Coupled Person Orientation Estimation and Appearance Modeling using Spherical Harmonics

Martijn C. Liem[a], Dariu M. Gavrila[a,b]

[a]*Intelligent Autonomous Systems Group, Informatics Institute, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands*
[b]*Environment Perception Department, Daimler Research and Development, Wilhelm Runge St. 11, 89081 Ulm, Germany*

Abstract

We present a novel approach for the estimation of a person's overall body orientation, 3D shape and texture, from overlapping cameras. A distinguishing aspect of our approach is the use of spherical harmonics for 3D shape- and texture-representation; it offers a compact, low-dimensional representation, which elegantly copes with rotation estimation. The estimation process alternates between the estimation of texture, orientation and shape. Texture is estimated by sampling image intensities with the predicted 3D shape (i.e. torso and head) and the predicted orientation, from the last time step. Orientation (i.e. rotation around torso major axis) is estimated by minimizing the difference between a learned texture model in a canonical orientation and the current texture estimate. The newly estimated orientation allows to update the 3D shape estimate, taking into account the new 3D shape measurement obtained by volume carving.

We investigate various components of our approach in experiments on synthetic and real-world data. We show that our proposed method has lower orientation estimation error than other methods that use fixed 3D shape models, for data involving persons.

Keywords: Person appearance modeling, Orientation Estimation, Spherical harmonics

Email addresses: M.C.Liem@uva.nl (Martijn C. Liem), D.M.Gavrila@uva.nl (Dariu M. Gavrila)

1. Introduction

A person's overall body orientation (i.e. rotation around 3D torso major axis, facing direction) conveys important information about the person's current activity and focus of attention. In this paper, we focus on the estimation of overall person body orientation from few, overlapping cameras. To obtain a more accurate orientation estimate, we jointly estimate it with a 3D shape and texture representation of a person's torso-head, under a rigidity assumption. By projection onto a basis of Spherical Harmonics (SH), a low dimensional appearance model is created that can cope with object deformations while retaining the spatial information captured in a textured 3D representation. By comparing the texture model of the person at consecutive points in time, the relative body orientation can be estimated elegantly using the properties of the SH. This in turn allows to update the 3D shape and texture model of a person.

Apart from facilitating an accurate orientation estimate, the proposed 3D shape and texture model has furthermore the potential to offer improved track disambiguation in a multiple person scenario, compared to less descriptive 2D view-based models. The 3D representation could also provide an initialization for applications aiming at full articulated 3D pose recovery.

The remainder of this paper is organized as follows. In section 2, we discuss the related work. Section 3 starts with an overview of the proposed approach and the basic properties of spherical harmonics. Thereafter, it describes the estimation of texture, orientation, and shape in alternating fashion. In section 4, we present experimental results on both artificial and real-world datasets. We conclude and suggest directions for future work in section 5.

2. Related Work

Extensive research has been performed in the areas of person appearance modeling [1, 2, 3, 4, 5], 3D body shape modeling [6, 7, 8, 9] and pose estimation [8, 9, 10, 11, 12, 13, 14, 15, 16]. This section focuses on the work that we consider to be most closely related to our paper.

Modeling person appearance is most commonly done based on single view information. Color histograms representing the full extent of a person's appearance are often used [1, 17], while more sophisticated methods split a single person's appearance up into several layers, incorporating some spatial

information in the descriptor [18, 19]. More recent methods make use of ensembles of different features to be more robust and cover different aspects of persons' appearance like color and texture information [2, 3].

Viewpoint invariance is addressed by Gray et al. [4], where AdaBoost is used to learn a good set of spatial and color features for representing persons. Some approaches combine histograms created at multiple viewpoints (e.g. Liem and Gavrila [18]). Others try estimating the full body texture of a person by projecting the person's appearance onto a 3D structure like a cylinder (e.g. Gandhi and Trivedi [5]). These multi-view types of methods provide robustness to perception from different angles, while the use of full body textures also maintains the spatial properties of the appearance per view.

Since the human body shape is largely non-rigid, modeling body texture is not straightforward. As we will see in the experiments, mapping the texture of a non-rigid shape onto a rigid object as done by Gandhi and Trivedi [5] may result in instable textures over time, introducing artifacts into learned texture models and making it hard to accurately compare body textures over time. Using a more accurate estimate of the 3D object shape could help in creating more discriminative appearance models.

Some research has specifically aimed to generate a more accurate 3D reconstruction of the visual hull of objects, computed using shape-from-silhouette methods. Kutulakos and Seitz [20] compute an improved voxel model of the visual hull by coloring all voxels using the camera views and checking color consistency among cameras for each voxel, eliminating inconsistent voxels. Cheung et al. [21] present a method for aligning rigid objects in multiple time steps and reconstructing an object using information from multiple time instances. Translation and rotation are estimated by matching and aligning colored surface points over time. By transforming the camera viewpoints according to the estimated transformation, new virtual viewpoints are created extending the number of viewpoints usable for shape-from-silhouette methods.

Mitzel and Leibe [7] learn a 3D shape model of a person over time based on stereo depth data. Using the Iterative-Closest-Point (ICP) algorithm, the model is aligned with the stereo data at each time step enabling person tracking and updating the model.

An alternative approach is to estimate fully articulated 3D body pose, either by lifting 2D part-based body models [10, 16] or by using 3D models directly [11, 8, 9]. This approach can in principle provide accurate 3D body

shape and texture models, but is computationally expensive and is hard to apply in uncontrolled, complex environments (e.g. dynamic background, multiple persons).

Some work has been done on estimating the body facing direction of persons without estimating the full articulated pose. Several 2D single frame approaches combine orientation-specific person detectors [12, 13, 14], while work by Chen and Odobez [15] estimates body and head pose in batch mode, coupling the output of underlying classifiers. These methods offer an absolute orientation estimate with respect to some global reference frame. In Gandhi and Trivedi [5], texture sampled from a cylinder surrounding a person is shifted along the rotational axis in a generate-and-test fashion to find the best matching orientation.

SH have been used in order to perform face recognition under varying lighting conditions (e.g. Yue et al. [22]). Representing 3D objects by projecting them onto an *SH* basis has been researched mainly with respect to exemplar based 3D object retrieval. A collection of rotationally invariant 3D object descriptors has been compared by Bustos et al. [23]. Recently, an *SH* decomposition of a 3D extension of the HOG descriptor was presented by Liu et al. [24]. Makadia et al. [25] present a method for the registration of 3D point clouds that uses an *SH* representation of Extended Gaussian Images (EGI) to estimate the difference in orientation between point clouds. Several rotation invariant 3D surface shape descriptors have been compared by Huang et al. [6], for the purpose of finding matching poses in sequences with high quality 3D data.

3. Appearance modeling and orientation estimation

3.1. Overview

Figure 1 gives a schematic overview of the proposed procedure for person appearance modeling and orientation estimation. Appearance is modeled as the combination of a 3D shape model and a texture model. We take an intermediate approach between modeling a person using a fixed body shape and doing full articulated pose estimation. We estimate the largest rigid partition (core) of the body over time by regarding all non-rigid elements of the human body (arms, legs) as 'noise' around the rigid body core (torso-head).

The preprocessing step uses multi-camera data to compute a volumetric 3D reconstruction of the scene. A person detection and tracking system lo-

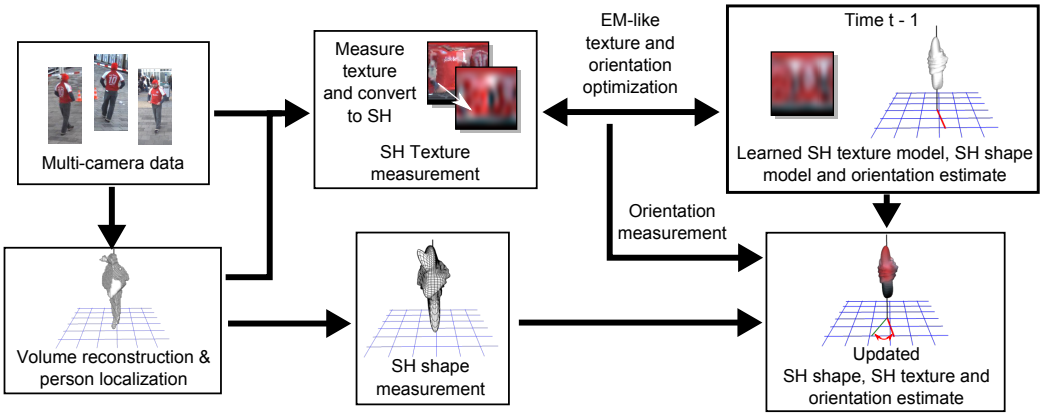
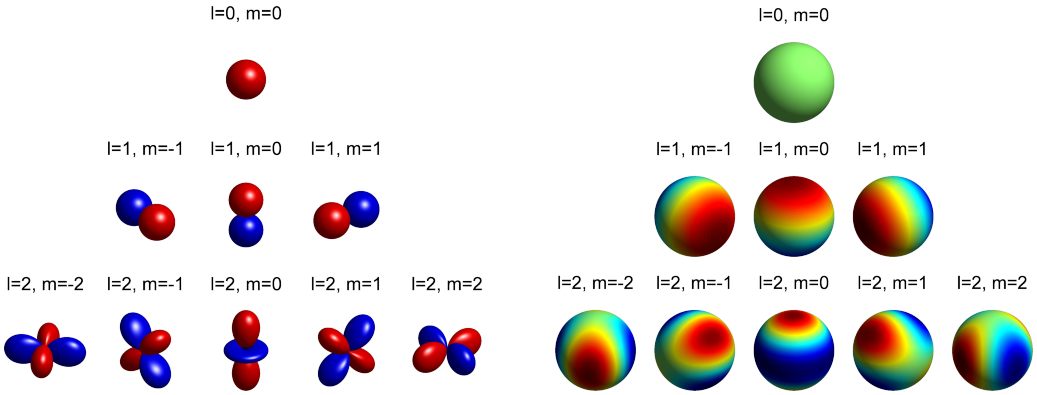


Figure 1: Overview of the appearance modeling and orientation estimation process.

calizes the person of interest, and the volume corresponding to this person is segmented from the volume space. In the next step, an *SH* based shape measurement is created using the segmented volume (section 3.3.2). Furthermore, an *SH* based texture measurement is computed using the learned shape model and Kalman filtered orientation estimate from the previous time step $t-1$ (section 3.3.1). Using an EM-like optimization scheme, an optimized orientation measurement is computed by iteratively comparing the texture measurement to the learned texture model from $t-1$, and adjusting the shape model's orientation to compute an improved texture measurement (section 3.3.3). When the orientation measurement has converged, it is used to correct the Kalman filtered orientation estimate from $t-1$, and the *SH* shape and texture models from $t-1$ are updated using the shape and texture measurements and the estimated orientation.

Our main contribution is a method for estimating a person's relative body orientation while simultaneously generating a basic model of the person's shape and texture. The estimate is made on a per-frame basis using an online learned appearance model consisting of low dimensional *SH* shape and texture representations. By using *SH* as a basis, orientation estimation can be performed elegantly, without the need of explicitly testing for different orientations and without a constraint on the maximum angular difference at successive image frames. The taking advantage of texture information and the SH formulation differentiates our approach from [7, 25]. This paper is based on our earlier work [26].



(a) *SH* components for geometric deformation

(b) *SH* components for gray values (using a gradient map)

Figure 2: Spherical basis functions for the first three *SH* bands ($L = 2$). *SH* components can be visualized by representing spherical function f as a deformation of a sphere (a) or as a texture (gray values visualized using a gradient map) (b).

3.2. Spherical Harmonics

In order to simplify the estimation of a person's orientation at time t based on a shape model $\mathcal{S}_t$ and texture model $\mathcal{T}_t$, we represent both in the linear space of Spherical Harmonics (*SH*). *SH* are the equivalent of Fourier transformations on a 3D sphere; they decompose a spherical function $f(u, v)$ (a function defined on the surface of a sphere) into a linear combination of orthonormal spherical basis functions. An *SH* subspace consists of $L + 1$ bands each characterized by l, $0 \leq l \leq L$, and M components (spherical basis functions) per band. For a certain band l there are $M = 2l+1$ components m in the range $[-l, l]$. If a 3D shape can be represented as a spherical function f_S by extruding a sphere along vertices (u, v), the *SH* decomposition of f_S can be seen as a linear combination of canonical shapes of which figure 2(a) shows the first 3 bands (9 components). Analogous, the *SH* decomposition of a texture represented by f_T as gray values at vertices (u, v) can be seen as a linear combination of textured spheres seen in figure 2(b).

The *SH* decomposition of a spherical function f computes *SH* parameters $\mathbf{A}$, where $\mathbf{A}^{l,m}$ is the weight of component m in band l. *SH* components with $m < 0$ are rotated $90/m$ degrees around the vertical axis compared to their positive counterparts labeled by $|m|$, as can be seen in figure 2. It is therefore common practice to merge the positive and negative order harmonics $\mathbf{A}^{l,m}$

and $\mathbf{A}^{l,-m}$ into a complex item $\mathcal{A}^{l,m}$ as follows:

$$\mathcal{A}^{l,m} = \mathbf{A}^{l,m} + i\mathbf{A}^{l,-m}, \tag{1}$$

where i is the imaginary unit. For $m = 0$, the imaginary part is 0. Since here $0 \leq m \leq l$, this leaves $l + 1$ parameters per band. A reconstruction $SF_{\mathcal{A}}$ of the original function f can be computed from the parameters $\mathcal{A}$ as a weighted sum of the *SH* basis functions (much like the spectral components of a Fourier decomposition are used in the Fourier reconstruction). In practice we use a limited number of bands such that $SF_{\mathcal{A}}$ is a band limited approximation of f. More details on *SH* and how to decompose a spherical function into *SH* components can be found in [27].

An interesting property of *SH* is that they are rotationally covariant. This means that rotating a 3D object and then projecting it onto an *SH* subspace gives exactly the same result as if the object was first projected onto an *SH* subspace and then rotated. Furthermore, the properties of the spherical basis functions and the rules of orthogonality make *SH* rotation a linear operation in which components between bands do not interact. The change in $\mathcal{A}^{l,m}$ is especially simple for rotations on the vertical axis that interest us the most. As suggested in [28], the vertical axis rotation over ϕ of a spherical function represented by *SH* can be represented simply as

$$\mathcal{B}^{l,m} = \mathcal{A}^{l,m} e^{-im\phi}. \tag{2}$$

Here $\mathcal{A}^{l,m}$ is the weight of component m in band l before rotation, i is the imaginary unit and $\mathcal{B}^{l,m}$ is the weight of component m in band l after rotation over angle ϕ. The non-complex *SH* parameters $\mathbf{B}^{l,m}$ and $\mathbf{B}^{l,-m}$ can be computed by taking the real part and the imaginary part of the complex *SH* parameters, respectively. For notational simplicity, we will sometimes leave out the superscript l, m when describing the rotation of a complete *SH* object and use $\mathcal{A}^{\phi}$ to denote $\mathcal{B}$. In those cases, the *SH* rotation operator is assumed to be applied to all components l, m of $\mathcal{A}$.

In case an *SH* parametrization $\mathcal{A}$ and $\mathcal{B}$ of same object is given, the rotation ϕ between them can be found by minimizing the following sum squared error (SSE) for ϕ using a quasi-Newton approach like BFGS [28]:

$$\sum_{1 \leq l \leq L} \sum_{1 \leq m \leq l} (\mathcal{A}^{l,m} e^{-im\phi} - \mathcal{B}^{l,m})^2. \tag{3}$$

The 0^{th} component of every band is not needed for this estimation since it is not influenced by rotation.

3.3. 3D Shape, Texture and Orientation Estimation

At each time step t, C images $I_t^{1:C}$ are captured from C different, overlapping viewpoints. In the experiments, $C = 3$. Foreground estimation is done for each image and a volumetric reconstruction of the scene is created using volume carving. Person detection and tracking is performed in the reconstructed volume space, using the RCTA$^+$ method described in [29]. This method divides the volume space into individual person hypotheses and performs tracking and detection by solving the assignment problem between known person tracks and individual hypotheses for each time step. The visual hull $\mathcal{H}$ of the person of interest is determined by selecting all voxels within a 1 meter diameter cylinder, positioned at the person's location estimated by RCTA$^+$. The measured person position $\hat{x}_t$ is the center of mass of $\mathcal{H}$ and is computed as the center of the principal axis of $\mathcal{H}$. A constant acceleration Kalman filter is used to filter $\hat{x}_t$ over time and gives the filtered person position x_t.

In order to use *SH* to model the person's shape, $\mathcal{H}$ is transformed into spherical shape function f_S. To create f_S, a unit sphere is centered on the person location x_t and the sphere's surface is discretized into a uniformly distributed set of surface points (in our experiments we use 55×55 surface points). For each surface point (u, v), a ray r is cast from x_t, through (u, v). The spherical function is defined by $f_S(u, v) = d$, where d is the distance between x_t and the most distant voxel in $\mathcal{H}$ along r. This is similar to the method described in [30] except that, in order to maintain rotational information, we do not normalize the 3D shape. To compensate for the fact that volume carving tends to over-estimate the shape, the spherical function is scaled down 10% to prevent sampling of the texture outside the object during texture generation.

A sampled texture consists of a spherical texture function f_T, created by projecting the surface points of a shape function f_S onto images $I_t^{1:C}$ and sampling the image values at these locations. Texture is only sampled from image regions containing foreground and the color of surface points visible in multiple cameras is averaged over these cameras.

The goal is to model the person's appearance, consisting of an *SH* shape model $\mathcal{S}_t$ and an *SH* texture model $\mathcal{T}_t$, and estimate the person's orientation ϕ_t (rotation along the vertical axis) based on the estimated appearance.

Algorithm 1 describes the *SH* based appearance modeling and orientation estimation method. The different parts of this method will be explained in the following sub-sections.

3.3.1. Texture Estimation

First, a constant acceleration Kalman filter is used to get a prediction ϕ_t^- of person orientation ϕ_t based on ϕ_{t-1}. Using the shape model $\mathcal{S}_{t-1}$ from the previous time step, the *SH* texture measurement $\hat{\mathcal{T}}_t$ at time t is computed and the person's orientation ϕ_t is estimated using this measurement.

Algorithm 1: Appearance modeling and orientation estimation

Input: $\mathcal{S}_{t-1}$, $\mathcal{T}_{t-1}$, x_{t-1}, ϕ_{t-1}, $I_t^{1:C}$, $\mathcal{H}$
Output: $\mathcal{S}_t$, $\mathcal{T}_t$, x_t, ϕ_t

$\hat{x}_t$ = person position based on average voxel position of $\mathcal{H}$;
x_t^- = Kalman filter prediction of person position using x_{t-1};
x_t = Kalman filter update of x_t^- using measurement $\hat{x}_t$;
ϕ_t^- = Kalman filter prediction of orientation using ϕ_{t-1};
$\Delta\phi = \infty$; Intermediate orientation difference;
`// Determine texture model` $\mathcal{T}_t$ `and orientation` ϕ_t
while $\Delta\phi \geq 0.001$ **do**
    if *not first iteration* **then** $\hat{\phi}_t = \hat{\phi}_t + \Delta\phi$;
    else $\hat{\phi}_t = \phi_t^-$;
    Rotate shape model: $\mathcal{S}_{t-1}^{\phi} = \mathcal{S}_{t-1} e^{im\hat{\phi}_t}$;
    Reconstruct spherical function $SF_{\mathcal{S}^\phi}$ from $\mathcal{S}_{t-1}^{\phi}$;
    Position $SF_{\mathcal{S}^\phi}$ at x_t;
    Project surface points $SF_{\mathcal{S}^\phi}(u,v)$ onto $I_t^{1:C}$;
    Sample texture from $I_t^{1:C}$ at projected surface points (i,j), creating $f_T(u,v) = \frac{1}{C}\sum_C I_t^c(i,j)$;
    Compute *SH* representation $\hat{\mathcal{T}}_t$ of f_T;
    $\Delta\phi = \arg\min_{\hat{\phi}}[\sum_\lambda \sum_l \sum_m (\hat{\mathcal{T}}_t^{l,m,\lambda} e^{-im\hat{\phi}} - \mathcal{T}_{t-1}^{l,m,\lambda})^2]$;
ϕ_t = Kalman filter update of ϕ_t^- using measurement $\hat{\phi}_t$;
if $t < 1/\alpha_{\mathcal{T}}$ **then**
    $\mathcal{T}_t = \frac{t-1}{t}\mathcal{T}_{t-1} + \frac{1}{t}\hat{\mathcal{T}}_t$;
else
    $\mathcal{T}_t = (1-\alpha_{\mathcal{T}})\mathcal{T}_{t-1} + \alpha_{\mathcal{T}}\hat{\mathcal{T}}_t$;
`// Compute shape model` $\mathcal{S}_t$ `based on` ϕ_t
Position a unit sphere with 55×55 discretized surface points at x_t;
foreach *surface point* (u,v) **do**
    Cast ray r from x_t trough (u,v);
    Determine distance d between x_t and the voxel in $\mathcal{H}$ along r furthest away from x_t;
    Compute scaled f_S using $f_S(u,v) = 0.9d$;
Compute the *SH* representation $\hat{\mathcal{S}}_t$ of f_S;
$\forall l, m \in \hat{\mathcal{S}}_t : \hat{\mathcal{S}}_t^{\phi} = \hat{\mathcal{S}}_t e^{-im\phi_t}$;
$\mathcal{S}_0 = \hat{\mathcal{S}}_0; \quad \mathcal{S}_t = (1-\alpha_{\mathcal{S}})\mathcal{S}_{t-1} + \alpha_{\mathcal{S}}\hat{\mathcal{S}}_t^{\phi}$;

Texture function f_T at t is created using a rotated spherical shape function $SF_{\mathcal{S}^\phi}$ reconstructed from the *SH* shape model $\mathcal{S}_{t-1}^{\phi}$, acquired by rotating $\mathcal{S}_{t-1}$

using the predicted orientation ϕ_t^- as follows:

$$\mathcal{S}_{t-1}^{\phi} = \mathcal{S}_{t-1} e^{im\phi_t^-}. \tag{4}$$

See also line 9 of algorithm 1. The exponent in (4) has been negated in order to rotate $\mathcal{S}_{t-1}$ with negative orientation angle $-\phi_t^-$.

When positioned at x_t, $SF_{\mathcal{S}^{\phi}}$'s discretized surface points can be used to sample the person's texture from images $I_t^{1:C}$. Texture regions without color information due to occlusions or sampling outside the foreground regions are filled using the average color along horizontal scanning lines over the texture. This way, artifacts in the *SH* texture representation due to lacking information are prevented while vertical color variance is maintained in the texture. $\hat{\mathcal{T}}_t$ is computed by projecting f_T onto a 9 band *SH* subspace ($L = 8$), reducing the dimensionality of the texture space by 97% and smoothing the texture. To use RGB color, the three color channels $\lambda = \{R, G, B\}$ are modeled as three separate *SH*.

Since rotation of the *SH* texture is simplified according to (2), finding the most likely orientation $\hat{\phi}_t$ given the *SH* texture model from the previous time step $\mathcal{T}_{t-1}$ and the current texture measurement $\hat{\mathcal{T}}_t$ is straightforward and can be solved by minimizing the SSE in equation (5) for $\hat{\phi}_t$ using a quasi-Newton approach like BFGS [28].

$$\sum_{\lambda} \sum_{l} \sum_{m} (\hat{\mathcal{T}}_t^{l,m,\lambda} e^{-im\hat{\phi}} - \mathcal{T}_{t-1}^{l,m,\lambda})^2 \tag{5}$$

A Kalman filter update of ϕ_t^- using $\hat{\phi}_t$ gives the final orientation ϕ_t.

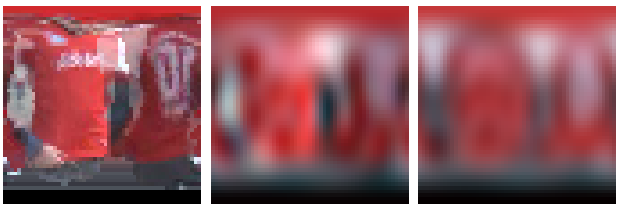
The texture model $\mathcal{T}_t$ is learned over time by exponential decay using learning rate $\alpha_{\mathcal{T}}$:

$$\mathcal{T}_t = (1 - \alpha_{\mathcal{T}})\mathcal{T}_{t-1} + \alpha_{\mathcal{T}} \hat{\mathcal{T}}_t. \tag{6}$$

However, to prevent over representation of the first sampled textures in $\mathcal{T}_t$, iterative averaging is used to learn $\mathcal{T}_t$ as long as $t < \frac{1}{\alpha_{\mathcal{T}}}$, as shown in equation (7).

$$\mathcal{T}_t = \frac{t-1}{t}\mathcal{T}_{t-1} + \frac{1}{t}\hat{\mathcal{T}}_t. \tag{7}$$

Combining models and measurements over time requires each measurement to be rotated into a canonical orientation, matching the model. By sampling the texture using the rotated shape model $\mathcal{S}_{t-1}^{\phi}$, $\hat{\mathcal{T}}_t$ is in a canonical orientation and can directly be combined with $\mathcal{T}_{t-1}$. An example of a sampled texture, its *SH* reconstruction and a learned texture after 100 frames can be found in figure 3.



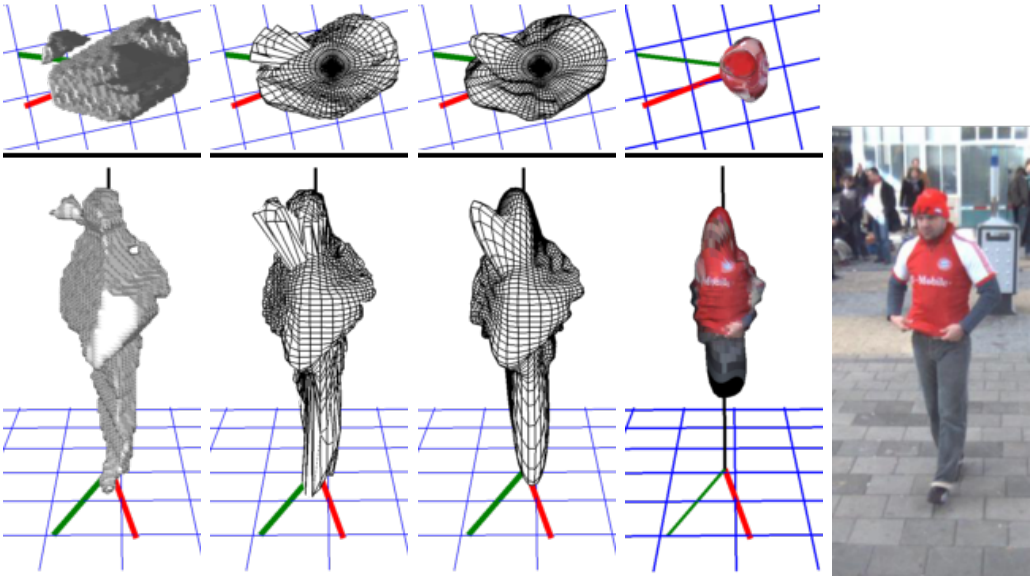
(a) (b) (c)

Figure 3: Examples taken from the person shown in figure 7(c) at frame 100, corresponding to figure 4. (a) Sampled texture (55×55 pixels, 3025 parameters). (b) Reconstructed 9 band (81 parameters) *SH* texture representation. (c) Learned *SH* texture after 100 frames (81 parameters).

3.3.2. 3D Shape Estimation

Using orientation ϕ_t, the *SH* shape model $\mathcal{S}_t$ can be computed. First, the *SH* shape measurement $\hat{\mathcal{S}}_t$ at time t is constructed by transforming visual hull $\mathcal{H}_t$ into a spherical function f_S and projecting this function onto the *SH* space. To reduce feature dimensionality and simultaneously smooth the shape representation, the first 17 bands of the *SH* space ($L = 16$) are used for projection, reducing feature dimensionality by 90%. Examples of a person's visual hull, the spherical function derived from that visual hull and the reconstruction of the *SH* representation of the spherical function can be seen in figure 4(a)-(c). The top-down views showing the person facing to the left clearly show the excess volume created by volume carving due to shape ambiguities. While figure 4(e) shows that, in this frame, the person stands straight with his arms alongside his torso, the top-down view of the 3D volume shows significant extrusions of the estimated shape on the front and back of the person.

Estimating the rigid body over time is done by averaging shape estimates over time, decreasing the influence of non-rigid body parts on the estimated body shape. Since arms and legs will have different positions over time, they will be averaged out in the final shape estimate as can be seen in figure 4(d). To combine shape models and measurements, they have to be in canonical orientation. Since the shape is represented in *SH* components, rotation is straightforward (as mentioned in sec. 3.2) and does not yield rotation artifacts which would occur when rotating objects in the discrete volume space consisting of cubic voxels in a regular grid. The rotated *SH* shape measurement $\hat{\mathcal{S}}_t^{\phi}$ is computed as shown in equation (8) (line 27 in alg. 1), in



(a) (b) (c) (d) (e)

Figure 4: Examples taken from the person shown in figure 7(c). Green lines on the groundplane show the ground truth orientation. Red lines show the estimated orientation. The vertical black line represents the person's principal axis and is 2 m high. (a)-(d) show a top-down view of the person facing to the left (top) as well as from a perspective matching (e) (bottom). All figures are shown at the same scale. (a) Person visual hull $\mathcal{H}$. (b) Spherical shape function using 55×55 surface points (3025 parameters). (c) Reconstructed 17 band (289 parameters) SH shape representation. (d) Learned shape model after 100 frames, mapped with sampled texture. (e) Person modeled.

accordance with (2):

$$\hat{\mathcal{S}}_t^{\phi} = \hat{\mathcal{S}}_t e^{-im\phi_t}. \tag{8}$$

The rigid body shape model $\mathcal{S}_t$ at time t is computed over time by iteratively combining $\hat{\mathcal{S}}_t^{\phi}$ for all time steps under exponential decay, as shown in equation (9)

$$\begin{aligned} \mathcal{S}_0 &= \hat{\mathcal{S}}_0; \\ \mathcal{S}_t &= (1-\alpha_{\mathcal{S}})\mathcal{S}_{t-1} + \alpha_{\mathcal{S}}\hat{\mathcal{S}}_t^{\phi} \end{aligned} \tag{9}$$

The shape learning rate, $\alpha_{\mathcal{S}}$, is split up in two parts: one for growing the model and one for shrinking it. Because volume carving reconstructions are convex, artifacts in the reconstruction are much more likely to consist of extrusions of the actual shape than of indentations. Therefore, measuring an indentation in $\hat{\mathcal{S}}_t^{\phi}$ compared to $\mathcal{S}_{t-1}$ conveys more information about the actual object shape than when measuring an extrusion. This is reflected in the learning rates. The learning rate for surface points that are more indented in $\hat{\mathcal{S}}_t^{\phi}$ than in $\mathcal{S}_{t-1}$ is set to 0.4. The learning rate for more extruded surface points is set to 0.01. In figure 4(a), a small artifact is visible near the head of the person. While this artifact gives rise to an erroneous extrusion in the spherical function (figure 4(b)) and its *SH* representation (figure 4(c)), the learned body model in figure 4(d) shows no influence of this artifact. The figure furthermore shows that the model was able to learn a good approximation of the actual person's volume.

3.3.3. Iterative Orientation Estimation

The estimate of the current body orientation made using equation (5) allows for an iterative orientation optimization scheme. Since the texture measurement $\hat{\mathcal{T}}_t$ is sampled using the reconstruction of $\mathcal{S}_{t-1}$, oriented according to the Kalman filter prediction ϕ_t^- instead of the final estimate ϕ_t, the sampled texture might be distorted.

In order to optimize the estimation of the person orientation, texture sampling and orientation estimation are repeated multiple times. Each time, the reconstruction of $\mathcal{S}_{t-1}$ is rotated using the most likely orientation estimate $\hat{\phi}_t$ of the previous iteration. While the estimate gets closer to the true orientation, the SSE gets smaller and the difference between orientation estimates reduces. Optimization is stopped when the difference between consecutive orientation estimates is less than 0.001 rad or 10 iterations have

been done. In figure 4, red lines on the ground plane show the resulting orientation estimate on a single frame, while green lines show the ground truth orientation.

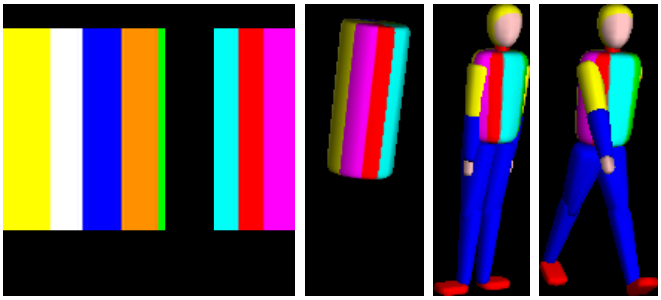
$\mathcal{S}_{t-1}$ could be updated every iteration using $\hat{\mathcal{S}}_t^{\phi}$, rotated using the previous iteration's estimate of $\hat{\phi}_t$. However, while the orientation estimate is suboptimal, an incorrectly rotated version $\hat{\mathcal{S}}_t^{\phi}$ might result in a more distorted shape estimate instead of an improved estimate.

4. Experiments

We evaluate how the *SH* texture representation, shape information and the iterative estimation procedure influence the quality of the estimated orientation. To this end, our method is compared to a state-of-the-art approach as well as to an alternative method. The first method is based on the Panoramic Appearance Map (PAM) [5]. A fixed size cylinder with a diameter of 35 cm and a height of 2 m, fitting tightly around a person's torso, is used for sampling the person's texture. This is much smaller than the 1 m diameter cylinder used for segmenting the person of interest from the volume space mentioned in section 3.3, but while the volume segmentation should contain all voxels belonging to the reconstructed person, the PAM cylinder should capture as little background as possible to generate stable textures. Orientation is estimated in a generate-and-test fashion by sampling 360 textures using 1° orientation difference and making a per-pixel weighted comparison between all textures and a learned texture model. The orientation of the best matching texture sample is used as the object orientation.

As a secondary, alternative method, we combine the cylinder based shape model with our *SH* based texture representation and use the iterative orientation estimation from section 3.3.3. This method will be referred to as 'cylinder shape with *SH* texture'. Like our method, both PAM and the cylinder shape with *SH* texture provide relative orientation estimates with respect to the initial person orientation.

The following settings are used for the experiments. Textures are sampled in standard RGB colorspace. Preliminary experiments were done using both normalized RGB and C-invariant color spaces [31], but the overall best results were obtained using standard RGB. The voxelspace has a resolution of $2 \times 2 \times 2$ cm/voxel. Textures and shapes are sampled at a resolution of 55×55 surface points (3025 parameters) and a texture learning rate $\alpha_{\mathcal{T}} = 0.05$ is used. For the methods using *SH* based textures 9 *SH* bands (81 parameters)



(a) (b) (c) (d)

Figure 5: (a) Synthetic texture with vertical blacked-out bar simulating missing information. (b) Rendered cylinder. (c) Rendered human. (d) Rendered articulated human.

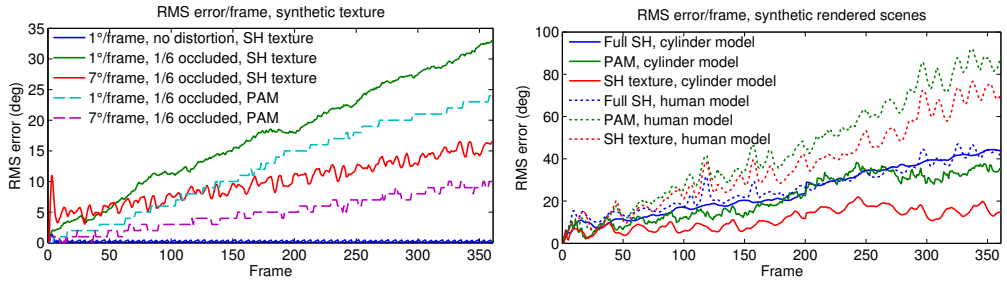
are used for texture representation. For the *SH* shape representation 17 *SH* bands (289 parameters) are used. For all methods, each frame's orientation estimate is constrained to be within 30° of the previous frame's estimate.

Experiments are done on three artificial datasets as well as on a real-world dataset. Doing experiments on artificial data allows us to test different aspects of the algorithms such as the influence of shape estimation and the use of *SH* for the texture representation.

As a measure of the orientation estimation performance, the Root Mean Squared (RMS) error between the ground truth orientation and the estimated orientation is used. The difference in orientation is computed using the angular difference between the two orientation angles. This ensures a correct error computation considering angular wrap-around ($0° = 360°$).

4.1. Synthetic Texture

The first artificial dataset uses a pre-generated 2D texture image, shown in figure 5(a), in order to test the performance of texture based orientation estimation. Using a fixed texture ensures that the measurements are not influenced by inaccurate shape estimates and the texture sampling method. A texture of 360×360 pixels is used, allowing for 1° rotations. Rotation is simulated by circular shifting this texture a number of pixels to the right at each time step. For two runs of the experiment a static black area is added to simulate a static sampling occlusion (e.g. a part of the object not seen by any camera), shown as a vertical blacked-out bar in figure 5(a). The orientation estimation methods treat this part of the texture as if no information is available for this region. Orientation is estimated using a



(a) (b)

Figure 6: (a) Difference between ground truth orientation and estimated orientation for artificially created textures. (b) RMS error per frame over cylinder shape based and human shape based synthetic experiments.

55×55 pixels resized version of the texture to simulate sampling. One run is done with a fully visible texture and 1° rotation per iteration. Two runs are done using the occluded texture using either 1° or 7° rotation per iteration. Since shape has no role in this experiment, we only compare between PAM using generate-and-test based orientation estimation, and the *SH* texture using our iterative orientation estimation.

Figure 6(a) shows the RMS error per frame for the different scenarios and methods. When using the fully visible texture, estimation results are as good as perfect. As is to be expected, the generate-and-test based method (not shown in the graph) does a perfect job here. The lightly jagged profile of the error of the *SH* based method on this texture is an effect of the fact that the downsampled texture is used to estimate 1° orientation shifts. When the original 360 pixels wide texture is shifted one pixel per degree, the changes in the downsampled texture (shifted about 0.15 pixels per degree) are very small. The *SH* based method does not pick up on this shift until the downsampled texture is shifted one pixel.

When a part of the texture is blacked-out, the *SH* based method is outperformed by the generate-and-test based method. The *SH* based texture model takes slightly longer to learn, causing a higher error for the first few frames. Tests with an increasing number of *SH* texture components produce similar results, suggesting that the *SH* texture combined with the iterative orientation estimation procedure is less accurate than the generate-and-test based method when the estimation task is this much simplified. After the initial offset in error, both methods show a similar gradual increase in the

orientation error over time. This is an artifact of doing relative orientation estimation. The use of an incrementally updated texture model as the reference makes it hard to compensate a drift in the estimated orientation over time. Drifting estimates are caused by inaccurate texture matches due to the blacked-out patch, combined with a learning rate that includes this shift into the model.

4.2. Rendered Scenes

The second artificial dataset shows a rendered scene containing a textured, rotating cylinder (figure 5(b)). For the last artificial dataset, a scene showing a rotating textured human model is rendered (figure 5(c)-(d)). The camera viewpoints and calibration for these scenes are the same as for the real-world data, as is the image resolution of 752×560 pixels. In all scenes, the object is kept positioned at one location in the scene. For both the rendered cylinder and the human models, sequences are generated using either 1° or 7° rotation per frame. In one cylinder sequence, the cylinder incrementally stands still, rotates left, stands still and rotates right for 50 frames each. In the last human model sequence, the model is rotated 1° per frame while the limbs show walking motion (fig 5(d)).

The system setup for the methods compared is the same as for the real-world experiments, except for the person tracking and detection part. Since foreground segmentation is provided and perfect, the volume reconstruction of the rendered scene only contains the reconstruction of the object of interest. Therefore, the object location is computed as the center of the principle axis of the reconstructed scene. In the cylinder sequences and the first two human sequences, the assumption of a rigid body is fully met. In the last human sequence, rigidity only holds for the torso and head. This allows evaluation of the influence of shape estimation under the rigid body assumption.

Figure 6(b) shows the RMS error per frame for the rendered cylinder data as well as for the human model data. For the cylinder scenario, the *SH* texture with cylindrical shape model benefits from its matching shape model. No time is needed for the methods to adapt to the object shape, resulting in well sampled textures from the first frame on. PAM also has this benefit, but suffers from texture sampling artifacts resulting in a higher error rate. The low dimensional *SH* texture is able to learn a more general texture model, resulting in better performance. For this scenario, our method has a drawback from having to estimate the object shape, resulting in lower

performance than the *SH* texture with cylindrical shape model, but it still performs on-par with PAM.

Like in the previous experiment, a gradual increase in error is visible for all methods for both the cylinder shape scenarios as well as the human shape scenarios. Picking the correct learning rate for the shape and texture model can compensate for this to some degree, but over time the error is likely to converge towards the random error.

For the human model, and specifically the articulated human model, the cylindrical assumption does not hold. It causes continuous distortions in the texture, giving rise to drifting orientation estimates. While the need to learn the shape of the object in our method seems to be a drawback for simple shapes, experiments on the more complicated human shape model show the benefit of modeling the object shape together with the texture. Our full *SH* model shows similar performance on the cylindrical shape and the human shape, but the methods using the cylindrical shape model have a clear decline in performance for the more complex object. The shape estimation still takes time to adapt to the shape, but in the end gives a more stable shape model to sample the texture from.

4.3. Real-World Scenes

The real-world data consists of about 2800 frames, distributed over 12 scenarios recorded in an open, outdoors environment with uncontrollable illumination. Three cameras with fully overlapping views are used recording at 20 Hz. While some scenarios feature multiple persons, orientation estimation is only performed for one person. The nine different persons involved in the scenes can be found in figure 7(a)-(i), while figure 7(j) shows sample frames from three scenarios and all camera viewpoints.

Ground-truth (GT) for the real-world data is created by annotating 17 distinctive locations (joint and head positions) on the body of each person of interest in each camera. A fully articulated 3D person model is fitted onto these annotated points using least-squares optimization. The model's torso orientation is taken to be the person body facing direction. Doing this for a regular interval of frames results in a positional accuracy of about 4 cm.

For the experiments on this dataset a third, classifier based orientation estimation method is added. This method, introduced by [13], uses a classifier combining HOG-based features to get an absolute orientation estimate per frame, with respect to a fixed 3D coordinate system. It is complementary to our method, using a fundamentally different way of orientation estimation,

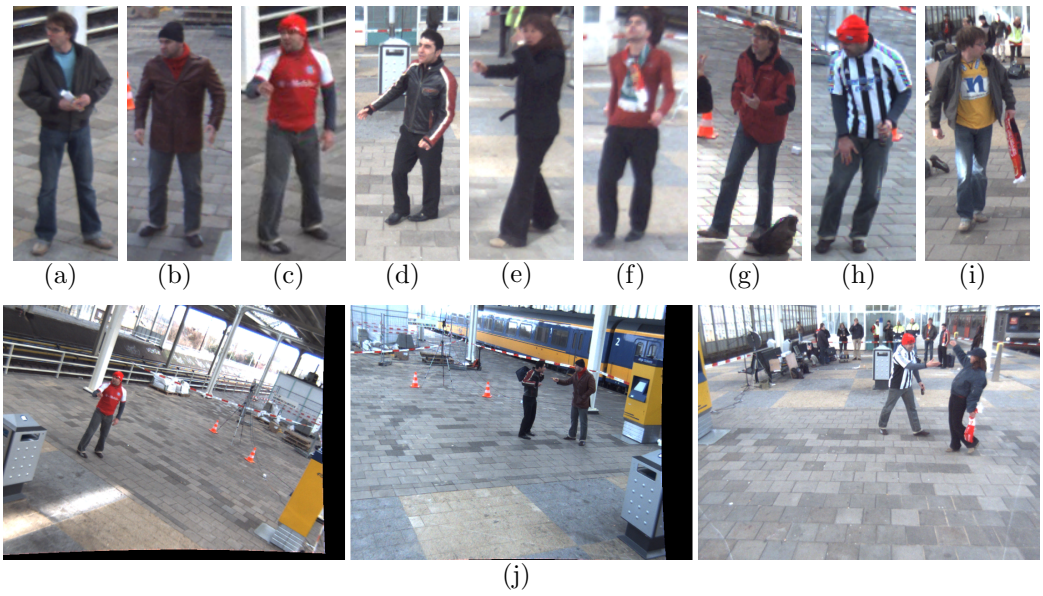


Figure 7: (a)-(i) The 9 persons used in the 12 scenarios. (j) Samples of scenarios, showing the 3 camera viewpoints and 3 scenarios.

and is added as a reference for absolute orientation estimation methods. It uses a mixture of four orientation experts, each trained on images showing pedestrians in one of four canonical orientations (front, back, left, right). The experts' classification results are used as weights in a Gaussian Mixture Model, creating a full 360° orientation probability density function (*pdf*). A maximum likelihood estimate derived from this *pdf* is used as the final orientation estimate. The experts are trained using data kindly provided by the authors of [13]. We generate regions of interest in the 2D images based on the projection of the volume reconstruction of the person of interest onto the camera plane. Orientation estimates per camera are combined into a 3D orientation estimate using the camera calibration. Applying the constraint on the maximum difference in estimated orientation between consecutive frames to the classifier based approach is done by limiting the range of the *pdf* when computing the maximum likelihood orientation. Since the classifier based approach is time independent by nature, we also test this method without applying the 30° orientation constraint. For both versions, the results have been Kalman filtered resulting in more stable and smooth orientation estimates over time.

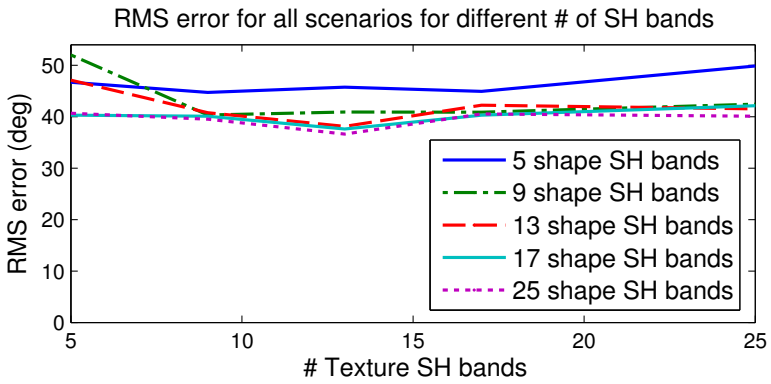


Figure 8: RMS error over all scenarios and all frames for different numbers of texture and shape *SH* bands. Showing RMS error for all combinations of 5, 9, 13, 17 and 25 *SH* texture and shape bands ($L \in \{4, 8, 12, 16, 24\}$).

Finally, results of the version of our method as presented in [26] are added for comparison. This version uses manually generated 'optimal' foreground segmentations to create the volume reconstruction and does not make use of RCTA$^+$ tracking results. Results from this method are referred to as '*SH* shape and texture (manual fg)'.

Figure 8 shows how the number of *SH* bands used influences orientation estimation performance for our method. Orientation estimation has been performed for all scenarios, using 5, 9, 13, 17 or 25 bands for both the *SH* texture and the *SH* shape ($L \in \{4, 8, 12, 16, 24\}$). This results in 25 performance measurements. Using a small number of bands has a clear negative impact on performance. This is shown by the blue line (only 5 shape bands) and the first part of the green and red lines (only 5 texture bands). However, a comparison with the cyan and magenta lines shows that using less bands for shape has a larger impact in performance than using less bands for texture. This is due to the fact that the texture depends on the shape for sampling, and gets more distorted when using less shape *SH*. For higher numbers of bands (17, 25), adding extra bands has limited influence on performance since the extra details modeled by the *SH* representation get decreasingly relevant. For the rest of the experiments 9 texture *SH* bands and 17 shape *SH* bands are used, balancing performance and computational efficiency. Using more bands results in a slight increase in performance at the cost of much more parameters (e.g. using 25 instead of 17 bands results in 625 instead of 289 parameters).

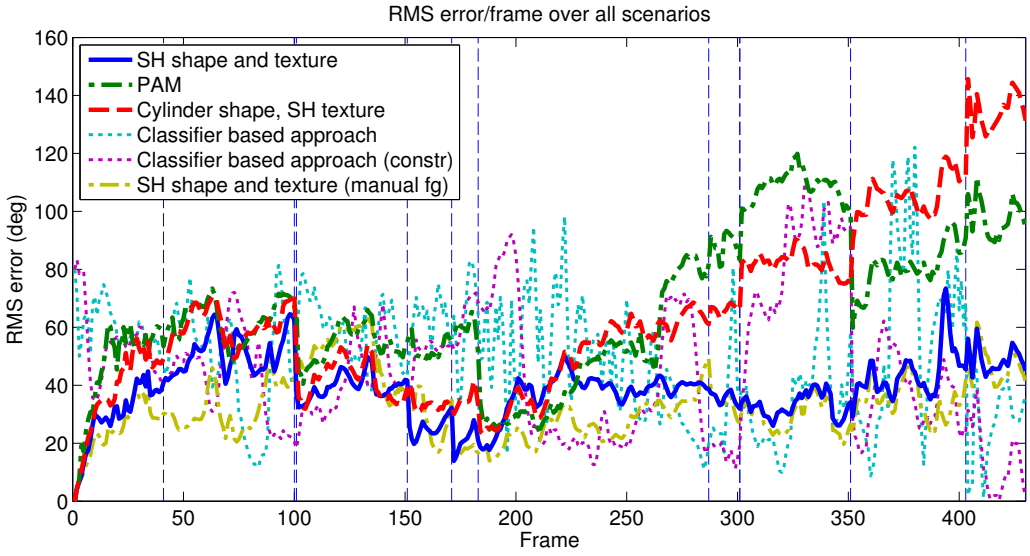


Figure 9: RMS error per frame over all scenarios, vertical lines mark scenario end frames.

Figure 9 shows the RMS error per frame for the real-world data, computed over all scenarios. This differs from the results shown in Figure 4(a) in [26] where the moving RMS error over time was used, giving a more smoothed view of the performance. Since not all sequences have the same length, vertical dashed lines indicate the points where scenarios end and the RMS error is taken over fewer scenarios. Please note that since our method, PAM and the cylinder shape with *SH* texture all provide relative orientation estimates w.r.t. the first frame, their error is defined to be 0 at that point. Because the classifier based approach gives an absolute orientation estimate for each frame, it also shows an error for the first frame.

Measured over all scenarios, our method shows better performance compared to the other relative orientation methods for almost all frames. Only the version of our method from [26] performs better, which is expected because of the 'optimal' foreground segmentations. In the first 40 frames the error rises quickly because of sub-optimal shape and texture models. After this, the error for our method seems to stabilize while the other errors keep rising. A comparison with the method using the cylinder together with *SH* texture shows that our method benefits from modeling the shape of the person. Representing the texture in low-dimensional *SH* space gives a comparable performance to PAM's full texture approach while the 9 *SH* bands

reduce the number of features by 97%. However, when the person being tracked has a low-contrast appearance like the one shown in figure 7(e), the *SH* texture lacks detail.

The relative orientation estimation methods using a fixed shape model exhibit a significantly stronger error increase over time. This effect is similar to the one seen in the experiments using the synthetically rendered human model. The incorrect shape model causes more noise in the sampled texture, causing a larger deviation in the estimated orientation over time. Around frame 350, a jump in the RMS error for PAM and the cylinder shape with *SH* texture is visible. This is caused by the ending of the scenario with the person shown in figure 7(h), who's repetitive shirt pattern poses an issue for PAM.

Directly comparing results between the classifier based, absolute estimation methods and the relative estimation methods is difficult, because of the different nature of these methods. In the short term, the frame-by-frame absolute orientation estimates show a bit more erratic behavior. The error spikes show the sensitivity of the classifier based method to orientation ambiguities between opposite orientations (errors of about 180°), caused by the ambiguity in person shape when viewed from the front or the back. In the long term however, the absolute nature of the method makes it invulnerable to drifting of the orientation estimate. For the relative methods, the error is likely to converge to the random error over time since they lack a fixed reference point. Constraining the orientation difference per frame for the classifier based method dampens the erratic behavior and seems to result in slightly more stable orientation estimates.

Figure 10 shows each method's cumulative error distribution, obtained by binning the errors over all frames. Our method clearly outperforms the other relative estimation methods with respect to the error distribution. Only our method using manually created foreground segmentations shows better performance. This graph confirms that PAM has the poorest error rates over all. The cylinder shape and *SH* texture method specifically shows less mid-range errors between 30° and 100° compared to PAM, while our method consistently shows less large errors than the other two relative estimation methods. The graph also confirms the classifier approaches' sensitivity to errors around 180°, shown by the final increase of the graph from 140° onwards. This matches the results from figure 6 of [13].

As a final measure, the RMS error measured over all scenarios and all frames together is computed for all methods. These results can be found in

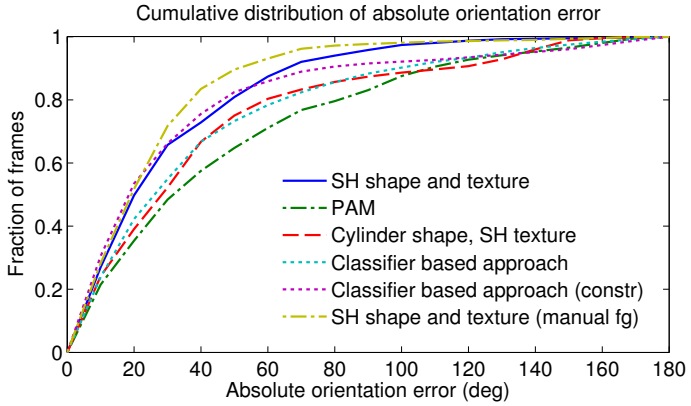


Figure 10: Cumulative absolute error distributions over all scenarios.

Method	RMS error
SH shape and texture (Our method)	40°
PAM [5]	62°
Cylinder shape, SH texture	56°
SH shape and texture (manual fg) [26]	34°
Classifier [13]	55°
Classifier (constrained)	50°

Table 1: RMS error for each method, measured over all scenarios and all frames together.

table 1. When comparing the relative orientation estimation methods, the method using manual foreground segmentations performs best. However, using tracking results instead of manually created foreground segmentations only gives a slight performance loss of 6°. Of the methods using real tracking results, our method shows the lowest average error, followed by the cylinder shape with *SH* texture with a 16° higher RMS error. These results show that relative orientation estimation benefits from the low dimensional *SH* texture representation and shape modeling. For the classifier based approaches, the constrained version shows the lowest error.

An example of the orientation estimation performance on a frame-to-frame basis in one scenario is shown in figure 11. In order not to clutter the graph too much, we only show the constrained classifier results here, since it gave the most stable results in the previous experiments. The graph demonstrates that our method follow the GT orientation closely, while both

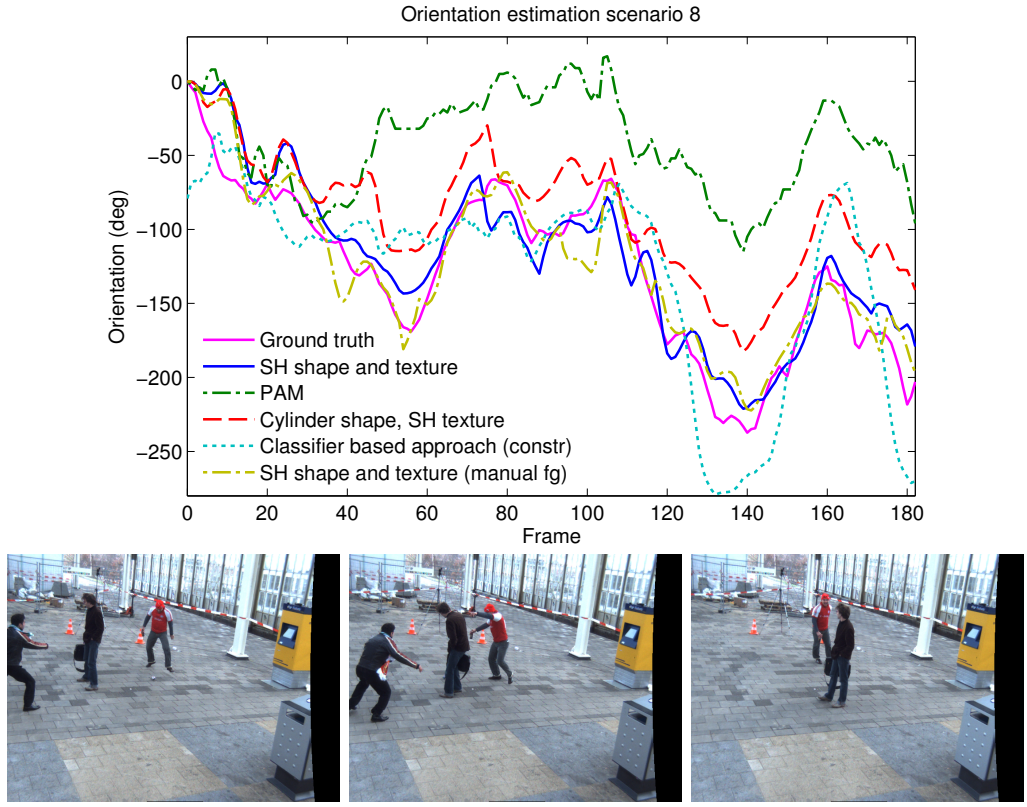


Figure 11: (top) Example of the estimated orientation of the person from figure 7(c) (with the red outfit) using the four methods. (bottom) Frames 48, 98 and 148 form this scenario. Notice some occlusion has occurred between frames 98 and 148.

PAM and the cylinder with *SH* texture method drift away from the GT. Around frame 40, the person jumps up and down and bends over, resulting in a bad match between the cylindrical shape model and the actual person shape. This causes a drift in the estimated orientation for PAM, which cannot be compensated. While PAM still follows the curve of the GT orientation for the rest of the scene, the offset remains. The *SH* texture model shows more robustness to this deformation, resulting in a much smaller drift for the cylinder model with *SH* texture. Around frame 110, the tracked person is occluded in one of the cameras causing a slight drift in orientation for the cylinder shape with *SH* texture model. Our method shows more robustness to this occlusion. The method using manually created foreground segmentations follows the GT a bit more closely than our method for most of

the scene, but does not show significantly better performance. The classifier based approach does a reasonable job following the GT, but its frame-wise estimation approach causes a bit more erratic orientation estimates resulting in less accurate Kalman filtered results.

All experiments were done on a single core of a 2.6 Ghz Intel Xeon CPU. Average processing times were: 9.5 s/fr for our method, 1 s/fr for PAM, 1.7 s/fr for the cylinder with *SH* texture and 0.7 s/fr for the classifier based approach. Volume carving and computation of the spherical shape function took about 8 s of the 9.5 s our method needs per frame, using a crude implementation. A large speed improvement is possible by using a GPU implementation. The classifier based approach was implemented in C++, while the other methods contain unoptimized Matlab code.

5. Conclusion

We presented a novel approach for estimating the relative person orientation, based on a low dimensional shape and texture representation using Spherical Harmonics; this involves a reduction in the number of appearance modeling parameters by $90 - 97\%$. Results on synthetic data show that, when the object shape is rigid and known in advance, fixed shape models combined with a learned *SH* texture model offer the best orientation estimation performance. However, when the object shape is not entirely rigid or precisely known, like in the case of a person's body, the addition of a learned SH shape model is beneficial.

Results could be improved by using more advanced inference methods like particle filters instead of the maximum likelihood approach employed here. The *SH* based orientation estimation could be extended to estimate the object's rotation along all axis. Finally, the fusion of our relative orientation estimates with classifier based absolute orientation estimates might further improve performance.

Acknowledgments

The authors would like to thank Leo Dorst from the University of Amsterdam for proof reading the paper. This research has received funding from the EC's Seventh Framework Programme under grant agreement number 218197, the ADABTS project.

References

[1] H. Ben Shitrit, J. Berclaz, F. Fleuret, P. Fua, Tracking multiple people under global appearance constraints, in: Proc. of the IEEE International Conference on Computer Vision, 2011, pp. 137–144.

[2] C.-H. Kuo, C. Huang, R. Nevatia, Multi-target tracking by on-line learned discriminative appearance models, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 685–692.

[3] M. Farenzena, L. Bazzani, A. Perina, V. Murino, M. Cristani, Person re-identification by symmetry-driven accumulation of local features, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2360–2367.

[4] D. Gray, H. Tao, Viewpoint invariant pedestrian recognition with an ensemble of localized features, in: Proc. of the European Conference on Computer Vision, no. 5302 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2008, pp. 262–275.

[5] T. Gandhi, M. M. Trivedi, Person tracking and reidentification: Introducing panoramic appearance map (PAM) for feature representation, Machine Vision and Applications 18 (3) (2007) 207–220.

[6] P. Huang, A. Hilton, J. Starck, Shape similarity for 3D video sequences of people, International Journal of Computer Vision 89 (2010) 362–381.

[7] D. Mitzel, B. Leibe, Taking mobile multi-object tracking to the next level: People, unknown objects, and carried items, in: Proc. of the European Conference on Computer Vision, no. 7576 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2012, pp. 566–579.

[8] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, H. P. Seidel, Motion capture using joint skeleton tracking and surface estimation, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1746–1753.

[9] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, H. W. Haussecker, Detailed human shape and pose from images, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[10] P. F. Felzenszwalb, D. P. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005) 55–79.

[11] M. Hofmann, D. M. Gavrila, 3D human model adaptation by frame selection and shape-texture optimization, Computer Vision and Image Understanding 115 (11) (2011) 1559–1570.

[12] T. Gandhi, M. M. Trivedi, Image based estimation of pedestrian orientation for improving path prediction, in: Proc. of the IEEE Intelligent Vehicles Symposium, 2008, pp. 506–511.

[13] M. Enzweiler, D. M. Gavrila, Integrated pedestrian classification and orientation estimation, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 982–989.

[14] H. Shimizu, T. Poggio, Direction estimation of pedestrian from multiple still images, in: Proc. of the IEEE Intelligent Vehicles Symposium, 2004, pp. 596–600.

[15] C. Chen, J. Odobez, We are not contortionists: Coupled adaptive learning for head and body orientation estimation in surveillance, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1544–1551.

[16] M. Andriluka, S. Roth, B. Schiele, Monocular 3D pose estimation and tracking by detection, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 623–630.

[17] B. Leibe, K. Schindler, N. Cornelis, L. V. Gool, Coupled object detection and tracking from static cameras and moving vehicles, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008) 1683–1698.

[18] M. Liem, D. M. Gavrila, Multi-person localization and track assignment in overlapping camera views, in: Proc. of the DAGM symposium on pattern recognition, no. 6835 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2011, pp. 173–183.

[19] A. Mittal, L. S. Davis, M 2 tracker: a multi-view approach to segmenting and tracking people in a cluttered scene, International Journal of Computer Vision 51 (3) (2003) 189–203.

[20] K. N. Kutulakos, S. M. Seitz, A theory of shape by space carving, International Journal of Computer Vision 38 (3) (2000) 199–218.

[21] K. Cheung, S. Baker, T. Kanade, Shape-From-Silhouette across time part i: Theory and algorithms, International Journal of Computer Vision 62 (3) (2004) 221–247.

[22] Z. Yue, W. Zhao, R. Chellappa, Pose-encoded spherical harmonics for face recognition and synthesis using a single image, EURASIP Journal on Advances in Signal Processing 2008 (1) (2008) 65:1–65:18.

[23] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, D. V. Vranić, Feature-based similarity search in 3D object databases, ACM Computing Surveys 37 (2005) 345–387.

[24] K. Liu, H. Skibbe, T. Schmidt, T. Blein, K. Palme, O. Ronneberger, 3D Rotation-Invariant description from tensor operation on spherical HOG field, in: Proc. of the British Machine Vision Conference, 2011.

[25] A. Makadia, A. Patterson, K. Daniilidis, Fully automatic registration of 3D point clouds, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2006, pp. 1297–1304.

[26] M. C. Liem, D. M. Gavrila, Person appearance modeling and orientation estimation using spherical harmonics, in: Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition, 2013, pp. 1–6.

[27] R. Green, Spherical harmonic lighting: The gritty details, in: Game Developers Conference, 2003.

[28] A. Makadia, K. Daniilidis, Direct 3D-rotation estimation from spherical images via a generalized shift theorem, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2003, pp. II– 217–24.

[29] M. C. Liem, D. M. Gavrila, A comparative study on multi-person tracking using overlapping cameras, in: Proc. of the International Conference on Computer Vision Systems, no. 7963 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 203–212.

[30] D. Saupe, D. V. Vranić, 3D model retrieval with spherical harmonics and moments, in: Proc. of the DAGM symposium on pattern recognition, no. 2191 in Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2001, pp. 392–397.

[31] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, H. Geerts, Color invariance, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (12) (2001) 1338–1350.