used to approximate real valued functions. After that, we briefly talk about the classification problem and how an optimal hyperplane can be estimated to separate two classes using support vector machines (SVM).

2.1 Ridge Regression

In a visual recognition task we often want to estimate a set of labels ($y \in \mathbb{R}$) from a representation of the input image ($\mathbf{x} \in \mathbb{R}^D$ where D is the feature dimension). We often assume that we can find a function mapping f such that

$$y = f(\mathbf{x}). \tag{2.1}$$

A particular class of functions to express this mapping is the class of linear models.

$$y = \mathbf{w} \cdot \mathbf{x} \tag{2.2}$$

A common approach to learn the parameters ($\mathbf{w}$) of such a model from a set of training examples $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ is by minimizing the least square error

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} ||\mathbf{w} \cdot \mathbf{x}_i - y_i||^2. \tag{2.3}$$

Unfortunately such a naive approach to parameter estimation often leads to very bad results in practice. The problem is that in many cases, we do not have enough examples to estimate the model parameters, making the optimization an ill-posed problem. Even when the number of examples is higher than the number of unknown parameters, we still run the risk of overfitting to the training data. The problem arises when we have noisy measurements and/or the model is too flexible. In such cases it is necessary to regularize the model by introducing a shrinkage term

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{N} ||\mathbf{w} \cdot \mathbf{x}_i - y_i||^2 + \lambda ||\mathbf{w}||^2, \tag{2.4}$$

where $\lambda \geq 0$ is the complexity parameter that is inversely proportional to the degrees of freedom of the model. This approach is commonly referred to as ridge regression[18], and has a simple closed form solution

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}, \tag{2.5}$$

where $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N]^T$ is a matrix consisted of training features, and $\mathbf{y} = [y_1, ..., y_N]^T$ is a vector of the corresponding labels.

2.2 Support Vector Machines

So far we have talked about solving general regression problems with linear models where we have a continuous target space. In classification problems however the target space is discrete and finite. For example, in a typical object classification task, we are interested in determining the object category corresponding to an input image from a limited set of possibilities (e.g. human, bird, cat, etc.). Classification problems are an important and well studied subject in machine learning.

For a special case of classification problems where we have binary labels $y \in \{-1, 1\}$, Vapnik [38] suggests a method for finding the optimal hyperplane that separates the two classes. This method, commonly referred to as support vector machines (SVM), defines the optimal hyperplane as the hyperplane that perfectly separates two classes with the maximum possible margin. The data samples that lie on the margin are called the support vectors. Such a hyperplane can be found by optimizing the following objective function ([18])

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w},b} \ ||\mathbf{w}||^2 \tag{2.6}$$

$$\text{subject to } \forall i, \ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \tag{2.7}$$

where b is the bias term. This optimization problem turns out to be convex, and standard quadratic programming techniques can be used to solve this problem.

Note that so far we assumed that the data is linearly separable. For cases where data is not linearly separable, we use a variant of SVM classifier called soft-margin SVM which allows for some outliers to appear on the wrong side of the margin. This is achieved by introducing a set of slack varriables $\{\xi_1, ..., \xi_N\}$, which correspond to the amount which each sample violates the margin, and altering the optimization problem as follows

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w},b} \ ||\mathbf{w}||^2 \tag{2.8}$$

$$\text{subject to } \forall i, \ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \text{ and} \tag{2.9}$$

$$\xi_i \geq 0. \tag{2.10}$$

At test time, we simply evaluate the svm model as follows

$$y = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b). \tag{2.11}$$

3 Tree Based Models

In the previous section we described two common and widely used linear methods for solving regression and classification problems. These models, however, have limited flexibility, which in many cases is not enough for modeling complex relations. There is a whole class of methods that deal with modeling nonlinear relations. But here we limit ourselves to the discussion of tree based methods as they are most used in this thesis. Methods presented in this section are based on the concept of decision trees. Decision trees are simple, intuitive, and efficient models that can be used for solving a variety of classification and regression tasks.

Decision trees consist of split nodes and leaves. The most common type of decision tree is the binary type, where decisions at each split node are based on binary tests, which are functions of the input feature. The result of these tests (whether it is true or false) determine the next node to visit (either right or left). Each leaf usually returns a single label corresponding to the most probable label which is determined based on the statistics of the training examples reaching that leaf. Starting from the root node, the decision tree is traversed until reaching a leaf node, the output of the decision tree is then simply the value which is stored at that leaf.

In computer vision, we often need to model very complex relations between the input features and the labels, and therefore it is often impossible to design a decision tree by hand. Instead, we use learning algorithms to build the tree automatically. CART (short for classification and regression trees) [3] is an example of such an algorithm, which we describe briefly next.

Assuming that we have access to a set Q of pairs of labeled examples $Q = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$. We start by building a pool of binary tests that split the

feature space into two partitions (Q_l, Q_r), and then select the split which maximizes the information gain ($\mathcal{IG}$) defined as follows

$$\mathcal{IG} = \mathcal{H}(\mathcal{Q}) - \sum_{s \in \{l,r\}} |\mathcal{Q}_s| \; \mathcal{H}(\mathcal{Q}_s), \tag{2.12}$$

where $\mathcal{H}(\mathcal{Q})$ represents the information entropy. For classification problems where we have a discrete set of labels $y \in \{1, ..., C\}$, we can define the entropy as

$$\mathcal{H}(\mathcal{Q}) = -\sum_{y=1}^{C} P_{\mathcal{Q}}(y) \log P_{\mathcal{Q}}(y), \tag{2.13}$$

where $P_{\mathcal{Q}}(y)$ corresponds to the ratio of label y in Q

$$P_{\mathcal{Q}}(y) = \frac{1}{|\mathcal{Q}|} \sum_{i \in \mathcal{Q}} \mathbb{1}(y_i = y), \tag{2.14}$$

where $\mathbb{1}$ is the indicator function. This procedure is repeated recursively for each node until we can no longer increase the information gain. At each leaf then we store the distribution of labels ($P_{\mathcal{Q}}$) from the training examples that reached that node.

3.1 Ensemble Methods

Decision trees are rarely used as a standalone machine learning tool to solve computer vision problems because of their poor generalization ability. Instead, decision trees are often used as a building block for ensemble methods. Ensemble methods combine multiple weak models to create a stronger model with higher predictive power. One should note though that the improvement can only be achieved if the base learners are diverse. In other words, it is crucial that the weak learners do not make the same mistakes. This can be done in various ways.

Bagging is a common technique to ensure diverse base models in an ensemble. The idea here is to split the training data into multiple subsets and build one base model for each part of data. Another technique, which is commonly used in conjunction with tree-base models, is to introduce randomness in feature selection. This can be achieved for example by reducing the pool of features during training of decision trees.

Both of the ideas mentioned above are used in random forests [4]. After building K decision trees $(T_1, ..., T_K)$ in this way, the outputs are simply averaged to produce the final prediction of the random forest f.

$$f(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} T_k(x) \tag{2.15}$$

In paper C, we describe how a random forest model can be used to classify pixels of an image of a football player to different body parts or background.

An alternative strategy to diversify the base models is through boosting. AdaBoost [17] is an example of boosting method that incrementally builds an ensemble from a set of weak classifiers. At the first stage, The ensemble is initialized with a base model that is slightly better than random guessing. At each later stage, the algorithm re-weights the examples based on the prediction error of the ensemble, and then trains and adds (to the ensemble) a new base model that focuses on the training examples that are not well explained by the ensemble.

Gradient tree boosting algorithm, introduced by Hastie et al. [18], is an alternative boosting method that can be used to solve both classification and regression problems. An overview of this algorithm is provided in paper D, where we use the gradient boosting algorithm to learn a global regression from an input face image to facial landmarks.

4 Pictorial Structures

For rigid objects, the pose of an object can simply be represented by a similarity matrix, including a translation and rotation. For non-rigid objects however we need a more flexible representation. This is often achieved by representing the pose with a set of landmarks[8]. For example in case of human body, we commonly define these landmarks over the major body joints. The problem of pose estimation is then reduced to locating these landmarks in the image.

In the last chapter we gave an example of how such a problem can be solved by learning classifiers that can identify each landmark independently. We also discussed the flaws of such a naive approach. Mainly the fact that the location of these landmarks are highly correlated and any independence assumption might produce inconsistent estimations of landmarks leading to invalid poses. One solution to this problem is to explicitly model the relation between these landmarks with a generative shape model. One such model, commonly referred to as *pictorial structure*, was introduced by Fischler and Elschlager ([16]) and later developed by Felzenszwalb et al. [14, 15] for the task of pose estimation and object detection.

A pictorial structure is a constellation of moving parts. In this model, each part has its own independent appearance model that estimates the likelihood of a part for each pixel on the image. The configuration of the parts is constrained by pairwise spatial constraints. This model can be best expressed with a graph structure, $G = (V, E)$ where each vertex $v_i \in V$ corresponds to a part, and each edge $(v_i, v_j) \in E$ corresponds to a connection between two parts. Let p_i be the coordinate of the center of ith part, then the pose of the object can be represented by a vector $p = (p_1, ..., p_K)$ where $K = |V|$ is the number of parts. Assuming that we have a function $s_a(p_i)$ that calculates the likelihood of ith part, and a deformation function $s_d(p_i, p_j)$ that assigns a likelihood to the configuration of each pair v_i and v_j, the pictorial structure model then assigns a global score to the configuration of

parts as follows

$$S(p) = \sum_{v_i \in V} s_a(p_i) + \sum_{(v_i, v_j) \in E} s_d(p_i, p_j). \tag{2.16}$$

This function can then be maximized to find the optimal configuration of parts

$$p^* = \arg\max_p S(p). \tag{2.17}$$

Felzenszwalb et al. [14] show that if we limit the connections of this graph to a tree structure, and use quadratic functions to model the deformation s_d, we can then solve the inference problem efficiently using generalized distance transforms [14]. This discovery in conjunction with a later paper on discriminative learning of pictorial structure model parameters [15] revolutionized the field of object detection and pose estimation and until recently pictorial structure based models were the state of the art in almost all the standard general object detection benchmarks such as PASCAL VOC [13]. We also extensively use this model throughout this thesis. In particular paper B and C use a pictorial structure model for human pose estimation, and A applies this model to tackle the problem of face alignment.

Chapter 3

Summary of papers

A Face Alignment with Part-Based Modeling

This paper addresses the problem of face alignment, that is given an image of a face we want to localize a set of landmarks on the image. These landmarks are defined over the boundaries of the eyes, mouth, and the nose as is shown in figure 1. The landmarks are chosen to capture the major deformations of the face, and therefore can be used for a variety of applications including facial expressing tracking and identity recognition.

Our aim in this paper is to learn a mapping from an image to the landmarks. We know though that this global mapping is highly nonlinear and therefore learning

(a) with parts (b) without parts

Figure 1: This figure shows the benefit of using parts in the performance of a regression function to accurately predict the location of landmark points. In both cases a linear regression model is learnt to map the appearance descriptors inside the patches to the location of the landmarks associated with the patch. Green lines represent the ground truth shape and the red lines are the prediction of the regression function. As can be seen greater accuracy is achieved when (a) using a separate regression function for each localized part as opposed to (b) one regression function from the global face patch.

Figure 2: The result of our method on a test set with ground truth information. In this figure the green lines show the ground truth landmarks and the red lines are the predictions of our method.

this function requires lots of training data that we assume we do not have access to. Instead, our strategy in this paper is to learn multiple simpler regressors for each individual facial part (i.e. eyes, nose, and the mouth) that can independently regress the location of corresponding landmarks. These individual partial models can be trained with much less examples, since the variation in appearance of local parts are much lower compared to that of the whole face. In fact we show that even linear regressors suffice to model the mapping in the examined datasets. Figure 1 shows how such a part based approach performs compared to a global regression approach.

Note though that this approach requires us to have a good estimate of the location of major facial parts. This is done by utilizing a pictorial structure model. We model the appearance of individual parts with a multivariate Gaussian distribution, and use a simple star model (with the nose at the root, and eyes and the mouth as leaves) to model the deformation of the parts.

B Using Richer Models for Articulated Pose Estimation of Footballers

In chapter 2, we briefly discussed pictorial structure models. A number of limitations are enforced on pictorial structure models to make the inference tractable (for example, we are limited to pairwise quadratic deformation functions and the dependency graph defined over parts can not contain a loop[14]). These limitations have a direct effect on the performance of pictorial structure models. In other words, the maximum scored pose[1] using a pictorial structure model in many cases is not the true pose of the object. Note that this problem is not limited to pictorial structures, this in fact is a general problem with generative modeling approach that often accurate modeling leads to intractable inference. Our solution to such a problem is very simple but we show that it can lead to significantly better performance in pose estimation problems. It is based on the observation that, although the global optimum of the pictorial structure model in many cases is not the true configuration, but nevertheless the true configuration consistently gets a high score.

Figure 3: Given an image of a football player, we want to determine a set of landmarks representing his pose. This is done by using a pictorial structure based model to generate multiple candidates for the pose and selecting the best one using a more accurate model.

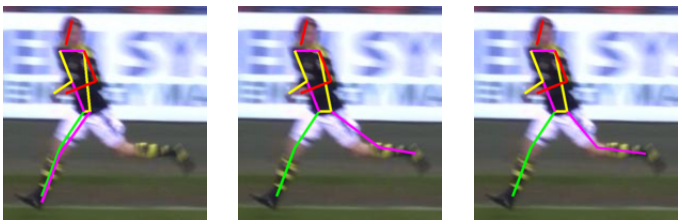
Assuming that we have access to two models, first a simple model that is fast to evaluate but not very accurate, and second an accurate model that is too expensive to exhaustively evaluate for all the possible configurations. Our solution is then to use the simple model to generate a set of highly likely candidates, and then only evaluate the expensive model on these configurations to pick the best one.

Generating multiple hypotheses might not be straightforward in a general case. The problem here is that the model might rank very similar poses as the top scoring poses. In these cases, we might need to introduce additional constraints to enforce diverse solutions[32]. Note that producing diverse solutions is essential to quickly explore a large portion of the parameter space.

Our more accurate model adds two additional components to a state of the art pictorial structure based model[39]. Firstly, we enforce an exclusion principle to avoid the problem of double counting of the limbs, which is a common problem

[1]The pose is expressed with a set of landmark locations corresponding to body joints.

with pictorial structure based models. In other words, our model doesn't assume that the location of the left and right limbs such as arms and legs are independent random variables. Furthermore our model includes a segmentation score that scores configurations that explain more foreground pixels higher. Inferring the most likely pose with respect to such a complex model is very expensive. But we use the framework of [32] to generate multiple hypotheses for the pose and only evaluate our model on the 1000 top scored configurations to pick the best one. Figure 4 shows an example where our reranking model is able to improve the result of a pictorial structure base model. On average, we show that the top scoring configuration returned by our model is 15% more likely to be the true configuration.

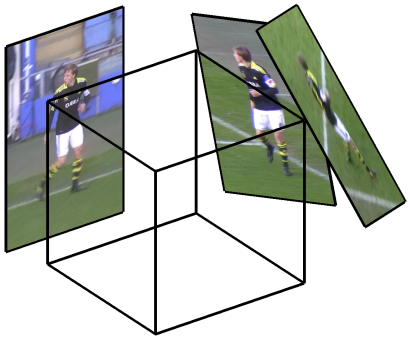


(a) FMP (b) Reranking (c) Closest to G.T.

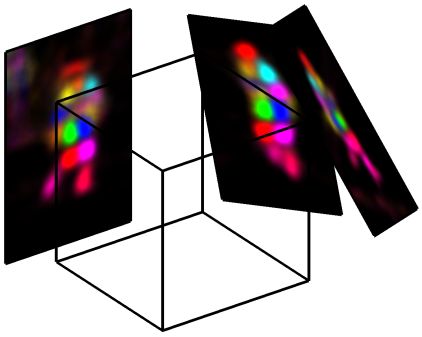
Figure 4: This figure shows (a) the result of FMP compared to (b) our reranking function, in addition to (c) the results of picking the closest configuration to the ground truth from a set top 1000 configurations.

C Multi-view Pose Estimation of Human Body

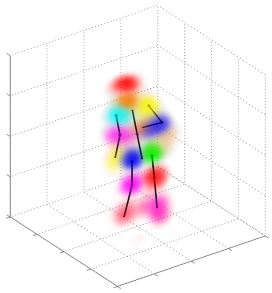
This paper focuses on the problem of multi-view pose estimation. Given images of a football player captured with multiple calibrated cameras, our goal is to reconstruct the pose of the body in 3D (See figure 5). Our strategy here is to use an efficient discriminative model to estimate the likelihoods of each body part over a 3D voxel, and then use a generative model based on 3D pictorial structures to produce a consistent hypothesis for the pose of the person across multiple views.



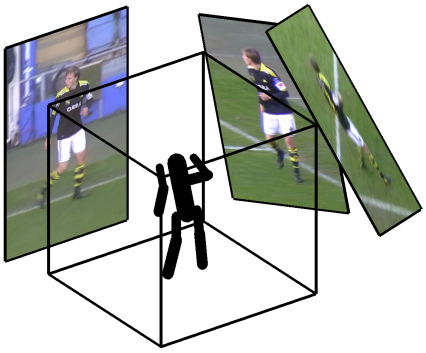
(a) Images are captured from three calibrated cameras



(b) Part scores are calculated with a single 2D discriminative model



(c) 2D part scores are aggregated over discrete 3D locations to generate consistent likelihoods across views



(d) Pose priros are used to infer a single 3D hypothesis

Figure 5: A general overview of our multi-view pose estimation framework. A 2D discriminative model is first used to classify pixels in each image as belonging to a part or the background. The results are then back-projected to a 3D volume. We find corresponding mirror symmetric parts across views by introducing a latent variable. Finally, a part-based model is used to estimate the 3D pose.

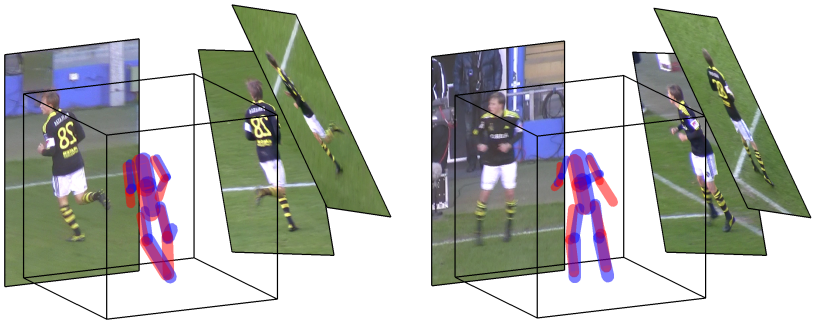


Figure 6: Final 3D poses obtained by taking, for each part independently, its most probable state over the grid. The mirror ambiguity is solved jointly. Estimation is red and ground truth is blue.

To learn a discriminative model that can directly estimate the likelihood of each part in 3D, one needs access to labeled 3D data and the associated calibrated cameras. There are two major problems with this approach, firstly collecting 3D data and labeling them is very expensive, and furthermore this approach requires fixed camera views – 3D data captured from a certain camera setup can only be used for that particular setup, and a small change in the pose of any of the cameras would require recollection of the training data.

In contast, our approach relies on 2D discriminative models that assign a likelihood to each pixel from each view independently, and we then project these scores on a discretized 3D space to produce 3D likelihoods. At the final stage a 3D pictorial structure model is used to select the optimal configuration of body parts with respect to simple limb length priors. Figure 6 shows examples of the final 3D pose estimated by our model comparing to the ground-truth annotations.

D One Millisecond Face Alignment with an Ensemble of Regression Trees

In this paper we again revisit the problem of face alignment, but with a completely different strategy. In section A, we briefly talked about the challenges of a global regression approach for estimating facial landmarks, namely the fact that the global mapping between the image features and the landmarks is highly nonlinear and difficult to model, and building such a model would require lots of training examples. Instead of averting the problem as we did in A, here, we choose to address the problem, and we show that such an approach can achieve state of the art performance for facial landmark detection while being much faster than any other previous method. (See figure 7)

Figure 7: Selected results on the HELEN dataset. An ensemble of randomized regression trees is used to detect 194 landmarks on face from a single image in a few milliseconds.

Our aim here is to learn a global regression model that directly estimates the location of landmarks from the input image. This is achieved by using an ensemble model consisting of thousands of shallow regression trees. To train such an ensemble we only use 2000 face images, but the training data is extended by a factor of 20 by warping images with random face shapes. The procedure for training the ensemble is based on the gradient boosting algorithm, which allows for quick reduction of training error by learning complementary decision trees. The gradient boosting algorithm though is prune to over-fitting problem and in the paper we discuss different strategies to avoid this problem. Furthermore we empirically show that

use of a cascade of smaller regressors is much more effective than using a single large regression model.

Figure 8 shows a few examples of the output of our method on random images from the HELEN[28] database.

Figure 8: Final results on HELEN[28] database.

E Real-time Face Reconstruction from a Single Depth Image

This paper presents a novel approach for dense correspondence estimation of deformable objects from single depth images. While our method is generic and can be applied to any deformable object, our experiments are limited to human faces. Final correspondences estimated by our method are highly accurate and can be used for a variety of applications, including 3D face shape and expression reconstruction, texture unwrapping, retexturing and retargeting in real-time. (See figure 9)

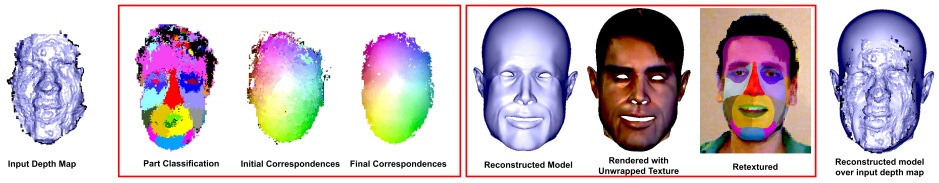


Figure 9: Our method starts with estimating dense correspondences on an input depth image, using a discriminative model. A generative model parametrized by blend shapes is then utilized to further refine these correspondences. The final correspondence field is used for per-frame 3D face shape and expression reconstruction, allowing for texture unwrapping, retexturing or retargeting in real-time.

We start off by defining a generative model based on blend-shape deformations. We represent the face with a mesh model consisted of $M = 11211$ vertices. Each blend-shape then contains a $3 \times M$ dimensional delta vector (each of these blend-shapes can correspond to a particular face shape or expression). Our generative model is simply defined by a linear combination of these blend-shapes transformed by a similarity matrix that corresponds to the pose of the head.

We discuss in the paper that trying to directly minimize the parameters of such a generative model based on the reconstruction error leads to very poor results. This is because the error function is highly nonlinear and very hard to optimize. Instead, we utilize a variant of ICP algorithm. Starting with an initial estimate of the correspondences, we estimate the model parameters, and then fix the model parameters and recalculate the correspondences. This procedure is repeated multiple times until convergence. But the problem is not solved yet, such an iterative procedure still requires a good initialization of parameters to avoid divergence. This problem is addressed by using a discriminatively trained regressor that directly estimates the correspondences from the input image. This initial estimate is very crude, but after a few iterations of the ICP procedure we are able to significantly reduce the error. At a final step, we use the model parameters estimated by the ICP procedure to initialize the particle swarm optimizer (PSO) [25]. PSO then explores the nearby solutions and pick the the best configuration according to the true reconstruction error.

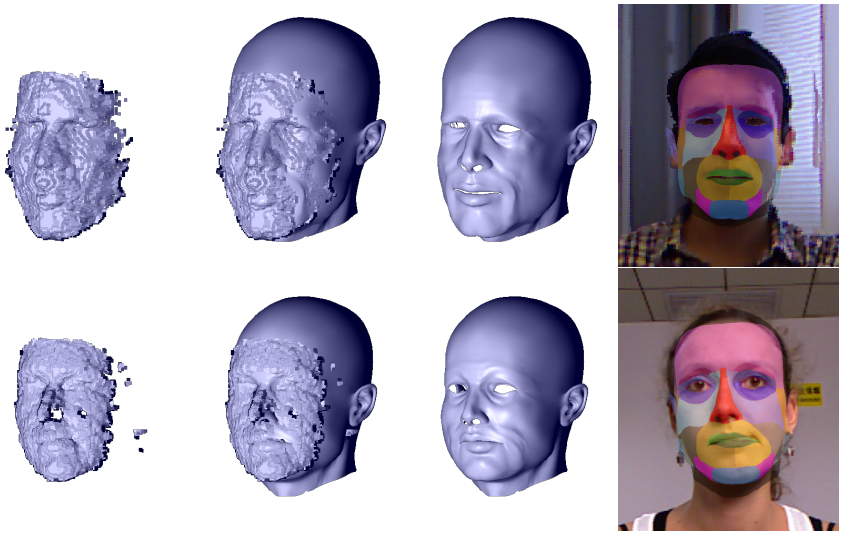


Figure 10: Qualitative results on real data captured using Kinect camera. From left to right, we show the input depth data, the depth data overlaid on the reconstructed model, the reconstructed model and the parts overlaid on the rgb image (which was only used for visualization). Some of these examples are from [5].

Figure 10 shows examples of the output of our method used for facial reconstruction and retexturing.

Chapter 4

Conclusion

As promised in the introduction, this thesis delivered practical solutions for some computer vision problems particularly around the subject of correspondence estimation. We would like to emphasize on the practicability of our solutions, which separates this work from many others. Many theoretically plausible solutions fail in practice because of a number of problems:

- In practice we often do not have access to infinite amount of training data and the labeled data is often very scarce.
- A particular kind of annotation might be easier to do by human labor than others.
- In practice we do not have perfect sensors and our measurements are often very noisy.
- We care about the training and test time in practice and our computational resources are limited.
- Often, our algorithms need to run on available consumer hardwares. The algorithms that can utilize the available hardware (multi-core CPUs, GPUs, and etc.) more efficiently are preferable.

These are a few examples of the limitations and concerns we face in practice, which must be taken into consideration when designing computer vision algorithms. All of the solutions provided in this thesis in one way or another are affected by these limitations, and in many cases they explain why we made the design choices and decisions we made throughout this thesis.

If we have access to a large dataset of labeled data, it makes sense to opt for a purely discriminative approach. The choice of features could potentially have a large impact on the performance of the method, but even if do not have any

information about the problem domain, we can still use end-to-end learning methods such as convolution neural networks[29, 30, 26] to learn the features and the classifiers/regressors jointly.

But how much training data is enough to learn such mapping? Of course the answer depends on the desired level of accuracy and also variation in the data. We will need more training examples to achieve higher accuracy. Also, high variance in the data often means that the relation between the input features and the labels is more complex. A more complex relation requires a more complex model, and consequently more examples are needed to learn the mapping.

When labeled data is scarce, a generative approach might be more appropriate. We can handcraft a generative model using our prior knowledge, or learn it entirely from the data. But the common approach is to use our prior knowledge to choose a proper statistical model, and learn the parameters of the model from the data. In the recent years, more researchers have shifted their focus from handcrafted methods to data-driven methods. The motivation behind this shift is twofold. Firstly, we are often interested in generic approaches that can be applied to many different problems. Data driven approaches are usually more generic and therefore more desirable in this respect. Furthermore, we can often make a more accurate model by increasing the number of training data.

We described pros and cons of both generative and discriminative approaches in the introduction, and discussed some practical solutions to these problems throughout the thesis. While the focus of this thesis has been on the subject of correspondence estimation, some of the ideas discussed in this work can be applied to many other applications even outside the computer vision domain, and our hope is that the thesis can be useful for a broader audience who have an interest in using machine learning tools to solve real-world problems.

1 Future Work

There are a number of subjects that we did not explore in this thesis, some of which we find interesting to investigate for a future work. One important subject is representation learning. While in this work we only used predefined features, a large body of papers have been recently published about learning feature representations from data particularly using convolutional neural networks (CNNs). This line of research was sparked by the impressive results reported by Krizhevsky et al. [26] on ImageNet classification challenge. While most of the research in this area has been focused on learning pose invariant features, we find it interesting to learn pose sensitive representations. A closely related idea that we find interesting to explore is about learning inter-class pose representations. We believe it is possible to learn generic pose features across multiple classes of objects, and such a representation can potentially help improve the performance of various recognition tasks.

Bibliography

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006. ISBN 0387310738.

[3] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman and Hall, 1984. ISBN 0-412-04841-8.

[4] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: a 3d facial expression database for visual computing. *Transactions on Visualization and Computer Graphics*, 2013.

[6] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2887–2894, 2012.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, 1998.

[8] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2005.

[10] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[11] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc J. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proceedings*

of the Conference on Computer Vision and Pattern Recognition, pages 2578–2585, 2012.

[12] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1078–1085, 2010.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Internation Journal of Computer Vision*, 61(1):55–79, 2005.

[15] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[16] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[17] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[18] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* New York: Springer-Verlag, 2001.

[19] Vahid Kazemi, Hossein Azizpour, Magnus Burenius, and Josephine Sullivan. Multi-view pose estimation of human body. In *Submission to International Journal of Computer Vision*, 2014.

[20] Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view body part recognition with random forests. In *Proceedings of the British Machine Vision Conference*, 2013.

[21] Vahid Kazemi, Cem Keskin, Johanatan Taylor, Pushmeet Kohli, and Shahram Izadi. Real-time face reconstruction from a single depth image. In *Proceedings of International Conference on 3D Vision*, 2014.

[22] Vahid Kazemi and Josephine Sullivan. Face alignment with part-based modeling. In *Proceedings of the British Machine Vision Conference*, pages 27.1–27.10, 2011.

[23] Vahid Kazemi and Josephine Sullivan. Using richer models for articulated pose estimation of footballers. In *Proceedings of the British Machine Vision Conference*, pages 6.1–6.10, 2012.

[24] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *CVPR*, 2014.

[25] James Kennedy, Russell Eberhart, *et al.* Particle swarm optimization. 4(2): 1942–1948, 1995.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[27] Julia Lasserre and Christopher M. Bishop. Generative or discriminative? getting the best of both worlds. *BAYESIAN STATISTICS*, 8:3–24, 2007.

[28] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir D. Bourdev, and Thomas S. Huang. Interactive facial feature localization. In *Proceedings of the European Conference on Computer Vision*, pages 679–692, 2012.

[29] Yann Lecunn, Fu-Jie Huang, and Leon Bottou. *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2004.

[30] Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML, pages 609–616, 2009. ISBN 978-1-60558-516-1.

[31] Andrew Y. Ng and Michael I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, pages 841–848. MIT Press, 2002.

[32] D. Park and D. Ramanan. N-best maximal decoders for part models. In *Proceedings of the International Conference on Computer Vision*, 2011.

[33] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems*, 2006.

[34] 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 *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[35] Min Sun, Pushmeet Kohli, and Jamie Shotton. Conditional regression forests for human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3394–3401, 2012.

[36] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The Vitruvian Manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2012.

[37] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2014.

[38] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

[39] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

Part II

Included Publications

Paper A

Face Alignment with Part-Based Modeling

Vahid Kazemi and Josephine Sullivan

Published in
British Machine Vision Conference, 2011

Face Alignment with Part-Based Modeling

Vahid Kazemi and Josephine Sullivan

Abstract

We propose a new method for face alignment with part-based modeling. This method is competitive in terms of precision with existing methods such as Active Appearance Models, but is more robust and has a superior generalization ability due to its part-based nature. A variation of the Histogram of Oriented Gradients descriptor is used to model the appearance of each part and the shape information is represented with a set of landmark points around the major facial features. Multiple linear regression models are learnt to estimate the position of the landmarks from the appearance of each part. We verify our algorithm with a set of experiments on human faces and these show the competitive performance of our method compared to existing methods.

1 Introduction

This paper presents a new method for accurate face alignment with part-based modeling. Although we focus on the class of human faces, this method could potentially be applied to any type of deformable object. We aim to learn a regression function mapping a feature representation of the appearance of the face to its shape represented by a set of connected landmarks forming contours around the major facial features. Ideally, we want to use linear regression functions to describe this mapping as they need less training data, have a lower chance of over-fitting, and are fast to compute. However, the relation between the global appearance of an object and its shape is highly nonlinear. Previously, piece-wise linear regression has been applied [15, 12] to deal with this non-linearity. Instead, in this work, the strategy is to learn regression functions for individual parts, see figure 1. The parts are chosen to ensure the linearity of the regression mapping. The main advantage of this approach is that it requires less training data, although it necessitates good part-detection.

Part-based methods, mostly used in recognition tasks, train different classifiers for each part of the model. Assuming each part is a rigid structure, deformation is

(a) with parts (b) without parts

Figure 1: This figure shows the benefit of using parts in the performance of a regression function to accurately predict the location of landmark points. In both cases a linear regression model is learnt to map the appearance descriptors inside the patches to the location of the landmarks associated with the patch. Green lines represent the ground truth shape and the red lines are the prediction of the regression function. As can be seen greater accuracy is achieved when (a) using a separate regression function for each localized part as opposed to (b) one regression function from the global face patch.

limited to the relative transformation of each part. The optimal solution for such a multi-part based matching problem can be found efficiently by simplifying the dependency of parts to form a tree structure as is presented by Felzenszwalb *et al.*[8]. This method cannot be directly applied to solve dense matching problems because it requires creating individual classifiers to detect each landmark which is not only computationally expensive but impractical, since most of the landmarks do not have a distinctive local appearance.

A large group of methods have been developed which rely on the global appearance of a deformable object to tackle the alignment problem, these include Active Shape Model(ASM) [4], and Active Appearance Model(AAM) [3]. Some attempts have been made to improve the robustness, and accuracy of these methods [9, 1, 10, 5, 13], but the main problem which remains unsolved in these methods is that they need a good initial estimate and are not able to adapt the model to fit a subject when the initial error is high. As an example in face tracking applications, AAM usually fails to converge when there is a sudden large deformation or motion of the subject.

One main advantage of our method compared to the global methods just described is that the regression functions directly estimate the landmark positions by-passing the need to perform iterative non-linear optimization on a complicated cost function. As a result, our method can be used on image sequences with fast motion, and it does not need any initialization. Furthermore the part-based nature of our method, and also the use of HOG descriptors [6] (in contrast to intensity vectors as in AAM), enhance the robustness and generalization ability of our algorithm. On the minus side the performance of our method is highly dependent

on a proper part detection and this requires strong and distinctive features in the appearance of the object. In the case of a human face which is the focus of our work, a sensible selection of parts is a division into the nose, mouth and the eyes. For the more general case we would need an automatic part selection method which is a an interesting and challenging topic investigated for future work.

2 Landmark localization via regression

In this work the localization of landmark points on the face is viewed as a regression problem. Assume we have N training image patches of the same size and that the appearance of each patch is described by a feature vector $\mathbf{f}_i \in \mathbb{R}^K$. Each patch contains L landmark points whose coordinates are defined relative to an origin at the centre of the patch and are then stacked into the vector $\mathbf{X}_i$. This training data is used to learn a regression function

$$q : \mathbb{R}^K \longrightarrow \mathbb{R}^{2L} \quad \text{s.t.} \quad q(\mathbf{f}) = \mathbf{X} \tag{1}$$

mapping a feature vector $\mathbf{f}$ to the coordinates of the landmark points $\mathbf{X}$. There are many options to approximate $q(\cdot)$ such as linear regression, nearest neighbour regression and relevance vector machines. We focus on modelling q as a linear function

$$q(\mathbf{f}) = W\,\mathbf{f} + \mathbf{w} \tag{2}$$

where $W \in \mathbb{R}^{2L\times K}$ and $\mathbf{w} \in \mathbb{R}^{2L}$. The question then remains which approach should be used to estimate $(W, \mathbf{w})$. The performance of several standard methods - ordinary least squares regression, ridge regression, principal component regression - are investigated and the results are reported upon and compared to the baseline of a nearest neighbour regression function in the experimental section. However, to summarize ridge regression, which is fast in both training and testing phases, was found to perform well. Ridge regression minimizes a least squares loss function combined with a regularization term:

$$W_{\text{ridge}}, \mathbf{w}_{\text{ridge}} = \arg\min_{W,\mathbf{w}} \left(\sum_{i=1}^{N} \|\mathbf{X}_i - W\,\mathbf{f}_i - \mathbf{w}\|^2 + \lambda(\text{trace}(WW^t) + \mathbf{w}^t\mathbf{w}) \right) \tag{3}$$

where λ is the non-negative regularization factor and defines the complexity of our model.

The approximated regression functions will be applied to arbitrary image patches $I_{\mathbf{b}}$. Such patches are defined by a bounding box $\mathbf{b} = (\mathbf{x}, s\,w, s\,h)$, where $\mathbf{x}$ is the centre of the patch, w and h are the width and height of the training patches and s is the ratio between the width of the test patch and the training patches. We assume test patches have the same aspect ratio as the training data. Then the

coordinates of the landmark points, estimated by our learnt regression functions, have to be rescaled and translated:

$$g(I_{\mathbf{b}}) = s\, q(\mathbf{f}_{I_{\mathbf{b}}}) + \mathbf{x} \tag{4}$$

where $\mathbf{f}_{I_{\mathbf{b}}}$ is the feature description of the image patch $I_{\mathbf{b}}$.

2.1 Feature description of an image patch

In this subsection we introduce the feature descriptor $\mathbf{f}_{I_{\mathbf{b}}}$ which is used to describe the appearance of an image patch. We use a variant of the PHOG [2] descriptor. Figure 2 shows a schematic representation of this descriptor. At the first level a histogram of gradient orientations of the whole patch is computed. While at the second level the patch is divided into 8 sub-regions. Each of these regions can once again be recursively divided into 8 more sub-regions until the required level of detail is obtained. However, our experiments show that it is more effective to just recursively subdivide the square sub-regions at each level of the pyramid into 8 more sub-regions. At the end, all the histograms are concatenated to form the final descriptor. This descriptor allows us to capture the appearance of an image

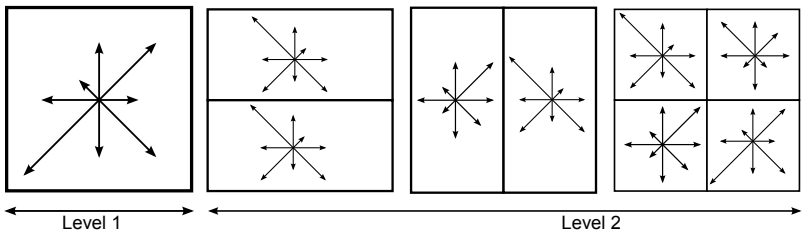


Figure 2: This figure demonstrates how an appearance descriptor is calculated from an image patch. The main patch is divided into 8 sub-regions. For each sub-region the histogram of gradient orientations is calculated. These histograms are then concatenated to form the final descriptor.

patch at different scales (same as PHOG) as well as the joint appearance of adjacent regions both horizontally and vertically. In this way we can better represent shape information while maintaining a degree of spatial invariance.

3 Part based regression

As stated we want to use linear regression to model the mapping from the appearance of a face to its shape. However, the relationship between the global appearance of the face and the position of all its landmark points is non-linear. We need, therefore, to model simpler relationships which can be better approximated with a linear function. Our strategy is to split the face into P parts, see figure

4, and learn separate regression functions, mapping each part's appearance to its associated landmark positions. This approach, of course, introduces a new set of problems. The first of these is: How do we define and set the number of parts?

Solving this problem automatically is non-trivial and is not tackled in this work. The parts and their number are set by hand, see figure 4. The parts are defined as a partition of the landmark points and the choice of this partition was guided by an effort to ensure

- the part landmark points can be well aligned across the training images,
- the part can be detected and accurately located in novel images and
- linear regression accurately models the mapping from each part's image feature space to the coordinates of its landmark points.

More formally we partition the set of L landmark points into P subsets with $L_1, \ldots, L_P$ points respectively and where each set of landmark points are spatially grouped. For instance in the face, the landmark points associated with the nose form one subset. The P regression functions we now learn are denoted by

$$q_p : \mathbb{R}^K \longrightarrow \mathbb{R}^{2L_p} \quad \text{s.t.} \quad q_p(\mathbf{f}) = \mathbf{X}_p \quad \text{for } p = 1, \ldots, P \tag{5}$$

where the learnt mapping is now from an image patch surrounding the particular subset of landmark points to their coordinates defined relative to the centre of the patch.

This means less training data is needed to learn each $q_p(\cdot)$ but they can cover a larger amount of variation for all the parts over learning one global regression function. Using parts, also, ensures that linear regression models are more likely to be a better approximation to the true mapping. As before, if w_p represents the width of the training image patches for part p, then the mapping from an image patch extracted from the bounding defined by $\mathbf{b} = (\mathbf{x}, s\,w_p, s\,h_p)$ is:

$$g_p(I_{\mathbf{b}}) = s\,q_p(\mathbf{f}_{I_{\mathbf{b}}}) + \mathbf{x}. \tag{6}$$

While the second problem introduced by this part based strategy is: Given a novel image, I, how do we find the location and size of each part within I? Fortunately, for us there are several standard approaches for achieving this and the one we will adopt is based on training part detectors and applying spatial constraints between parts in the manner of pictorial structures [8] which is described next.

3.1 Part detection

We model the appearance of an individual part by fitting a multivariate Gaussian distribution to the part's appearance descriptor. This is a very simple model and forms the basis for the match score, more sophisticated classification scores could be

used, but it is sufficient for the face data-sets we examine. The spatial constraints between parts of the face are represented in the form of a tree. In our case, the nose is defined as the root of the tree, and the mouth and eyes are the leaves. The relative location of each pair of connected nodes are then modeled with a 2D Gaussian model. To detect the location of parts in a novel image, a window of a fixed size is slid across the image, and the Mahalanobis distance of the appearance descriptors from the mean model is calculated for each window to create a distance map. The method presented by Felzenszwalb *et al.*[8] is then used to find an optimal solution for the matching problem.

The details are as follows let $\mathbf{b}_p = (\mathbf{x}_p, s\,w_p, s\,h_p)$ denote the bounding box of the pth part. The scale factor s is the same for each part and is found by finding the global scale of the face via a frontal face detector such as the one defined by the Viola and Jones face detector. Then the unknown parameters of the bounding boxes $\mathbf{x}_p$ are found by solving:

$$\min_{\mathbf{x}_1,\ldots,\mathbf{x}_P} \left(\sum_{p=1}^{P} m_p(\mathbf{x}_p, I) + \sum_{(i,j)\in E} d_{ij}(\mathbf{x}_i, \mathbf{x}_j) \right) \tag{7}$$

where each $m_p(\mathbf{x}_p, I)$ computes a score of how well the appearance of image patch $I_{\mathbf{b}_p}$ matches the model of part p's appearance

$$m_p(\mathbf{x}_p, I) = (\mathbf{f}_{I_{\mathbf{b}_p}} - \boldsymbol{\mu}_p)^t \, \Sigma_p^{-1} \, (\mathbf{f}_{I_{\mathbf{b}_p}} - \boldsymbol{\mu}_p) \tag{8}$$

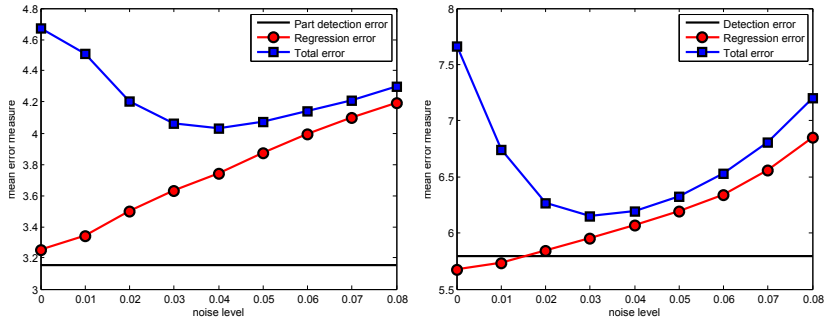
and the appearance is described with the same type of descriptor, $\mathbf{f}_{I_{\mathbf{b}_p}}$, as used in the regression functions. Each $d_{ij}(\mathbf{x}_i, \mathbf{x}_j)$ measures the likelihood of the relative layout of the ith and jth part:

$$d_{ij}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j - \mu_{ij})^t \, \Sigma_{ij}^{-1} \, (\mathbf{x}_i - \mathbf{x}_j - \mu_{ij}) \tag{9}$$

and E is the list of edges in the tree structure defining the face. As the spatial deformation scores are modelled with a Mahalanobis distance the optimization problem defined in equation (7) can be efficiently solved using distance transforms and dynamic programming as described in [8].

3.2 Learning the part regression functions

For test images there will, in general, be small inaccuracies in the localization of each part and estimation of its scale. Therefore during the training of the individual regression functions we compensate for this fact by augmenting the training data. To do this a zero mean Gaussian noise was added to the location of patches and we created a variety of patches with slightly different scales and positions for every image. The regression model then can learn the mapping between shifted appearance descriptors to the correct shape data. Figure 3 shows a benchmark of algorithm with different levels of noise. As can be seen adding about 3 to 4% noise to the position of parts in the training set, improves the final results.



(a) with parts (b) without parts

Figure 3: This figure shows the effect of adding noise to the location of the patches used in training on the performance of the regression functions. The *Total error* curve displays the average error in the prediction of the landmark positions when the regression functions use the output from the automatic part detection and the *Regression error* curve the error when instead the ground truth location of the part is used. Observe that perturbing the training data up to a certain noise level increases the regression functions robustness to inaccuracies in the part detection. Note the use of parts (a) significantly increases the accuracy of the algorithm.

4 Experiments and results

In our experiments we have used the IMM face database [14] which contains 240 still images of 40 different human faces with the resolution of 640×480 pixels with the average head size 240×320. The database includes a variety of facial expressions and head orientations, and contains both male and female subjects. Each image comes with 58 annotated landmarks, outlining the major facial features including eyebrows, eyes, nose, mouth, and jaw. Although in our experiments we only use 44 landmarks (excluding the landmarks around the jaw).

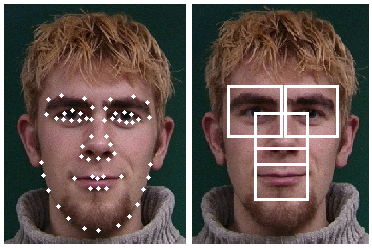


Figure 4: An annotated image from the IMM dataset. Each image comes with 58 annotated landmarks outlining the major facial features. We calculate the bounding boxes for each part based on the location of these landmarks.

We performed 40-fold cross-validation on the IMM dataset to benchmark the performance of our method. 40 different models were trained, for each model we excluded all of the images from one of the test subjects. Furthermore, we have tested our algorithm on novel image sequences and the qualitative results are presented in figure 7.

Four individual part detectors are trained for detecting the left and right eyes, nose, and mouth. We have used 90×90 pixel patches to model each part which we found to be optimal in our dataset. After locating the parts, we extract the appearance parameters around the detected point and compute the shape parameters based on the regression model.

The optimal configuration for the appearance descriptor which we found by experiment is $L = 3$, and $b = 5$, where L is the number pyramid levels, and b is the number of histogram bins, which will give a feature descriptor with length equal to $b(1 + \sum_{l=1}^{L} 4^{2l}) = 845$. For part detection also we have used similar parameters except the pyramid level which was set to $L = 2$.

There are many options for finding the parameters of the linear regression. Figure 5 shows a comparison of the results from nearest neighbor, ordinary regression, and principal component regression with different dimension reduction rates. The results show that regularization significantly improves the performance of the

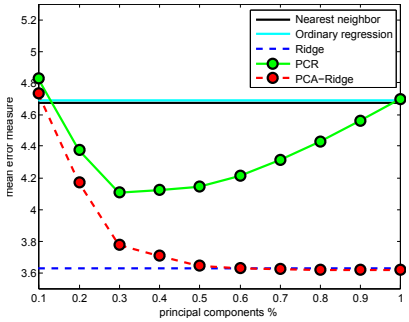


Figure 5: Comparison of performance of different regression methods. The results show that proper regularization significantly improves the results.

regression model. The regularization factor is set to $\lambda = 200$ by inspecting its performance on a few validation images. Using PCA with ridge regression gives the top performing results with enough components but the results are not significantly better than ridge regression, although it can potentially improve the speed because of the reduction of the feature dimensionality.

Table 1 shows a comparison of performance of our method with several existing methods on the same dataset. The error measure used in benchmarking is the mean of the Euclidean distance between detected landmarks and their ground truth location. The resolution of the images used in these experiments is 640×480 pixels, and size of the head is approximately 240×320 pixels. The average mean error as

a result of cross-validation on the whole dataset in our method is about 4 pixels. This is better than the results of various implementations of AAM and ASM that we found. Figure 6 shows the detected and ground truth location of the landmarks on a different set of test images. We have compared the results with standard AAM implementation by Stegmann *et al.*[14], and CCA-AAM by Donner *et al.*[7], in addition to the results from [11] (ASM, DFM). The results show the superior precision of our algorithm despite the fact that the other methods need initialization with ±10% error.

Method	Input Type	Mean error
Our method	Gray	**4.03**
DFM	Gray	4.80
CCA-AAM	Gray	5.70
AAM	Color	5.92
AAM	Gray	6.03
ASM	Gray	6.20

Table 1: Mean point to point error of detected landmarks in full size images. The error measure is the mean of the Euclidean distance between the detected landmarks and their ground truth location in pixels.

We have also tried our method on a variety of novel videos, and image sequences and an example is demonstrated in figure 7.

5 Conclusion

In this paper we have introduced a new method for the alignment of faces which has proven to be effective in real world applications. Given 240 images of human faces, we were able to create a model that can be used to align arbitrary faces with acceptable precision. The experiments prove that the method has a good generalization ability and is competitive in terms of precision with global methods such as Active Appearance Models.

As mentioned before the choice of parts is a critical factor in the performance of the algorithm. Further improvement could be achieved by designing an automated method to find the best part configuration. The parts need to be reliably detected and should be evenly spread across the face to give a complete representation of the face's appearance. Furthermore the algorithm can be extended to improve the precision by doing a local search around the detected landmarks to find the optimal match. Alternatively the detected landmarks could be used as an initial estimate for an iterative matching process such as AAM; this can potentially lead to better results since by providing a good initialization to AAM the chances of converging to the global optima increases.

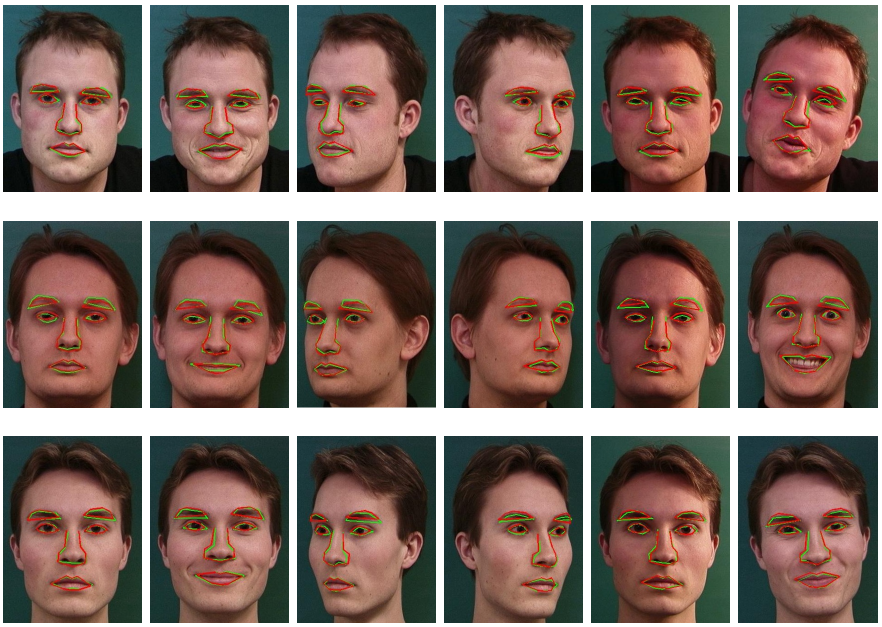


Figure 6: The result of our method on a test set with ground truth information. In this figure the green lines show the ground truth landmarks and the red lines are the predictions of our method.

References

[1] B. Amberg, A. Blake, and T. Vetter. On compositional image alignment, with an application to active appearance models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009.

[2] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

[3] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, 1998.

[4] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[5] D. Cristinacce and T. F. Cootes. Boosted regression active shape models. In *Proceedings of the British Machine Vision Conference*, 2007.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2005.

Figure 7: The result of our method, trained using IMM dataset, on an unknown subject, with unknown expressions. The results show that our model covers unseen faces with novel expressions and lighting conditions.

[7] R. Donner, M. Reiter, G. Langs, P. Peloschek, and H. Bischof. Fast active appearance model search using canonical correlation analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1690–1694, 2006.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Internation Journal of Computer Vision*, 61(1):55–79, 2005.

[9] X. Hou, S. Z. Li, H. Zhang, and Q. Cheng. Direct appearance models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2001.

[10] Y. Huang, Q. Liu, and D. N. Metaxas. A component-based framework for generalized face alignment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 41(1):287 –298, 2011.

[11] A. Kuhl, T. Tan, and S. Venkatesh. Automatic fitting of a deformable face mask using a single image. In *MIRAGE*, pages 69–81. Springer-Verlag, 2009.

[12] R. Okada and S. Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. In *Proceedings of the European Conference on Computer Vision*, 2008.

[13] Julien Peyras, Adrien Bartoli, Hugo Mercier, and Patrice Dalle. Segmented aams improve person-indepedent face fitting. In *Proceedings of the British Machine Vision Conference*, 2007.

[14] M. B. Stegmann, B. K. Ersbøll, and R. Larsen. Fame - a flexible appearance modelling environment. *IEEE Transactions on Medical Imaging*, 22(10):1319–1331, 2003.

[15] A. Thayananthan, R. Navaratnam, B. Stenger, Ph. H. S. Torr, and R. Cipolla. Multivariate relevance vector machines for tracking. In *Proceedings of the European Conference on Computer Vision*, 2005.

Paper B

Using Richer Models for Articulated Pose Estimation of Footballers

Vahid Kazemi and Josephine Sullivan

Published in
British Machine Vision Conference, 2012

Using Richer Models for Articulated Pose Estimation of Footballers

Vahid Kazemi and Josephine Sullivan

Abstract

We present a fully automatic procedure for reconstructing the pose of a person in 3D from images taken from multiple views. We demonstrate a novel approach for learning more complex models using SVM-Rank, to reorder a set of high scoring configurations. The new model in many cases can resolve the problem of double counting of limbs which happens often in the pictorial structure based models. We address the problem of flipping ambiguity to find the correct correspondences of 2D predictions across all views. We obtain improvements for 2D prediction over the state of art methods on our dataset. We show that the results in many cases are good enough for a fully automatic 3D reconstruction with uncalibrated cameras.

1 Introduction

This work tackles the problem of automatically reconstructing the 3D pose of a person, in particular a football player, from multiple images taken from uncalibrated affine cameras. We adopt a bottom up approach, summarized as, localize the skeletal 2D joints in each image independently and then perform factorization with limb length constraints to estimate the 3D pose. The joint localization task is the more challenging part and is the work's main focus.

Localization of a person's limbs in an image is very difficult for a myriad of reasons most notably the range of articulations of the person (especially true in sports footage), self-occlusion, foreshortening of limbs and motion blur. However, in recent years significant progress has been made with the introduction of pictorial structure type models using discriminatively learned parts [6, 4, 15]. These models compromise between accurate modeling of the underlying flexibility in the appearance and spatial configuration of the person's limbs and computational concerns to make the parameter learning and the inference tractable.

Despite this progress, though, the results are far from perfect in real world scenarios. Figure 1(a) shows the results from the state-of-the-art *Flexible Mixture of Parts* (FMP) model [15] on images from our football dataset. The right of figure 1(a) shows an example of a common failure. The problem is partly due to the simplifications made in the modeling. However, the main observation exploited in this work is that while the

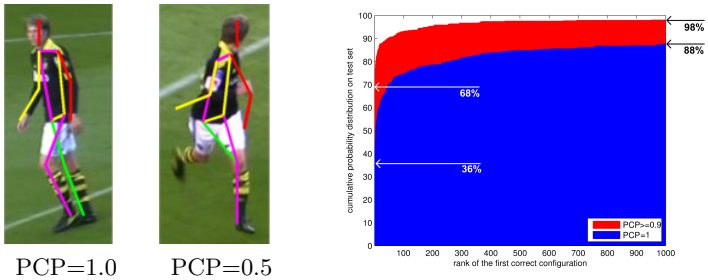


Figure 1: **(a)** Shown is the top scoring configuration returned by the FMP model and its PCP score for two images. The PCP score is the proportion of correctly localized limbs. **(b)** This is a cumulative histogram of the rank of the first correctly predicted pose by the FMP model. In 36% of the test cases the top scoring configuration has PCP=1. While 88% of the time there exists a configuration with PCP=1 in the top scoring 1000 configurations. These percentages change to 68% and 98% when the definition of a correct configuration is lowered to having PCP ≥ 0.9.

true configuration might not always correspond to the global optimum of the FMP's cost function, it frequently gets a high score. One can observe this by examining figure 1(b). It shows that on our football dataset a correct configuration - all the parts are localized correctly - is in the top 1000 scoring configurations w.r.t. the FMP cost function 88% of the time, while the top scoring configuration is a correct configuration only 36% of the time.

As a correct configuration is frequently in the set of the top n scoring configurations w.r.t. the simplified (FMP) scoring function and it is straightforward to obtain these configurations [9], we only need to evaluate a more accurate and arbitrarily complex scoring/re-ranking function on this small set. This is the general strategy we adopt. In this work we learn this re-ranking function and describe the components it includes. While the latter part of this paper presents a road map of how to put the arms and legs in correspondence (solving the left/right ambiguity) across the multiple images in order to allow a 3D reconstruction.

Our main contributions are: **1)** We introduce a new model which is an extension to [15]. It utilizes a global segmentation score, extra pairwise terms, and an exclusion principle to avoid double counting the score of overlapping parts. The overhead of our model over the FMP model is very small as our search space is a relatively small constant number. **2)** We present an effective parameter learning procedure based on the SVM-Rank formulation [7] to calibrate the factors included in our re-ranking function. **3)** We present a first attempt to automatically and accurately solve the 3D reconstruction from multiple view images in a non-studio environment. **4)** We present a new dataset of 771 images of football players taken from 3 views at 257 time instances, which will be publicly available on the author's website.

1.1 Related Work

By imposing a few assumptions on the pictorial structure model - independent appearance scores, quadratic deformation function - [4] developed an algorithm that finds the global optimum of the pictorial structure energy function in linear time complexity to the number of locations on the image. Using discriminatively trained parameters [5, 1] within this model produces very good results. There has been a few attempts on extending the model to handle inaccurate annotations using latent parameters [5, 8]. [10] tries to improve the pose priors by using a local kernel regression model. [11] proposes a cascade model for enabling the use of more sophisticated appearance models. [12] uses a more complicated graphical model to model extra dependencies between parts, and utilizes an approximate belief propagation algorithm to do the inference. Flexible Mixture of Parts (FMP) model [15] uses multiple linear models to represent the appearance of the object. We use the FMP model as the base of our work which has outperformed all the previous work by a significant margin. The paper [9] describes an efficient algorithm to approximately compute a set of high scoring configurations with almost no extra cost. Commonly automatic 3D pose reconstruction is performed by tracking with a 3D model [2] or applying a learnt regression function which maps an extracted image feature to a 3D pose [13]. However, due to the developments in 2D pose estimation it has allowed us to explore in this work the automatization of previously semi-manual based algorithms using 2D joints [14].

2 Components of a more accurate scoring function

Given the n-best configurations returned by the FMP model, the challenge is to re-score them in order to identify the ones which are closest to a correct configuration. The re-ranking function we learn is a linear combination of different features which indicate - weakly or strongly - the plausibility of a hypothesized configuration. In this section we describe the features and measurements which are extracted. These include a global segmentation score measuring how compatible a hypothesized configuration is with a segmentation of the image into foreground and background based on colour and a re-weighting of part appearance scores to impose an exclusion principle to avoid double counting the score of overlapping parts. First, though, we review the scoring function of the FMP model [15]. Many of its individual components are included in our re-ranking function but computed on a graph defining the dependency structure which includes loops.

2.1 Review of the flexible mixture of parts model

In the flexible mixture of parts (FMP) model [15] the object is divided into multiple parts, and each part is modelled by a set of templates. A graph structure, $G = (V, E)$, represents the dependencies used when fitting this model. V is the set of parts and E is the set of edges indicating which parts are linked. The coordinates of the centre of the ith part is denoted by p_i and $p = (p_1, \ldots, p_K)$ is the vector of all the part centres. Each part is also assigned a template t_i where each $t_i \in \{1, \ldots, T\}$ and let $t = (t_1, \ldots, t_K)$. The FMP model then scores a configuration p and its associated part types t with

$$S_{\mathrm{fmp}}(p, t) = S_{\mathrm{a}}(p, t) + S_{\mathrm{d}}(p, t) + S_{\mathrm{c}}(t). \quad (1)$$

which has three distinct components. $S_{\mathrm{a}}(p,t)$ is a weighted sum of appearance scores for each part

$$S_{\mathrm{a}}(p,t) = \sum_{i \in V} s_{\mathrm{a}}(p_i, t_i) = \sum_{i \in V} w_i^{t_i} \cdot \phi(I, p_i), \tag{2}$$

where $\phi(I, p_i)$ is a HOG descriptor of the image patch centred at p_i and $w_i^{t_i}$ is the template for ith part of type t_i. $S_{\mathrm{d}}(p,t)$ is the deformation score

$$S_{\mathrm{d}}(p,t) = \sum_{e \in E} s_{\mathrm{d}}(p_e, t_e) = \sum_{e=(i,j) \in E} w_{ij}^{t_i,t_j} \cdot \psi(p_i - p_j), \tag{3}$$

and is a sum of quadratic functions ($\psi(dx, dy) = [dx\ dx^2\ dy\ dy^2]$) modeling the deformation between connected parts. While $S_{\mathrm{c}}(t)$ is the score which consists of a prior for each part type and a compatibility score between the types of connected parts

$$S_{\mathrm{c}}(t) = \sum_{i \in V} s_{\mathrm{c}}(t_i) = b_{\mathrm{root}}^{t_{\mathrm{root}}} + \sum_{i \in V \setminus \mathrm{root}} \left(b_i^{t_i} + b_{i,\mathrm{parent}(i)}^{t_{i,\mathrm{parent}(i)}} \right). \tag{4}$$

Using the generalized distance transform and assuming G is a tree one can efficiently find the configuration, $(p_{\mathrm{fmp}}, t_{\mathrm{fmp}})$ which maximizes $S_{\mathrm{fmp}}(p,t)$ and the configurations corresponding to the n-best scores of $S_{\mathrm{fmp}}(p,t)$. The top scoring configuration frequently has a high PCP score and in general the head, torso and one leg are reliably detected. The problem of *double counting*, though, is prevalent. To help combat this issue, we include in our re-ranking function the same individual deformation scores, defined in equation (3), but augment these with deformation scores between pairs of left and right parts, see figure 4.

2.2 Modelling the correlation between parts

As we only focus on the n-best configurations returned by the FMP model we are at liberty to exploit more complicated and computationally expensive scoring of a configuration. Here we describe the scores we compute that are not facsimile of those in the FMP model. The first is a re-weighting of the individual appearance scores in equation (2) to prevent the double counting of evidence. The second is one based on performing crude segmentation. The crucial factor in both is that we allow ourselves to consider the global configuration p simultaneously as opposed to only considering pairs of parts at a time.

2.2.1 Enforcing an exclusion principle

Double counting occurs frequently in the football data, for instance when one of the legs is in motion and appears blurry while the other is stationary. In this situation the FMP or any pictorial structure model commonly double counts the strong evidence (usually the stationary limb) due to the independence assumptions they make. It is necessary to take the visibility of each part into account to allow for a more accurate modeling of the underlying situation and to implicitly enforce an exclusion principle. We employ probabilistic reasoning to do this modeling. Let sets $\mathcal{S}_{p,1}, \ldots, \mathcal{S}_{p,L}$ partition the set of K parts. Each $\mathcal{S}_{p,l}$ either contains the left and right versions of a part or just one single part for the parts associated with the head and torso. Let $p_{\mathcal{S}_{p,l}}$ denote the positions of the

parts in $\mathcal{S}_{p,l}$, similarly for $t_{\mathcal{S}_{p,l}}$ and $I_{\mathcal{S}_{p,l}}$ is the region of the image I which corresponds to where the parts in $\mathcal{S}_{p,l}$ occur. If the parts in $\mathcal{S}_{p,l}$ do not overlap then the likelihood of $I_{\mathcal{S}_{p,l}}$ is

$$p(I_{\mathcal{S}_{p,l}} \,|\, p_{\mathcal{S}_{p,l}}, t_{\mathcal{S}_{p,l}}) = \prod_{k \in \mathcal{S}_l} p(I_{p_k} \,|\, p_k, t_k) \tag{5}$$

However, if the parts in $\mathcal{S}_{p,l}$ overlap then the likelihood is calculated differently. As we do not know which part is the closest to the camera, we cycle through the different possibilities to get

$$p(I_{\mathcal{S}_{p,l}} \,|\, p_{\mathcal{S}_{p,l}}, t_{\mathcal{S}_{p,l}}) = \sum_{k \in \mathcal{S}_l} p(I_{p_k} \,|\, p_k, t_k)\, P(\text{part } p_k \text{ is the most visible part in } \mathcal{S}_{p,l}) \tag{6}$$

where for simplicity it is assumed that only one of the parts in $\mathcal{S}_{p,l}$ is visible at a time. If it is assumed that each $p(I_{p_k} \,|\, p_k, t_k) \propto \exp(s_{\mathrm{a}}(p_k, t_k))$ and each part in $\mathcal{S}_{p,l}$ is equally likely to be the one visible, then we can define scores which mimic $p(I_{\mathcal{S}_{p,l}} \,|\, p_{\mathcal{S}_{p,l}}, t_{\mathcal{S}_{p,l}})$:

$$s_{l,\mathrm{joint}}(p,t) = \begin{cases} \log\left(\frac{1}{|\mathcal{S}_l|} \sum_{k \in \mathcal{S}_l} \exp(s_{\mathrm{a}}(p_k, t_k))\right) & \text{if parts in } \mathcal{S}_l \text{ overlap} \\ \sum_{k \in \mathcal{S}_l} \; s_{\mathrm{a}}(p_k, t_k) & \text{otherwise} \end{cases} \tag{7}$$

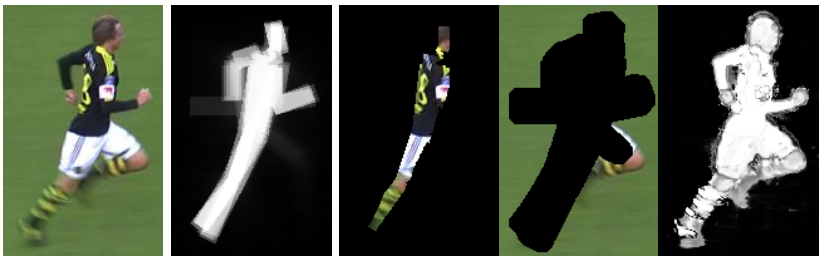
2.2.2 Segmentation score

A configuration p produces a segmentation of the image into background and foreground pixels. One can then measure the plausibility of configuration p by comparing this segmentation to one produced by another independent process. In our case this independent process segments based on comparing the colour of each pixel to learnt distributions of the colour for background and foreground pixels. We learn these foreground and background distributions for each test image with the following procedure. The high scoring configurations returned by the FMP model are used to create an initial estimate of the segmentation into foreground and background, see figure 2. This is done simply by averaging the foreground masks created from the boxes representing the parts in each configuration. The result is a rough estimate of the probability of a pixel belonging to the foreground. Thresholding these probabilities with separate criteria gives an under and over-segmentation. The foreground pixels from the under-segmentation are used to fit a GMM distribution for foreground pixels

$$p(c_x \,|\, l_x = f) = \sum_{i=1}^{M_f} \alpha_i^f \, \mathcal{N}(c_x \,|\, \mu_i^f, \Sigma_i^f) \tag{8}$$

where c_x is the RGB colour of a pixel at location x and l_x is the pixel's label as foreground or background. Similarly the background pixels from the over-segmentation are then used to fit a GMM distribution representing $p(c_x \,|\, l_x = b)$. Assuming a uniform prior probability, the posterior probability of pixel being foreground given its colour is

$$P(l_x = f \,|\, c_x) = \frac{p(c_x \,|\, l_x = f)}{p(c_x \,|\, l_x = f) + p(c_x \,|\, l_x = b)} \tag{9}$$



(a) input (b) initial mask (c) under seg. (d) over seg. (e) $p(l_x = f \mid c_x)$

Figure 2: To estimate the initial segmentation of an image (a) we use the top scoring configurations from the FMP model to get an initial estimate of the probability of a pixel belonging to the foreground (b). The results are then used to create under (c) and over (d) segmentation masks. A GMM is fit to both the colours of the foreground pixels and the background pixels. These distributions are then used to compute the posterior probability of each pixel being foreground (e).

We aggregate these individual posterior probabilities into a plausibility score of p based on its agreement with the segmentation

$$s_{\text{seg}}(p) = \frac{1}{N}\left(\sum_{x \in \mathcal{F}_p} P(l_x = f \,|\, c_x) + \sum_{x \in \mathcal{B}_p} P(l_x = b \,|\, c_x)\right) \tag{10}$$

where N is the total number of pixels, $\mathcal{F}_p$ is the set of pixels labeled as foreground according to p and similarly $\mathcal{B}_p$ is the background set.

3 Learning the parameters of the re-ranking function

In the previous section we introduced scores which indicate the plausibility of the person's hypothesized 2D pose. The next task is to combine these within one single function which can be used to re-rank the n-best configurations output by the FMP model. To this end we construct a feature vector $x_{p,t}$ for each (p, t) by concatenating the different components already described:

$$x_{p,t} = (\, s_{\text{seg}}(p), s_{1,\text{joint}}(p,t), \ldots, s_{L,\text{joint}}(p,t), s_{\text{d}}(p_{e_1}, t_{e_1}), \ldots, s_{\text{d}}(p_{e_l}, t_{e_l}), s_{\text{c}}(t_1), \ldots, s_{\text{c}}(t_K)\,)$$

where the edges $e_i \in E$ are now taken from a graphical model of the pairwise dependencies between parts with loops, see figure 4. We let the final scoring function take the form of a weighted sum of the individual components of $x_{p,t}$:

$$\text{score}(p, t) = w \cdot x_{p,t}$$

Our objective is to learn the linear weights w such that configurations closer to the ground truth are scored more highly. Closeness to the ground truth can be measured by

the PCP score [3]. This measure returns 1 if each part of the hypothesized configuration overlaps significantly with its corresponding part in the ground truth configuration. Our training data consists of N training images. For each training image I_k we calculate the n-best configurations returned by the FMP model. Each of these configurations generates a feature vector x_{ki} and let y_{ki} denote its PCP score. Let r_k be a subset of the pairwise constraints imposed by the ranking of x_{ki}'s based on y_{ik}:

$$r_k = \{(x_{ki}, x_{kj}) \ : \ y_{ki} > .9 \ \text{ and } \ y_{kj} \leq .9\}$$

Then we find the optimal w by minimizing the SVM-Rank[7] objective function:

$$\arg \min_{w,\, \xi_{ijk}} \frac{1}{2}\|w\|^2 + C \sum_{i,j,k} \xi_{ijk} \tag{11}$$

subject to for $k = 1, \ldots, N$

$$w \cdot x_{ki} \geq w \cdot x_{kj} + 1 - \xi_{ijk} \quad \forall\, (x_{ki}, x_{kj}) \in r_k \quad \textbf{and} \quad \xi_{ijk} \geq 0 \quad \forall\, (i, j, k) \tag{12}$$

Note the formulation is similar to that of the SVM, but that the set of constraints has been extended to enforce the correct ordering between all pair of configurations within each r_k. The main reason for using the SVM-rank model instead of a regular SVM is that the absolute value of our target function is not an accurate quantitative measure, but we assume the measure is accurate enough for comparing two configurations from the same image. To do the optimization we used the publicly available cutting-plane solver from [7].

4 3D Reconstruction

To estimate a player's 3D pose we must put his arms and legs in correspondence across the three views. This is because the current 2D pose model cannot distinguish between the real left and right limbs. There are 32 possible correspondences, ignoring mirrored configurations. We reconstruct the position of the skeletal joints in 3D for each of these combinations and the 2D joint locations highlighted by our re-ranking function. The correspondence which results in a plausible 3D pose - estimated 3D skeleton has limb lengths similar to those estimated during training - and gives the smallest re-projection error is then chosen. To do the reconstruction, first we compute an initial estimate of the 3D pose, $\tilde{X}$, and camera matrices, $\tilde{M}$, using the affine factorization algorithm. These quantities must then be rectified and therefore we seek an affine transformation, A, which transforms $\tilde{X}$ and $\tilde{M}$ to the true 3D locations and camera matrices. A is estimated by minimizing a cost function which softly enforces that each limb of the rectified 3D skeleton has the same length as observed in the training data. We use MATLAB's standard nonlinear optimization toolbox to perform this.

5 Results

We have annotated a total of 771 images of football players, which includes images taken from 3 views at 257 time instances. We used 180 of the images for training our model and the rest for testing. Figure 3 shows three annotated examples from our football dataset.

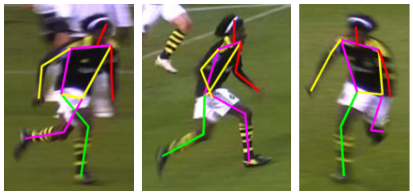


Figure 3: Three annotated examples from our football dataset which are taken at the same time instance.

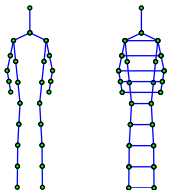


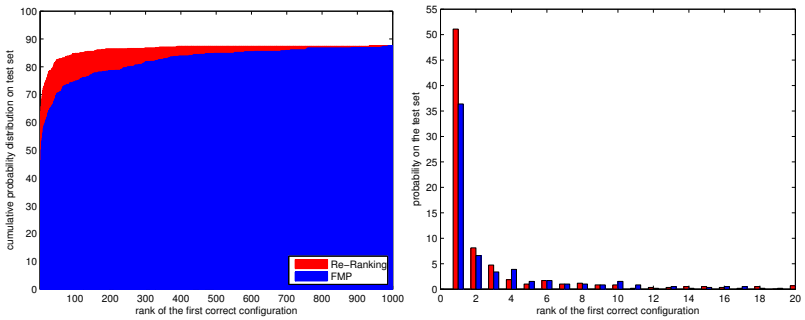
Figure 4: The pairwise dependencies in the FMP model (left), compared to the one used in the re-ranking function (right).

Ranking function	left/right flips **not** ignored	left/right flips ignored
Flexible Mixture of Parts	0.884	0.895
Re-ranking SVM-Rank	0.917	0.936
Oracle re-ranking	0.982	0.982

Table 1: Summary of the results on our football dataset with and without the re-ranking function. The first column of numbers displays the average PCP score of the top scoring configuration returned by the FMP model, our learnt re-ranking function and an oracle re-ranking function. The second column is the average PCP score when the left and right labels for the arms and legs are ignored.

Table 1 summarizes the results on our dataset with and without using the re-ranking function, as well as the results of picking the closest configuration to the ground truth between top 1000 configurations. In addition to the standard PCP score, we have provided the PCP scores ignoring the left/right limb assignments. This criteria is more accurate for our dataset since the limbs annotated as left/right on 2D images do not represent the real left/right limbs of the person. The results are improved by 3.3% with the PCP score criteria and 4.1% if we ignore the flipping. Figure 5a shows the cumulative probability distribution of rank of the true configuration across the top 1000 configurations given by the FMP model, in comparison with the results with our model. Figure 5b shows the same results on a finer scale. We can observe that the probability of the true configuration getting the top score based on FMP model is 36%, while this probability is increased to 51% using our model (an oracle ranking function in this case could improve the results up to 88%).

Figure 6 shows some qualitative results from our experiments on our football dataset. We observed that in many cases the double counting problem is fixed using our model (1-2nd rows). While in some cases the predicted flip is not compatible with the ground truth (2nd row) and this is the reason for the additional improvements if we ignore the flipping. The measurements in some cases are too noisy for our model, and we do not observe much of an improvement in these cases (3rd row). Finally, we have used the 2D estimates from



(a) Cumulative histogram (b) Histogram on finer scale

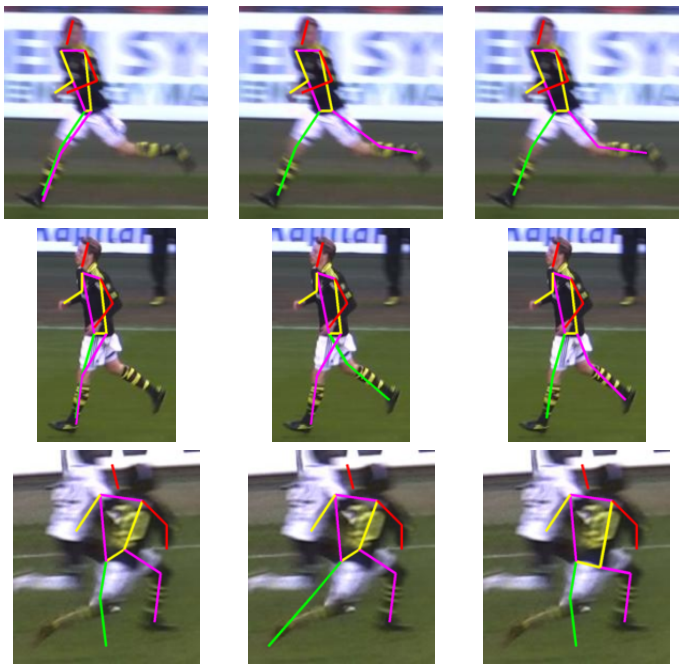
Figure 5: (a) The cumulative histogram of the rank of the first correctly predicted pose by the flexible mixture of parts model before (blue) and after reranking (red). (b) The same histogram on a finer scale.

our model to reconstruct the configuration of the player in 3D. With no assumptions about the pose of the player this is an extremely difficult task. However, when we have fairly good 2D estimates across all views we are able to get reasonable results. Figure 7 shows a stick figure of the 3D reconstruction of the top scoring 2D configurations, along with the back projected 2D estimates. An extended set of results is provided as supplementary materials.

5.1 Conclusions

We described a simple and efficient way of improving the performance of part based models by evaluating a more complicated scoring function to reorder a set of high scoring configurations. With good enough predictions of the location of a set of body joints across three images, we can obtain fairly accurate estimation of camera parameters and 3D joint positions. We believe by enforcing the temporal continuity constraints over sequences of images we can expect a boost in robustness and accuracy of our 3D predictions, which will be the subject for a future work. We would also like to exploit a multi-modal ranking function as opposed to a linear model which we have utilized in this work.

Acknowledgement: This work has been funded by the European Commission within the project FINE (Free Viewpoint Immersive Networked Experience).



(a) FMP (b) Reranking (c) Closest to G.T.

Figure 6: This figure shows (a) the result of FMP compared to (b) our reranking function, in addition to (c) the results of picking the closest configuration to the ground truth from a set top 1000 configurations. In many cases (row 1-2) we can solve the double counting problem, but sometimes (row 2) we have problem with the flipping ambiguity. In the last case the measurement is too noisy for our model and we are not able to improve the results.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009.

[2] Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *Internation Journal of Computer Vision*, 61(2):185–205, 2005.

[3] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proceedings of the British Machine Vision Conference*, 2009.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Internation Journal of Computer Vision*, 61(1):55–79, 2005.

Figure 7: The result of the 3D reconstruction of the body joints computed from the top scoring 2D configurations, along with the back projected 2D estimates.

[5] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[6] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[7] T. Joachims. Optimizing search engines using clickthrough data. In *Knowledge Discovery and Data Mining*. ACM, 2002.

[8] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[9] D. Park and D. Ramanan. N-best maximal decoders for part models. In *Proceedings of the International Conference on Computer Vision*, 2011.

[10] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2010.

[11] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proceedings of the European Conference on Computer Vision*, 2010.

[12] L. Sigal and M.J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006.

[13] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and P. Cipolla. Pose estimation and tracking using multivariate regression. *Pattern Recognition Letters*, 29(9): 1302–1310, 2008.

[14] P. Tresadern and I. Reid. Uncalibrated and unsynchronized human motion capture: A stereo factorization approach. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2004.

[15] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

Paper C

Multi-view Pose Estimation of Human Body

Vahid Kazemi, Hossein Azizpour, Magnus Burenius, and Josephine Sullivan

Submitted to
International Journal of Computer Vision

Multi-view Pose Estimation of Human Body

Vahid Kazemi, Hossein Azizpour, Magnus Burenius, and Josephine Sullivan

Abstract

This paper presents a complete framework for performing pose estimation from images captured with multiple calibrated cameras. This is done using a hybrid discriminative/generative approach. We use randomized decision trees to create a discriminative model that classifies each patch from the input images as a body part or background. The output of the classifier is aggregated over a discretized 3D voxel to produce consistent part likelihoods across multiple views. We then use a generative model based on pictorial structures to produce a single hypothesis for the configuration of body parts in 3D. We benchmark the performance of our method on a large dataset of football players and show that our method achieves state of the art results.

1 Introduction

In this paper[1] we address the problem of automatically estimating the 3D pose of a person from multiple calibrated cameras. While design decisions made in this work are based on our particular scenario, namely pose estimation of football players in a professional game, the presented framework can be customized to perform multi-view pose estimation for arbitrary applications.

Football footage have several key characteristics some of which are shared between different sports. Most notably the images are commonly disturbed by motion blur because of the fast moving players and cameras. There is also a large variation in the players' 3D pose. On the other hand the variation in the players' clothing is limited and background clutter is not as severe as in less structured environments. Yet, low quality images and fast motion make it hard to perform background subtraction reliably.

Currently, the most successful solutions to 2D pose estimation are discriminatively trained part-based models [19, 73, 2, 52, 31]. This class of methods are attractive as they enable efficient inference by reducing the conditional dependencies between parts, and demand less labeled training data as they can generate new poses at test time. Part-based models have also been used for 3D pose estimation

[1] This paper is an extended version of [32] presented at the British Machine Vision Conference.

[6, 60, 13, 53, 33], but to our knowledge good performance has only been reported in studio environments. This paper falls into the same category but we show that by using strong discriminative 3D part likelihoods our approach can be successfully applied to solve real world body pose estimation problems without imposing strong pose priors.

To discriminatively learn the 3D part likelihoods directly for the individual parts would require labeled 3D data and the associated calibrated views. We want to avoid this potentially expensive and non-trivial labeling task. Therefore, in this paper we discriminatively learn 2D part likelihoods for each part and aggregate the likelihoods from the different views to obtain the 3D part likelihoods. This means that we only need labelled 2D images from uncalibrated cameras. However, to get good performance this requires solving a part correspondence problem across views during the aggregation phase. We return to this issue later in the introduction, but now we turn to the issue of how to learn and compute the 2D part likelihoods.

State of the art 2D part based models for human pose estimation rely on SVM classifiers applied to a HOG descriptor of an image patch [73]. However we opt to use a more efficient random forest approach for estimating the part likelihoods. We take our inspiration from the recent success of the Kinect system. Shotton *et al* [55] use a random forest to estimate a person's 3D pose from a depth image. They divide the human body into a set of parts and a random forest is used to estimate the probability of each pixel belonging to each part. From these probabilities the 3D location of the skeletal joints are then independently estimated. Their work clearly demonstrates that given sufficiently diverse training data, one can learn a compact random forest classifier which at test time efficiently recognizes parts across a very varied set of 3D poses. In this paper we consider ordinary visual images, as opposed to depth images, but similarly use a random forest to assign to every pixel a probability of being a particular part or background. These probabilities form the basis for our part likelihood scores in 2D and 3D.

We create 3D part appearance likelihoods by aggregating the 2D likelihoods across all camera views. Care must then be taken regarding the correspondence of joints across the views. Because of the similar appearance of mirror symmetric parts, such as left and right arms and legs, and also the local nature of our part detectors, we can not directly distinguish the correct correspondences for each part. In this paper this issue is handled by introducing a latent variable into our model which represents the correspondence. At inference time we optimize for both the best pose and the best values of our latent variable. We show that this approach is both feasible and effective (fig. 2).

2 Background

Both 2D and 3D human pose estimation from visual images have for several decades been well-established research problems within the field of computer vision. An extensive body of work is devoted to both and several review articles give a thorough

overview of the area [24, 59, 38, 39]. What these review articles highlight [39] is that the strategies to tackle both problems have frequently reflected the prevailing algorithmic trends within computer vision. There has been a to-and-fro between exploiting generative [61, 16, 18] and discriminative [54, 63, 66] models and the compromise of generative models whose parameters are learnt discriminatively [73, 3], and also between modelling the human as a collection of pixels [34, 55], small [73] or mid-level parts [9, 72] or as one global entity [1]. There has, however, been the constant allure, especially in 3D, of the generative part based model because it is such a good fit to the human skeleton's articulated structure and it allows one to handle the variability of a human's appearance in an efficient way.

Generative part-based models Human pose estimation algorithms based on a generative part-based model have the following elements: a procedure to define the model's parts, an objective function to score a hypothesized pose, and an optimization strategy to find an optimum (local or global) of the objective function. The objective function has two components. One compares the anticipated appearances of each part, derived from an appearance model, to the image's actual appearance at the part's hypothesized image location. The other is derived from a prior detailing how parts should be arranged relative to one another. Each generative based algorithm is determined by how each of these elements and components are defined. Integral to all is the ability to effectively use the cues in the image.

In the early to mid noughties generative part based models dominated 3D pose estimation [25, 16, 61, 57, 4, 11, 60]. These algorithms assume a realistic 3D human model of the anatomical parts, calibrated camera(s) and an initial estimate of the 3D pose. Typically the objective functions, derived using Bayesian probabilistic modelling [57] and/or hand-crafted energy functions[61, 16], impose loops on the dependency structure between the model's parameters making it computationally infeasible to locate a global optimum. Instead the initial pose estimate is updated by finding a nearby local optimum through either iterative optimization [25, 11] or stochastic search [16, 23]. These approaches had success for tracking applications with multiple cameras [16]. However, regardless of the improvements made such as stronger motion priors [70, 40, 69], better appearance modelling [58] or more sophisticated optimization [61, 16], they were always susceptible to losing track and did not have the ability to re-initialize.

The challenge of initializing generative part models, in non-controlled environments, was taken up by those working in 2D [48, 18]. The most influential work is that of Felzenswalb and Huttenlocher [18] where they adapted the pictorial structure (PS) model [22] to 2D human pose estimation. In [18] they discretize the 2D pose space and then use dynamic programming and the generalized distance transform to find the global optimum of their objective function. Crucially, in constructing the objective function a tree structure is imposed on the dependencies between the parts' pose parameters and the observable image features. The clean pictorial structure formulation and the ability to find a global optimum sparked a

flurry of activity.

This activity has led to more sophisticated modeling of part appearances [47] (iteratively learning color histograms for each part), incorporating part detectors [2] (boosted part detectors applied to shape context descriptors of edge maps) and discriminative learning of the model parameters [17, 2, 3, 73], learning the best tree structure to model the dependencies between parts [37], synthetic augmentation of the labelled training data [45] and adding edge contour connectivity in the objective function [68].

Alongside pictorial structures have been 2D generative part approaches which have loops in the graph representing the dependencies between poses' parts [30, 67, 64] and the image data[21, 51, 50, 52, 6]. These more accurate models sacrifice the ability to find the globally optimal pose of their objective functions in a reasonable time and must instead rely on clever approximate inference such as pruning [21] and multi-scale inference[51].

The parts in a pictorial structure model normally correspond to the human's anatomical parts, but frequently it is not these parts that can be most reliably detected in images. Poselets were introduced by Bourdev [10, 9] and automatically identify and learn detectable parts from annotated training data. Each poselet is a detector, typically a SVM classifier in tandem with a HOG descriptor, of a part that is associated with a specific pose (*e.g.* arms-crossed) of a subset of the limbs (or partial limbs) seen from a specific viewpoint. It is possible to aggregate the responses of multiple poselet detectors to predict a person's 2D pose [72]. The poselet idea has also been incorporated into the pictorial structure model [44, 43, 37] to help bias the solution towards a valid 2D pose.

One inherent deficiency of the PS model is that background pixels are not modelled and image evidence is only included from the regions covered by parts. A set of methods combat this weakness by combining background and foreground segmentation with pose estimation[36, 21, 71, 49].

All these developments in global fitting of generative part based models in 2D have begun to see their way into 3D. In particular, with the increased RAMs and speed of modern computers it has become possible for the pictorial structure approach to be extended from 2D to 3D. Several research groups have pounced on this opportunity [13, 42, 41, 5, 53] and produced promising results. The adaptation from 2D to 3D results in an increase of the dimensionality of the search space, but there is the appealing advantage that geometric constraints on the human skeleton make more sense in 3D than 2D. It is also makes tackling the double counting problem easier because there is the hard constraint to exploit that limbs in 3D cannot physically intersect.

Discriminative based methods Alongside the developments in pictorial structures others explored the idea of using machine learning techniques to directly learn a mapping from 2D image features to either the 3D pose [54, 1, 7, 8] or the 2D pose [15, 26, 66, 74]. The computational burden of these methods are typically at

training time and the pay off is very fast execution at test time. These methods are, however, hampered by several issues. The first is that the mapping to the 3D pose is multi-valued and there is no generic regression function designed for this purpose. Next learning both the 2D and (especially) 3D regression functions can require an infeasibly large number of diverse labelled training examples to adequately cover the input space. Lastly, they are not robust to background and foreground clutter, especially the methods that rely on the extraction of clean silhouette data. Therefore it was considered that perhaps a purely learning based approach to either 2D or 3D pose estimation was somewhat of a dead-end.

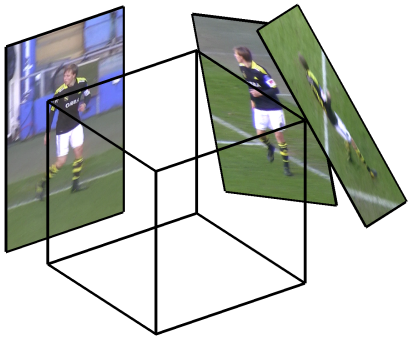
However, two recent developments have pushed purely discriminative based approaches back into the spotlight. The first is the remarkable Kinect system [55]. In this case the 3D pose estimation problem is more-or-less solved given RGB-D camera (and an environment where it is effective) and a random forest classifier [55, 62, 56]. An interesting facet of this proposed solution is that generative modelling was used to augment the labelled training data - pairs of a depth image and the corresponding 3D position of the body parts. Therefore the discriminative classification model can be considered to have efficiently encoded the elements of the generative model needed for joint prediction.

The second development is the emergence of deep learning and the ability of convolutional neural networks (CNN) to learn very powerful and generic image representations [35] for visual recognition tasks. Already the CNN structure has been exploited to learn a regression function that predicts the 2D coordinates of the skeletal joints from the pixel intensity data [66] within a bounding box. The community awaits to see how far deep learning can be pushed to solve the 2D pose estimation problem and see even if they can used to predict 3D pose from 2D image data.

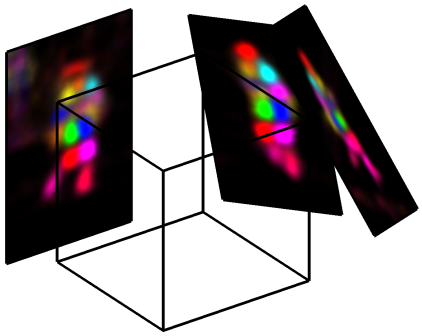
The work presented in this paper continues the line of research of marrying generative part based models with the discriminative learning of part appearances to tackle human pose estimation. Excitingly, it now appears that this approach can be applied with success to both 2D and 3D pose estimation in non-controlled environments without the need for initialization.

3 Method

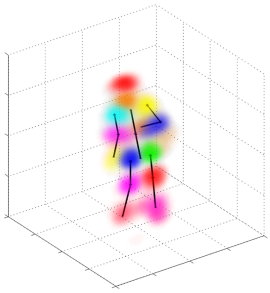
Given a set of calibrated cameras viewing a person, our goal is to estimate the location of body joints in 3D. Figure 1 shows a general overview of our framework. First a random forest is used to classify each pixel in each image as a part or the background, as described in section 3.1. We then discuss how the resulting 2D part appearance likelihoods can be used for 2D pose estimation in section 3.2. This process is performed so that we can compare 2D part detectors to previous work for 2D pose estimation. The results from section 3.2 are not used for performing 3D inference. For 3D part appearance likelihoods we back-project the result of the random forest pixel classification to a 3D volume, as described in section 3.3.



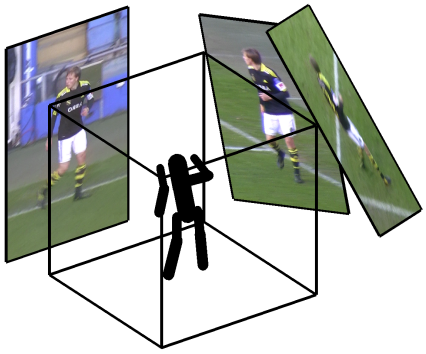
(a) Images are captured from three calibrated cameras



(b) Part scores are calculated with a single 2D discriminative model



(c) 2D part scores are aggregated over discrete 3D locations to generate consistent likelihoods across views



(d) Pose priors are used to infer a single 3D hypothesis

Figure 1: A general overview of our multi-view pose estimation framework. A 2D discriminative model is first used to classify pixels in each image as belonging to a part or the background. The results are then back-projected to a 3D volume. We find corresponding mirror symmetric parts across views by introducing a latent variable. Finally, a part-based model is used to estimate the 3D pose.

We then discuss how our 3D part appearance likelihoods can be plugged into any multi-view part-based model in section 3.4. The problem of mirror ambiguity for symmetric parts is addressed in section 3.5.

3.1 Appearance likelihoods in 2D using random forests

We use a random forest of classification trees to estimate the probability that a pixel $\mathbf{v}$ belongs to a skeletal joint or the background class. The split decisions made in

each tree are based on thresholding a dimension of the HOG descriptor [20] of the image window. This dimension is defined by three numbers as follows. First there is a 2D offset vector $\mathbf{u}$. The point $\mathbf{u}+\mathbf{v}$ then falls within a certain cell of the HOG descriptor and a record is kept of this cell. The final number defines the dimension of $(\mathbf{u}+\mathbf{v})$'s cell descriptor to be accessed. It is this entry which is thresholded in the split decision. The offsets, $\mathbf{u}$, considered are constrained to be within a certain distance of $\mathbf{v}$.

We have training images that have the position of the 2D skeleton joints labelled. From these labelled images we generate a new labelled dataset $\{(\mathbf{h}_k, \mathbf{v}_k, y_k)\}_{k=1}^{K}$ where $\mathbf{h}_k$ is the HOG descriptor of an image and $\mathbf{v}_k$ is a pixel with class label $y_k \in \{0, 1, \ldots, N\}$. The label 0 corresponds to the background class and the other numbers to the skeletal joints. This is the labelled data we use to train the random forest. Algorithm 1 summarizes the procedure for training each decision tree in the random forest.

When we apply a learnt decision tree to a test image i and a pixel location $\mathbf{v}$ in an image with HOG descriptor $\mathbf{h}$ we will reach a leaf node m. The posterior probability of pixel $\mathbf{v}$ having label y is equal to the proportion of the training samples that reach node m and have label y. The output of our random forest is the average of the probabilities returned by the trees in the forest. The final response image for each part n is denoted by $f_n(i, \mathbf{v})$.

3.2 Inferring the 2D pose

We first formulate the pose estimation problem in 2D. This is done so we can introduce our notation for part-based models and can compare the random forest results to previous work for 2D pose estimation. However, the results from this sub-section are not used when performing the multi-view 3D inference.

Let $\mathbf{V}_n$ be a random variable representing the 2D position of joint n. The 2D pose of the person is then $\mathbf{V} = (\mathbf{V}_1, \ldots, \mathbf{V}_N)$. Let I be a random variable representing the image evidence. We consider part-based models that assume there is some image evidence for each joint I_n and that these are conditionally independent given the position of the joints

$$P(i \mid \mathbf{v}) = \prod_n P(i_n \mid \mathbf{v}_n) \tag{6}$$

where lower cases are used for outcomes of the random variables. We use the response from the random forest as the 2D joint appearance likelihoods

$$P(i_n \mid \mathbf{v}_n) \propto f_n(i, \mathbf{v}_n) \tag{7}$$

We infer the pose by finding the most probable state of $\mathbf{V}$ given the measurement data

$$\max_{\mathbf{v}} P(\mathbf{v} \mid i) = \max_{\mathbf{v}} \left[\ln P(\mathbf{v}) + \sum_n \ln P(i_n \mid \mathbf{v}_n) \right] \tag{8}$$

where $P(\mathbf{v})$ describes an arbitrary 2D pose prior. This optimization can be solved in different ways, depending on the form of the 2D pose prior $P(\mathbf{v})$. In our implementation we first find the modes of the joint appearance likelihoods $P(i_n \mid \mathbf{v}_n)$. To make the process more efficient we first sample pixels with high probabilities to find a small set of modes. In practice we use the meanshift algorithm for this. In many cases, taking the mode with the highest probability for each joint independently leads to a valid configuration (This corresponds to the pose prior $P(\mathbf{v}) = \prod_n P(\mathbf{v}_n)$ where each $P(\mathbf{v}_n)$ is uniform for all n). This is because the random forest is able to aggregate information from a relatively large neighbourhood around each joint and produce confident joint hypotheses. There are, however, some cases where this approach fails. To find the estimated joints which have both a spatial configuration consistent with a valid 2D pose and high appearance scores, we search for the optimal combination of body joints from a small set of highly probable modes. This is done efficiently by using dynamic programming to minimize a cost function which is factorized over a tree.

To define $P(\mathbf{v})$ we follow the pose prior proposed by [73]. Similar to [73] we use a mixture model to model the distribution of the joint offsets. If we let the vector $\mathbf{t} = (t_1, \ldots, t_N)$ represent the mixture component chosen for each joint then

$$\ln P(\mathbf{v}) = \max_{\mathbf{t}} \; S_{\text{def}}(\mathbf{v}, \mathbf{t}) + S_{\text{co}}(\mathbf{t}) + \text{const} \tag{9}$$

We model the offset of each joint $\mathbf{v}_n$ from its parent $pa(\mathbf{v}_n)$ with a mixture of Gaussians. The deformation score S_{def} is defined over pairs of joints as follows:

$$S_{\text{def}}(\mathbf{v}, \mathbf{t}) = \sum_{n \neq \text{root}} \ln \left(\mathcal{N}(\mathbf{v}_n - pa(\mathbf{v}_n);\; \mu_{n,t_n}, \Sigma_{n,t_n}) \right)$$

where μ_{n,t_n} and Σ_{n,t_n} are the mean and variance of joint offset n for component t_n. S_{co} scores each configuration based on the co-occurrence of pairs of mixture components:

$$S_{\text{co}}(\mathbf{t}) = \sum_n b_n^{t_n} + \sum_{n \neq \text{root}} b_n^{t_n, t_{pa(n)}} \tag{10}$$

where $b_n^{t_n}$ is the probability of the occurrence of component t_n and $b_n^{t_n, t_{pa(n)}}$ is the probability of the co-occurrence of components t_n and $t_{pa(n)}$. The parameters of this prior are estimated from the statistics of the training data annotations.

3.3 Appearance likelihoods in 3D

Let the 3D position of joint n be the random variable $\mathbf{X}_n$ and the 3D pose $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$. The image evidence from view c for joint n is represented by the random variable $I_{c,n}$ and the evidence of all joints for a single view is $I_c = (I_{c,1}, \ldots, I_{c,N})$. Let $\mathbf{V}_{c,n}$ be the 2D position of joint n in view c. Let T_c be the projective transformation of camera c. We assume the 2D position $\mathbf{v}_{c,n}$ of joint n

in view c is deterministically calculated as $\mathbf{v}_{c,n} = T_c(\mathbf{x}_n)$. The part appearance likelihood for view c is computed by projecting $\mathbf{X}_n$ to that view

$$P(i_{c,n} \mid \mathbf{x}_n) = P(i_{c,n} \mid T_c(\mathbf{x}_n)) \propto f_n(i_c, T_c(\mathbf{x}_n)) \tag{11}$$

We assume the image evidence across views is conditionally independent given $\mathbf{x}_n$ and thus compute the multi-view 3D appearance likelihood (see figures 1 and 2) as

$$P(i_{1,n}, \ldots, i_{C,n} \mid \mathbf{x}_n) = \prod_{c=1}^{C} P(i_{c,n} \mid \mathbf{x}_n) \tag{12}$$

3.4 Inferring the 3D pose

Similar to 2D we estimate the pose by computing the most probable state of $\mathbf{X}$ given the measurement data. This equates to finding the maximum of the posterior distribution

$$\begin{aligned} &\max_{\mathbf{x}} P(\mathbf{x} \mid i_1, \ldots, i_C) = \\ &\max_{\mathbf{x}} \left[\ln P(\mathbf{x}) + \sum_n \sum_c \ln P(i_{c,n} \mid T_c(\mathbf{x}_n)) \right] \end{aligned} \tag{13}$$

where $P(\mathbf{x})$ describes an arbitrary 3D pose prior. This optimization can be solved in different ways, depending on the choice of the state space for $\mathbf{X}$ and the form of the 3D pose prior $P(\mathbf{x})$. If the pose prior can be factored according to a tree graph and every X_n is considered as a discrete random variable, then it is feasible to find a global optimum using dynamic programming [6, 13, 53].

Our 3D appearance likelihoods can be used by any multi-view part-based model. To demonstrate the performance of a full system we follow the approach of [13] and discretize the state space. We assume the person is within a bounding cube and create a uniform grid covering this cube. The appearance likelihoods are then evaluated for all grid points. We consider two different pose priors $P(\mathbf{x})$. The first is $P(\mathbf{x}) = \prod_n P(\mathbf{x}_n)$ with $P(\mathbf{x}_n)$ uniform over its state space. Then the global optimum can be found by optimizing equation (12) for each joint independently. The second pose prior imposes limb length constraints as in [13]. We define $P(\mathbf{x})$ as follows:

$$P(\mathbf{x}) = \prod_n \mathcal{N}(\|\mathbf{x}_n - pa(\mathbf{x}_n)\|;\ \mu_n, \sigma_n) \tag{14}$$

where μ_n and σ_n here are the mean and variance of limb lengths calculated from the training data annotations. This is a rather simplistic prior, but we found it to be adequate for our purposes. One can use mixture of Gaussians to model the distribution of joint offsets the same way as described in section 3.2 but in our experiments we only used the simple limb length prior.

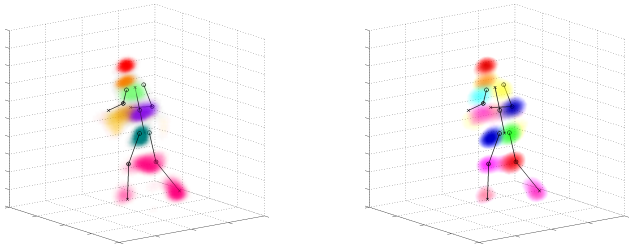


Figure 2: Overcoming ambiguities introduced by symmetric appearances. The left image shows the 3D appearance likelihoods computed from part detectors that ignore the left and right label of the parts. The right image shows the result of finding corresponding parts across views by maximizing a latent variable. The ground truth pose is shown in black.

3.5 Overcoming ambiguities introduced by symmetric appearances

In equation (11) we have assumed that the mapping between the labels for the 2D joints and the 3D joint labels is consistent across views and that it is one-to-one. However, this is not necessarily the case especially for the mirror symmetric joints, i.e. joints associated with the right and left legs (arms). For such joints, the classifier can either be trained to

- just detect the joints and ignore their label as left or right or
- recognize the left and right label *of the image*

In the latter scenario we do not know if the joints labelled as left in two views correspond to the same physical 3D joints. Therefore to match the left and right legs *of an image* with the left and right *of the person* we have two choices. If we also try to match the arms we have a total of $2^2 = 4$ choices per image. Considering all C views gives a total of 2^{2C} choices.

To handle this mirror ambiguity we introduce a discrete latent random variable $\mathbf{M}_c = (M_{c,1}, \ldots, M_{c,N})$ which represents the mapping of the labels from the 3D joint labels to the 2D joint labels in view c. We assume M_c is uniformly distributed over its 4 states. For non-limb joints the mapping is considered unambiguous. Instead of using (11) we thus let the image evidence of each joint depend on $M_{c,n}$ as follows

$$\begin{aligned} &P(i_{c,n} \mid m_{c,n}, \mathbf{x}_n) = \\ &\quad P(i_{c,n} \mid m_{c,n}, T_c(\mathbf{x}_n)) \propto f_{m_{c,n}}(i_c, T_c(\mathbf{x}_n)) \end{aligned} \tag{15}$$

Then the optimum of the full posterior distribution for $\mathbf{X}$ and $\mathbf{M} = (\mathbf{M}_1, \ldots, \mathbf{M}_C)$ assuming a uniform prior over $\mathbf{M}$ is given by

$$\max_{\mathbf{x},\mathbf{m}} P(\mathbf{x}, \mathbf{m} \mid i_1, \ldots, i_C) = \max_{\mathbf{m}} \max_{\mathbf{x}} \left[\ln P(\mathbf{x}) + \sum_n \sum_c \ln P(i_{c,n} \mid \mathbf{m}_{c,n}, T_c(\mathbf{x}_n)) \right] \tag{16}$$

and this becomes the optimization problem we solve at inference time as opposed to (13). See figure 2. We perform the outer optimization over $\mathbf{m}$ by exhaustive search, independently of the method used for the inner optimization over $\mathbf{x}$. This approach can therefore be applied to any part-based model. When we solve this optimization problem the joints across the views will be in correspondence, but there may still be an unresolved front/back ambiguity in 3D.

4 Experiments

To benchmark the performance of our approach in a realistic outdoor scenario we have created the publicly available KTH Multiview Football Dataset from a professional football game. The dataset consists of images of two different players of the same team. The dataset is divided in two sets.

For the first set, we annotated the 2D pose of the players for 5907 images. We used the first 3900 images to train the random forest and the rest for testing the 2D pose estimation performance. The results presented in section 4.1 are based on this data.

The second set contains an annotated video sequence consisted of 214×3 images where the player was captured by three moving cameras. We used the 2D annotation to synchronize and calibrate the cameras and the human pose is reconstructed in 3D using the affine factorization algorithm [12, 65, 46, 28]. We used the 3D reconstruction of the this sequence as the ground truth for testing the 3D pose estimation performance. These results are in presented in section 4.2.

Finally, in section 4.3 we analyze our dataset in terms of the difficulty of pose estimation. Then, using different similarity based transformations we give a detailed indication of variations of footballers between the training set and the evaluation set.

4.1 Scoring and inference in 2D

What follows contains an analysis of the effect of different parameters on the performance of the random forest, as well as a comparison with the state of the art *Flexible Mixture of Parts (FMP)* model [73] trained and tested on our football dataset.

Number of trees: It is well known that decision trees are prone to overfitting and combining multiple trees can significantly help in regularizing their outcome

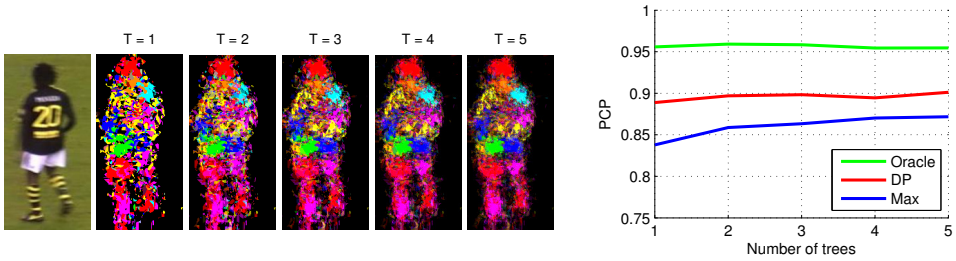


Figure 3: The effect of the number of trees on performance of random forest. PCP score is used for quantitative evaluation of the end results with three different matching methods. These are taking the modes with maximum probability (blue curve), using dynamic programming with a simple shape prior (red curve), and an oracle matching method on the highest (5-10) probability modes (green curve). The increase in performance is minimal with the addition of more trees to the forest after the first two.

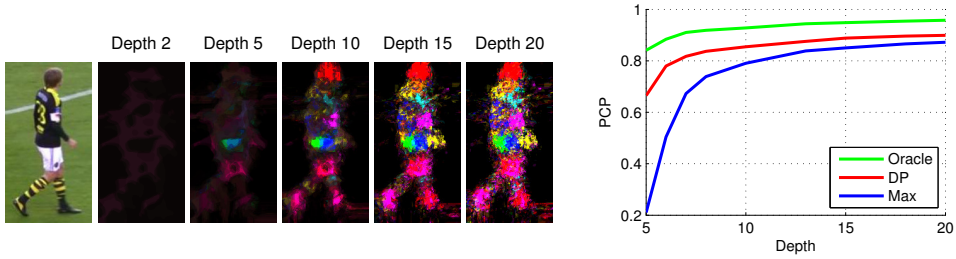


Figure 4: Qualitative and quantitative results show how the depth of the tree affects the output of the random forest. See figure 3 for explanation of quantitative results.

[29]. However, we observe that in our case the improvement with more than two trees is not drastic, see figures 3 (a). In our experiments we fixed the number of trees to 5.

Depth of trees: Figures 4 (b) show how the depth of the trees affects the performance of random forest. It can be observed that with a random forest of depth 10, we can already correctly classify pixels belonging to easy to detect parts like head, hips, and knees. The depth of each tree was set to 20 in our experiments. It is worth mentioning that the resulting decision trees are not balanced. The decision trees trained on our dataset have around 10% of the nodes of a balanced tree with equal depth.

Feature pool: Decisions at each node are made by thresholding HOG [20, 14] dimensions in a neighbourhood of each pixel. To increase randomization, at each node a pool of features is created by selecting a random subset of all the available

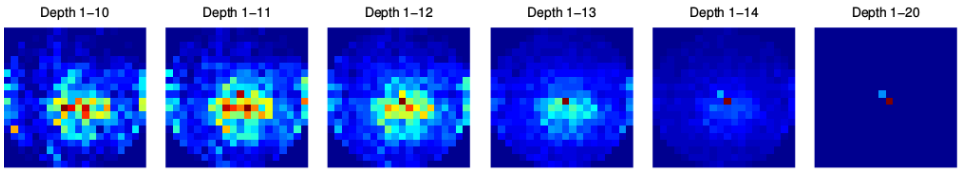


Figure 5: The 2D histogram of the offsets selected at the decision tree splits at different depth levels. The initial splits use information from wider neighbourhoods.

features. The optimal feature and threshold are then chosen from this pool. We set the feature pool size to 25000. Figure 5(c) shows the distribution of the offsets chosen at different depths levels of the random forest. The results show that at the earlier levels of the tree a wide exploration of the surrounding area is performed, but as we move down to the bottom of the tree most of the selected features are centered at the probe pixel. In our experiments we allow for offsets up to 50 pixels. The height of the person is about 180 pixels.

Comparison to state-of-the-art in 2D pose estimation: We compare our results to *Flexible Mixture of Parts(FMP) [73]* which achieves state of the art performance on general 2D human pose estimation tasks. We have trained and tested their method using the original code provided by authors on our football dataset.

Table 1 shows a summary of results on our football dataset. The results show that our 2D part detector outperforms FMP [73] on this dataset. It is also worth mentioning that our 2D part detector is at least an order of magnitude faster than FMP. This is due the fact we only need to evaluate a few (5) decision trees to classify each pixel while FMP requires convolving 138 filters. Figure 6 shows some qualitative results comparing to the current state of the art. The major problems for our method seem to be caused by unseen poses, lack of strong features for parts such as lower arms, and the absence of contextual support, e.g. for outstretched limbs. The latter can be potentially solved by using higher offsets (as described in section 3.1).

We also tried our random forest on some standard datasets, which were smaller than our football dataset and had more background clutter. Under those conditions FMP still outperforms our random forest. We believe that the difficulty to deal with severe background clutter is a disadvantage of the current version of our part detectors. However, a recent work [15] shows state of the arts performance within a very similar random forest framework. Although, this approach still seems to require considerably more training data than FMP.

4.2 Scoring and inference in 3D

To perform 3D pose estimation we follow the approach of [13] and discretize the search space. We assume that the person is within a bounding cube (fig. 1) and

Figure 6: A qualitative comparison of random forests with a state of the art pose estimation method on our dataset. The top row shows the modes of probabilities output from the random forest. A point's size indicates its certainty level. The second row is the result of inferring the configuration by imposing 2D pose priors. The last row is the result of FMP [73].

	Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	Average
FMP [73]	**.97**	**.99**	.92	.66	.94	.80	.86
RF	.94	.96	.90	.69	.94	.84	.87
RF + Pose Prior	.96	.98	**.93**	**.71**	**.97**	**.88**	**.89**
RF + Oracle Matching	.97	.99	.94	.82	.98	.97	.94

Table 1: A comparison of PCP scores of different baselines on our football dataset. The rows represent the results of the following methods. (1) FMP [73] trained and tested on our dataset. (2) Taking the optimal modes for each joint independently. (3) Taking the modes that maximise a shape prior. (4) Taking the optimal modes wrt the ground truth. For the last two baselines the matching is performed only on a few of the most probable modes (5-10).

create a $64 \times 64 \times 64$ grid covering this cube. We compute our 3D part appearance likelihoods for all grid points. We perform inference with and without the pose prior discussed in section 3.4. The former imposes limb length and intersection constraints. We also perform inference with and without the latent variable handling the mirror ambiguity as discussed in section 3.5.

The results are summarized in table 2. The performance is measured using 3D PCP scores with $\alpha = 0.5$ [13]. The table shows that introducing the latent variable to deal with the mirror ambiguity significantly improves the final results. On this dataset it is surprisingly much more important than the pose prior.

Figure 7 shows our estimated 3D poses (red) compared to the ground truth (blue), for six different frames. For this figure the inference was performed using the latent mirror variable but without any pose prior (uniform). The figure shows that our 3D appearance likelihoods accurately detect most of the body parts, even without imposing any pose prior. If we add the limb length and intersection constraints we are able to correct for some of the limited double counting that occurs for the lower legs, which is reflected by numbers in table 2.

	Upper Arms	Lower Arms	Upper Legs	Lower Legs	Average
RF	.02	.03	.86	.57	.37
RF + Pose Prior	.16	.07	.91	.87	.50
RF + Mirror Latency	.87	.68	1.00	.96	.88
RF + Mirror Latency + Pose Prior	**.89**	**.68**	**1.00**	**.99**	**.89**
Burenius et al. [13]	.60	.35	1.00	.90	.71
Belagiannis et al. [5]	.68	.56	.78	.70	.68

Table 2: An evaluation of our 3D pose estimation results in terms of PCP scores. The rows represent the results of the following methods. (1) Taking the maximum probability estimates for each part independently over the 3D grid. (2) Taking the pose priors into account. (3) Handling mirror ambiguity without pose priors and (4) with pose priors.

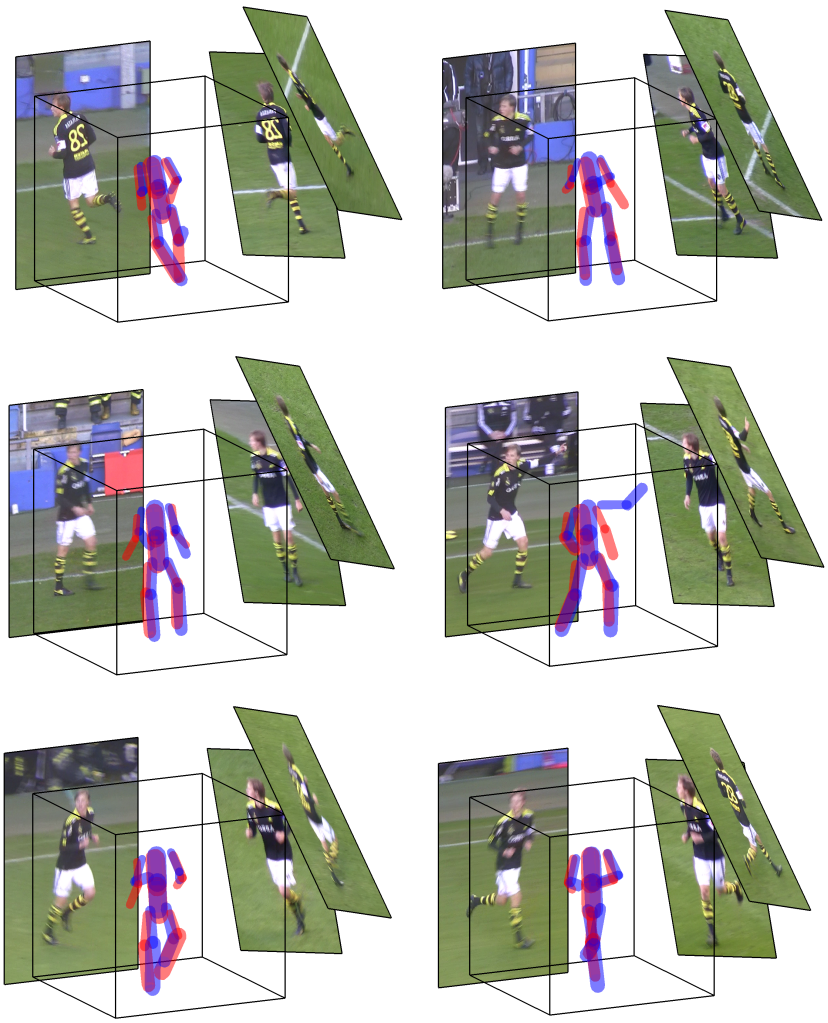


Figure 7: Final 3D poses obtained by taking, for each part independently, its most probable state over the grid. The mirror ambiguity is solved jointly. Estimation is red and ground truth is blue.

4.3 Dataset Analysis

In an attempt to quantify the difficulty of the poses in our dataset (Football5907) we have plotted the frequency of test samples based on their difficulty (Figure 8). Our measure of difficulty is defined as the sum of the residuals of the joint locations of a test example from its closest nearest neighbour in the training set. Observe that more than half the training data have average difficulty of around 100 pixels, where a person's height is around 200 pixels.

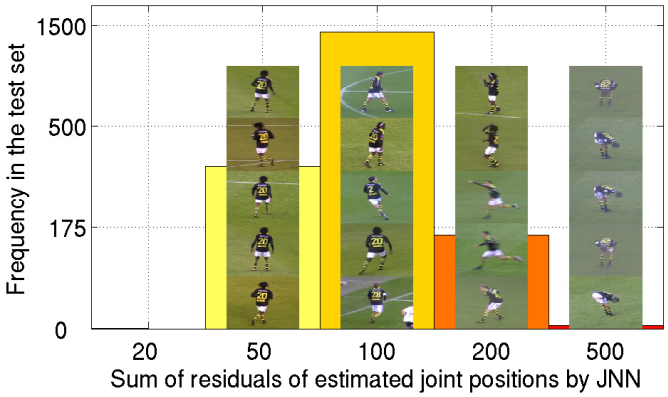


Figure 8: The frequency of the test samples according to their difficulty. The difficulty of a test image is measured by the sum of residuals from its best matched training sample (using the annotations). Some random samples are shown for each interval. The x, y axes are exponentially labelled.

Furthermore, we estimate 2D pose estimation results by finding the nearest neighbors of the test images in the training set using different similarity measures and report the PCP in Table 3, different similarity measures used are as follows.
Appearance Based Nearest Neighbour (ANN): A naïve appearance similarity measure based on the HOG feature will be sensitive to noise and clutter. Thus, a feature selection technique is adopted to highlight its discriminative dimensions. We extract a HOG descriptor ϕ_p for each positive training patch p and use Linear Discriminant Analysis (LDA) to train a linear filter w_p. This means

$$w_p = \Sigma^{-1}(\phi_p - \mu_N) \tag{17}$$

where μ_N is the mean of descriptors extracted from the background patches and Σ is a full rank covariance matrix computed from all the positive and negative patch descriptors. The hyperplane w_p now encodes the dimensions of ϕ_p which best separate it from the negative class. As analyzed in [27] w_p can also be seen as a whitened version of the original descriptor. A new representation d_p is developed for the positive patch p by the following equation which its vectorized version is

	JNN+T	JNN+TS	JNN+TSR	ANN	FMP	**Ours**	Oracle
PCP	.94	.95	.96	.80	.86	**.89**	.94

Table 3: Final PCP results for the baselines and different versions of our method on the [33] football dataset.

further normalized to unit length.

$$d_p = \Sigma^{-1/2}(\phi_p - \mu_N) \tag{18}$$

The similarity between every pair of patches is computed by applying the histogram intersection kernel,

$$K(p_1, p_2) = \sum_{i=1}^{D} \min(d_i^{p_1}, d_i^{p_2}) \tag{19}$$

where d_i^p is the ith element of the new representation d for sample p and D is its dimensionality. The covariance matrix Σ is estimated using the approach of [27]. To ensure better alignments we search over slightly offset patches for each example.
Joint Nearest Neighbour(JNN): Here, by using each *test sample landmark annotation* we search the training set to find the closest match in terms of joint positions. This is done by optimizing a similarity matrix for each of the training samples which transforms the test image joint annotations to the corresponding joints in the training sample. The least-square optimization is solved by Procrustes analysis closed form solution. For each test example, we look for an optimal translation (JNN+T), and scale (JNN+TS), and rotation (JNN+TSR) matrix which transforms an example to the closest sample in training data. Due to the fact that these methods use ground truth annotation of *test samples* they act as an upper-bound for a perfect global nearest neighbour method and gives an indication of the difficulty of the dataset.

5 Conclusion

We presented a complete framework for performing 3D pose estimation from multiple calibrated cameras. We utilized an efficient discriminative model based on randomized decision trees with a pictorial structure based generative model to produce consistent poses in 3D. Our algorithm achieves state-of-the-art performance on our large football dataset. Yet, dealing with small datasets with severe background clutter can be challenging for our method. When combining the 2D part detectors over multiple views for 3D part detection, the similar appearance of mirror symmetric body parts is a problem. We have highlighted this and presented a simple solution based on a latent variable formulation.

Two major parts of the presented framework can be altered to match the needs of the application.

Firstly, we chose to use random forests for training 2D part detectors. Random forests are both effective when trained on large image datasets and are also very efficient. Yet, we have observed that they are not robust to severe background clutter. In our particular football scenario this is not a limitation, since the background is mostly grass and the classifier easily distinguishes the players from the background. For other scenarios one might to either consider performing background subtraction (this can be easily done for static cameras) or use more robust features. We are particularly interested to see how deep neural network based representations will perform if used as part detectors in our framework.

The choice of the generative model is also another part of the presented framework which can be easily altered. In this work we used a simple pictorial structure based model based on limb-lengths. This model is too flexible and can produce invalid configurations. Restricting the model by applying kinematic constraints to ensure feasibility of the solution can be an interesting direction for extending this work.

Finally we presented a large multi-view football dataset. We made the dataset publicly available and we hope that it will stimulate more research of 3D pose estimation in realistic outdoor environments.

References

[1] A. Agarwal and B. Triggs. Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):44–58, 2006.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009.

[3] Mykhaylo Andriluka, Stefan Roth, and Bernt Schiele. Discriminative appearance models for pictorial structures. *Internation Journal of Computer Vision*, 99(3):259–280, 2012.

[4] A. Balan and M. Black. An adaptive appearance model approach for model-based articulated object tracking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006.

[5] V. Belagiannis, S. Amin, M. Andriluka, B. Schiele, N. Navab, and S. Ilic. 3d pictorial structures: From single to multiple human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2014.

[6] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr. A study of parts-based object class detection using complete graphs. *Internation Journal of Computer Vision*, 87 (1):93–117, 2010.

[7] A. Bissacco, M. H. Yang, and S. Soatto. Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2007.

[8] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *Internation Journal of Computer Vision*, 87(1-2):28–52, 2010.

[9] Lubomir Bourdev, Subhransu Maji, Thomas Brox, and Jitendra Malik. Detecting people using mutually consistent poselet activations. In *Proceedings of the European Conference on Computer Vision*, 2010.

[10] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *Proceedings of the International Conference on Computer Vision*, 2009.

[11] Christoph Bregler, Jitendra Malik, and Katherine Pullen. Twist based acquisition and tracking of animal and human kinematics. *Internation Journal of Computer Vision*, 56(3):179–194, 2004.

[12] M. Burenius, J. Sullivan, and S. Carlsson. Motion capture from dynamic orthographic cameras. In *4DMOD - ICCV Workshop*, 2011.

[13] M. Burenius, J. Sullivan, and S. Carlsson. 3d pictorial structures for multiple view articulated pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2005.

[15] M. Dantone, J. Gall, C. Leistner, and L. Van Gool. Human pose estimation using body parts dependent joint regressors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[16] Jonathan Deutscher and Ian Reid. Articulated body motion capture by stochastic search. *Internation Journal of Computer Vision*, 61(2):185–205, 2005.

[17] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proceedings of the British Machine Vision Conference*, 2009.

[18] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Internation Journal of Computer Vision*, 61(1):55–79, 2005.

[19] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[20] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), Sept. 2010.

[21] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[22] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[23] J. Gall, B. Rosenhahn, T. Brox, and H.-P. Seidel. Optimization and filtering for human motion capture. *International Journal of Computer Vision*, 87(1-2):75–92, 2010.

[24] D. M. Gavrila. The visual analysis of human movement: a survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.

[25] D. M. Gavrila and L. Davis. 3-d model-based tracking of humans in action: a multi-view approach. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 1996.

[26] K. Hara and R. Chellappa. Computationally efficient regression on a dependency graph for human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[27] Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. In *Proceedings of the European Conference on Computer Vision*, pages 459–472, 2012.

[28] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[29] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. New York: Springer-Verlag, 2001.

[30] H. Jiang and D. R. Martin. Globel pose estimation using non-tree models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[31] S. Johnson and M. Everingham. Learning effective human pose estimation from inaccurate annotation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[32] Vahid Kazemi, Magnus Burenius, Hossein Azizpour, and Josephine Sullivan. Multi-view body part recognition with random forests. In *Proceedings of the British Machine Vision Conference*, 2013.

[33] Vahid Kazemi and Josephine Sullivan. Using richer models for articulated pose estimation of footballers. In *Proceedings of the British Machine Vision Conference*, pages 6.1–6.10, 2012.

[34] P. Kohli, J. Rihan, M. Bray, and P. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *Internation Journal of Computer Vision*, 79(3):285–598, 2008.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[36] L. Ladicky, A. Zisserman, and P.H.S. Torr. Human pose estimation using a joint pixel-wise and part-wise formulation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[37] Y. Li and F. Wang. Beyond physical connections: Tree models in human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[38] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 103(2-3):90–126, 2006.

[39] T.B. Moeslund, A. Hilton, V. Krüger, and L. Sigal. *Visual Analysis of Humans: Looking at People*. Springer, 2011. ISBN 9780857299963.

[40] K. Moon and V. Pavlovic. Impact of dynamics on subspace embedding and tracking of sequences. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006.

[41] Bojan Pepik, Peter Gehler, Michael Stark, and Bernt Schiele. 3d2pm - 3d deformable part models. In *Proceedings of the European Conference on Computer Vision*, 2012.

[42] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3d geometry to deformable part models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2012.

[43] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[44] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Strong appearance and expressive spatial models for human pose estimation. In *Proceedings of the International Conference on Computer Vision*, 2013.

[45] L. Pishchulin, A. Jain, M. Andriluka, T. Thormählen, and B. Schiele. Articulated people detection and pose estimation: Reshaping the future. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2012.

[46] Long Quan. Self-calibration of an affine camera from multiple views. *Internation Journal of Computer Vision*, 19:93–105, July 1996.

[47] D. Ramanan. Learning to parse images of articulated bodies. In *Advances in Neural Information Processing Systems*, 2006.

[48] D. Ramanan and D.A. Forsyth. Finding and tracking people from the bottom up. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2003.

[49] B. Rothrock, S. Park, and S.-C. Zhu. Integrating grammar and segmentation for human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

[50] B. Sapp, C. Jordan, and B. Taskar. Adaptive pose priors for pictorial structures. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2010.

[51] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *Proceedings of the European Conference on Computer Vision*, 2010.

[52] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[53] H. Sarmadi. Human detection and pose estimation in a multi-camera system. *Master's Thesis at KTH Royal Institute of Technology, Sweden*, 2013.

[54] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In *Proceedings of the International Conference on Computer Vision*, 2003.

[55] 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 *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[56] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, and Andrew Blake. Efficient human pose estimation from single depth images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99:1, 2012.

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

[58] Hedvig Sidenbladh and Michael J. Black. Learning the statistics of people in images and video. *Internation Journal of Computer Vision*, 54(1-3):183–209, 2003.

[59] L. Sigal and M. Black. Guest editorial: state of the art in image- and video-based human pose and motion estimation. *Internation Journal of Computer Vision*, 87 (1-2):1–3, 2010.

[60] L. Sigal, M. Isard, H. Haussecker, and M. Black. Loose-limbed people: Estimating 3d human pose and motion using non-parametric belief propagation. *Internation Journal of Computer Vision*, 98(1):15–48, 2012.

[61] C. Sminchisescu and B. Triggs. Estimating Articulated Human Motion with Covariance Scaled Sampling. *International Journal of Robotics Research*, 22(6):371–393, 2003.

[62] Min Sun, Pushmeet Kohli, and Jamie Shotton. Conditional regression forests for human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 3394–3401, 2012.

[63] A. Thayananthan, R. Navaratnam, B. Stenger, P. Torr, and P. Cipolla. Pose estimation and tracking using multivariate regression. *Pattern Recognition Letters*, 29(9): 1302–1310, 2008.

[64] T.-P. Tian and S. Sclaroff. Fast globally optimal 2d human detection with loopy graph models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2010.

[65] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Internation Journal of Computer Vision*, 9:137–154, November 1992.

[66] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2014.

[67] D. Tran and D. Forsyth. Improved human parsing with a full relational model. In *Proceedings of the European Conference on Computer Vision*, 2010.

[68] N. Ukita. Articulated pose estimation with parts connectivity using discriminative local oriented contours. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2012.

[69] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2006.

[70] R. Urtasun and P. Fua. 3d human body tracking using deterministic temporal motion models. In *Proceedings of the European Conference on Computer Vision*, 2004.

[71] H. Wang and D. Koller. Multi-level inference by relaxed dual decomposition for human pose segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[72] Y. Wang, D. Tran, and Z. Liao. Learning hierarchical poselets for human parsing. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[73] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[74] T. H. Yu, T. K. Kim, and R. Cipolla. Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2013.

Algorithm 1 Training a classification tree

Have labelled training data $\{(\mathbf{h}_k, \mathbf{v}_k, y_k)\}_{k=1}^{K}$. Let $\mathcal{Q}$ be the indices of the training examples associated with a node. A split at this node is found by the following process:

1. Randomly select a pool of features $\theta_1, \ldots, \theta_M$ where each $\theta_m = (\mathbf{u}_m, d_m)$. $\mathbf{u}_m$ is a 2D offset vector and d_m is the dimension of a HoG cell descriptor.
2. Let $F(\mathbf{h}_k, \mathbf{v}_k, \theta_m)$ represent the value of the d_mth dimension of pixel $(\mathbf{v}_k + \mathbf{u}_m)$'s HoG cell descriptor.
3. For each θ_m select potential thresholds $\tau_{m,1}, \ldots, \tau_{m,s}$ by dividing the interval between the minimum and maximum values for $\{F(\mathbf{h}_k, v_k, \theta_m)\}_{k \in \mathcal{Q}}$ into equal sub-intervals.
4. Each potential split (θ, τ) partitions $\mathcal{Q}$:

$$\mathcal{Q}_l(\theta, \tau) = \{k \mid k \in \mathcal{Q} \text{ and } F(\mathbf{h}_k, \mathbf{v}_k, \theta) < \tau\} \tag{1}$$

$$\mathcal{Q}_r(\theta, \tau) = \mathcal{Q} \setminus \mathcal{Q}_l(\theta, \tau) \tag{2}$$

5. Choose the parameters (θ^*, τ^*) that maximize the information gain which corresponds to:

$$(\theta^*, \tau^*) = \arg\min_{(\theta,\tau)} \sum_{s \in \{l,r\}} |\mathcal{Q}_s(\theta, \tau)|\, \mathcal{H}(\mathcal{Q}_s(\theta, \tau)) \tag{3}$$

   where $\mathcal{H}(Q)$ is the entropy of the probabilities for the class labels in $\mathcal{Q}$:

$$\mathcal{H}(\mathcal{Q}) = \sum_{y=0}^{N} P_{\mathcal{Q}}(y) \log P_{\mathcal{Q}}(y) \tag{4}$$

   and $P_{\mathcal{Q}}(y)$ is ratio of examples with label y:

$$P_{\mathcal{Q}}(y) = \frac{1}{|\mathcal{Q}|} \sum_{k \in \mathcal{Q}} \text{Ind}(y_k = y) \tag{5}$$

6. If the information gain induced by (θ^*, τ^*) is sufficiently large make the split. If the split is made, try to further split $\mathcal{Q}_l(\theta^*, \tau^*)$ if both the maximum depth of the tree is not exceeded and $|\mathcal{Q}_l(\theta^*, \tau^*)|$ is sufficiently large. Do the same for $\mathcal{Q}_r(\theta^*, \tau^*)$.

Paper D

One Millisecond Face Alignment with an Ensemble of Regression Trees

Vahid Kazemi and Josephine Sullivan

Published in
Computer Vision and Pattern Recognition Conference, 2014

One Millisecond Face Alignment with an Ensemble of Regression Trees

Vahid Kazemi and Josephine Sullivan

Abstract

This paper addresses the problem of Face Alignment for a single image. We show how an ensemble of regression trees can be used to estimate the face's landmark positions directly from a sparse subset of pixel intensities, achieving super-realtime performance with high quality predictions. We present a general framework based on gradient boosting for learning an ensemble of regression trees that optimizes the sum of square error loss and naturally handles missing or partially labelled data. We show how using appropriate priors exploiting the structure of image data helps with efficient feature selection. Different regularization strategies and its importance to combat overfitting are also investigated. In addition, we analyse the effect of the quantity of training data on the accuracy of the predictions and explore the effect of data augmentation using synthesized data.

1 Introduction

Face alignment corresponds to automatically finding the location of a set of predefined landmarks on a human face. Accurate face alignment is useful for many applications including markerless performance capture for creating realistic animations for computer games and movies. It is also key to improving the performance on high level problems such as person and expression recognition.

Faces are easy to detect due to the more or less fixed spatial arrangement of its main parts - eyes, nose and mouth - relative to one another. However, it is not so straightforward to precisely locate the landmark points (the shape), because at a more micro level there is significantly more variation. Firstly, faces can deform in a highly non-rigid fashion as the dozens of the muscles under the skin move to form different facial expressions. Additionally, the appearance of faces can significantly change across people and under different illumination conditions which we term the nuisance factors.

Figure 1: Selected results on the HELEN dataset. An ensemble of randomized regression trees is used to detect 194 landmarks on face from a single image in a few milliseconds.

In this work we show, as others have [7, 2], though that face alignment can be solved with a cascade of regression functions. In our case each regression function in the cascade efficiently estimates the shape from an initial estimate and the intensities of a sparse set of pixels indexed relative to this initial estimate. Our work builds on the large amount of research over the last decade that has resulted in significant progress for face alignment [8, 4, 12, 6, 14, 1, 15, 17, 3] and to the simultaneous face detection and alignment [18, 5]. In particular, we incorporate into our learnt regression functions two key elements that are present in several of the successful algorithms cited and we detail these elements now.

The first revolves around the indexing of pixel intensities relative to the current estimate of the shape. The extracted features in the vector representation of a face image can greatly vary due to both shape deformation and the nuisance factors. This make accurate shape estimation using these features difficult. The dilemma then is that we need reliable features to accurately predict the shape, and on the other hand we need an accurate estimate of the shape to extract reliable features. Previous work [4, 8, 7] as well as this work, use an iterative approach (the cascade) to

deal with this problem. Instead of regressing the shape parameters based on features extracted in the global coordinate system of the image, the image is transformed to a normalized coordinate system based on a current estimate of the shape, and then the features are extracted to predict an update vector for the shape parameters. This process is usually repeated several times until convergence.

The second considers how to combat the difficulty of the inference/prediction problem. At test time, an alignment algorithm has to estimate the shape, a high dimensional vector, that best agrees with the image data and our model of shape. The problem is non-convex with many local optima. Successful algorithms [4, 8] handle this problem by assuming the estimated shape must lie in a linear subspace, which can be discovered for example by finding the principal components of the training shapes. This assumption greatly reduces the number of potential shapes considered during inference and can help to avoid local optima. Recent work [7, 10, 2] use the fact that a certain class of regressors are guaranteed to produce predictions that lie in a linear subspace defined by the training shapes and there is no need for additional constraints.

Crucially, our algorithm has these two elements, but within a fully data driven framework that performs the shape invariant feature selection by minimizing the same loss function during training as we want to minimize at test time, and this is what separates this work from earlier work. The proposed framework, produces high quality predictions while being highly efficient (Figure 1). It also has the advantage of naturally handling missing or uncertain labels. In this work, we provide experimental results showing the contribution of major components of our method on final predictions. Furthermore, we analyse the effect of quantity of training data, use of partially labelled data and synthesized data on quality of predictions.

2 Method

This work presents an algorithm to precisely estimate the position of facial landmarks in a computationally efficient way. Similar to previous works [7, 2] our proposed method utilizes a cascade of regressors. In the rest of this section we describe the details of the form of the individual components of the cascade and how we perform training.

2.1 The cascade of regressors

To begin we introduce some notation. Let $\mathbf{x}_i \in \mathbb{R}^2$ be the x, y-coordinates of the ith facial landmark in an image I. Then the vector $\mathbf{S} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_p^T)^T \in \mathbb{R}^{2p}$ denotes the coordinates of all the p facial landmarks in I. Frequently, in this work we refer to the vector $\mathbf{S}$ as the shape. We use $\hat{\mathbf{S}}^{(t)}$ to denote our current estimate of $\mathbf{S}$. Each regressor, $r_t(\cdot, \cdot)$, in the cascade predicts an update vector from the image and $\hat{\mathbf{S}}^{(t)}$ that is added to the current shape estimate $\hat{\mathbf{S}}^{(t)}$ to improve the estimate.

$$\hat{\mathbf{S}}^{(t+1)} = \hat{\mathbf{S}}^{(t)} + r_t(I, \hat{\mathbf{S}}^{(t)}) \tag{1}$$

The critical point of the cascade is that the regressor r_t makes its predictions based on features, such as pixel intensity values, computed from I and indexed relative to the current shape estimate $\hat{\mathbf{S}}^{(t)}$. This introduces some form of geometric invariance into the process and as the cascade proceeds one can be more certain that a precise semantic location on the face is being indexed. Later we describe how this indexing is performed.

Note that the range of outputs expanded by the ensemble is ensured to lie in a linear subspace of training data if the initial estimate $\hat{\mathbf{S}}^{(0)}$ belongs to this space. We therefore do not need to enforce additional constraints on the predictions which greatly simplifies our method. The initial shape can simply be chosen as the mean shape of the training data centred and scaled according to the bounding box output of a generic face detector.

To train each r_t we use the gradient tree boosting algorithm with a sum of square error loss as described in [9]. We now give the explicit details of this process.

2.2 Learning each regressor in the cascade

Assume we have training data $(I_1, \mathbf{S}_1), \ldots, (I_n, \mathbf{S}_n)$ where each I_i is a face image and $\mathbf{S}_i$ its shape vector. To learn the first regression function r_0 in the cascade we create from our training data triplets of a face image, an initial shape estimate and the target update step, that is, $(I_{\pi_i}, \hat{\mathbf{S}}_i^{(0)}, \Delta\mathbf{S}_i^{(0)})$ where

$$\pi_i \in \{1, \ldots, n\} \tag{2}$$

$$\hat{\mathbf{S}}_i^{(0)} \in \{\mathbf{S}_1, \ldots, \mathbf{S}_n\} \backslash \mathbf{S}_{\pi_i} \text{ and} \tag{3}$$

$$\Delta\mathbf{S}_i^{(0)} = \mathbf{S}_{\pi_i} - \hat{\mathbf{S}}_i^{(0)} \tag{4}$$

for $i = 1, \ldots, N$. We set the total number of these triplets to $N = nR$ where R is the number of initializations used per each image I_i. Each initial shape estimate for an image is sampled uniformly from $\{\mathbf{S}_1, \ldots, \mathbf{S}_n\}$ without replacement.

From this data we learn the regression function r_0 (see algorithm 2), using gradient tree boosting with a sum of square error loss. The set of training triplets is then updated to provide the training data, $(I_{\pi_i}, \hat{\mathbf{S}}_i^{(1)}, \Delta\mathbf{S}_i^{(1)})$, for the next regressor r_1 in the cascade by setting (with $t = 0$)

$$\hat{\mathbf{S}}_i^{(t+1)} = \hat{\mathbf{S}}_i^{(t)} + r_t(I_{\pi_i}, \hat{\mathbf{S}}_i^{(t)}) \tag{5}$$

$$\Delta\mathbf{S}_i^{(t+1)} = \mathbf{S}_{\pi_i} - \hat{\mathbf{S}}_i^{(t+1)} \tag{6}$$

This process is iterated until a cascade of T regressors $r_0, r_1, \ldots, r_{T-1}$ are learnt which when combined give a sufficient level of accuracy.

As stated each regressor r_t is learned using the gradient boosting tree algorithm. It should be remembered that a square error loss is used and the residuals computed in the innermost loop correspond to gradient of this loss function computed at each training sample. Included in the statement of the algorithm is a learning rate

parameter $0 < \nu \leq 1$ also known as the shrinkage factor. Setting $\nu < 1$ helps combat against over-fitting and usually results in regressors which generalize much better than those learnt with $\nu = 1$ [9].

Algorithm 2 Learning r_t in the cascade

Have training data $\{(I_{\pi_i}, \hat{\mathbf{S}}_i^{(t)}, \Delta\mathbf{S}_i^{(t)})\}_{i=1}^N$ and the learning rate (shrinkage factor) $0 < \nu < 1$

1. Initialise

$$f_0(I, \hat{\mathbf{S}}^{(t)}) = \arg\min_{\boldsymbol{\gamma} \in \mathbb{R}^{2p}} \sum_{i=1}^{N} \|\Delta\mathbf{S}_i^{(t)} - \boldsymbol{\gamma}\|^2$$

2. for $k = 1, \ldots, K$:
   (a) Set for $i = 1, \ldots, N$

$$\mathbf{r}_{ik} = \Delta\mathbf{S}_i^{(t)} - f_{k-1}(I_{\pi_i}, \hat{\mathbf{S}}_i^{(t)})$$

   (b) Fit a regression tree to the targets $\mathbf{r}_{ik}$ giving a weak regression function $g_k(I, \hat{\mathbf{S}}^{(t)})$.
   (c) Update

$$f_k(I, \hat{\mathbf{S}}^{(t)}) = f_{k-1}(I, \hat{\mathbf{S}}^{(t)}) + \nu\, g_k(I, \hat{\mathbf{S}}^{(t)})$$

3. Output $r_t(I, \hat{\mathbf{S}}^{(t)}) = f_K(I, \hat{\mathbf{S}}^{(t)})$

2.3 Tree based regressor

The core of each regression function r_t is the tree based regressors fit to the residual targets during the gradient boosting algorithm. We now review the most important implementation details for training each regression tree.

2.3.1 Shape invariant split tests

At each split node in the regression tree we make a decision based on thresholding the difference between the intensities of two pixels. The pixels used in the test are at positions $\mathbf{u}$ and $\mathbf{v}$ when defined in the coordinate system of the mean shape. For a face image with an arbitrary shape we would like to index the points that have the same position relative to its shape as $\mathbf{u}$ and $\mathbf{v}$ have to the mean shape. To achieve this the image can be warped to the mean shape based on the current shape estimate before extracting the features. Since we only use a very sparse

representation of the image it is much more efficient to warp the location of points as opposed the whole image. Furthermore, a crude approximation of warping can be done using only a global similarity transform in addition to local translations as suggested by [2].

The precise details are as follows. Let $k_{\mathbf{u}}$ be the index of the facial landmark in the mean shape that is closest to $\mathbf{u}$ and define its offset from $\mathbf{u}$ as

$$\delta\mathbf{x}_{\mathbf{u}} = \mathbf{u} - \bar{\mathbf{x}}_{k_{\mathbf{u}}}$$

Then for a shape $\mathbf{S}_i$ defined in image I_i, the position in I_i that is qualitatively similar to $\mathbf{u}$ in the mean shape image is given by

$$\mathbf{u}' = \mathbf{x}_{i,k_{\mathbf{u}}} + \frac{1}{s_i} R_i^T \delta\mathbf{x}_{\mathbf{u}} \tag{7}$$

where s_i and R_i are the scale and rotation matrix which define the similarity transform which transforms $\mathbf{S}_i$ to $\bar{\mathbf{S}}$, the mean shape, and minimizes

$$\sum_{j=1}^{p} \|\bar{\mathbf{x}}_j - (s_i R_i \, \mathbf{x}_{i,j} + \mathbf{t}_i)\|^2 \tag{8}$$

the sum of squares between the mean shape's facial landmark points, $\bar{\mathbf{x}}_j$'s, and those of the warped shape. $\mathbf{v}'$ is similarly defined. Then formally each split is a decision involving 3 parameters $\boldsymbol{\theta} = (\tau, \mathbf{u}, \mathbf{v})$ and is applied to each training and test example as

$$h(I_{\pi_i}, \hat{\mathbf{S}}_i^{(t)}, \boldsymbol{\theta}) = \begin{cases} 1 & I_{\pi_i}(\mathbf{u}') - I_{\pi_i}(\mathbf{v}') > \tau \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\mathbf{u}'$ and $\mathbf{v}'$ are defined using the scale and rotation matrix which best warp $\hat{\mathbf{S}}_i^{(t)}$ to $\bar{\mathbf{S}}$ according to equation (7).

Note that in practice the assignments, and local translations are determined during training phase. Calculating the similarity transform which is the most computationally expensive part of this process at test time is only done once at each level of the cascade.

2.3.2 Choosing the node splits

For each regression tree we approximate the underlying function with a piecewise constant function where a constant vector is fit to each leaf node. To train the regression tree we randomly generate a set of candidate splits, that is $\boldsymbol{\theta}$'s, at each node. We then greedily choose the $\boldsymbol{\theta}^*$, from these candidates, which minimizes the sum of square error. If $\mathcal{Q}$ is the set of the indices of the training examples at a node this corresponds to minimizing

$$E(\mathcal{Q}, \boldsymbol{\theta}) = \sum_{s \in \{l,r\}} \sum_{i \in \mathcal{Q}_{\boldsymbol{\theta},s}} \|\mathbf{r}_i - \boldsymbol{\mu}_{\boldsymbol{\theta},s}\|^2 \tag{10}$$

where $\mathcal{Q}_{\boldsymbol{\theta},l}$ is the indices of the examples that are sent to the left node due to the decision induced by $\boldsymbol{\theta}$, $\mathbf{r}_i$ is the vector of all the residuals computed for image i in the gradient boosting algorithm and

$$\boldsymbol{\mu}_{\boldsymbol{\theta},s} = \frac{1}{|\mathcal{Q}_{\boldsymbol{\theta},s}|} \sum_{i \in \mathcal{Q}_{\boldsymbol{\theta},s}} \mathbf{r}_i, \quad \text{for } s \in \{l, r\} \tag{11}$$

The optimal split can be found very efficiently because if one rearranges equation (10) and omits the factors not dependent on $\boldsymbol{\theta}$ then one can see that

$$\arg\min_{\boldsymbol{\theta}} E(\mathcal{Q}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \sum_{s \in \{l,r\}} |\mathcal{Q}_{\boldsymbol{\theta},s}| \, \boldsymbol{\mu}_{\boldsymbol{\theta},s}^T \boldsymbol{\mu}_{\boldsymbol{\theta},s}$$

Here we only need to compute $\boldsymbol{\mu}_{\boldsymbol{\theta},l}$ when evaluating different $\boldsymbol{\theta}$'s, as $\boldsymbol{\mu}_{\boldsymbol{\theta},r}$ can be calculated based on the average of targets at the parent node $\boldsymbol{\mu}$ and $\boldsymbol{\mu}_{\boldsymbol{\theta},r}$ as follows

$$\boldsymbol{\mu}_{\boldsymbol{\theta},r} = \frac{|\mathcal{Q}|\boldsymbol{\mu} - |\mathcal{Q}_{\boldsymbol{\theta},l}|\boldsymbol{\mu}_{\boldsymbol{\theta},l}}{\mathcal{Q}_{\boldsymbol{\theta},r}}$$

2.3.3 Feature selection

Recall that decisions at each node are based on thresholding the difference of intensity values of pairs of pixels. This is a rather simple test, but it is much more powerful than single intensity thresholding because of its relative insensitivity to changes in global lighting. Unfortunately, the drawback of using pixel differences is the number of potential split (feature) candidates is quadratic in the number of pixels in the mean image. This makes is difficult to find good $\boldsymbol{\theta}$'s without searching over a very large number of them. However, this limiting factor can be eased, to some extent, by taking the structure of image data into account. One can simply introduce an exponential prior

$$P(\mathbf{u}, \mathbf{v}) \propto e^{-\lambda \|\mathbf{u}-\mathbf{v}\|} \tag{12}$$

over the distance between the pixels used in a split to encourage closer pixel pairs to be chosen.

We found using this simple prior slightly reduces the prediction error on a number of face datasets. Figure 4 compares the features selected with and without this prior, where the size of feature pool is fixed to 20 in both cases.

2.4 Handling missing labels

The objective of equation (10) can be easily extended to handle the case where some of the landmarks are not labelled in some of the training images (or we have a measure of uncertainty for each landmark). Introduce variables $w_{i,j} \in [0, 1]$ for each training image i and each landmark j. Setting $w_{i,j}$ to 0 indicates that the

landmark j is not labelled in the ith image while setting it to 1 indicates that it is. Then equation (10) can be updated to

$$E(\mathcal{Q}, \boldsymbol{\theta}) = \sum_{s \in \{l,r\}} \sum_{i \in \mathcal{Q}_{\boldsymbol{\theta},s}} (\mathbf{r}_i - \boldsymbol{\mu}_{\boldsymbol{\theta},s})^T W_i (\mathbf{r}_i - \boldsymbol{\mu}_{\boldsymbol{\theta},s})$$

where W_i is a diagonal matrix with the vector $(w_{i1}, w_{i1}, w_{i2}, w_{i2}, \ldots, w_{ip}, w_{ip})^T$ on its diagonal and

$$\boldsymbol{\mu}_{\boldsymbol{\theta},s} = \left(\sum_{i \in \mathcal{Q}_{\boldsymbol{\theta},s}} W_i \right)^{-1} \sum_{i \in \mathcal{Q}_{\boldsymbol{\theta},s}} W_i \, \mathbf{r}_i, \quad \text{for } s \in \{l, r\} \tag{13}$$

Subsequently the gradient boosting algorithm is modified to account for the weights. This can be done simply by initializing the ensemble model with the weighted average of targets, and fitting regression trees to the weighted residuals in algorithm 2 as follows

$$\mathbf{r}_{ik} = W_i(\Delta \mathbf{S}_i^{(t)} - f_{k-1}(I_{\pi_i}, \hat{\mathbf{S}}_i^{(t)})) \tag{14}$$

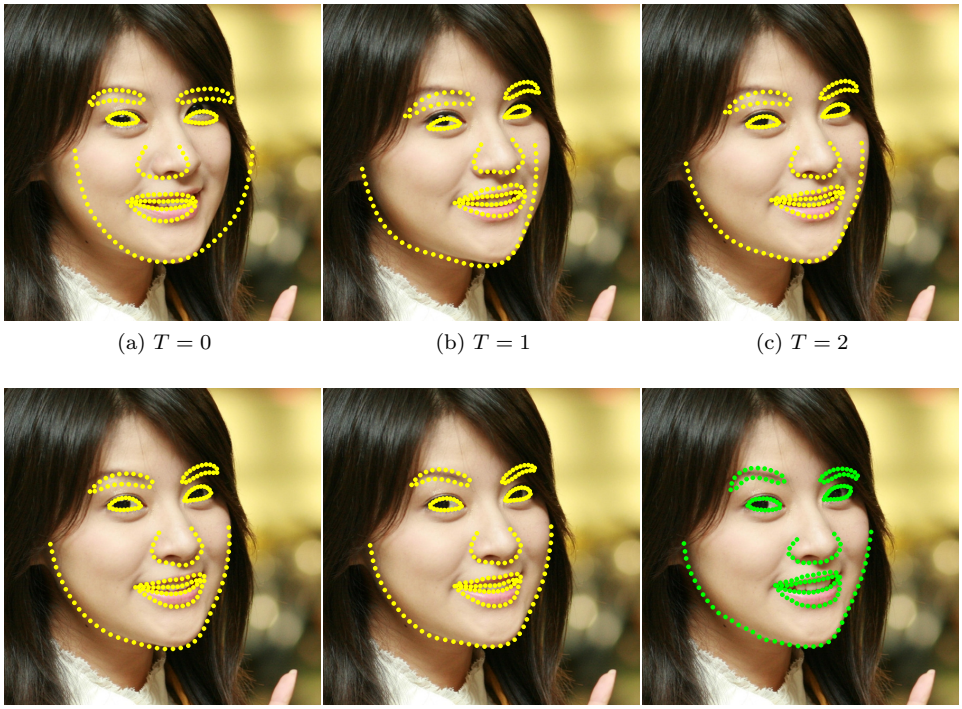
3 Experiments

Baselines: To accurately benchmark the performance of our method, in addition to implementation of the our proposed ensemble of regression trees (*ERT*) we created two more baselines. The first is based on randomized ferns with random feature selection (*EF*) and the other is a more advanced version of this with correlation based feature selection (*EF+CB*) which is our re-implementation of [2]. All the parameters are fixed for all three approaches.

EF uses a straightforward implementation of randomized ferns as the weak regressors within the ensemble and is the fastest to train. We use the same shrinkage method as suggested by [2] to regularize the ferns.

EF+CB uses a correlation based feature selection method that projects the target outputs, $\mathbf{r}_i$'s, onto a random direction, $\mathbf{w}$, and chooses the pairs of features $(\mathbf{u}, \mathbf{v})$ s.t. $I_i(\mathbf{u}') - I_i(\mathbf{v}')$ has the highest sample correlation over the training data with the projected targets $\mathbf{w}^T \mathbf{r}_i$.

Parameters: Unless specified, all the experiments are performed with the following fixed parameter settings. The number of strong regressors, r_t, in the cascade is $T = 10$ and each r_t comprises of $K = 500$ weak regressors g_k. The depth of the trees (or ferns) used as to represent g_k is set to $F = 5$. At each level of the cascade $P = 400$ pixel locations are sampled from the image. To train the weak regressors we randomly sample a pair of these P pixel locations according to our prior and choose a random threshold to create a potential split as described in equation (9). The best split is then found by repeating this process $S = 20$ times, and choosing the one that optimizes our objective. To create the training data to learn our model we use $R = 20$ different initializations for each training example.



(a) $T = 0$ (b) $T = 1$ (c) $T = 2$

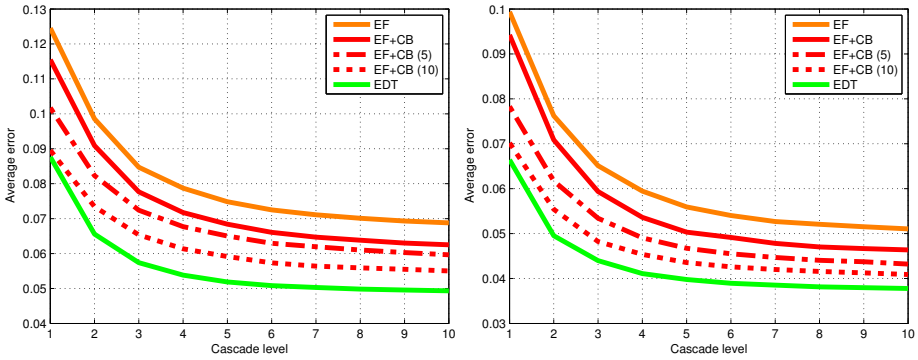
(d) $T = 3$ (e) $T = 10$ (f) Ground truth

Figure 2: Landmark estimates at different levels of the cascade initialized with the mean shape centered at the output of a basic Viola & Jones[16] face detector. Note that after the first level of the cascade, the error is already greatly reduced.

Performance: The runtime complexity of the algorithm on a single image is constant $O(TKF)$. The complexity of training time depends linearly on the number of training data $O(NDTKFS)$ where N is the number of training data and D is dimension of the targets. In practice with a single CPU our algorithm takes about an hour to train on the HELEN[11] dataset and at runtime it only takes about a few milliseconds per image.

Database: Most of the experimental results reported are for the HELEN[11] face database which we found to be the most challenging publicly available dataset. It consists of a total of 2330 images, each of which is annotated with 194 landmarks. As suggested by the authors we use 2000 images for training data and the rest for testing.

We also report final results on the popular LFPW[1] database which consists of 1432 images. Unfortunately, we could only download 778 training images and 216 valid test images which makes our results not directly comparable to those



(a) HELEN (b) LFPW

Figure 3: A comparison of different methods on HELEN(a) and LFPW(b) dataset. *EF* is the ensemble of randomized ferns and *EF+CB* is the ensemble of ferns with correlation based feature selection initialized with the mean shape. We also provide the results of taking the median of results of various initializations (5 and 10) as suggested by [2]. The results show that the proposed ensemble of regression trees (*ERT*) initialized with only the mean shape consistently outperforms the ensemble of ferns baseline and it can reach the same error rate with much less computation.

previously reported on this dataset.

Comparison: Table 1 is a summary of our results compared to previous algorithms. In addition to our baselines, we have also compared our results with two variations of Active Shape Models, *STASM*[13] and *CompASM*[11].

	[13]	[11]	EF	EF+CB	EF+CB (5)	EF+CB (10)	ERT
Error	.111	.091	.069	.062	.059	.055	**.049**

Table 1: A summary of the results of different algorithms to the HELEN dataset. The error is the average normalized distance of each landmark to its ground truth position. The distances are normalized by dividing by the interocular distance. The number within the bracket represents the number of times the regression algorithm was run with a random initialization. If no number is displayed then the method was initialized with the mean shape. In the case of multiple estimations the median of the estimates was chosen as the final estimate for the landmark.

The ensemble of regression trees described in this work significantly improves the results over the ensemble of ferns. Figure 3 shows the average error at different levels of the cascade which shows that *ERT* can reduce the error much faster than other baselines. Note that we have also provided the results of running *EF+CB*

multiple times and taking the median of final predictions. The results show that similar error rate to *EF+CB* can be achieved by our method with an order of magnitude less computation.

We have also provided results for the widely used LFPW[1] dataset (Table 2). With our *EF+CB* baseline we could not replicate the numbers reported by [2]. (This could be due to the fact that we could not obtain the whole dataset.) Nevertheless our method surpasses most of the previously reported results on this dataset taking only a fraction of computational time needed by any other method.

	[1]	[2]	EF	EF+CB	EF+CB (5)	EF+CB (10)	ERT
Error	.040	**.034**	.051	.046	.043	.041	.038

Table 2: A comparison of the different methods when applied to the LFPW dataset. Please see the caption for table 1 for an explanation of the numbers.

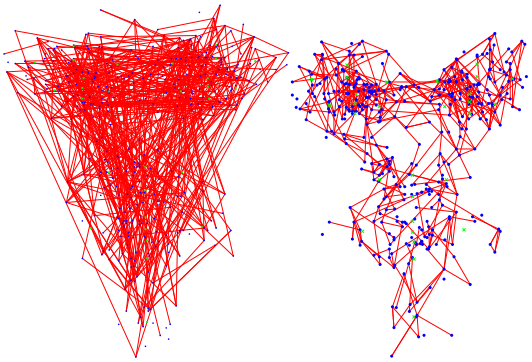
Feature Selection: Table 3 shows the effect of using equation (12) as a prior on the distance between pixels used in a split instead of a uniform prior on the final results. The parameter λ which determines the distribution of range of features was set to 0.1 in our experiments. Selecting this parameter by cross validation when learning each strong regressor, r_t, in the cascade could potentially lead to a more significant improvement. Figure 4 is a visualization of the selected pairs of features when the different priors are used.

	Uniform	Exponential
Error	.053	.049

Table 3: The effect of using different priors for selecting pairs of features on final average error. The exponential prior is applied on euclidean distance between pairs and is defined by equation 12.

Regularization: When using the gradient boosting algorithm one needs to be careful to avoid overfitting. To obtain lower test errors it is necessary to perform some form of regularization. The simplest approach is shrinkage. This applies setting the learning rate ν in the gradient boosting algorithm to less than 1 (Here we set $\nu = 0.1$). Regularization can also be achieved by averaging the predictions of multiple regression trees. This way, g_k correspond to a random forest as opposed to one tree and we set $\nu = 1$. Therefore at each iteration of the gradient boosting algorithm instead of fitting one regression tree to the residuals, we fit multiple trees (10 in our experiments) and average the results. (Note that the total number of trees is fixed in all the cases.)

In terms of bias and variance trade off, the gradient boosting algorithm always decreases the bias but increases the variance while regularizing by shrinkage or averaging effectively reduces the variance by learning multiple overlapping models.



(a) Uniform prior (b) Exponential prior

Figure 4: Different features are selected if different priors are used. The exponential prior biases the selection towards pairs of pixels which are closer together.

	Unregularized	Shrinkage	Averaging
Error	.103	.049	.049

Table 4: A comparison of the results on the HELEN dataset when different forms of regularization are applied. We found similar results using shrinkage and averaging given the same total number of trees in the ensemble.

We achieved similar results using the averaging regularization compared to the more standard shrinkage method (table 4). However, regularization by averaging has the advantage of being more scalable, as it enables parallelization during training time which is especially important for solving large scale problems.

Cascade: At each level of the cascade the second level regressors can only observe a fixed and sparse subset of the shape indexed features. Indexing the features based on the current estimate is a crude way of warping the image with a small cost. Table 5 shows the final error rate with and without using the cascade. We found significant improvement by using this iterative mechanism which is in line with previously reported results [7, 2] (Note that for a fair comparison here we fixed the total number of observed features to 10×400 points).

# Trees	1×500	1×5000	10×500
Error	.085	.074	.049

Table 5: The results show the importance of using a cascade of regressors as opposed to a single level ensemble.

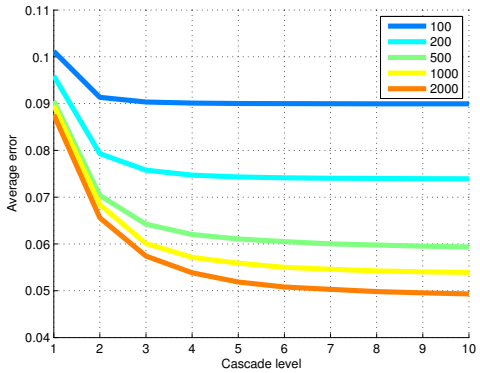


Figure 5: Average error at each level of cascade is plotted with respect to number of training examples used. Using many levels of regressors is most useful when the number of training examples is large.

Training Data: To test the performance of our method with respect to the number of training data, we trained different models from different sized subsets of the training data. Table 6 summarizes the final results and figure 5 is a plot of the error at each level of the cascade. Using many levels of regressors is most useful when we have large number of training examples.

# Examples	100	200	500	1000	2000
Error	.090	.074	.059	.054	.049

Table 6: Final error rate with respect to the number of training examples. When creating training data for learning the cascade regressors each labelled face image generated 20 training examples by using 20 different labelled faces as the initial guess for the face's shape.

We repeated the same experiments with the total number of augmented examples fixed but varied the combination of initial shapes used to generate a training example from one labelled face example and the number of annotated images used to learn the cascade (Table 7).

Augmenting the training data using different initial shapes is a way of expanding the dataset in terms of shape. To achieve invariance to appearance (texture and lighting) changes we still need to use more annotated images. Although the rate of improvement gained by increasing training data quickly slows after the first few hundred images.

Partial annotations: Table 8 shows the results of using partially annotated data. 200 training examples are fully annotated and the rest are only partially annotated.

# Examples	100	200	500	1000	2000
# Initial Shapes	400	200	80	40	20
Error	.062	.057	.054	.052	.049

Table 7: Here the effective number of training data is fixed but we use different combinations of the number of training images and number of initial shapes used for each labelled face image.

# Examples	200	200+1800(25%)	200+1800(50%)	2000
Error	.074	.067	.061	.049

Table 8: Results of using partially labelled data. 200 examples are always fully annotated. The values inside the parenthesis show the percentage of landmarks observed.

The results show that we can gain substantial improvement by using partially labelled data. Yet the improvement displayed may not be saturated because we know that the underlying dimension of shape parameters are much lower than the dimension of the landmarks (194×2). There is, therefore, potential for a more significant improvement with partial labels by taking explicit advantage of the correlation between the position of landmarks. Note that the gradient boosting procedure described in this work does not take advantage of the correlation between landmarks. This issue will be addressed in a future work.

4 Conclusion

We have described how an ensemble of regression trees can be used to regress the location of a set of landmarks from a sparse subset of intensity values extracted from the image. The presented framework has the advantage of being faster in reducing the error compared to the previous work and can also handle partial or uncertain labels. While major components of our algorithm treat different target dimensions as independent variables, for a future work we intend to take advantage of the correlation of shape parameters for more efficient training and a better use of partial labels.

References

[1] Peter N. Belhumeur, David W. Jacobs, David J. Kriegman, and Neeraj Kumar. Localizing parts of faces using a consensus of exemplars. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 545–552, 2011.

[2] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2887–2894, 2012.

[3] Timothy F. Cootes, Mircea C. Ionita, Claudia Lindner, and Patrick Sauer. Robust and accurate shape model fitting using random forest regression voting. In *Proceedings of the European Conference on Computer Vision*, pages 278–291, 2012.

[4] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[5] Matthias Dantone, Juergen Gall, Gabriele Fanelli, and Luc J. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2578–2585, 2012.

[6] Liya Ding and Aleix M. Martínez. Precise detailed detection of faces and facial features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2008.

[7] Piotr Dollár, Peter Welinder, and Pietro Perona. Cascaded pose regression. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 1078–1085, 2010.

[8] Gareth J. Edwards, Timothy F. Cootes, and Christopher J. Taylor. Advances in active appearance models. In *Proceedings of the International Conference on Computer Vision*, pages 137–142, 1999.

[9] T. Hastie, R. Tibshirani, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* New York: Springer-Verlag, 2001.

[10] Vahid Kazemi and Josephine Sullivan. Face alignment with part-based modeling. In *Proceedings of the British Machine Vision Conference*, pages 27.1–27.10, 2011.

[11] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir D. Bourdev, and Thomas S. Huang. Interactive facial feature localization. In *Proceedings of the European Conference on Computer Vision*, pages 679–692, 2012.

[12] Lin Liang, Rong Xiao, Fang Wen, and Jian Sun. Face alignment via component-based discriminative search. In *Proceedings of the European Conference on Computer Vision*, pages 72–85, 2008.

[13] Stephen Milborrow and Fred Nicolls. Locating facial features with an extended active shape model. In *Proceedings of the European Conference on Computer Vision*, pages 504–513, 2008.

[14] J. Saragih, S. Lucey, and J. Cohn. Deformable model fitting by regularized landmark mean-shifts. *Internation Journal of Computer Vision*, 91:200–215, 2010.

[15] Brandon M. Smith and Li Zhang. Joint face alignment with non-parametric shape models. In *Proceedings of the European Conference on Computer Vision*, pages 43–56, 2012.

[16] Paul A. Viola and Michael J. Jones. Robust real-time face detection. In *Proceedings of the International Conference on Computer Vision*, page 747, 2001.

[17] Xiaowei Zhao, Xiujuan Chai, and Shiguang Shan. Joint face alignment: Rescue bad alignments with good ones by regularized re-fitting. In *Proceedings of the European Conference on Computer Vision*, 2012.

[18] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 2879–2886, 2012.

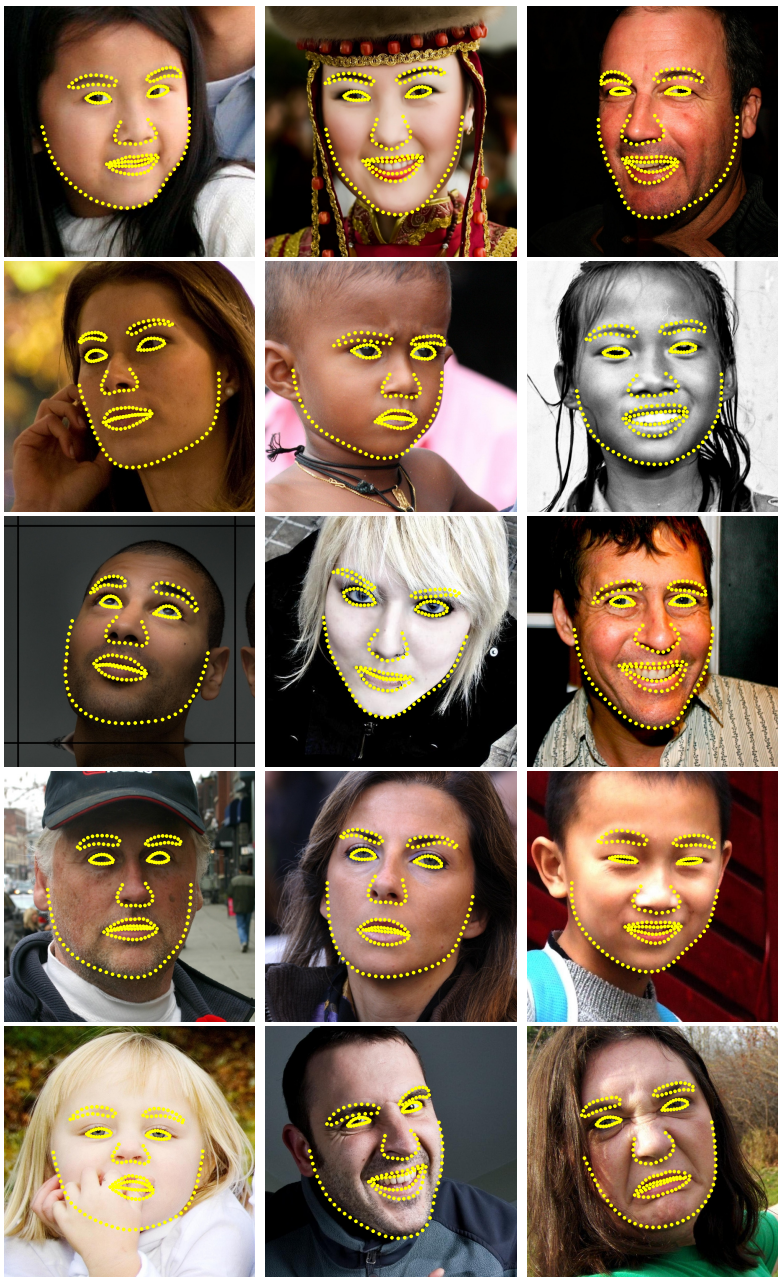


Figure 6: Final results on HELEN database.

Figure 7: Final results on HELEN database.

Paper E

Real-time Face Reconstruction from a Single Depth Image

Vahid Kazemi, Cem Keskin, Jonathan Taylor,
Pushmeet Kohli, and Shahram Izadi

Published in
International Conference on 3D Vision, 2014

Real-time Face Reconstruction from a Single Depth Image

Vahid Kazemi, Cem Keskin, Jonathan Taylor,
Pushmeet Kohli, and Shahram Izadi

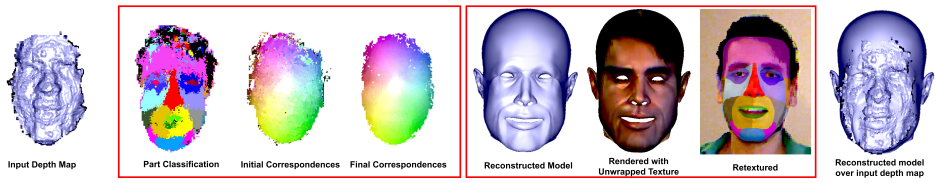


Figure 1: Our method starts with estimating dense correspondences on an input depth image, using a discriminative model. A generative model parametrized by blend shapes is then utilized to further refine these correspondences. The final correspondence field is used for per-frame 3D face shape and expression reconstruction, allowing for texture unwrapping, retexturing or retargeting in real-time.

Abstract

This paper contributes a real time method for recovering facial shape and expression from a single depth image. The method also estimates an accurate and dense correspondence field between the input depth image and a generic face model. Both outputs are a result of minimizing the error in reconstructing the depth image, achieved by applying a set of identity and expression blend shapes to the model. Traditionally, such a generative approach has shown to be computationally expensive and non-robust because of the non-linear nature of the reconstruction error. To overcome this problem, we use a discriminatively trained prediction pipeline that employs random forests to generate an initial dense but noisy correspondence field. Our method then exploits a fast ICP-like approximation to update these correspondences, allowing us to quickly obtain a robust initial fit of our model. The model parameters are then fine tuned to minimize the true reconstruction error using a stochastic optimization technique. The correspondence field resulting from our hybrid generative-discriminative pipeline is accurate and useful for a variety of applications such as mesh deformation and retexturing. Our method works in real-time on a single depth image i.e. without temporal tracking, is free from per-user calibration, and works in low-light conditions.

1 Introduction

We address the problem of reconstructing 3D face shape and expressions in real-time, given only a single depth image. As with the Kinect [36], we are motivated by interactive gaming scenarios, in our case face retargeting or retexturing, where often users will play in low-lighting conditions where color data is unavailable or limited. Our method provides a per-frame estimate of the 3D face shape and expressions using depth data only, avoiding any temporal information or tracking (which is often prone to errors during large motions). It also avoids per-user calibration, which can be a costly step in prior systems.

Per-frame, our method computes a dense correspondence field between an input depth image and a canonical face model in real-time. We fit a deformable face model, parameterized by a set of identity and expression blend shapes, to the data. Minimizing the error in reconstructing the face in the observed depth image lets us estimate the true parameters of the face model and in turn the dense correspondence field. Traditionally this generative approach has been shown to be computationally expensive and non-robust because of the non-linear nature of the reconstruction error. To overcome this problem, we use a discriminatively trained prediction pipeline which provides a robust initial solution. The correspondences are then updated using a variant of the iterative closest point (ICP) algorithm to get an initial fit and then further refined by minimizing the true reconstruction error using particle swarm optimization (PSO).

Our discriminative pipeline first estimates an initial set of correspondences between the deformable model of the face and the data using random forests. Previous methods for computing dense correspondences for deforming objects use the structure of a classification tree for constructing the regression forest [37, 35]. However, we find that employing a joint classification and regression objective [21] leads to more accurate correspondences. The correspondence field resulting from our hybrid generative-discriminative (Figure 1) pipeline is accurate and useful for a variety of applications such as mesh deformation and retexturing.

Related work: Early work on facial tracking and model fitting typically used monocular RGB video sequences and tracked the motion of *sparse* 2D facial features or triangulated 3D points across frames [3, 29]. Approaches typically used parametric 2D or 3D shape models, which were matched against these sparse correspondences in the video sequence. Approaches are therefore typically generative, but some adopt discriminative methods for feature detection. Early work on facial tracking used variants of active appearance models [12] for parameterizing the face in 2D. Whilst powerful for the initial detection of the face, these 2D linear approaches fail to model complex motions or large deformations of the face. [4] proposed a morphable 3D parametrization for the face, which has been adopted as a richer representation in more recent work.

[10, 5, 39, 14] also extract blend shape parameters but from a sparse set of visually tracked landmarks, to fit 3D morphable models to video sequences. They demonstrate a variety of video editing tasks such as facial animation transfer and

face replacement. Kemelmacher-Shlizerman et al. [27] and Li et al. [32] propose purely discriminative approaches, which use sparse feature tracks and matching to retrieve a 3D face model from a single RGB image. [19] use a pipeline that combines sparse feature matching, blend shape estimation, and dense geometry reconstruction (using optical flow and a shading based refinement step) to demonstrate impressive 3D facial reconstructions from a single monocular sequence.

The RGB systems so far are non real-time in terms of performance. [44] use a coarse 3D morphable model in combination with a 2D active appearance model and sparse features for real-time facial tracking in video. More recent work has shown how regression forests can learn to find a sparse set of facial features in real-time [15, 26]. In [8], the 3D positions of facial landmark points are inferred by a regressor from 2D video frames of an off-the-shelf web camera or mobile phone. From these 3D points, the pose and expressions of the face are recovered by fitting a user-specific 3D morphable model.

In the computer graphics community, facial tracking and modeling has received much attention. Here algorithms aim at dense detailed facial capture for performances. Given the desire for high-quality, complex multi-camera and motion capture rigs, costly scanner systems, custom lighting and studio conditions are required [34]. Multi-camera rigs have been used to track markers or find dense correspondences using invisible make-up [43, 18, 22]. [24] combines marker-based motion capture with high quality 3D scanning for detailed capture of facial expressions. Other dense 3D methods, track shape templates from a dynamic active 3D scanner [42, 40], including non-facial shapes [30]. Whilst these methods exploit the dense depth data only, they rely on very high quality input for robust tracking and estimation. Our method works with commodity but noisy depth cameras. High-quality facial performances have also been demonstrated with passive stereo camera setups [7, 2, 38]. All these dense approaches produce high-quality results, but most require complex, expensive setups and high computational costs.

With the advent of consumer depth cameras, many real-time head pose, facial tracking and modeling pipelines have been proposed [41, 31, 6, 17]. Whilst demonstrating impressive results, these real-time methods rely on both 2D sparse RGB features and depth data. The RGB data is typically used to increase robustness, given the noisy depth data. As such, these methods are limited to visible lighting conditions, and are non-robust to extreme changes in illumination. Further, all these systems use a personalized blend shape model, which requires either online or offline per-user calibration. In contrast, recent work in full body pose estimation [36] has freed itself from the RGB constraints by considering only depth images. The seminal work uses a random forest to rapidly label the identity of every depth pixel [36]. Our work is most similar to [37] that instead uses a forest to predict a dense set of correspondences back to a canonical human body model. The model is then fit to make the corresponding model and data points agree, but no update to the correspondences is considered. The work [35] demonstrated that improved pose accuracy can be obtained by updating the correspondences in addition to the model parameters. We take a similar approach for the discriminative part of our

pipeline, with the advantage that our method is more efficient and operates in real-time. Note that we additionally employ PSO for tracking and further refinement. Our contributions can be therefore summarized as follows:

1. We present a unified framework for dense correspondence estimation, and facial shape and expression reconstruction. Our method operates in real-time, requires only depth data, and reconstructs each frame independently. Therefore our method is not prone to failures due to fast motion, is largely invariant to different lighting conditions, and enables new interactive scenarios. Our method also avoids expensive per-user calibration steps, and uses only a generic face model for fitting.
2. Quantitative and qualitative results are presented verifying that our estimated correspondence field is accurate enough for facial shape and expression reconstruction, and retexturing in real-time.
3. In contrast to both [37] and [35] which have used similar approach for human pose estimation, we directly minimize a true measure of reconstruction error with PSO while still maintaining real-time speeds.
4. We demonstrate that using a classification objective only in the upper levels of tree training helps with the multimodality of the correspondence distributions.

2 The Generative Model

We will use $O = \{z_n\}_{n\in\mathcal{I}}$ to represent the observed depth image where z_n is the depth of pixel n in the set $\mathcal{I}$ of image pixels. Similarly, we will use $f(\theta) = \{\hat{z}_n(\theta)\}_{n\in\mathcal{I}}$ to represent the depth of the pixels in the image rendered from the face model (see below) with parameters θ. Given the observed depth image O, the posterior distribution over the parameters θ of the face model is then defined as

$$\Pr(\theta|I) \propto \exp(-E(O, f(\theta))) \tag{1}$$

where E is the reconstruction error which measures the distances between the observation and rendered image under the parameters θ. The Maximum a Posteriori configuration of the face model parameters can be computed by solving the inverse problem:

$$\theta* = \arg\min_{\theta\in\Theta} E(O, f(\theta)) \ . \tag{2}$$

For the remainder of this manuscript, we will make the dependence of the energy on O and f implicit and simply write $E(\theta)$.

Parameterizing and rendering the 3D face model: We use a blend shape based model for synthesizing 3D faces. This model is able to account for variation in 3D structure caused by both the identity of the user as well as his or her expression. In this model, a base mesh $\{v^b_m\}^M_{m=1}$ consisting of M vertices is deformed using a linear combination of N_{identity} identity blend shapes and $N_{\text{expression}}$ expression blend shapes. The i'th identity blend shape contains a 3D offset v^s_{mi} for each vertex m

and likewise the i'th expression blend shape contains an offset v^e_{mi} for vertex m. A set of coefficients $\{\alpha_i\}_{i=1}^{N_{\text{identity}}}$ determine how much of the identity blend shape i to add to the base mesh. Similarly the set $\{\beta_i\}_{i=1}^{N_{\text{expression}}}$ determine the same for the expression blend shapes.

Having specified the face model, we now describe how we use it to render an image that can be matched with the given observations. We first generate a deformed mesh $\{v_m\}_{m=1}^{M}$ from the face model by applying the weighted vertex offsets to the base mesh, and applying a global scaling s through

$$v_m = s\left(v^b_m + \sum_{i=1}^{N_{\text{identity}}} \alpha_i v^s_{mi} + \sum_{i=1}^{N_{\text{expression}}} \beta_i v^e_{mi}\right) . \tag{3}$$

The deformed mesh is then positioned in 3D using a rotation $R \in SO(3)$ and translation t generate a set of $3D$ points $\{p_m\}_{m=1}^{M}$ via

$$p_m = Rv_m + t . \tag{4}$$

In total, the parameter vector θ of our model is simply the concatenation of the identity blend shape weights $\{\alpha_i\}_{i=1}^{N_{\text{identity}}}$, the expression blend shape weights $\{\beta_i\}_{i=1}^{N_{\text{expression}}}$, global scale s, rotation R and translation t.

The observations $f(\theta) = \{\hat{z}_n(\theta)\}_{n\in\mathcal{I}}$ ultimately take the form of a rendered image that is produced by sweeping over each image pixel index $n \in \mathcal{I}$ and calculating a depth $\hat{z}_n(\theta)$. If the pixel index back projects to a point within the bounds of our model (i.e. into the convex hull of the positioned vertex set $\{p_m\}_{m=1}^{M}$), we simply employ standard graphics techniques to render the depth $\hat{z}_n(\theta)$. We denote this set of foreground pixel indices as $\mathcal{I}_{\text{fg}} \subseteq \mathcal{I}$. For a pixel n in the background $\mathcal{I}_{\text{bg}} = \mathcal{I} - \mathcal{I}_{\text{fg}}$, we simply render a fixed background depth $\hat{z}_n(\theta) = 5000mm$ far behind the mesh to simulate a wall.

The golden energy: We now describe the reconstruction error that implicitly encodes the likelihood of seeing an observed image given the model parameters θ. Given a depth image $\{z_n\}_{n\in\mathcal{I}}$, we assume that pixels within the foreground come from our generative model, whereas pixels in the background are not likely to. We thus use a truncated L-1 difference between the rendered and the observed image as the reconstruction error:

$$E_{gold}(\theta) = \sum_{n\in\ \mathcal{I}} \min(|z_n - \hat{z}_n(\theta)|, \zeta) \tag{5}$$

We refer to this error as the 'golden' energy of the model parameters, as it represents how well the model, under parameters θ, fits the observed image modulo the known deficiencies of our model (e.g. our naive constant background model).

Substituting equation 5 into 2, we get the model fitting optimization problem:

$$\theta* = \arg\min_{\theta\in\Theta} \sum_{n\in\ \mathcal{I}} \min(|z_n - \hat{z}_n(\theta)|, \zeta). \tag{6}$$

This is a hard non-linear optimization problem as it has numerous local minima and even locally differentiating it is non-trivial [16]. One way to handle such problems is to use a derivative free optimizer with a good initial guess. For this we employ the PSO method [28] that works by evolving a population of P particles (i.e. solutions) $\{\theta_1, ..., \theta_P\}$. The rules for updating these particles are standard and we refer the reader to [28] for more details. Briefly though, the swarm's movement is designed to strike a balance between global exploration of the parameter space and local exploitation of the collective knowledge that it has obtained from each particle's evaluation. In theory, PSO is capable of performing a robust global optimization when enough particles are allowed to evolve for sufficiently many generations, however, doing so is prohibitive for a real-time application. It is thus crucial that we have a good initial guess and that we only use PSO to perform a fast local derivative free optimization of (5). We thus return to the use of PSO in section 4 and now consider an alternative energy which we can use to obtain such a good initial guess.

The silver energy: To this end, we back project the foreground depth pixels to obtain a 3D data point cloud $\{x_n\}_{n=1}^{N_{fg}}$, where N_{fg} is the number of such pixels. We assume that each data point x_n is a noisy observation of a point $S(u;\theta)$ on the surface of our model. Here $u \in \Omega$ is a coordinate (*i.e.*a triangle index and barycentric coordinate) in the (2D) surface domain Ω of our surface. Assuming a Gaussian noise model, this allows us to define a new energy based on the distance from each observation to the models surface as

$$E_{silver}(\theta) = \sum_{n=1}^{N_{fg}} \min_{u \in \Omega} \|S(u;\theta) - x_n\|^2 \ . \tag{7}$$

By naming, for each data point x_n, a corresponding model coordinate u_n we can pass the inner minimizations through the summation and rewrite this as

$$E_{silver}(\theta) = \min_{u_1,...,u_{N_{fg}}} \sum_{n=1}^{N_{fg}} \|S(u_n;\theta) - x_n\|^2 \ . \tag{8}$$

This allows us to define yet another energy

$$E'_{silver}(\theta, U) = \sum_{n=1}^{N_{fg}} \|S(u_n;\theta) - x_n\|^2 \tag{9}$$

defined both on the block of parameters θ and a block of surface coordinates $U = \{u_1, ..., u_{N_{fg}}\} \subseteq \Omega$. Importantly $E_{silver}(\theta) = \min_U E'_{silver}(\theta, U) \leq E'_{silver}(\theta, U)$ for any U, and thus we can approach minimizing (7) by minimizing (9). When $S(u;\theta)$ is differentiable and there is a procedure available for calculating $u^*(x;\theta) = \arg\min_{u \in \Omega} \|S(u;\theta) - x\|^2$, one can perform coordinate descent on the θ and U to obtain a classical iterative closest point method [13].

In our case, we use the M vertices of our mesh to provide a discretization $\Omega' = \{u'_1, ..., u'_M\}$ of Ω where $S(u'_m; \theta) = p_m$ is well defined from (4). We then desire to minimize (9) but use a set of surface coordinates U' restricted to this discretization (i.e. $U' \subseteq \Omega'$). Importantly,

$$E_{silver}(\theta) = \min_{U \subseteq \Omega} E'_{silver}(\theta, U) \leq \min_{U' \subseteq \Omega'} E'_{silver}(\theta, U') \tag{10}$$

where the first bound gets tighter as we optimize for U and the second bound can be made very tight using a dense enough mesh. This is the case for our discretization where $M = 11211$, a number close to the typical number of pixels on the face.

To minimize $E'_{silver}(\theta, U')$ we observe that our restriction allows us to satisfy both properties needed to craft a classical iterative closest point algorithm. The function $S(u'; \theta)$ for fixed $u' \in U'$ defined by (4) and (3) has well defined derivatives that are straightforward to compute. The function $u'^*(x; \theta) = \arg\min_{u' \in \Omega'} \|S(u'; \theta) - x\|^2$ is now approachable by, for example, iterating over the M possible values in Ω'. We show in the next subsection, how we obtain a good initial guess for U' and now provide more details about how we efficiently perform coordinate descent on $E'_{silver}(\theta, U')$.

Our procedure for optimizing over θ while holding U' fixed exploits the availability of derivatives and the squared error terms in the energy. This allows us to exploit the Gauss-Newton approximation $J(\theta)^t J(\theta)$ of the Hessian $H(\theta)$, where $J(\theta)$ is the Jacobian, and perform powerful second order Gauss-Newton step $\theta_{k+1} = (J(\theta_k)^t J(\theta_k))^{-1} J(\theta_k)^t r(\theta_k)$ where $r(\theta_k)$ is the vector of residuals at step k. We use the publicly available Ceres implementation [1] of the popular Levenberg-Marquardt variant [33] that simply damps the $J(\theta_k)^t J(\theta_k)$ matrix when the quadratic approximation fails to yield a good step. This variant combines the advantage of quadratic convergence when the quadratic approximation is valid (e.g. provably so near local minima) with a graceful degradation to first order gradient descent when the approximation fails to allow progress to be made.

Our procedure for optimizing over U' while holding θ' fixed is carefully designed to maintain real-time speeds. Indeed, the naive method of calculating $u^*(x; \theta) = \arg\min_{u' \in \Omega'} \|S(u'; \theta) - x\|^2$ by iterating linearly over the elements of Ω' results in an algorithm with a $O(NM)$ complexity. It has been suggested [35] to use a KD-Tree to reduce the complexity to $O(NlogM)$, but it is not obvious how to implement the tree construction at real-time speeds. In fact, as we are searching over model points dependent on θ, and not data points which are independent of θ, one has to construct a new KD-tree at every iteration. In order to obtain real-time speeds, we instead perform a simple but effective local approximation that is easily parallelizable. We rely on the GPU to quickly render our model into 3D and only search for rendered vertices in a small local neighbourhood that back projects from a rectangular patch surrounding each depth pixel.

3 Discriminative Model

Our discriminative model consists of a random forest of binary decisions trees. For each pixel in the input depth image, the forest predicts its corresponding position $u' \in \Omega'$ on the canonical face model. This approach is similar to [37] that has been applied to body pose estimation task. To train their forest, Taylor *et al.* use a surrogate classification objective based on body parts, which has been shown to achieve higher accuracies than a pure regression objective in [20]. In this paper, we show that a hybrid objective yields better results than both a pure classification and a pure regression objective for our application.

The decisions that each split node of the trees make are based on the simple depth-invariant depth comparison features ($\mathbf{f}_\phi$) proposed in [36]. Although extremely lightweight to compute, these features have been shown to be powerful for a variety of tasks [36, 37, 20]. At each node, our training algorithm processes a sample set Q as follows:

1. A pool of features $\phi = \{\phi_i\}_{i=1}^{|\phi|}$ is randomly selected.
2. For each feature ϕ_i, a set of candidate thresholds $\{\tau_{ij}\}_{j=1}^{|\tau|}$ is selected.
3. For each set of split parameters $z = (\phi, \tau)$, samples Q are divided into left and right partitions: $Q_l(z) = \{Q : \mathbf{f}_\phi < \tau\}$ and $Q_r(z) = Q \setminus Q_l(z)$.
4. The optimal parameter z^* is chosen to maximize the information gain ($G(z)$)

$$\mathcal{G}(z) = \mathcal{H}(Q) - \sum_{s \in (l,r)} \frac{|Q_s(z)|}{|Q|} \mathcal{H}(Q_s(z)) \tag{11}$$

$$\mathcal{H}(Q) = \alpha \mathcal{H}^* + (1 - \alpha)\mathcal{H}^\dagger \tag{12}$$

with $\alpha = \mathbb{1}(\mathit{depth} \leq L)$. In the above, L indicates the depth at which we switch the objective from that of classification to regression. The classification objective is based on the Shannon entropy defined over part classes whereas the regression objective is simply a measure of correspondence variation

$$\mathcal{H}^*(Q) = -\sum_{c \in \mathit{classes}} P(c) \log P(c) \tag{13}$$

$$\mathcal{H}^\dagger(Q) = \mathrm{Tr}(\Lambda(Q)) \ . \tag{14}$$

In the above, $P(c)$ is the proportion of samples in Q with class label c, and $\Lambda(Q)$ is the covariance of the regression target labels in Q.

5. If the appropriate information gain $\mathcal{G}(z^*)$ can be made sufficiently large, the split is accepted. The algorithm then continues recursively down the right and left branches until a maximum depth is achieved or the sample set in a node becomes small.

Algorithm 1 Pseudo-code of our hybrid single frame model fitting and correspondence finding procedure. The same algorithm is used for both identity and expression fitting by setting $N_{\text{expression}}$ or N_{identity} to zero respectively. In the latter case, the base mesh is assumed to have been morphed to incorporate the identity.

Initialize scalars $\{\alpha_i\}_{i=1}^{N_{\text{identity}}}$ and $\{\beta_i\}_{i=1}^{N_{\text{expression}}}$ to zero.
Evaluate forest on depth image to obtain initial U'.
Solve for optimal R, t, s holding everything else fixed.
for $i = 1$ to N_{ICP} **do**
  Optimize $\{\alpha_i\}_{i=1}^{N_{\text{identity}}}$ and $\{\beta_i\}_{i=1}^{N_{\text{expression}}}$, R, s and t using LM.
  Update U' using closest point approximation.
end for
Initialize PSO by sampling near current solution.
for $i = 1$ to N_{PSO} **do**
  Evolve PSO Swarm.
  Update U' using closest point approximation.
end for

Lastly, we use the mean-shift algorithm [11] on the empirical distribution that ends up in each leaf to find the modes of the distribution. At test time, each decision tree is traversed based on its selected features until a leaf node is reached and the set of all modes found by mean shift are aggregated. The final output of the forest is the correspondence u' closest to the strongest mode in this aggregation.

4 Hybrid Method

We now return to our original task of minimizing the golden energy (5) to recover both the model parameters and a good set of correspondences. Although this energy is difficult to optimize directly, we have now developed the necessary tools in the previous section to develop our hybrid method that can rapidly obtain a good minimum.

We start by detecting the head using a standard skeleton tracker [36] and removing the outliers by simple distance thresholding. Then, the algorithm, described in Algorithm 1, leverages our discriminative correspondence to obtain a good initial guess for the correspondences U' used in our proxy generative model described by the silver energy. We set all of the blend shape weights to zero and solve simultaneously for an initial optimal global scale, translation and rotation [23]. We then perform N_{ICP} iterations of an iterative closest point algorithm by alternating between a continuous optimization of θ and a discrete update of U' as described in Section 2 to get close to a local minimum of (7). Note that, in practice, we add a small regularizer $\lambda \sum_{i=1}^{N_{\text{expression}}} \rho(\beta_i)$ to (7) [1] which we find helps condition the

[1] Here ρ is the Huber error functional [25]

optimization. We then sample near the current solution to construct a population of PSO particles, which rapidly refines the solution locally to drive down (5).

Our algorithm can be used in two different modes of operation: identity fitting and expression fitting. In identity fitting, only identity blend shapes are used (i.e. $N_{\text{expression}} = 0$) to fit the shape of the user in a neutral pose. These identity blend shapes can then be incorporated into the base mesh by setting $v_m^b \leftarrow v_m^b + \sum_{i=1}^{N_{\text{identity}}} \alpha_i v_m^s$ for $m \in \{1, ..., M\}$. The algorithm, can then be switched to expression fitting mode where only expression blend shapes are used (i.e. $N_{\text{identity}} = 0$) as the base mesh has been fit to the identity of the user.

5 Experiments

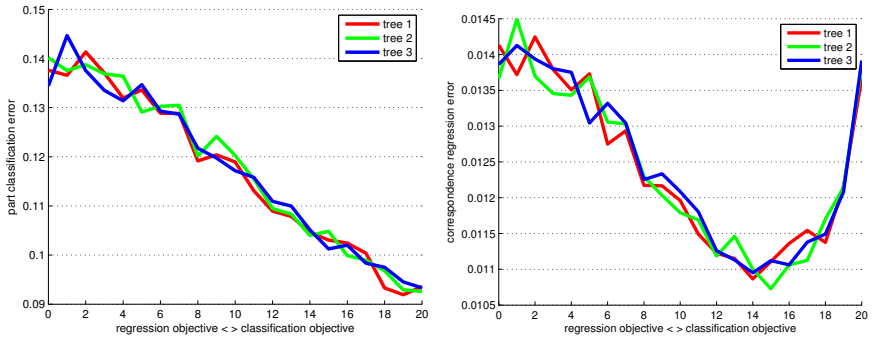
This section details a set of experiments that we have performed to evaluate different components of our system individually and as a whole. We use synthetic data to train and test our system, and also provide qualitative generalization results on real depth images 5.

Evaluation of correspondence prediction: To train our discriminative model, we use third-party software to generate synthetic images. For each image, we randomize the model parameters between reasonable limits and render a synthetic depth image, an image with part annotations and an image that encodes ground truth correspondences. We synthesize $10,000$ images of size 320×240 and sample 2000 pixels from each to train the random forest. Features at each split node are selected from a pool of 5000 random features. Our final forest consists of 3 trees of depth 20.

As described in Section 3, we train our random forest with a combined objective (12), which includes a classification term for the coarser part labels and a regression term for the finer correspondence labels. Intuitively, the signal that the classification objective provides, helps regularize the tree in the initial levels, which implicitly isolates the multiple modes of the distribution of the regression labels. This in turn makes the regression objective more effective in the deeper levels of the tree. Figure 2 shows that the hybrid objective function performs better than using a pure classification or pure regression objective.

Evaluation of model fitting strategy: Our generative model uses a total of 50 blend shapes to represent the face. These include $N_{\text{identity}} = 40$ and $N_{\text{expression}} = 10$. The silver and golden energy described in Section 2 allows us to evaluate the fit of our generative model to the observed depth data. In addition, we are specifically interested in the inference of the expression weights $\{\beta_i\}_{i=1}^{N_{\text{expression}}}$ and data model correspondences $U' = \{u'_i\}_{n=1}^{N_{fg}}$. In a single image, we measure the expression error as

$$e_{expression} = \sum_{i=1}^{N_{\text{expression}}} (\beta_i - \beta_i^{\text{gt}})^2 \ . \tag{15}$$



(a) Part classification error (b) Correspondence error

Figure 2: Effect on classification (a) and correspondence regression error (b) resulting from varying the switching depth L. Note that naturally using a pure part classification objective ($L = 20$) does lead to better classification but as we desire to minimize correspondence error, we should switch objectives for the last few levels ($L = 15$).

For a single foreground pixel, we measure the correspondence error as

$$e_{correspondence} = \|S(u;\theta_0) - S(u^{gt};\theta_0)\| \tag{16}$$

where θ_0 is simply the parameter setting that yields the undeformed base mesh model.

Evaluation of silver energy optimization: We begin by demonstrating how optimizing the proxy objective (9) allows us to rapidly reduce the error in (7), the silver energy. This is demonstrated in panel (a) of Figure 3, where we can see a significant decrease as the first ICP step corrects the errors in the forest predictions (see panel (c)). As further ICP steps are taken, the model parameters slowly adjust in tandem so that more accurate correspondences can be acquired (see panel (c)). As expected, there is high correlation between the silver and golden energies, and we manage to greatly decrease the golden energy by minimizing the proxy silver energy, which can be seen in panel (b). Not surprisingly, a better fit of our model also allows us to acquire more accurate expressions as shown in panel (d). Again, the key result here is that by optimizing (9) we are able to simultaneously drive down all relevant energies and errors.

Evaluation of golden energy optimization: We now analyze the ability of our complete hybrid method that refines the result of the previous section by directly optimizing the golden energy with PSO. This is summarized in Figure 4, where it can be seen that 10 iterations of PSO brings us substantially further down in energy. In panel (c) it can be seen that the expression error actually increases after 10 iterations of PSO. In panel (d), we see that the expression error lowers again if we continue to 100 iterations. This is not unexpected behavior, as the

	forehead	eye	temple	cheek	ear	nose	mouth	jaw	average
RF	14.453	9.274	12.871	9.575	14.687	9.634	11.141	9.551	10.554
ICP 1	5.086	4.637	4.934	5.037	4.546	5.413	6.017	5.227	5.079
ICP 5	3.824	3.761	4.268	4.215	3.747	4.676	5.261	4.613	4.299
PSO 1	3.729	3.673	4.252	4.120	3.803	4.476	5.145	4.705	4.260
PSO 5	3.600	3.560	4.266	3.965	3.829	4.190	4.828	4.704	4.151
PSO 10	3.580	3.528	4.308	3.926	3.825	4.134	4.732	4.667	4.115

Table 1: Average correspondence error over facial parts at different stages of the pipeline. The error is given in millimeters.

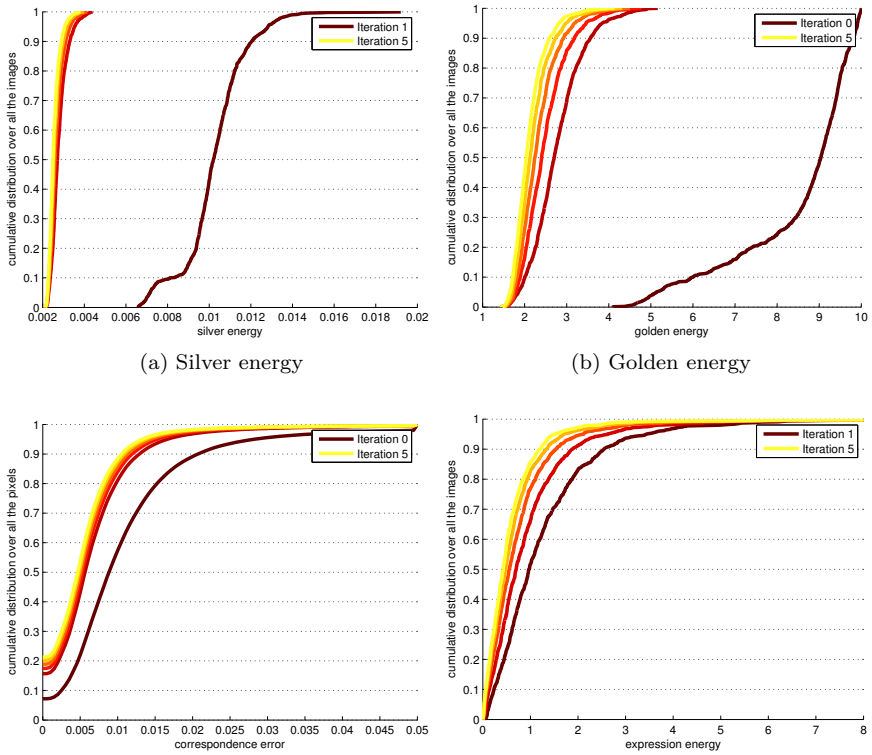
blend shape regularization contained in the silver energy is not contained in the golden energy. Although it is helpful to quickly get to a stable and robust result, PSO takes a bit of time to undo this overfitting. Interestingly, a better fit to the data only translates into a moderately better correspondence error (panel (b)) but closer inspection shows that the error actually does drop significantly in key regions of the face. This is demonstrated in Table 1 shows that the correspondence error in the important mouth and nose regions is reduced considerably.

Qualitative results on real data: We demonstrate qualitative results on real data in Figure 5 and supplementary material. In panel (e) we can see the reconstructed face model and in panel (d) the final set of correspondences. To demonstrate the accuracy of the correspondence field, we use our method to extract textures from users' faces and also paint on these to create visual effects. This can be done in real-time ($>25Hz$), as shown in the performance section of the supplementary material and accompanying video.

6 Conclusion

We presented a real-time algorithm for fitting a complex but generic face model to a single depth image of an arbitrary face. In addition to the fit model, we also are able to infer the expression weights and a dense data-model correspondence field. Our system is real-time allowing 3D facial shape and expressions to be used for interactive scenarios such as retargeting and retexturing. In addition, we demonstrate empirically how the various components of our algorithm drive down the "golden energy", which we argue is the natural energy to minimize. We show that this energy is highly correlated with the other two quantities that we are interested in estimating, namely the correspondence error and expression error. Unlike other related methods, our method relies only on per-frame depth, avoiding tracking failures due to fast motions, working in low-lighting conditions, and removing the need for per-user calibration. Moreover, our discriminative pipeline estimates a dense correspondence field, making it more robust than methods that rely on a small number of landmarks which can easily be occluded.

Naturally, our method is only robust to moderate occlusions. This is due to our use of a truncated loss function in the final optimization of the golden energy and due to the locality of forest features. The latter helps the forest provide a "good



(a) Silver energy (b) Golden energy

(c) Corr. error (d) Expression error

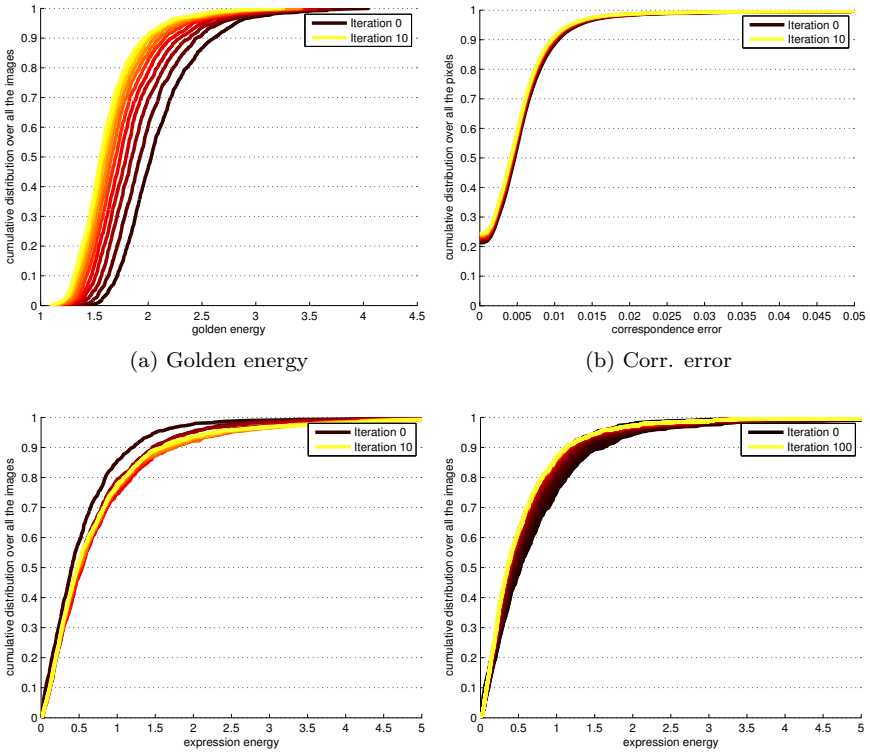
Figure 3: Cumulative distribution of (a) silver energy, (b) golden energy, (c) correspondence error, and (d) expression error at different iterations of ICP. ICP procedure reduces all the error measures while optimizing over the silver energy.

enough" initialization to the (largely local) optimization of the former. Larger occlusions are not currently handled, but could be alleviated by synthesizing occluded faces for training. Additionally, as with other machine learning techniques, we are of course limited by the expressiveness of our model and the variety of our training data. Improving the richness of our model and handling occlusions remains interesting areas of future work.

References

[1] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `https://code.google.com/p/ceres-solver/`.

[2] Thabo Beeler, Fabian Hahn, Derek Bradley, Bernd Bickel, Paul Beardsley, Craig



(a) Golden energy (b) Corr. error

(c) Expression error (d) Exp. error (100)

Figure 4: Cumulative distribution of (a) golden energy, (b) correspondence error, (c) expression error at different iterations of PSO, and (d) expression error up to 100 iterations of PSO.

Gotsman, Robert W Sumner, and Markus Gross. High-quality passive facial performance capture using anchor frames. In *ACM Transactions on Graphics*, 2011.

[3] Michael J Black and Yaser Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of the International Conference on Computer Vision*, 1995.

[4] V. Blanz and T. Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of SIGGRAPH*, 1999.

[5] Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. Reanimating faces in images and video. In *Computer graphics forum*, 2003.

[6] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Transactions on Graphics*, 2013.

[7] Derek Bradley, Wolfgang Heidrich, Tiberiu Popa, and Alla Sheffer. High resolution passive facial performance capture. *ACM Transactions on Graphics*, 2010.

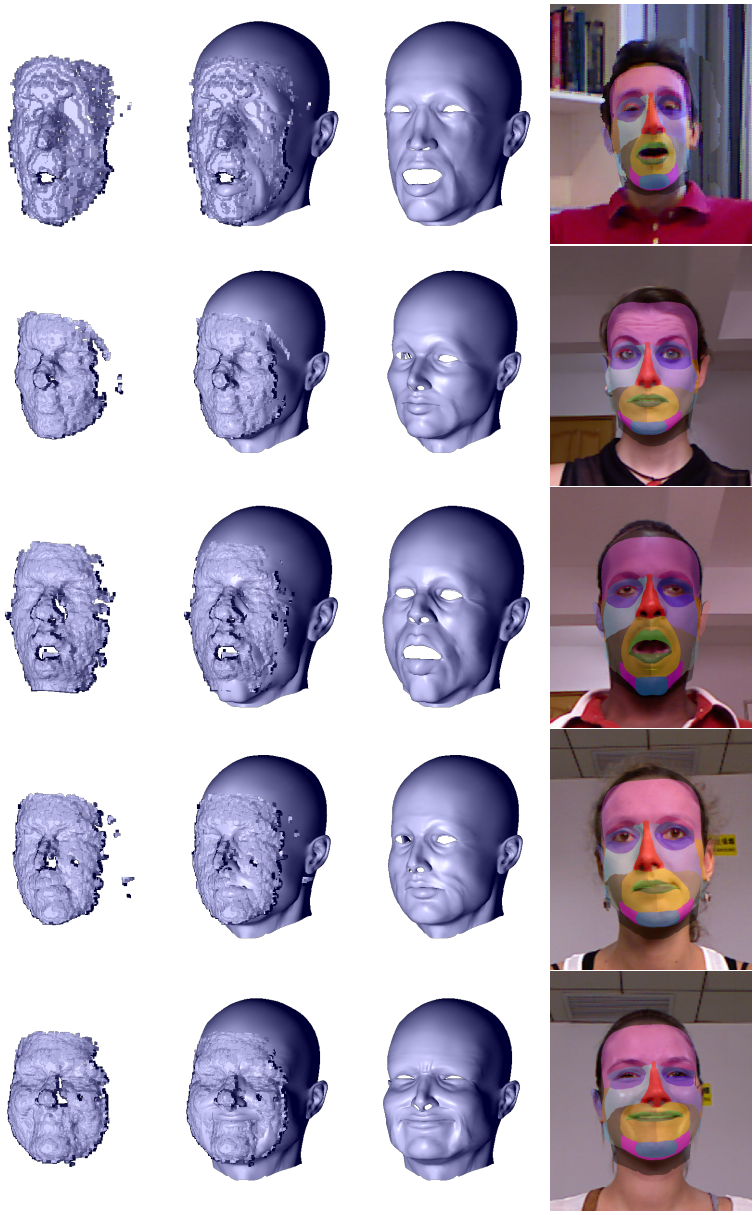


Figure 5: Qualitative results on real data captured using Kinect camera. From left to right, we show the input depth data, the depth data overlaid on the reconstructed model, the reconstructed model and the parts overlaid on the rgb image (which was only used for visualization). Some of these examples are from [9].

[8] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics*, 2013.

[9] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: a 3d facial expression database for visual computing. 2013.

[10] Jinxiang Chai, Jing Xiao, and Jessica Hodgins. Vision-based control of 3d facial animation. In *Computer Animation*, 2003.

[11] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002.

[12] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, 1998.

[13] Stefano Corazza, Lars Mündermann, Emiliano Gambaretto, Giancarlo Ferrigno, and Thomas P Andriacchi. Markerless motion capture through visual hull, articulated icp and subject specific model generation. *International Journal of Computer Vision*, 2010.

[14] Kevin Dale, Kalyan Sunkavalli, Micah K Johnson, Daniel Vlasic, Wojciech Matusik, and Hanspeter Pfister. Video face replacement. In *ACM Transactions on Graphics*, 2011.

[15] M. Dantone, J. Gall, G. Fanelli, and L. J. Van Gool. Real-time facial feature detection using conditional regression forests. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2012.

[16] Amaël Delaunoy and Emmanuel Prados. Gradient flows for optimizing triangular mesh-based surfaces: Applications to 3d reconstruction problems dealing with visibility. *International Journal of Computer Vision*, 2011.

[17] Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *IJCV*, 2013.

[18] Yasutaka Furukawa and Jean Ponce. Dense 3d motion capture for human faces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2009.

[19] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Transactions on Graphics*, 2013.

[20] R. B. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. W. Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *Proceedings of the International Conference on Computer Vision*, 2011.

[21] Ben Glocker, Olivier Pauly, Ender Konukoglu, and Antonio Criminisi. Joint classification-regression forests for spatially structured multi-object segmentation. In *Proceedings of the European Conference on Computer Vision*. Springer, 2012.

[22] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Fredric Pighin. Making faces. In *ACM Transactions on Graphics*, 1998.

[23] Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 1987.

[24] Haoda Huang, Jinxiang Chai, Xin Tong, and Hsiang-Tao Wu. Leveraging motion capture and 3d scanning for high-fidelity facial performance acquisition. *ACM Transactions on Graphics*, 2011.

[25] Peter J Huber *et al.* Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 1964.

[26] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2014.

[27] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[28] James Kennedy, Russell Eberhart, *et al.* Particle swarm optimization. In *IEEE Neural Networks*, 1995.

[29] Haibo Li, Pertti Roivainen, and Robert Forchheimer. 3-d motion estimation in model-based facial image coding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1993.

[30] Hao Li, Bart Adams, Leonidas J Guibas, and Mark Pauly. Robust single-view geometry and motion reconstruction. *ACM Transactions on Graphics*, 2009.

[31] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. 2013.

[32] Kai Li, Feng Xu, Jue Wang, Qionghai Dai, and Yebin Liu. A data-driven approach for facial expression synthesis in video. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2012.

[33] Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial & Applied Mathematics*, 1963.

[34] F Pighin and JP Lewis. Performance-driven facial animation. In *ACM Transactions on Graphics*, 2006.

[35] Gerard Pons-Moll, Jonathan Taylor, Jamie Shotton, Aaron Hertzmann, and Andrew Fitzgibbon. Metric regression forests for human pose estimation. In *Proceedings of the British Machine Vision Conference*, 2013.

[36] 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 *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2011.

[37] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The Vitruvian Manifold: Inferring dense correspondences for one-shot human pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2012.

[38] Levi Valgaerts, Chenglei Wu, Andrés Bruhn, Hans-Peter Seidel, and Christian Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics*, 2012.

[39] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. In *ACM Transactions on Graphics*, 2005.

[40] Yang Wang, Xiaolei Huang, Chan-Su Lee, Song Zhang, Zhiguo Li, Dimitris Samaras, Dimitris Metaxas, Ahmed Elgammal, and Peisen Huang. High resolution acquisition, learning and transfer of dynamic 3-d facial expressions. In *Computer Graphics Forum*, 2004.

[41] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics*, 2011.

[42] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. Face/off: Live facial puppetry. In *Computer Animation*. ACM, 2009.

[43] Lance Williams. Performance-driven facial animation. 1990.

[44] Jing Xiao, Simon Baker, Iain Matthews, and Takeo Kanade. Real-time combined 2d+3d active appearance models. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2004.