Int J Comput Vis (2009) 83: 72–84
DOI 10.1007/s11263-009-0218-0

Real-Time Body Pose Recognition Using 2D or 3D Haarlets

Michael Van den Bergh · Esther Koller-Meier · Luc Van Gool

Received: 25 April 2008 / Accepted: 21 January 2009 / Published online: 21 February 2009

Abstract This article presents a novel approach to markerless real-time pose recognition in a multicamera setup. Body pose is retrieved using example-based classification based on Haar wavelet-like features to allow for real-time pose recognition. Average Neighborhood Margin Maximization (ANMM) is introduced as a powerful new technique to train Haar-like features. The rotation invariant approach is implemented for both 2D classification based on silhouettes, and 3D classification based on visual hulls.

Keywords Pose estimation · Pose recognition · Silhouettes · 3D hulls · LDA · ANMM · Haarlets

1 Introduction

Posture recognition has received a significant amount of attention given its importance for human-computer interfaces, teleconferencing, surveillance, safety control, animation, and several other applications. The context of this work is a virtual reality system where the user walks on an omnidirectional treadmill, and can interact with the virtual world using body pose commands. For this application a markerless pose detection subsystem has to be fast and robust for detecting a predefined selection of poses.

We present an example-based technique for real-time markerless rotation-invariant pose recognition using Average Neighborhood Margin Maximization (ANMM) (Wang and Zhang 2007) and Haar wavelet-like features (Viola and Jones 2001). The latter will be called *Haarlets* from now on for brevity.

The setup consists of a 4 m × 4 m working space on which the user can walk, and several cameras placed around this working space. We propose both a 2D system based on silhouettes of the user, which can work with 1 or more cameras, and a 3D system based on visual hulls, which works with multiple cameras. Silhouettes are extracted based on color (Griesser et al. 2005). The visual hulls are extracted based on these silhouettes and using voxel carving and a fixed lookup table (Kehl et al. 2005).

In example-based approaches, observations are compared and matched against stored examples of human body poses. This is done here in real-time using Haarlets. ANMM is introduced as a powerful approach to train these Haarlets. We will show that using ANMM yields higher performance than Linear Discriminant Analysis (LDA) (Van den Bergh et al. 2008), and better performance than AdaBoost (Viola and Jones 2001), which can only train 2D Haarlets. Where classic AdaBoost runs into memory issues when training 3D rather than 2D Haarlets (Ke et al. 2005), the weakened memory requirements of ANMM allow for a straightforward implementation of a 3D pose detector based on 3D Haarlets. The benefit of classifying 3D hulls rather than silhouettes, is that the orientation of the hulls can be normalized in a straightforward manner. We evaluate the 2D and

M. Van den Bergh (✉) · L. Van Gool
Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland
e-mail: vamichae@vision.ee.ethz.ch

L. Van Gool
e-mail: vangool@vision.ee.ethz.ch

E. Koller-Meier · L. Van Gool
ESAT-PSI/VISICS, Katholieke Universiteit Leuven, Leuven, Belgium

E. Koller-Meier
e-mail: ebmeier@vision.ee.ethz.ch

L. Van Gool
e-mail: Luc.VanGool@esat.kuleuven.be

3D based systems for performance and complexity of the system setup.

2 Background

Example-based The introduction above contains a number of choices that have been made. The first is one in favour of example-based rather than model-based techniques. Model-based approaches typically rely on articulated 3D body models (Bregler and Malik 1998; Delamarre and Faugeras 1999; Gavrila and Davis 1996; Kakadiaris and Metaxas 2000; Papageorgiou et al. 1998; Sminchisescu and Triggs 2003; Yamamoto et al. 1998). In order to be effective they need to have a high number of degrees of freedom, in combination with non-linear, anatomical constraints. Consequently, they require time-consuming per-frame optimization and the resulting trackers are too slow for real-time approaches. They are also very sensitive to fast motions and segmentation errors. To relieve the speed problem 2D model-based approaches have been proposed. Baumberg and Hogg (1994) use active shape models to track pedestrians, however pose is not recovered. Ioffe and Forsyth (2001) infer likely body configurations using a tree model based on candidate body parts and feature points to perform coarse 2D tracking in a single camera. Still, these systems remain only near real-time.

In the example-based approaches, instead of tracking articulated body models, observations are compared and matched against stored examples of human body poses. Mori and Malik (2002) propose a technique where example 2D views of human body poses are stored together with manually marked and labeled positions of body joints. Poses can then be recovered using shape context matching. Rosales and Sclaroff (2000) train a neural network to map example 2D silhouettes to 2D positions of body joints. Shakhnarovich et al. (2003) outline a framework for fast pose recognition using parameter sensitive hashing. In their framework image features such as edge maps, vector responses of filters and edge direction histograms can be used to match silhouettes against examples in a database.

These example-based methods benefit from the fact that the set of typical or interesting poses is far smaller than the set of anatomically possible ones, which is good for robustness. Also, not needing an explicit parametric body model makes them more amenable to real-time implementation and application to the pose analysis of other structures than human bodies, e.g. animals.

Silhouettes and Visual Hulls Silhouettes and their derived visual hulls seem to capture the essence of human body poses well. The model-based approach proposed by Delamarre and Faugeras (1999), as well as the example-based method proposed by Rosales and Sclaroff (2000), are other examples of silhouette-based approaches, however they are not real-time. There are also some examples of methods based on 3D hulls of the human body. Mikíc et al. (2001) and Cheung et al. (2003) propose model-based tracking approaches using 3D voxel reconstructions, but they are not real-time either. Cohen and Li (2003) propose a near-real-time example-based approach where 3D hulls are matched by a support vector machine (SVM).

LDA and Haarlets As the survey by Yang et al. (2002) points out, LDA provides superior performance to SVMs in many vision tasks. Belhumeur et al. (1997) have proven LDA to be superior to principal component analysis (PCA) for a similar task of face recognition, as can be expected given that LDA is a kind of refined PCA. LDA is frequently used in face recognition (Belhumeur et al. 1997; Yang et al. 2002; Zhao et al. 1998), but to the best of our knowledge it has not yet been applied to pose recognition. Wang and Zhang (2007) present ANMM as a variation of LDA which has higher performance and has fewer limitations.

Moreover, ANMM, which is rather slow, lends itself well to combine its strength with the speed of Haarlets. Indeed, fast integral image based Haarlets can be used to approximate the ANMM components. Haarlets were introduced by Papageorgiou et al. (1998), and Ren et al. (2005) trained Haarlets for pose recognition using AdaBoost. Our approach, which approximates ANMM features with Haarlets, provides a multi-class alternative to AdaBoost. As it can deal with a much larger number of candidate Haarlets in the training set, our method can also be extended to 3D, volumetric Haarlets for the classification of 3D voxel hulls. Of those there are many more than of the 2D Haarlet type. The main advantages of switching to 3D are rotation invariance and increased robustness. In 3D, the speed advantage of the Haarlet approximation becomes even more apparent and is crucial to keep the system real-time.

3 Classification

In example-based approaches, observations are compared and matched against stored examples of human body poses. In the 2D approach we explain in this article, these observations consist of silhouettes of the user. These silhouettes are extracted from videos of several fixed cameras around the person. To extract the silhouettes from the camera views, we use the background subtraction algorithm by Griesser et al. (2005). Some examples of such silhouettes are shown in Fig. 1. The extracted silhouettes are normalized to a fixed resolution and position, by cutting a square bounding box around the top and bottom pixels of the silhouette, and centered horizontally to the center of gravity of the silhouette.

Therefore, it is possible for the user to change position in the scene, without significantly affecting the resulting silhouettes. The images containing the silhouettes from the different camera views are concatenated to one single image, which the classifier can process, as illustrated in Fig. 2.

The 3D approach aims to classify poses based on 3D hulls of the user, rather than silhouettes. Several cameras are placed around the person. Any number of cameras can be chosen, but it is best to deploy sufficient cameras to make a good 3D voxel reconstruction. Using background subtraction (Griesser et al. 2005) the silhouettes are extracted from each camera view. These silhouettes are then used for the 3D voxel reconstruction based on the method proposed by Kehl et al. (2005). A lookup table (LUT) is created to map each pixel in each camera view to a projection into the voxel space. A voxel carving technique then generates the reconstructed hull. These resulting hulls are normalized to a fixed resolution, rotation and position, which allows for the subject to not only change the position, but also the orientation. An example of such a 3D hull is shown in Fig. 3.

In Fig. 4 the basic classifier structure is shown, where T denotes a transformation which is found using Average Neighborhood Margin Maximization (ANMM) (Wang and Zhang 2007). This transformation projects the input samples (silhouettes or hulls) onto a lower dimensional space where the different pose classes are maximally separated and easier to classify. Using a nearest neighbors (NN) approach these projected samples are matched to stored poses in a database and the closest match is the output of the system. In order to improve the speed of the system, the transformation T can be approximated using *Haarlets*, which will be explained in Sect. 4.

In this classifier, each sample is classified independently from the previous ones. The training samples are divided into different pose classes. Depending on the 2D or 3D case, the training examples consist of silhouettes or hulls. The pixel, respectively voxel values of these silhouettes or hulls are stored in an n-dimensional vector, where n is the total number of pixels, respectively voxels in the input sample. The goal of the training step is to find a linear transformation T which will project the input samples onto a lower dimensional space where they are maximally separated.

Fig. 1 Examples of silhouettes which are used for classification. Note the holes in the segmentation and the artifacts due to reflections on the floor

3.1 Linear Discriminant Analysis (LDA)

The idea is to find a linear transformation such that the classes are maximally separable after the transformation (Fukunaga 1990). The class separability can be measured by the ratio of the determinant of the between-class scatter matrix S_B and the within-class scatter matrix S_W. The optimal projection W_{opt} is chosen as the transformation that maximizes the ratio,

$$W_{opt} = \arg\max_W \frac{|WS_BW^T|}{|WS_WW^T|}, \tag{1}$$

and is determined by calculating the generalized eigenvectors of S_B and S_W. Therefore,

$$W_{opt}^T = [\mathbf{w}_1\mathbf{w}_2\ldots\mathbf{w}_m], \tag{2}$$

where $\mathbf{w}_i$ are the generalized eigenvectors of S_B and S_W corresponding to the m largest generalized eigenvalues λ_i. The eigenvalues represent the weights of the different eigenvectors, and are stored in a diagonal matrix D, while the eigenvectors $\mathbf{w}_i$ represent characteristic features of the different pose classes.

A solution for the optimization problem in (1) is to compute the inverse of S_W and solve an eigenproblem for the matrix $S_W^{-1}S_B$ (Fukunaga 1990). Unfortunately S_W will be singular in most cases, because the number of training examples is smaller than the number of dimensions in the sample vector, and thus inverting S_W will be problematic. There are several solutions proposed to circumvent this small sample size problem, such as direct LDA (Yang et al. 2000), but they don't yield a significant performance increase over LDA.

3.2 Average Neighborhood Margin Maximization (ANMM)

LDA aims to pull apart the class means while compacting the classes themselves. This introduces the small sample

Fig. 2 Example of 3 camera views, foreground-background segmentation, and their concatenation to a single normalized sample

Fig. 3 Example of a reconstructed 3D hull of the user

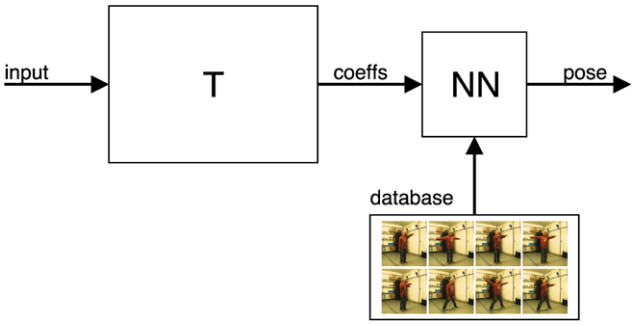


Fig. 4 Basic classifier structure. The input samples (concatenated silhouettes or 3D hulls) are projected with transformation T onto a lower dimensional space, and the resulting coefficients are matched to poses in the database using nearest neighbors (NN)

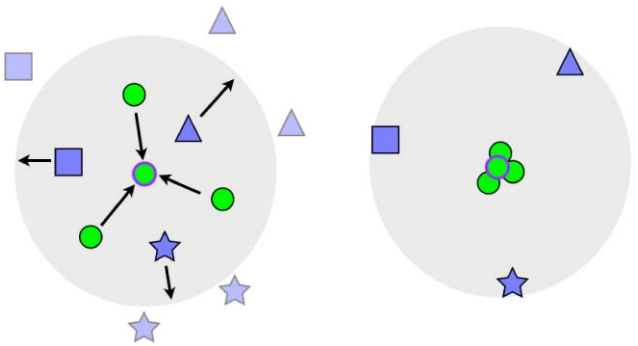


Fig. 5 An illustration of how ANMM works. For each sample, within a neighborhood (marked in *gray*), samples of the same class are pulled towards it, while samples of a different class are pushed away, as shown in the *left*. The figure on the *right* shows the data distribution in the projected space

size problem which renders the within-class scatter matrix singular. Furthermore LDA can only extract $c-1$ features (where c is the number of classes), which is suboptimal for many applications.

ANMM, as proposed by Wang and Zhang (2007), is a similar approach which avoids these limitations. For each data point, ANMM aims to pull the neighboring points with the same class label towards it as near as possible, while simultaneously pushing the neighboring points with different labels away from it as far as possible. This principle is illustrated in Fig. 5.

Instead of using the between-class scatter matrix S_B and the within-class scatter matrix S_W, ANMM defines a *scatterness matrix* as,

$$\mathbf{S} = \sum_{i,k:\mathbf{x}_k \in \mathcal{N}_i^e} \frac{(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_j)^T}{|\mathcal{N}_i^e|}, \tag{3}$$

and a *compactness matrix* as,

$$\mathbf{C} = \sum_{j:\mathbf{x}_j \in N_i^o} \frac{(\mathbf{x}_i - \mathbf{x}_k)(\mathbf{x}_i - \mathbf{x}_j)^T}{|\mathcal{N}_i^o|}, \tag{4}$$

where $\mathcal{N}_i^o$ is the set of the n most similar data which are in the same class as $\mathbf{x}_i$ (n nearest homogeneous neighborhood) and where $\mathcal{N}_i^e$ is the set of the n most similar data which are in a different class than $\mathbf{x}_i$ (n nearest heterogeneous neighborhood). The ANMM eigenvectors W_{opt} can then be found by the eigenvalue decomposition of $\mathbf{S}-\mathbf{C}$.

ANMM introduces 3 main benefits compared to traditional LDA: it avoids the small sample size problem since it does not need to compute any matrix inverse; it can find the discriminant directions without assuming a particular form of class densities (LDA assumes a Gaussian form); and finally much more than $c-1$ feature dimensions are available. Some examples of resulting ANMM eigenvectors are shown in Fig. 6.

4 Haarlet Approximation

In order to improve the speed of the system, the ANMM transformation can be approximated using *Haarlets*, as shown in Fig. 7. In this case the transformation T is approximated by a linear combination C of Haarlets. An optimal set of Haarlets is selected during the training stage and stored. Computing this stored set of features on the input image results in a number of coefficients. Transforming these coefficients with C results in new coefficients, which approximate the coefficients which would result from the transformation T on the same input data, and subsequently can be used for classification in the same manner as in the pure ANMM case.

Haarlets are very popular for real-time object detection and real-time classification. The ANMM approximation approach provides a new and powerful method for selecting or training Haarlets. Especially in the 3D case, where existing methods fail because of the large amount of candidate Haarlets (Ke et al. 2005), our approach makes it possible to train 3D Haarlets selecting from the full set of candidates.

4.1 2D Haarlets

Papageorgiou et al. (1998) proposed a framework for object detection based on Haarlets, which can be computed

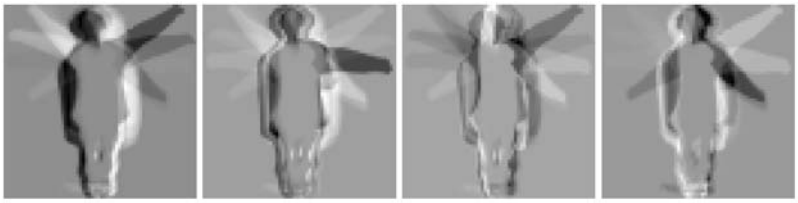


Fig. 6 The first 4 eigenvectors for the frontal view only, after training for a 12 pose set, using the ANMM algorithm

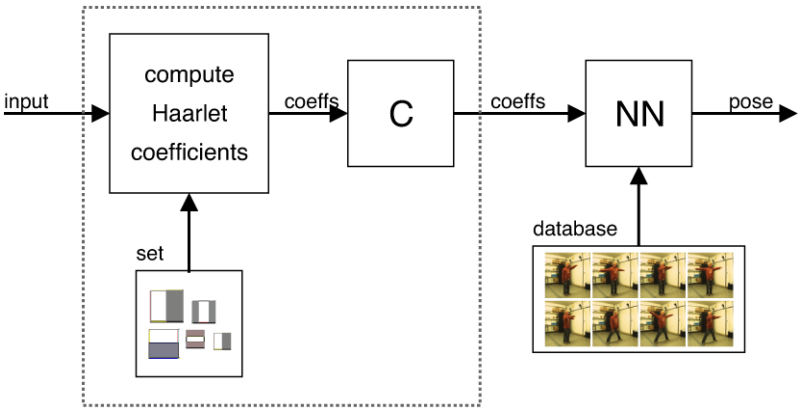


Fig. 7 Classifier structure illustrating the Haarlet approximation. The pre-trained set of Haarlets are computed on the input sample (silhouette or hull). The approximated coefficients are computed as a linear combination C of the Haarlet coefficients. The contents of the *dotted line box* constitute an approximation of T in Fig. 4

Fig. 8 The set of possible Haar-like feature types

with a minimum of memory accesses and CPU operations using the integral image. Viola and Jones (2001) used AdaBoost to select suitable Haarlets for object detection. The same approach was used for pose recognition by Ren et al. (2005). In our approach similar Haarlets are used, however we introduce a new selection process based on ANMM. The Haarlets are selected to approximate W_{opt} as a linear combination thereof. The particular set of Haarlets used here, was carefully selected by Lienhart and Maydt (2002) and is shown in Fig. 8.

Besides the feature type, the other parameters are width, height and position in the image. All combinations are considered. At a resolution of 24×24 pixels and using 3 cameras views, there are over a million candidate Haarlets. The best Haarlets are obtained from this set by convolving all candidate Haarlets with the vectors in W_{opt} and selecting those with the highest coefficients, i.e. the highest response magnitudes. This score is found for each candidate Haarlet by calculating the dot product of that Haarlet with each ANMM vector (each row in W_{opt}), and calculating the weighted sum using the weights of those ANMM vectors, as stored in the diagonal matrix D (i.e. the eigenvalues serve as weights). Thus, the entire ANMM eigenspace is approximated as a whole, giving dimensions with a higher weight higher priority when selecting Haarlets. This dot product can be computed very efficiently using the integral image.

However, most selected Haarlets will be redundant unless W_{opt} is adapted after each new Haarlet is selected before choosing the next. Let F be a matrix containing the already selected Haarlets in vector form, where each row of F is a Haarlet. F can be regarded as a basis that spans the feature space that can be represented by the Haarlet vectors selected so far. Basically, in our iterative solution toward the final W_{opt}, we don't want the next W'_{opt} to span space that is already represented by F. Let N be a basis of the null space of F,

$$N = \text{null}(F). \tag{5}$$

N forms a basis that spans everything that is not yet described by F. To obtain the new optimal transformation we project $D \cdot W_{opt}$ onto N, where D is the diagonal matrix containing the weights of the eigenvectors $\mathbf{w}_i$ in W_{opt}.

$$D' \cdot W'_{opt} = D \cdot W_{opt} \cdot N \cdot N^T, \tag{6}$$

or,

$$W'_{opt} = D \cdot D'^{-1} \cdot W_{opt} \cdot N \cdot N^T, \tag{7}$$

where D' is a diagonal matrix containing the new weights λ'_i of the new eigenvectors $\mathbf{w}_i$ in W'_{opt},

$$\lambda'_i = \left\| \lambda_i \cdot \mathbf{w}_i \cdot N \cdot N^T \right\|. \tag{8}$$

Every time a new Haarlet is selected based on W'_{opt}, F is updated accordingly and the whole process is iterated until the desired number of Haarlets is selected. Examples of selected Haarlets are shown in Fig. 9.

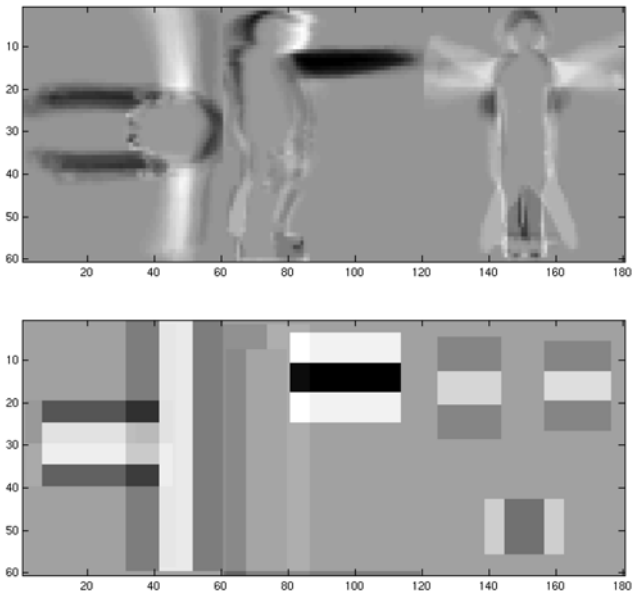


Fig. 9 The *top* figure shows one ANMM vector, featuring the overhead, profile and frontal views side by side. The *bottom* figure shows the Haarlet approximation of this ANMM vector, using the 10 best Haarlets selected to approximate W_{opt}. It can be seen how the Haarlets look for arms and legs in certain areas of the image

After the ANMM vectors have been computed and the Haarlets have been selected to approximate them, the next step is to actually classify new silhouettes. This process uses the Haarlets to extract coefficients from the normalized silhouette image, and then computes a linear combination of these coefficients to approximate the coefficients that would result from the ANMM transformation. An example of such an approximated ANMM feature vector is shown in Fig. 9. The resulting coefficients can be used to classify the pose of the silhouette. Given the coefficients $\mathbf{h}$ extracted with the Haarlets, the approximated ANMM coefficients $\mathbf{l}$ can be computed as

$$\mathbf{l} = C \cdot \mathbf{h}, \tag{9}$$

where C is an $m \times l$ matrix where m is the number of ANMM eigenvectors and l is the number of Haarlets used for the approximation. C can be obtained as the least squares solution to the system

$$W_{opt} = C \cdot F^T. \tag{10}$$

The least squares solution to this problem yields

$$C = W_{opt} \cdot \left(\left(F^T F \right)^{-1} F^T \right)^T. \tag{11}$$

C provides a linear transformation of the feature coefficients $\mathbf{h}$ to a typically smaller number of ANMM coefficients $\mathbf{l}$. This allows for the samples to be classified directly based on these ANMM coefficients, whereas an AdaBoost-based method needs to be complemented with a detector cascade (Viola and Jones 2001), or with a hashing function (Ren et al. 2005). Finally, using nearest neighbors search, the new silhouettes can be matched to the stored examples, i.e., the mean coefficients of each class.

4.2 Introduction of 3D Haarlets

The concepts of an integral image and Haarlets can be extended to three dimensions. The 3D integral image, or integral volume, is defined as,

$$ii(x, y, z) = \sum_{x' \le x, y' \le y, z' \le z} i(x', y', z'). \tag{12}$$

Using the integral volume, any rectangular box sum can be computed in 8 array references as shown in Fig. 10. Accordingly, the integral volume makes it possible to construct

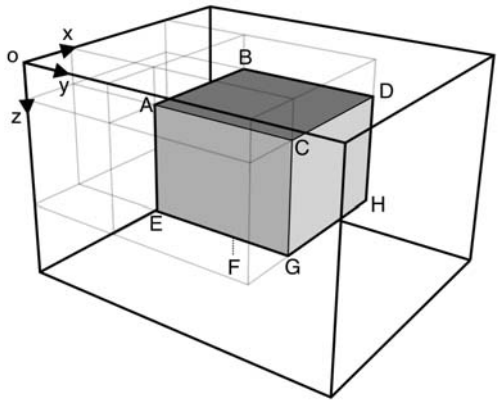


Fig. 10 The sum of the voxels within the gray cuboid can be computed with 8 array references. If A, B, C, D, E, F, G and H are the integral volumes at shown locations, the sum can be computed as (B + C + E + H) − (A + D + F + G)

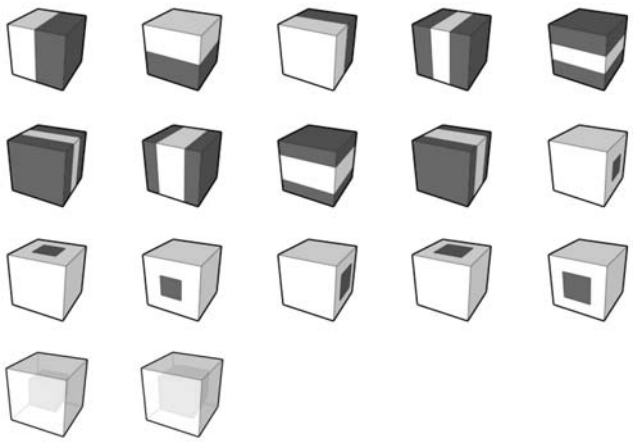


Fig. 11 The proposed 3D Haarlets. The first 15 features are extruded versions of the original 2D Haarlets in all 3 directions, and the other 2 are true 3D center-surround features

Table 1 Outline of the training algorithm for the 3D approach

Input: training set of 3D hulls separated into pose classes.

Output: set of n3D Haarlets, C.

1. Read and normalize the hulls.
2. Apply ANMM on training data to obtain W_{opt} and D. Set initial W'_{opt} and D' to W_{opt} and D respectively.
3. Repeat until n Haarlets are selected:
   - Calculate all candidate features on the vectors in W'_{opt}, weighted with D'.
   - Select the Haarlet with the highest response magnitude, add it to F.
   - Calculate N, null space of F.
   - Update W'_{opt} and D', $D' \cdot W'_{opt} = D \cdot W_{opt} \cdot N \cdot N^T$.
4. Compute approximation C using $C = W_{opt} \cdot ((F^T F)^{-1} F^T)^T$.

Table 2 Outline of the classification algorithm

Input: 3D hull.

Output: pose.

1. Calculate n feature coefficients and put them in vector $\mathbf{h}$.
2. Calculate approximated ANMM feature coefficients $\mathbf{l} = C \cdot \mathbf{h}$
3. Find nearest neighbor match between $\mathbf{l}$ and stored examples.

volumetric box features similar to the Haarlets in Viola and Jones (2001). We introduce the 3D Haarlet set as illustrated in Fig. 11. Besides in feature type, the Haarlets can vary in width, height, depth and position inside the voxel space. At $24 \times 24 \times 24$ resolution, this results in hundreds of millions of candidate features. The Haarlet selection process and approximation are similar to what is explained in the 2D case in Sect. 4.1. For clarity, the process is summarized in Table 1.

The 3D Haarlets require twice as much memory accesses and computations as their 2D counterparts. However, they often contain more information and are more efficient in representing body parts. Whereas the 2D approach would need a Haarlet in each camera view, one 3D Haarlet might be sufficient. Some examples of the approximated ANMM feature vectors using the selected 3D Haarlets are shown in Fig. 12.

After the ANMM transformation is computed and 3D Haarlets are selected to approximate that transformation, the next step is to classify new hull samples. The classification is analogous to the 2D approach. Coefficients are extracted for each 3D feature, and then a linear combination of those coefficients is made to approximate the original ANMM transformation. The outline of the classification algorithm is shown in Table 2.

5 Rotation Invariance

The pose classification problem becomes quite a bit more difficult when the subject can not only change position freely, but also orientation. While a change of position can be normalized easily, in the 2D case it is impossible to normalize the rotation of the subject. In the 3D approach however, it is possible to normalize the rotation of the 3D hulls before classifying them.

The orientation of the user is estimated using a simple overhead tracker. Our visual tracker is based on a color-based particle filter (Nummiaro et al. 2003). The tracker uses a set of particles to model the posterior distribution of the likely state of the subject. During each iteration of the tracker, a set of new hypotheses is generated for the state by propagating the particles using a dynamic model. This generates a prior distribution of the state, which is then tested using the observation of the image. A person is modeled by a circle and an ellipse, representing the head and shoulders respectively. The head is modeled separately to deal with changes in perspective. As the head is closer to the overhead camera than the shoulders, its relative position will change depending on the position of the person, and thus change the appearance of the shoulder/head region in the overhead camera. The color distributions of these two regions are compared to a stored model histogram to yield the likelihood for the state of each particle. An example of the overhead tracker tracking the orientation of a person is shown in Fig. 13.

The tracker is initialized automatically by fitting an ellipse to a silhouette of the user at the beginning of the algorithm. The estimation of the orientation of the person is defined as the angle of the minor axis of this ellipse. During the initialization, the user is assumed to be in the middle of the working volume and facing a certain direction, so that the polarity of the orientation can be determined. The tracker runs at real-time and continuously provides the orientation of the person.

5.1 3D Approach

Normalizing the rotation of the hull consists of measuring the angle of its orientation, and then rotating it to a standard orientation. The goal is that regardless of the orientation of the subject, the resulting normalized hull will look the same, as shown in Fig. 14.

5.2 2D Approach

As the 2D classifier cannot classify the pose of a person with changing orientation as is, it is impossible to compare the 3D directly to the 2D approach. It is however possible to redesign the 2D system to classify different orientations. The

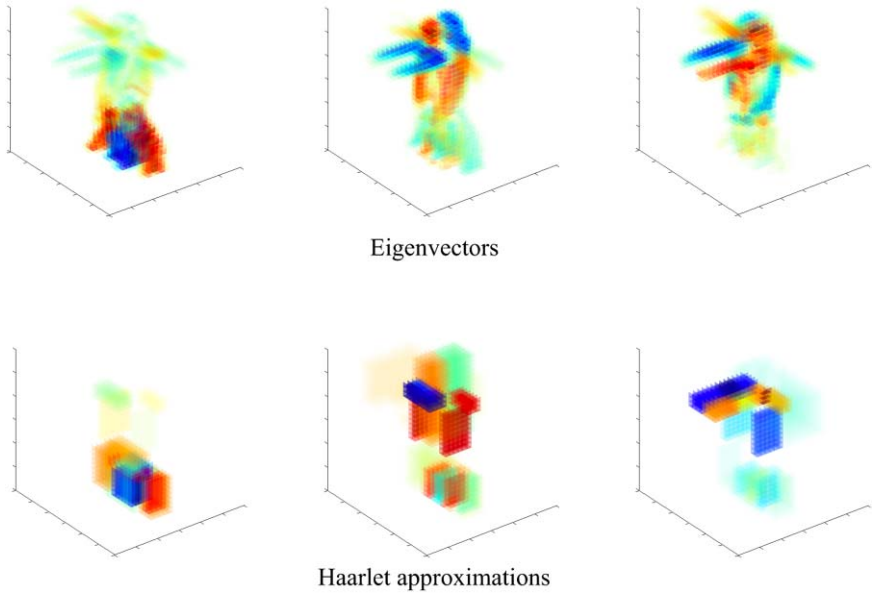


Fig. 12 The first row shows 3 example vectors from an ANMM eigenspace. The second row shows the approximation using not more than 10 Haarlets. The first example shows how a feature is selected to inspect the legs, while the last example shows a feature that distinguishes between left and right arm stretched out forward

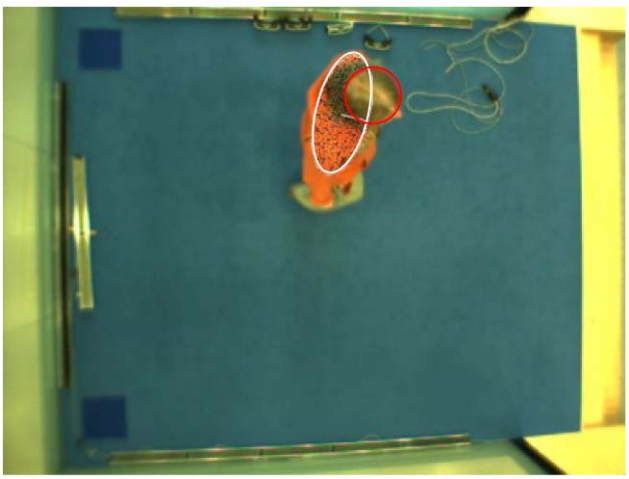


Fig. 13 Example of the tracker tracking the orientation of a person

angle of orientation of the subject can be measured from the top view camera as in Sect. 5.1. Then, the training samples are divided into 36 individual bins, depending on the angle of orientation. For each bin a separate 2D classifier is trained. In the classification stage, depending on the measured angle of orientation, the appropriate 2D classifier is used. This results in a pseudo-rotation invariant implementation.

6 Experiments

Our test setup consists of 6 cameras connected to 6 standard 3 GHz PCs which are placed in a network. One camera is placed overhead, while the other cameras are placed sideways around the working volume. The working area has a cluttered background, and the floor has some reflections, so the resulting hulls contain some noise and holes. Using this setup 6000 samples were recorded of a user in 50 unique poses and in varying orientations. We defined 50 pose classes as shown in Fig. 15. The silhouettes are extracted from the camera views and then the position and size are normalized resulting in silhouettes of 24×24 pixels. The hulls are reconstructed and then the position, size and orientation are normalized resulting in $24 \times 24 \times 24$ voxel hulls. Of these 6000 samples, one third are used for training, and the remaining two thirds are used for validation. Classification is run on a single 3 GHz PC.

In the first experiment we show that ANMM is indeed better than LDA at classifying 3D hulls over a large number of poses. Using up to 50 pose classes, the test hulls are classified using both the LDA and the ANMM approach. When using less than 50 pose classes, a random selection of pose classes is made, and the results are averaged over 5 random samplings of pose classes. The results are shown in Fig. 16. ANMM is more consistent and maintains high correct classification rates of around 97% even when all 50 pose classes are used. The LDA-based approach drops down to 80% correct classification.

In a second experiment we compare classifying 3D hulls to classifying 2D silhouettes, as shown in Fig. 17. The 3D classification is based on 3D hulls which are reconstructed from 6 cameras and results in around 97% correct classification over 50 pose classes. The 2D classification is based on 2D silhouettes which are taken from 6 camera views, and using 36 orientational bins results in around 95% correct classification. Using only 3 camera views, the 2D system achieves around 91% correct classification. The 3D hull-based system indeed has better classification rates than the

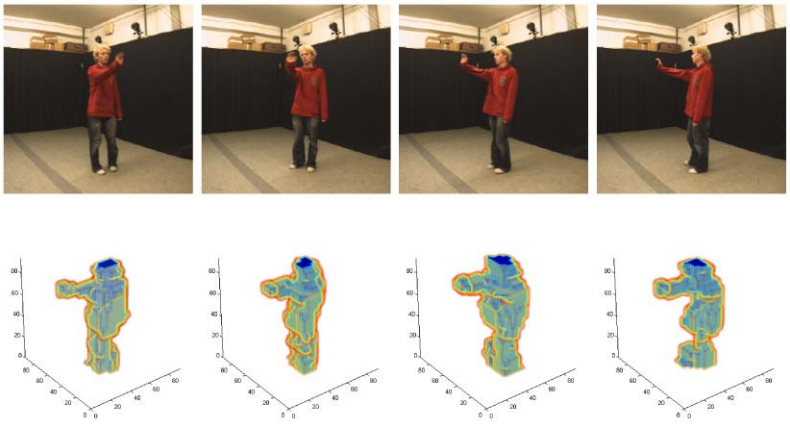


Fig. 14 (*Top*) Examples of different orientations of the user, resulting in (*bottom*) similar rotation-normalized voxel hulls

2D silhouette-based system. However, using less hardware (3 cameras and 1 PC) it is still possible to get high correct classification rates using the 2D silhouette-based approach.

In a third experiment we evaluate how many Haarlets are needed to approximate the ANMM transformation. The results of this experiment are shown in Fig. 18. It shows that the 3D case converges faster than the 2D case, which makes sense as 3D Haarlets contain more information than their 2D counterparts. Both the 2D and 3D cases converge to their optimum with about 100 Haarlets. The 3D approach is able to produce reasonable results with only 15 Haarlets.

In this experiment we also attempted to train a 2D classifier using AdaBoost, using the same training and test data as in the previous experiment. As AdaBoost is a 2-class approach and we are using 50 classes, a trick is applied similar to Ren et al. (2005) to turn the problem into a 2-class problem. The data is rearranged in two classes as pairs of normalized silhouette images: matching pose (positive training examples) and non-matching pose (negative training examples). This results in hundreds of thousands of training examples, so a resampling step is made as in Ren et al. (2005) to reduce the number of training samples in order for all the data for the AdaBoost algorithm to be able to fit in the memory of the computer.

The classification results using the features trained with AdaBoost are not as good as the ANMM-based approach. There are several reasons for this. Firstly, the reductions in the resampling step which are needed to fit the data in memory are rather drastic, discarding a lot of training information. Another issue is that there seems to be overfitting as we increase the number of Haarlets, as the nearest neighbor search becomes too high in dimensionality. In the ANMM case this problem is solved by reducing the number of dimensions by approximating the original ANMM transformation. In Ren et al. (2005) this problem is solved by introducting a hashing function to reduce the number of dimensions in the search space. An alternative might be to use an algorithm that extends the boosting approach to a multi-class setting, which to our knowledge have not been applied to pose classification.

As pointed out by Ke et al. (2005), the above mentioned memory constraint problems render it out of the question to train 3D Haarlets using a method based on AdaBoost.

In the last experiment we demonstrate the speed improvement in using 3D Haarlets to approximate the ANMM transformation. The results of this experiment are shown in Fig. 19. We show how the computation time increases almost linearly for the ANMM transformation as the number of pose classes are increased. This is because increasing the number of pose classes increases the number of ANMM feature vectors almost linearly. Using the ANMM approximation, the integral volume of the hull has to be computed once, after which computing additional Haarlets coefficients requires virtually no computation time (relative to the time of computing the integral volume).

As the classifier is part of a bigger online system, we have to add computation time for segmentation, reconstruction and sending the data over the network. in this case 50 ms for classification in the pure ANMM case is too slow, while the 3 ms Haarlet approximation allows for a real-time implementation of the pipeline.

7 Summary and Conclusion

This work introduced ANMM as a new and powerful approach to training Haarlets for human pose classification. This approach was implemented and tested for pose recognition on 2D silhouettes, and compared to classic AdaBoost. The approach was extended to classifying 3D voxel hulls, and consequently 3D Haarlets were introduced.

First, this article provides a proof-of-concept that Haarlets can be trained using ANMM, introducing interesting advantages as it is a true multi-class approach, and it has virtually no memory restrictions on the resolution or number of candidate Haarlets to train from. It also offers a complete

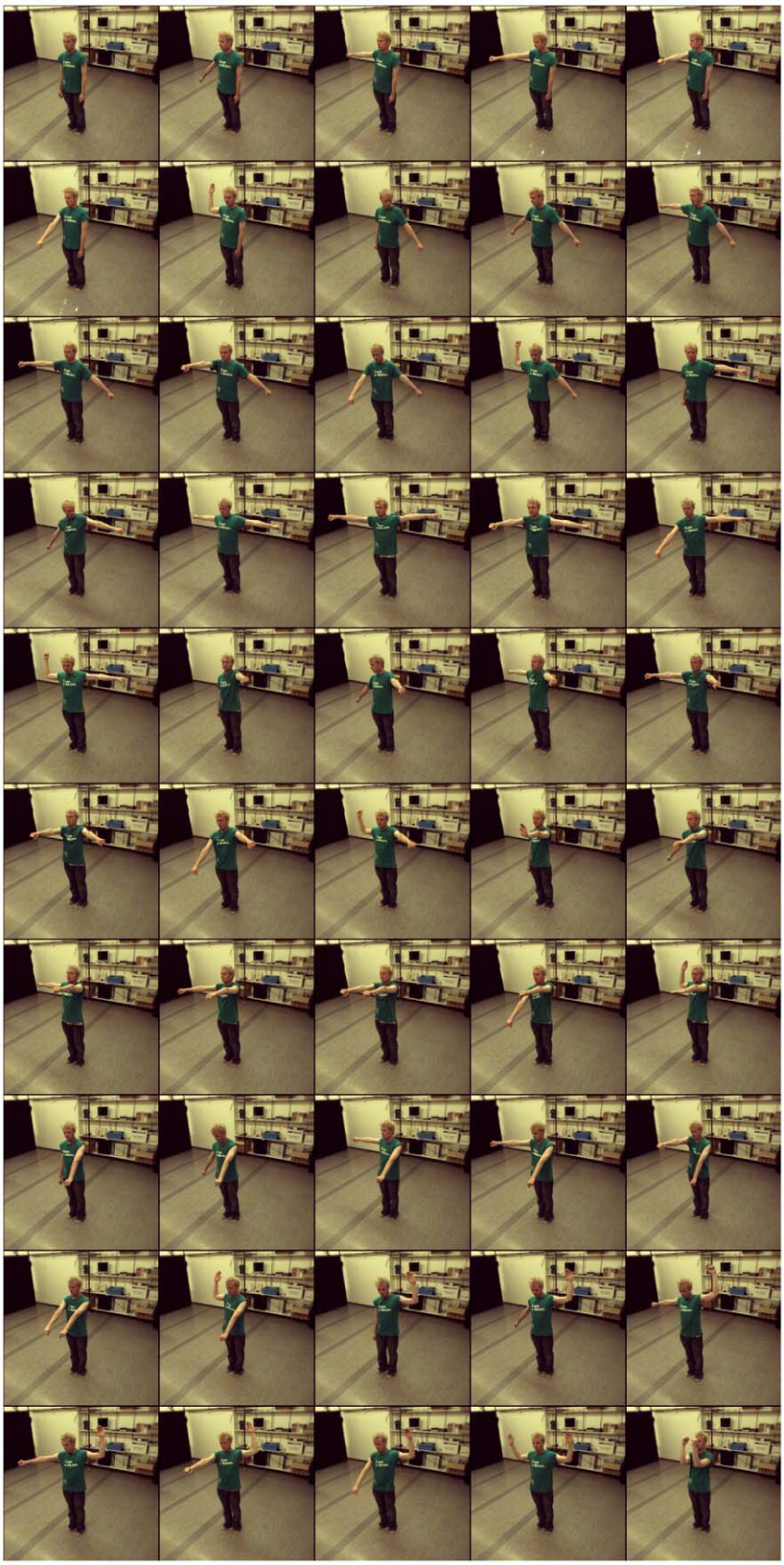


Fig. 15 The 50 pose classes used in this article, differing by arm directions

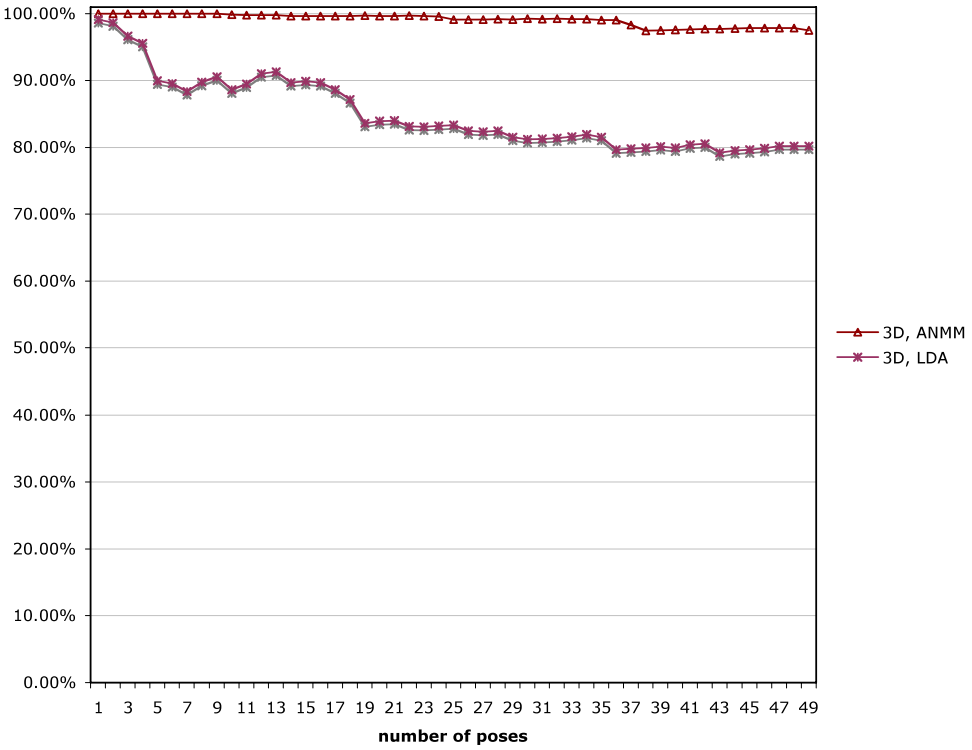


Fig. 16 Correct classification rates comparing LDA and ANMM for the classification of 3D hulls

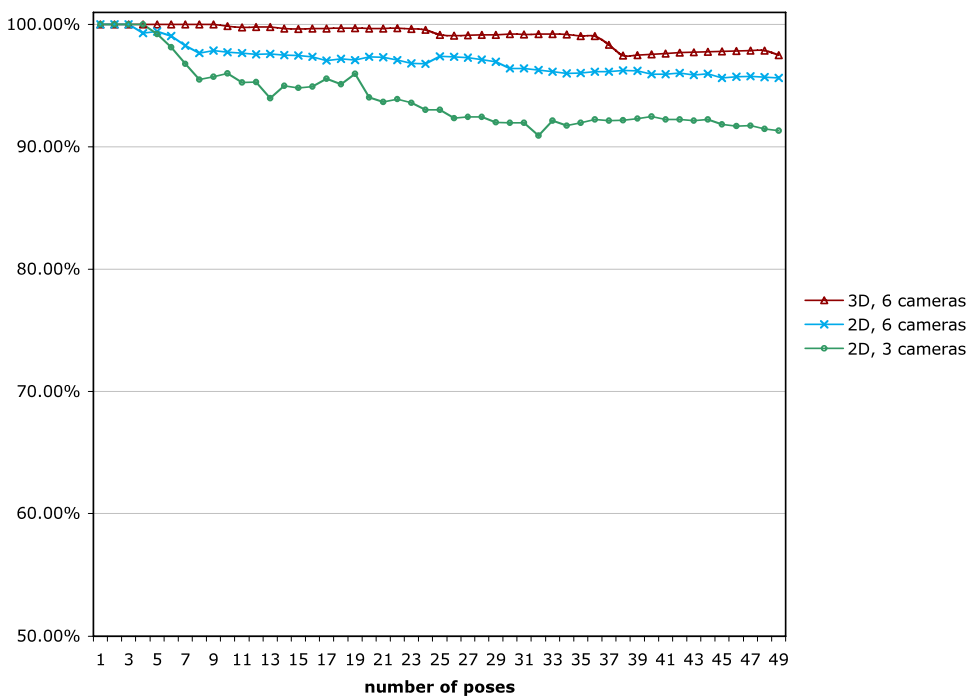


Fig. 17 Correct classification rates comparing classification based on 2D silhouettes and 3D hulls using ANMM

solution: after the Haarlets are selected they are used to approximate the ANMM transformation, which allows for the samples to be classified directly, whereas AdaBoost needs to be complemented with a detector cascade (Viola and Jones 2001), or with a hashing function (Ren et al. 2005). The second contribution of this article is the extension of the ANMM-based algorithm to three dimensions and the introduction of 3D Haarlets for pose recognition. Unlike AdaBoost, training can be based on the full set of candidate 3D Haarlets. The 3D approach has new, interesting properties such as rotation invariance and increased performance. The result is pose classification with at least comparable perfor-

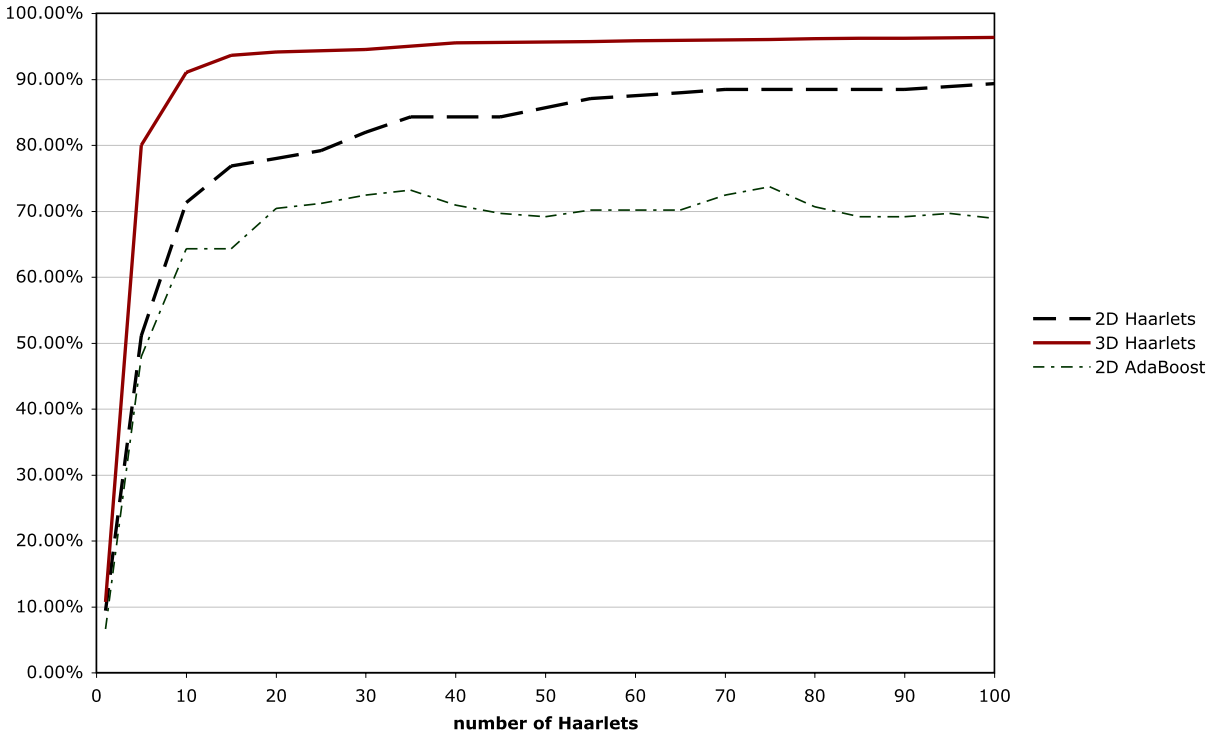
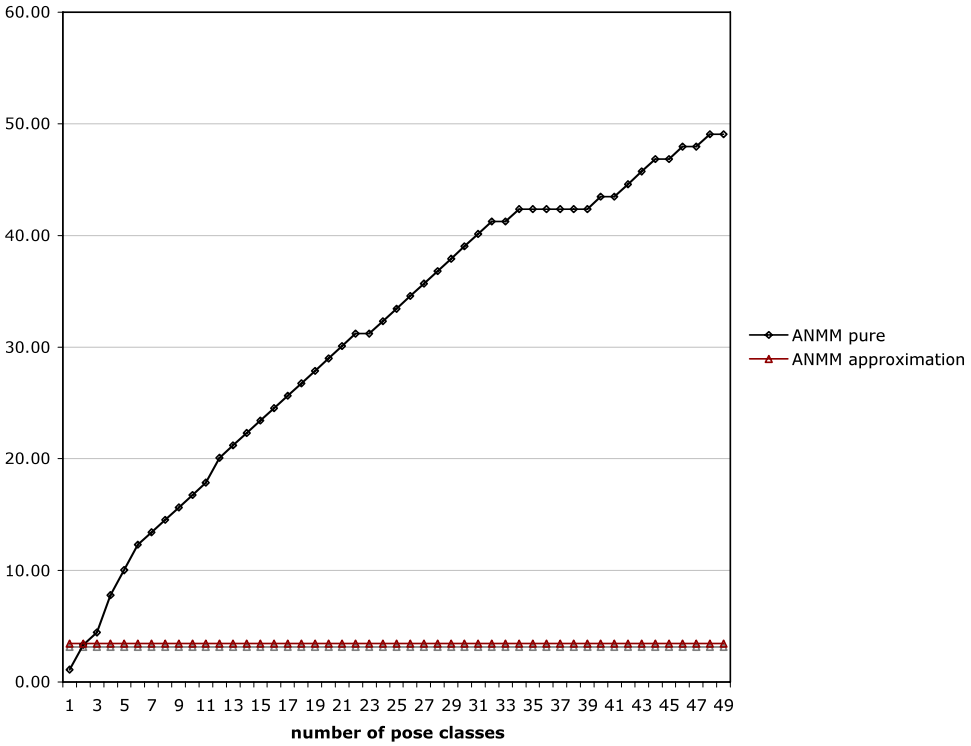


Fig. 18 Correct classification rates using up to 100 Haarlets for classification. *Solid line*: shows correct classification rates using ANMM approximation with 3D Haarlets. *Dashed line*: correct classification rates using ANMM approximation with 2D Haarlets. *Dashed-dotted line*: correct classification rates using 2D Haarlets trained with AdaBoost

Fig. 19 (Color online) Classification times in milliseconds for the pure ANMM classifier (*blue*) and the classifier using 100 3D Haarlets to approximate the ANMM transformation (*red*)



mance when compared to the state-of-the-art, but at interactive speeds.

The methods described in this article can be ported to other classification problems as well, such as hand gesture recognition, object detection and recognition, face detection and recognition, and even event detection where the 3rd dimension of the 3D Haarlets is a time dimension.

Acknowledgements This work has been carried out in the context of the Sixth Framework Programme of the European Commission: EU Project FP6 511092 (CyberWalk) and Swiss NCCR project IM2.

References

Baumberg, A., & Hogg, D. (1994). Learning flexible models from image sequences. *Lecture Notes in Computer Science*, *800*, 299–308.

Belhumeur, P. N., Espanha, J., & Kriegman, D. (1997). Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711–720. Special Issue on Face Recognition.

Bregler, C., & Malik, J. (1998). Tracking people with twists and exponential maps. In *IEEE conference on computer vision and pattern recognition* (pp. 8–15).

Cheung, K. M., Baker, S., & Kanade, T. (2003). Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *IEEE computer society conference on computer vision and pattern recognition* (Vol. 1, pp. 77–84).

Cohen, I., & Li, H. (2003). Inference of human postures by classification of 3D human body shape. In *IEEE international workshop on analysis and modeling of faces and gestures* (p. 74).

Delamarre, Q., & Faugeras, O. (1999). 3D articulated models and multi-view tracking with silhouettes. In *International conference on computer vision* (pp. 716–721).

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd edn.). New York: Academic Press.

Gavrila, D. M., & Davis, L. (1996). 3D model-based tracking of humans in action: a multi-view approach. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 73–80).

Griesser, A., De Roeck, S., Neubeck, A., & Van Gool, L. (2005). GPU-based foreground-background segmentation using an extended colinearity criterion. In *Proceedings of vision, modeling, and visualization (VMV)* (pp. 319–326).

Ioffe, S., & Forsyth, D. (2001). Human tracking with mixtures of trees. In *International conference on computer vision* (Vol. 1 pp. 690–695).

Kakadiaris, I., & Metaxas, D. (2000). Model-based estimation of 3D human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), 1453–1459.

Ke, Y., Sukthankar, R., & Hebert, M. (2005). Efficient visual event detection using volumetric features. In *International conference on computer vision* (pp. 166–173).

Kehl, R., Bray, M., & Van Gool, L. (2005). Full body tracking from multiple views using stochastic sampling. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 129–136).

Lienhart, R., & Maydt, J. (2002). An extended set of Haar-like features for rapid object detection. In *IEEE international conference on image processing* (Vol. 1, pp. 900–903).

Mikíc, I., Trivedi, M., Hunter, E., & Cosman, P. (2001). Articulated body posture estimation from multi-camera voxel data. In *IEEE conference on computer vision and pattern recognition* (pp. 455–462).

Mori, G., & Malik, J. (2002). Estimating human body configurations using shape context matching. In *European conference on computer vision* (Vol. 3, pp. 666–680).

Nummiaro, K., Koller-Meier, E., & Van Gool, L. (2003). An adaptive color-based particle filter. *Image Vision Computing*, *21*(1), 99–110.

Papageorgiou, C., Oren, M., & Poggio, T. (1998). A general framework for object detection. In *International conference on computer vision* (pp. 555–562).

Ren, L., Shakhnarovich, G., Hodgins, J. K., Pfister, H., & Viola, P. (2005). Learning silhouette features for control of human motion. *ACM Transactions on Graphics*, *24*(4), 1303–1331.

Rosales, R., & Sclaroff, S. (2000). Specialized mappings and the estimation of body pose from a single image. In *IEEE human motion workshop* (pp. 19–24).

Shakhnarovich, G., Viola, P., & Darrell, T. (2003). Fast pose estimation with parameter-sensitive hashing. In *IEEE international conference on computer vision* (pp. 750–757).

Sminchisescu, C., & Triggs, B. (2003). Estimating articulated human motion with covariance scaled sampling. *International Journal of Robotics Research*, *22*(6), 371–393.

Van den Bergh, M., Koller-Meier, E., & Van Gool, L. (2008). Fast body posture estimation using volumetric features. In *IEEE visual motion computing*.

Viola, P., & Jones, M. J. (2001). Robust real-time object detection using a boosted cascade of simple features. In *IEEE computer society conference on computer vision and pattern recognition* (Vol. 1, pp. 511–518).

Wang, F., & Zhang, C. (2007). Feature extraction by maximizing the average neighborhood margin. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 1–8).

Yamamoto, M., Sato, A., Kawada, S., Kondo, T., & Osaki, Y. (1998). Incremental tracking of human actions from multiple views. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 2–7).

Yang, J., Yu, H., & Kunz, W. (2000). An efficient LDA algorithm for face recognition. In *International conference on control, automation, robotics and vision*.

Yang, M.-H., Kriegman, D., & Ahuja, N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(1), 34–58.

Zhao, W., Chellappa, R., & Nandhakumar, N. (1998). Empirical performance analysis of linear discriminant classifiers. In *IEEE conference on computer vision and pattern recognition* (pp. 164–169).