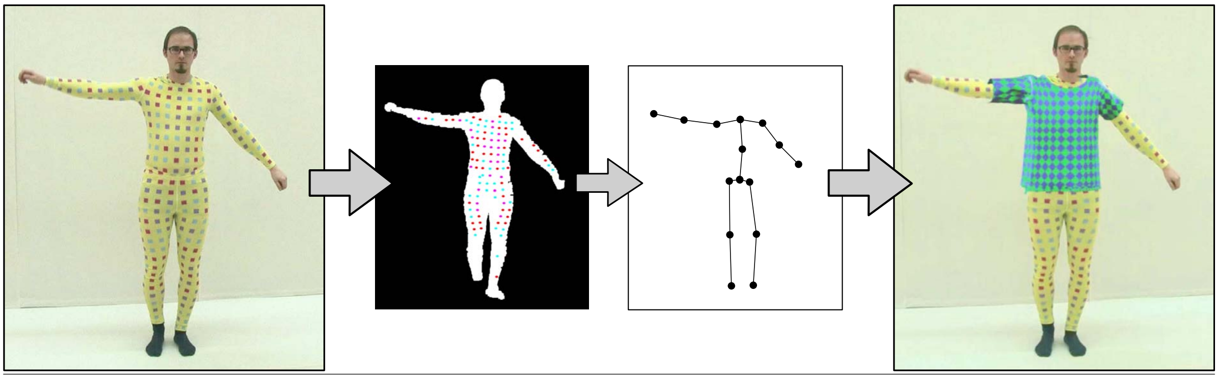
Vision, Modeling, and Visualization (2011)
Peter Eisert, Konrad Polthier, and Joachim Hornegger (Eds.)

Monocular Pose Reconstruction for an Augmented Reality Clothing System

L. Rogge[1] and T. Neumann[2] and M. Wacker[2] and M. Magnor[1]

[1]Institut für Computergraphik, TU Braunschweig, Germany
[2]University of Applied Sciences, Dresden, Germany



Abstract

In this paper, we present an approach for realizing an augmented reality system for try-on of apparel. The core component of our system is a quick human pose estimation algorithm based on a single camera view only. Due to monocular input data, pose reconstruction may be ambiguous. We solve this problem by using a markered suit, though not relying on any specific marker layout. To recover 3D joint angles of the person using the system we use Relevance Vector Machine regression with image descriptors that include neighborhood configurations of visible colored markers and image gradient orientations. This novel combination of image descriptors results in a measurable improvement in reconstruction quality. We initialize and evaluate our algorithm with pose data acquired using a motion capture system. As the final step, we simulate a cloth draped around a virtual character adopting the estimated pose. Composing the original view and the rendered cloth creates the illusion of the user wearing virtual garments.

Categories and Subject Descriptors (according to ACM CCS): I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Motion, H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1. Introduction

Pose and motion reconstruction methods have a wide field of application, predominantly in the film and gaming industry where captured motions are used for animation of characters. Our motivation is to use pose reconstruction to enable a user to try on virtual clothing. As a key component of such a system, a fast and simple to use pose estimation method is required. However, most pose reconstruction approaches require a system with multiple views. This complicates the setup significantly, since time synchronization and camera calibration are required. Although new sensors such as Microsoft's Kinect allow pose estimation using a single sensor, such a sensor would still have to be coupled and calibrated with a camera that delivers a high resolution color image to be used as a basis for high-quality augmentation. To this end,

DOI: 10.2312/PE/VMV/VMV11/339-346

we decided to focus on monocular pose reconstruction from a single high definition camera image, thereby keeping the system very flexible and allowing quick set up at different locations without labour intensive preparations. The major problem in our setting, caused by the restriction to monocular input, is the reconstruction of a correct 3D pose from 2D information from the single input video. We start by analyzing the image data to detect features that are likely to be very pose specific. Instead of only using silhouettes, we also employ information about image gradient orientations and marker configurations. This novel combination helps to resolve pose ambiguities and results in a more accurate pose estimation. In contrast to multi view marker-based approaches, we do not rely on direct marker identification and reprojection, but only use information about neighborhood configurations. In our approach, we propose to use Relevance Vector Machine (RVM) regression, which allows, after a short training phase, to reconstruct 3D joint angles from a computed pose descriptor. These reconstructed pose angles are applied to the forward kinematic of a human body model acting as collision body in a cloth simulation. Composing the original image and the cloth rendering creates the illusion of wearing virtual garments. The proposed steps of the algorithm are chosen and designed to be simple, fast, and to depend on very little user interaction, which makes them suitable for an augmented reality application. Such a system could be used as a virtual fitting room, or may speed up the process of prototyping of apparel by enabling design customization with instant visual feedback.

2. Related Work

The realization of a augmented reality clothing system such as the proposed one requires components that combine ideas from usually separated fields of research, such as pose and surface reconstruction, image segmentation, and video augmentation.

Regarding pose reconstruction from monocular images, many different approaches have been developed. Since important data like depth cues is missing, most of the reconstruction techniques use a human motion model. To recover a human pose from a single view, several parameters, for example body height, have to be either known a priori or estimated from cues in the image. In general there are two types of pose reconstruction methods that only rely on image information and no special hardware (such as depth sensors like Microsoft's Kinect). Some try to recover the posture by computing joint angles for a given hierarchical body model [AT04, MM02, LC85, BM98, BYS07], like in conventional motion capture systems. Usually a database or trained functional is used to map image features to a set of pose angles, or a set of pose templates exists to find the best matching pose to a given image. For example, Wang et al. used a colored glove [WP09] to compare a captured 26-DOF hand pose with a database containing 100,000 templates. An approach to recover a full human pose was presented by Agarwal and Triggs [AT04, AT06]. Using Relevance Vector Machine (RVM) regression [Tip00], they computed a functional mapping silhouette information to a vector of pose angles. Dalal and Triggs also presented a similar technique that relies on image gradient information instead of silhouette features [DT05] allowing detection of human shapes or specific poses in given images.

Belongie et al. developed a technique to identify certain shapes and objects using only their silhouette called *Shape Context* [SB02]. Looking at a set of silhouette points, a log-polar histogram is build regarding angle and distance between two silhouette points. An adaption of this technique was used by Mori and Malik [MM02] to identify specific human postures, since different poses correspond to different silhouette shapes.

Full reconstruction of a detailed 3D surface using only a single image is an even more complex problem than pose reconstruction because of the higher dimensionality of the output space. This search space can be reduced significantly by using parametrized 3D body models, often generated from large databases of human shapes. The SCAPE database [ASK*05] contains 71 different scans of poses of a person and additionally several scans of different persons in a single pose. It is used in many monocular reconstruction algorithms [BSB*07,BBHS07]. Guan et al. [GWBB09] require a coarse manually fitted skeleton to initialize their reconstruction algorithm. They then use SCAPE to find the best fitting body to the pose observed in the given input image by finding body parameters minimizing an energy functional which describes the pose fitting. Unfortunately, these parameters are not directly related to separate body features like gender, height, or weight. Hasler et al. [HSS*09] removed this problem by adding semantics to the human model allowing to control shape deformations directly. Such approaches still require a suitable initial pose to start the surface estimation. For example, Sigal et al. [SBB07] use a Shape Context driven pose reconstruction as a prior for optimizing body shape parameters (using SCAPE) to find a best fitting pose and body shape from a single image.

Most of these approaches require accurate silhouettes of the observed body. Extracting these is a challenging problem itself. Image segmentation has been of interest from the beginning of computer vision [HS85] and various approaches have been developed using contour information, texture similarity [MBLS01], level-set-based methods [VC02], expectation maximization [CBGM02] and recently more and more graph-based image representations [FH04, RKB04, CCBK06]. Since one of our requirements was a quick segmentation of a person's silhouette, computationally complex techniques (e.g. texture-based) had to be avoided. On the other hand, simple chroma keying or static background subtraction was not an option, since for an augmented reality system we expect dynamic environments. An approach based on graph cuts is well suited in this case, since it is robust and capable of running in realtime on graphics hardware [VN08].

The proposed system is meant to be a virtual try-on system to test and virtually wear clothing not yet produced. Existing techniques replace textures of cloth worn by a person in front of the camera. Scholz et al. [SSK*05] used a colored dot pattern to track distinct surface positions in 2D allowing to blend in a new garment texture replacing the pattern. Shading reconstruction allows to visually reintroduce the garment's wrinkles and folds to the replaced texture. Hilsmann et al. [HE08, HE09] presented methods to replace features of a worn garment by tracking an arbitrary shape. While Scholz' system operates offline, the technique proposed by Hilsmann et al. allows to blend in new textures in real time, but only on a distinct patch on the cloth (e.g. a logo on a shirt) instead of the whole piece of clothing.

Instead of modifying an extisting surface, our goal is to simulate a completely new piece of cloth. We therefore reconstruct 3D data of an observed pose and do not operate in 2D when augmenting the input video. In the following sections we will describe the steps of our solution in detail.

3. Pose Reconstruction

The reconstruction of a human pose from a single image is not trivial, since the missing depth information requires to extract the 3D position and orientation of body parts from 2D image data. Several constraints such as invariant bone length or angular limits for joints can be used to reduce the search space for the correct pose. The proposed pose estimation algorithm consists of three major steps. First of all, necessary features to describe an observed pose in a 2D image have to be computed. We lay out the details for that in Section 3.1. From the extracted features, a distinct feature descriptor can be computed that describes the image captured by the camera in such a way that, ideally, the 3D pose can be inferred directly from that feature descriptor even when other variables such as lighting, subject, or 3D translation change. Having computed the feature descriptor, the 3D pose can be inferred from a set of sample poses and suitable interpolation by a functional. We use RVM regression [Tip00] so this functional has the form

$$y = X_f \cdot d(X_b, x) \tag{1}$$

where $x \in \mathbf{R}^M$ is the feature descriptor recovered from a frame, $X_b \in \mathbf{R}^{M \times S}$ is a matrix of M-dimensional feature descriptor support vectors and $X_f \in \mathbf{R}^{N \times S}$ a matrix mapping support vector configurations to a N-dimensional pose vector. The functional d yields the distance of the measured feature descriptor to all precomputed support vectors. How to train this mapping from example data is explained in Section 3.2. We describe a pose using a body model in the Biovision BVH format.

3.1. Feature Detection and Processing

Using only monocular image data complicates reconstruction, since information of how different body parts are layered in the observed posture is not readily available. For example, when only silhouette information is used (e.g. Agarwal and Triggs [AT04]) it is hard to distinguish if an arm is in front of, or behind the torso. These ambiguities must be resolved somehow to correctly reconstruct the pose. We address this problem by using a suit printed with markers. In contrast to other marker based approaches for motion capture, these markers do not have to be placed exact and do not require calibration. Using a random pattern of three different colors it is possible to distinguish several marker distrubutions. These are different if occlusions of body parts occur and allow for solving ambiguous cases. The used markers are uniformly colored 4 × 4 cm squares and randomly distibuted on a regular grid with a 3 cm gap between markers. Three colors (cyan, purple, and red) are used as marker colors, which are printed on a yellowish green background. These four colors are chosen to have a hue distance of 90° in HSV color space, making identification of different markers robust against illumination changes.

3.1.1. Background Segmentation

To reduce the search space when looking for marker positions in an image, the silhouette of the person is extracted first. To quickly separate foreground and background we apply an algorithm based on graph cuts which is optimized to detect the suit colors in front of an arbitrary background. We adapted the technique proposed by Criminisi et al. [CCBK06], which uses likelihood lookup tables regarding color, motion and gradients. In their work a small adaption and self-training sequence is applied to initialize likelihoods which are then used as weights in a graph cut optimization. Employing a short training we created a set of likelihood lookup tables corresponding to the suit which are then applied in the actual image segmentation process.

In the training phase of the algorithm, a small set of images and user-created masks is required. In our case a set of 20 frames with corresponding masks was sufficient, from which several lookup tables were trained and saved as the pixel's negative log likelihood for a certain condition. We use the hue and saturation channel of the HSV colorspace to learn the likelihood of a certain color to be either foreground or background over a sequence of successive frames.

$$C(\alpha, X) = -\sum_{n}^{N} log(p(c(x_n)|\alpha)) \tag{2}$$

Where $c(x)$ is a pixel's color and $\alpha \in \{FG, BG\}$ a label denoting foreground or background. Then we evaluate the 3×3 neighborhood $N(x)$ of each pixel x for correlation of hue, saturation and change in label. A transition probability table is built for the transitions $FG \rightarrow FG, FG \rightarrow BG, BG \rightarrow$

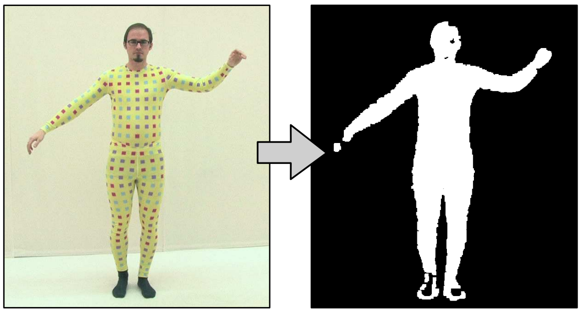


Figure 1: *Our segmentation algorithm separates a given image into foreground and background. The foreground area is used as a mask for feature detection.*

$FG, BG \rightarrow BG$ being the labelling configuration of neighboring pixels and the difference regarding hue and saturation.

$$N(\alpha_x, \alpha_y, X) = -\sum_{n}^{N} \sum_{i}^{\|N(x_n)\|} log(p(d(x_n, y_{n,i})|\alpha_n, \alpha_i) \quad (3)$$

Where $y_{n,i} \in N(x_n)$ is a neighboring pixel of pixel x_n, and $d(x,y) = (d_H(x,y), d_S(x,y))^T$ is a vector of two pixel's distance in hue and saturation. Another lookup table is created from horizontal and vertical gradient magnitude, $g_h(x)$ resp. $g_v(x)$, of each pixel x and it's likelihood to be foreground or background.

$$G(\alpha, X) = -\sum_{n}^{N} log(p(\ (g_h(x), g_v(x))^T |\alpha_n)) \quad (4)$$

A fourth table represents the correlation between gradient magnitude $g(x)$ and the change of labelling over time. This is important to allow a label change in boundary regions and preventing it in smooth contiguous areas.

$$M(\alpha^t, \alpha^{t-1}, X) = -\sum_{n}^{N} log(p(g(x_N)|\alpha_n^t, \alpha_n^{t-1})) \quad (5)$$

The last energy term is learned from temporal transitions of labels and describes how likely it is that a pixel keeps a new label after a label change. It is realized using a second-order Markov chain requiring two previous frames and their segmentation.

$$T(\alpha^t, \alpha^{t-1}, \alpha^{t-2}) = -\sum_{n}^{N} log(p(\alpha_n^t|\alpha_n^{t-1}, \alpha_n^{t-2})) \quad (6)$$

These lookup tables define the energy to be minimized by a two label graph cut algorithm using the labels of $\alpha \in \{FG, BG\}$.

$$\begin{aligned} E(\alpha^t, \alpha^{t-1}, \alpha^{t-2}, x, y) &= \sigma_C \cdot C(\alpha^t, x) \\ +\sigma_N \cdot N(\alpha^t, x, y) &+ \sigma_G \cdot G(\alpha^t, x) \quad (7) \\ +\sigma_M \cdot M(\alpha^t, \alpha^{t-1}, x) &+ \sigma_T \cdot T(\alpha^t, \alpha^{t-1}, \alpha^{t-2}) \end{aligned}$$

When using a graph cut algorithm to minimize this energy, the term evaluating the neighborhood configuration of pixels $N(\alpha_x, \alpha_y, X)$ is used to define the pairwise weights between neighboring pixels which are now treated as graph nodes. All other terms contribute to the weights of label edges connecting pixels with each of the labels α. We weight the energy terms using σ_C, σ_N, σ_G, and σ_M. Since all the terms are based on lookup tables, the graph setup is very fast and only a gradient image of the current frame has to be computed additionally. All other data is used from previous frames or the current frame directly. Finally a fast graph cut algorithm like Cuda Cuts [VN08] may be applied to separate foreground from background in real time. The background segmentation algorithm yields a binary map where the detected person is foreground and everything else is marked as background. Figure 1 shows an example of the extracted silhouette. In contrast to other silhouette-based pose reconstruction algorithms such as [AT04, AT06], our method is much more robust against "dirty", erronous silhouettes, since it is only used as a mask for a marker detector presented in the next section.

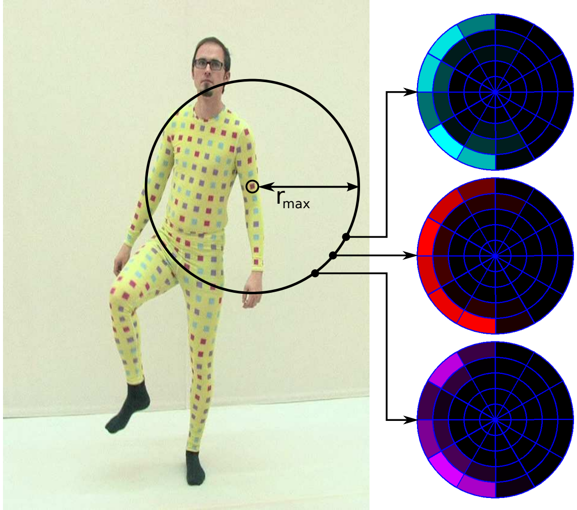


Figure 2: *Depicted is a Shape Context histogram describing the neighborhood configuration of a certain marker with respect to orientation, distance, and color. Per frame these histograms are built for every detected marker and compressed into a single Feature Context Descriptor using vector quantization.*

3.1.2. Feature Context Descriptor

To generate a descriptor based on marker information in the observed image, a vector containing neighborhood information of detected markers is constructed, similar to the Shape Context descriptor proposed by Belongie et al. [SB02]. Having a marker m_i of color class $c_i \in C = \{cyan, purple, red\}$ all markers m'_j within a specified neighborhood $N(m_i, R)$ of radius R around marker m_i are inserted into a histogram

H_i regarding their log distance $r_{i,j} = log(\|m'_j - m_i\|)$, angle to the horizontal axis in image space $\theta_{i,j}$ and color class c_j. See Figure 2 for a sample histogram. With 5 radial bins, 12 angular bins and three color bins, all histograms $H_i, i \in (1,\ldots,n)$ (n being the number of visible markers) contain $5*12*3 = 180$ values. Since the number of visible markers varies from frame to frame, we cannot simply concatenate all the histograms - this would form a vector of different length for each frame which is incompatible with our regression-based pose inference. We solve this problem by vector quantizing the histograms of all visible markers into a set H_{ref} of N reference points in the same space. These are built from training data using the k-means algorithm. Marker histograms then vote into their 5 nearest neighbors in H_{ref}. Using this technique, for every frame a N-dimensional vector describes the pose corresponding to the observed marker configuration. We use $N = 200$ reference points for the final histogram, normalize it and call this vector H_{FC} the *Feature Context Descriptor* (*FC Descriptor*).

3.1.3. HOG Descriptor

Similar to the *FC Descriptor* we generate a histogram of oriented gradients [DT05] to incorporate gradient information into our pose reconstruction process. We only use the region of the detected person, i.e. the axis aligned bounds of the extracted silhouette, which is subdivided into $m \times n$ blocks. For each block a histogram of gradient orientations is built. The input image I is filtered using conventional Sobel kernels in horizontal and vertical direction, S_h resp. S_v, yielding gradient orientations for each pixel by computing $\theta_{x,y} = \arctan(S_v * I / S_h * I)$. Using 36 bins for angle orientations $\Theta \in [-\pi..\pi]$ and soft binning, dominant angle orientations can be detected for each block. (See Figure 3). We again use vector quantization and soft voting into a set of reference histograms computed using k-means to reduce the dimensionality of the data. This way the M most significant angle distibutions are extracted. We used $M = 100$ for quantization, which seems to be enough to provide pleasing results. We refer to the normalized histogram H_{HOG} as *HOG Descriptor* throughout the rest of the paper.

3.1.4. Frame Descriptor

To finally describe an observed pose, we combine the previously mentioned feature descriptors. We compose our final frame descriptor from the 200D *FC Descriptor*, the 100D *HOG Descriptor* as well as the silhouette's relative position and size and get a 304D vector describing the pose seen in an image. This vector is processed by a Relevance Vector Machine to reconstruct pose angles as described in the following section.

3.2. Joint Angle Reconstruction

To obtain a training set of 3D pose data corresponding to a sequence of frames and the frame descriptors computed for each frame, we use a markerless motion capture system by Organic Motion [Org]. Thus while capturing a video test-sequence a corresponding sequence of pose angle data was captured. The captured motion was exported using the BVH format supported by many modelling and rigging tools (i.e. MotionBuilder). We calibrate a HD video camera into the coordinate system of the motion capture system using checkerboard patterns, the OpenCV library and the SDK provided by the Organic Motion system. This provides a one-to-one mapping of 3D-pose data and feature descriptor vectors, allowing computation of a functional which maps the feature descriptor space to the pose angle space.

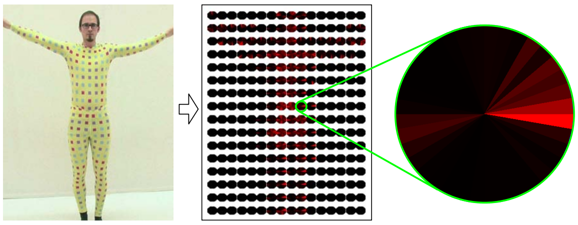


Figure 3: *For every frame, a grid of* 16×16 *histograms of oriented gradients (HOG) is generated. The silhoutte's bounding box is subdivided into blocks used for computing gradients. Thus the descriptor is pose specific but independent of position and size of the person.*

To reduce the size of data points used for regression, we applied a k-means clustering on the feature descriptor data of our training sequence. We tried four different sample sizes $S \in \{512, 768, 1024, 1500\}$ and stored the S most significant feature descriptors most similar to the S cluster centers as a matrix $X_b \in \mathbf{R}^{304 \times S}$ of basis vectors. To make reconstruction more robust, we split up every joint angle into sine and cosine part, allowing smooth wrap around compared to the $0 = 2\pi$ gap when using angular values. These joint angles, complemented by the 2D offset (with respect to the silhouette) and depth of the body model, form a 123D vector describing the body pose which we aim to reconstruct from our feature descriptor as described in the next section. We tried to use quaternions to represent the joint angles, but since quaternions would require spherical linear interpolation to yield acceptable results, we could not use the described RVM as it linearly weights exemplar poses to interpolate a plausible result.

To reconstruct pose angles from a feature descriptor based on 2D image information, we compute the euclidean distance d of this descriptor vector x to all reference vectors X_b selected for RVM regression. We compute the euclidean distance matrix $D = d(X_b, x) \in \mathbf{R}^{1 \times M}$ containing distances between all vectors in $X_b \in \mathbf{R}^{U \times M}$ and $x \in \mathbf{R}^{U \times 1}$, where X_b contains M support vectors as colums, and x contains the U dimensional descriptor vector to be converted into the target space of V dimensions. This is done by solving

$$y = X_f \cdot D^T \tag{8}$$

where $X_f \in \mathbf{R}^{V \times M}$ is a matrix of kernel functions transforming all M-d column vectors of D into V dimensional space. In this case we compare our feature descriptor $x \in \mathbf{R}^{304 \times 1}$ to a set of basis vectors in $X_b \in \mathbf{R}^{304 \times S}$, we get a distance matrix $D \in \mathbf{R}^{1 \times S}$ describing the observed pose with respect to the training samples in X_b. Using a matrix of kernel functions $X_f \in \mathbf{R}^{123 \times S}$ we map x to to actual pose angles $y \in \mathbf{R}^{123}$. This matrix has been precomputed using a standard RVM regression as in [AT04] from the set of corresponding feature descriptors and pose angle vectors selected from our test sequence using k-means clustering. The reconstructed vector y is composed of the relative 2D screen position of the body model offset and the approximate depth followed by 6 values per joint being the sine and cosine parts of the three Yaw-Pitch-Roll angles per body joint. Since our body model uses 20 joints, we get a vector containing 123 values. This reconstruction process yields only interpolated values, so each $(\sin(\theta), \cos(\theta))$ pair has to be rescaled to unit length. Also the reconstructed vector does not contain the actual 3D offset of the body model's root joint. Using the extrinsic parameters of the camera used for capturing the training sequence, the body model is reprojected into 3D space from the reconstructed 2D position and the estimated depth value. Like the background segmentation in Section 3.1.1, the projection matrix has to be computed only once. Since this projection is highly linear and the RVM inference is, in contrast to the joint angles, not able to reliably produce the correct body position we automatically adjust this position in a postprocessing step. Using a simple and fast color classification algorithm [JR02], we estimate the positions of head, hands and feet in 2D and rigidly move the body to reduce reprojection error. This corrects the position of the body, but in the future the 2D positions of hand and feet should be included into the feature descriptor. The BVH body model can then be driven to simulate the recovered pose by applying the reconstructed angles to the forward kinematic of a virtual character, which can act as a collision body in a cloth simulation.

4. Cloth Simulation and Image Compositing

The final step of the proposed system is the simulation of a piece of garment draped over a pose data driven human body model. Since the pose data has been estimated in 3D by our system, we can use a physical cloth simulation to animate a piece of cloth adopting this pose in a realistic way. A rigged body model is driven by the pose data and used as collision body for a dynamic piece of cloth, such as a T-Shirt or a dress. Using the extrinsics of the camera recovered from our training sequence, we are able to render the animated garment from the original camera viewpoint. The final rendering is then composed into the original frame to create the augmented video, Figure 6.

5. Results

In our setup we used a Canon XH A1 HD video camera providing images in the format of 1440 × 1080 px at a framerate of 25 fps. We used this video data for all training sequences as well as the reconstructed motion sequences. The training of the image segmentation algorithm used only 20 frames, for which reference masks had to be created manually. The set of poses used for training our pose reconstruction algorithm created by using a markerless motion capture system consisted of 5666 frames. For each 3D pose an image captured by our camera was available. By computing our feature descriptors for each of these frames, we could select several sample sub-sets of this data using k-means, thereby selecting N reference poses. The RVM was trained using these sample data sets in Matlab [AT04]. The RGB frames of our camera capture were then fed into the trained system. The reconstruction results were pleasing and temporally consistent, even though we did not employ any temporal tracking. After the reconstruction of the 3D pose data we exported the pose data as a BVH file. This can be imported by Autodesk Maya, where we used the particle-based cloth simulation *nCloth* to render a piece of garment according to the observed poses. When rendering the animated cloth over the input image, the illusion of the person standing in front of the camera wearing a virtual piece of garment is created. Figure 6 shows a final rendering of our system.

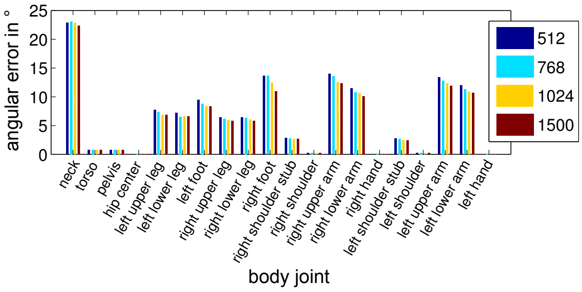


Figure 4: *Average angular error compared to ground truth data for every joint of the body model in our main reconstruction sequence. Reconstruction quality improves with increasing number of sample poses used for training. Notice that some angular errors, such as rotations around the roll axis of a bone, do not affect the visual result.*

We evaluate the reconstruction result of our system by using the motion capture data of the initial training as ground truth data to the video frames of that training sequence. The reconstruction quality is measured as the angular error of the reconstructed joints compared to ground truth data. To evaluate the correlation of reconstruction quality and training sample set size, we tested sample sets of $N \in \{512, 768, 1024, 1500\}$ poses. The more training samples were used, the more accurate the reconstructed pose was (see Figure 4). However, this improvement was not significant.

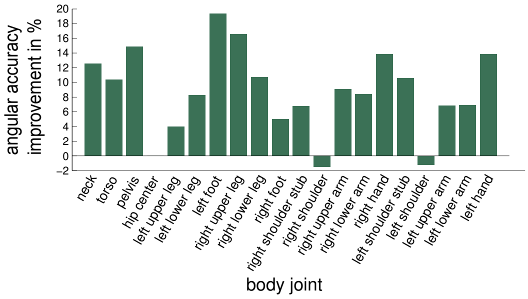


Figure 5: *Improvement of angular error compared to the silhouette-based method from [AT04] applied to the exact same sequence of our data set. Our marker-based method improves the detection of important body joints as arms and legs. Accuracy at the shoulder joints presumably suffers from the low marker density used.*

To evaluate how much our feature descriptor affects the reconstructed pose quality, we compared the performance of the parts of our feature descriptor separately to the silhouette-feature-based method of Agarwal and Triggs [AT04]. Using only the HOG descriptor we observe worse reconstruction (angular error about 4 times as high). With the marker-based FC descriptor alone no significant improvement can be seen. However, the combination of FC and HOG descriptors, as proposed by our method, yields an average improvement of the angular error of about 9% compared to the silhouette-based approach. Figure 5 shows the angular accuracy improvement per joint compared to [AT04]. The pose reconstruction accuracy of important body parts, such as arms or legs, improves using our novel feature descriptor. Body parts covered by only a few markers, like the shoulders, do not improve in reconstruction accuracy.

To further evaluate the system, we used another test sequence not used for any training. So feature descriptor matches to actual support vectors during reconstruction could not happen. The resulting pose estimation was again smooth and reconstructed the observed poses as expected. Therefore our system seems to be suitable to reconstruct 3D poses from monocular views of a person wearing the markered suit.

6. Conclusion and Future Work

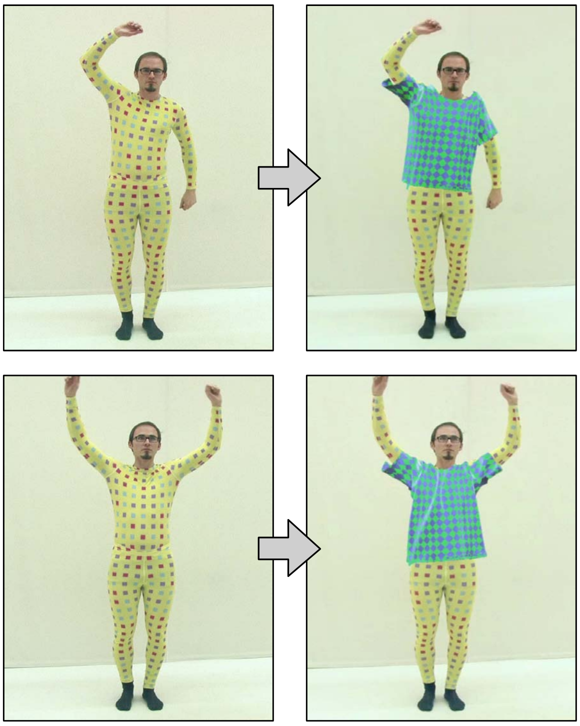


Figure 6: *Final results of our system. Using the reconstructed pose data, a piece of cloth is simulated and composed into the original image.*

The presented system allows to quickly recover a 3D human pose from single images. Using a short initial training, the system is able to recover 3D pose data from image features. To provide these features, we use a special suit printed with a random marker pattern. Our algorithms are trained to detect these markers using fast image processing and to recover a pose from a feature descriptor built for each frame. The 3D pose data is then used to simulate a virtual garment on a human body model adopting the pose of the person seen in the processed image. Composing a rendering of the garment with the original image data creates the illusion of wearing virtual cloth. The algorithms used for recovering the pose are designed to be simple and process data in parallel, which makes them suitable for implementation on graphics hardware allowing them to run at real time framerates. The major bottleneck so far is the cloth simulation, which requires to interpolate additional frames from the capture framerate of 25 fps to achieve stable simulation results. This step is also the most complex and computationally intense part of the system, confining the runtime of the whole system.

One aspect of future work will be removing as much user interaction as possible. A promising approach would be to replace the segmentation training in Section 3.1.1 by a motion-based background segmentation technique like the one presented by Scharfenberger et al. [SCF09]. Segmentations supplied by this algorithm can then be used to initialize and train the more robust graph cut algorithm already used in our current setup. Even though this step is only neccessary once (when training the graph cut algorithm to a new suit), automated training in each new environment should improve the robustness and accuracy of the image segmentation. A more important aim for the future is to speed up the cloth simulation, and thus enabling the whole system to run at real time framerates. Existing physics simulation libraries like Havok ^{TM} or nVidia PhysX ^{TM} made recent advances in cloth simulation and are used in real time multimedia applications. Incorporating such a library into our system could solve the performance bottleneck of the presented system.

References

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: Scape: shape completion and animation of people. *ACM Trans. Graph. 24* (July 2005), 408–416. 2

[AT04] Agarwal A., Triggs B.: 3d human pose from silhouettes by relevance vector regression. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on 2* (2004), 882–888. 2, 3, 4, 6, 7

[AT06] Agarwal A., Triggs B.: Recovering 3d human pose from monocular images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 1 (January 2006), 44–58. 2, 4

[BBHS07] Balan A., Black M., Haussecker H., Sigal L.: Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on* (October 2007), pp. 1 –8. 2

[BM98] Bregler C., Malik J.: Tracking people with twists and exponential maps. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on* (June 1998), pp. 8 –15. 2

[BSB*07] Balan A., Sigal L., Black M., Davis J., Haussecker H.: Detailed human shape and pose from images. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (June 2007), pp. 1 –8. 2

[BYS07] Bissacco A., Yang M.-H., Soatto S.: Fast human pose estimation using appearance and motion via multi-dimensional boosting regression. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on* (June 2007), pp. 1 –8. 2

[CBGM02] Carson C., Belongie S., Greenspan H., Malik J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2002), 1026–1038. 2

[CCBK06] Criminisi A., Cross G., Blake A., Kolmogorov V.: Bilayer segmentation of live video. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (June 2006), vol. 1, pp. 53 – 60. 2, 3

[DT05] Dalal N., Triggs B.: Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on 1* (2005), 886–893. 2, 5

[FH04] Felzenszwalb P., Huttenlocher D.: Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2 (2004), 167–181. 2

[GWBB09] Guan P., Weiss A., Balan A. O., Black M. J.: Estimating human shape and pose from a single image. In *Computer Vision, 2009 IEEE 12th International Conference on* (October 2009), vol. 29, pp. 1381 –1388. 2

[HE08] Hilsmann A., Eisert P.: Tracking deformable surfaces with optical flow in the presence of self occlusion in monocular image sequences. *Computer Vision and Pattern Recognition Workshop 0* (2008), 1–6. 3

[HE09] Hilsmann A., Eisert P.: Tracking and retexturing cloth for real-time virtual clothing applications. In *Proceedings of the 4th International Conference on Computer Vision/Computer Graphics CollaborationTechniques* (Berlin, Heidelberg, 2009), MIRAGE '09, Springer-Verlag, pp. 94–105. 3

[HS85] Haralick R., Shapiro L.: Image segmentation techniques. *Computer vision, graphics, and image processing 29*, 1 (1985), 100–132. 2

[HSS*09] Hasler N., Stoll C., Sunkel M., Rosenhahn B., Seidel H.-P.: A statistical model of human pose and body shape. *Computer Graphics Forum 28*, 2 (2009), 337–346. 2

[JR02] Jones M. J., Rehg J. M.: Statistical color models with application to skin detection. *Int. J. Comput. Vision 46* (January 2002), 81–96. 6

[LC85] Lee H.-J., Chen Z.: Determination of 3d human body postures from a single view. *Computer Vision, Graphics, and Image Processing 30*, 2 (1985), 148 – 168. 2

[MBLS01] Malik J., Belongie S., Leung T., Shi J.: Contour and texture analysis for image segmentation. *International Journal of Computer Vision 43*, 1 (2001), 7–27. 2

[MM02] Mori G., Malik J.: Estimating human body configurations using shape context matching. In *Computer Vision - ECCV 2002*, Heyden A., Sparr G., Nielsen M., Johansen P., (Eds.), vol. 2352 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2002, pp. 150–180. 2

[Org] Organic Motion, Inc.: Stage markerless motion capture system. http://www.organicmotion.com/solutions/stage. 5

[RKB04] Rother C., Kolmogorov V., Blake A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 309–314. 2

[SB02] S. Belongie J. Malik J. P.: Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 4 (April 2002), 509–522. 2, 4

[SBB07] Sigal L., Balan A., Black M. J.: Combined discriminative and generative articulated pose and non-rigid shape estimation. *Advances in neural information processing systems* (2007). 2

[SCF09] Scharfenberger C., Chakraborty S., Farber G.: Robust image processing for an omnidirectional camera-based smart car door. In *Embedded Systems for Real-Time Multimedia, 2009. ESTIMedia 2009. IEEE/ACM/IFIP 7th Workshop on* (October 2009), pp. 106 –115. 7

[SSK*05] Scholz V., Stich T., Keckeisen M., Wacker M., Magnor M.: Garment motion capture using color-coded patterns. *Computer Graphics Forum 24*, 3 (2005), 439–447. 3

[Tip00] Tipping M. E.: The relevance vector machine. In *Advances in Neural Information Processing Systems 12* (2000), MIT Press, pp. 652–658. 2, 3

[VC02] Vese L., Chan T.: A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision 50*, 3 (2002), 271–293. 2

[VN08] Vineet V., Narayanan P. J.: Cuda cuts: Fast graph cuts on the gpu. *Computer Vision and Pattern Recognition Workshop 0* (2008), 1–8. 3, 4

[WP09] Wang R. Y., Popović J.: Real-time hand-tracking with a color glove. *ACM Trans. Graph. 28* (July 2009), 63:1–63:8. 2