Real-time Volumetric Reconstruction and Tracking of Hands in a Desktop Environment

Christoph John[1,2], Ulrich Schwanecke[2], and Holger Regenbrecht[1]

[1] University of Otago, New Zealand
[2] University of Applied Sciences Wiesbaden, Germany

Abstract. A probabilistic framework for vision based volumetric reconstruction and marker free tracking of hand and face volumes is presented, which exclusively relies on off-the-shelf hardware components and can be applied in standard office environments. Here a 3D reconstruction of the interaction environment (user-space) is derived from multiple camera viewpoints which serve as input sources for mixture particle filtering to infer position estimates of hand and face volumes. The system implementation utilizes graphics hardware to comply with real-time constraints on a single desktop computer.

Key words: Probabilistic Shape From Silhouette, Mixture Particle Filtering

1 Introduction

Virtual and mixed reality environments rely on the implementation of (tele) presence: the perceived sense that a user's own body and body parts belong to the artificial world presented. Of paramount importance here is the efficient and accurate registration, tracking, reconstruction and display of the head and hands of a human operator.
In the following we present an approach for the reconstruction and tracking of hands and head (skin-colored objects) in a potential standard office environment which works with off-the-shelf hardware components. The supervised volume in our table-top environment (see figure 1) has a hand tracking volume size of $1.0\text{m} \times 1.0\text{m} \times 0.75\text{m}$. The lighting conditions have been constrained to controlled and reasonably well lit office room, following the recommendations of IEEE Std. 241 [1].
The system consists of a flock of six color cameras which are utilized to compute a volumetric reconstruction of the user-space. A variant of a *probabilistic Shape from Silhouette* (pSfS) algorithm, first introduced by Landabaso and Pardas [4] has been developed. Unlike *traditional SfS* (tSfS), which performs object segmentation in the image domain, pSfS utilizes a 3D probabilistic background model. This shifts object segmentation into the spatial domain and leads to improved segmentation results in presence of image noise or clutter. We have extended pSfS by imposing constraints on the 3D foreground process in terms of anticipated color and occupancy of hand and face volumes, thus limiting volumetric reconstruction to skin-colored foreground regions. This leads to more detailed reconstructions and an increased probabilistic distance to the background scene. In addition we allow dynamic per pixel on/off switching of cameras to allow the integration of occlusion masks and to stabilize reconstruction results in

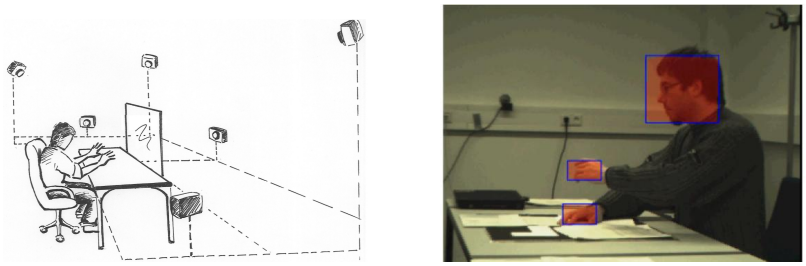

Fig. 1. Proposed environment. Hands and face are tracked in front of the projection screen.

presence of occlusion.
In [4] volumetric reconstructions have been projected into images to generate occlusion masks needed for background model update. We instead utilize the derived visual hulls as input source for a variant of mixture particle filtering [7] to estimate positions of hand and face volumes. Occlusion masks are then generated from tracked bounding boxes. This has the advantage that masks can be computed efficiently and that volumetric reconstruction errors do not degenerate the background model. Finally we present a GPU implementation of the presented system, which in contrast to [4] permits the whole system to run in real-time on a consumer graphics card and a single desktop computer.

2 Probabilistic Volume Reconstruction

The reconstruction algorithm presented below is based on probabilistic reasoning and can be subdivided into an image and volume based classification part. In the image based part a measure is assigned to each pixel which exhibits its probability of belonging to a skin-colored foreground silhouette. In the volume based part these silhouettes are utilized to derive volumetric reconstructions.

2.1 Image based Likelihood Evaluation

The task of skin-colored foreground object segmentation can be formulated as a classification problem at pixel level. A pixel may belong to one of four groups which are given as the possible combinations of fore-/background and skin/non-skin color. Pixel likelihood evaluations are casted as *maximum a posteriori* (MAP) assignments in a discriminative model. I.e., the model expresses the per pixel probability of belonging to the foreground with the skin-colored class $P'(F, S|\mathbf{c})$ as a function of its observed color vector $\mathbf{c}$. The prime denotes augmentation with an outlier model which will be described in detail later. Here $\mathbf{c} = [\mathrm{r}, \mathrm{g}]^T$ is represented in the normalized-rg color space.
A combination of two classifiers constitutes our discriminative model (see figure 2). The first classifier $P'(F|\mathbf{c})$ is based on a model of the background process and estimates per pixel foreground probabilities by combining the MAP assignment of being foreground $P(F|\mathbf{c})$ with an outlier model. Equivalently the second classifier $P'(S|\mathbf{c})$

(a) $P'(F|\mathbf{c})$ (b) $P'(S|\mathbf{c})$ (c) $P'(F,S|\mathbf{c})$

Fig. 2. Left: foreground classification; Middle: skin classification; Right:combined classification

augments the MAP estimate of being skin color $P(S|\mathbf{c})$. The final per pixel classification scheme is thus given by:

$$P'(F,S|\mathbf{c}) = P'(F|\mathbf{c}) \cdot P'(S|\mathbf{c})$$

Our setup has been constrained to office environments with a fixed camera setup. This leads to a relatively static background scene which can be modeled with a *single Gaussian model* (SGM) [3]. A SGM is defined in the bivariate case with the mean normalized-rg color vector μ and covariance matrix Σ as:

$$P(\mathbf{c}|\mu,\Sigma) = \frac{1}{2\pi\left|\Sigma\right|^{\frac{1}{2}}} \cdot \exp\left(-\frac{1}{2}\left(\mathbf{c}-\mu\right)^T \cdot \Sigma^{-1} \cdot \left(\mathbf{c}-\mu\right)\right) \tag{1}$$

The background likelihood is obtained from equation (1) as $P(\mathbf{c}|\bar{F}) = P(\mathbf{c}|\mu_{\bar{F}},\Sigma_{\bar{F}})$ and is used to derive the MAP foreground likelihood $P(F|\mathbf{c})$. Assuming equal likelihood of color appearance in the foreground i.e. $P(\mathbf{c}|F) = \frac{1}{256^2}$, we obtain:

$$P(F|\mathbf{c}) = \frac{P(\mathbf{c}|F)\cdot P(F)}{P(\mathbf{c})} = \frac{\frac{1}{256^2}\cdot P(F)}{\frac{1}{256^2}\cdot P(F) + P(\mathbf{c}|\bar{F})\cdot P(\bar{F})}$$

Priors of fore-/background are derived from the expected volume occupancy of foreground objects and will be discussed in the next section.
For skin color classification we follow Caetano *et al.* [2] and model skin color with a static *Gaussian Mixture Model* (GMM) with $I = 2$ basis functions. This has been reported as a good tradeoff between accuracy and efficiency. The GMM is derived from equation (1) and associated weights w_i as $P(\mathbf{c}|S) = \sum_{i=1}^{I} w_i \cdot P(\mathbf{c}|\mu_i,\Sigma_i)$. The parameters have been trained with Expectation Maximization from a set of labeled skin-color images. Skin color classification can thus be cast in a Bayesian formulation by assuming equal likelihood of color appearance in non skin-colored regions $P\left(\mathbf{c}|\bar{S}\right) = \frac{1}{256^2}$ resulting in the MAP assignment:

$$P(S|\mathbf{c}) = \frac{P(\mathbf{c}|S)\cdot P(S)}{P(\mathbf{c})} = \frac{P(\mathbf{c}|S)\cdot P(S)}{P(\mathbf{c}|S)\cdot P(S) + \frac{1}{256^2}\cdot P(\bar{S})}$$

Notice that the models introduced so far do not permit for any type of classification error. Following [6], a more robust classification scheme is formulated by reverting to

the prior in case of an outlier. Let $e_F, e_S \in [0,1]$ be the probabilities of being outlier in the foreground and skin color model respectively. Then the classifier augmentations are:

$$P'(F|\mathbf{c}) = e_F \cdot P(F) + (1-e_F) \cdot P(F|\mathbf{c}) \text{ and } P'(S|\mathbf{c}) = e_S \cdot P(S) + (1-e_S) \cdot P(S|\mathbf{c})$$

2.2 Volume based Classification

pSfS has been adapted to combine the previously described image based classifiers. The difference between the presented algorithm and [4] is the definition of ϕ and β. In our setting ϕ describes a skin colored foreground and β a group of classes given as the remaining combinations of being fore-/background and skin/non-skin color. This leads to the introduction of multiple priors into pSfS.
Now let $\{\Gamma_1, \cdots, \Gamma_N\}$ be the set of super classes representing all $N = 2^S$ possible combinations of skin-colored foreground or background classifications of all S sensors.

$$\begin{aligned}
\Gamma_1 &= \{\ \phi, \quad \phi, \quad \phi, \quad \ldots, \quad \phi \quad \} \\
\Gamma_2 &= \{\ \beta, \quad \phi, \quad \phi, \quad \ldots, \quad \phi \quad \} \\
\Gamma_3 &= \{\ \phi, \quad \beta, \quad \phi, \quad \ldots, \quad \phi \quad \} \\
&\vdots \\
\Gamma_{S+2} &= \{\ \beta, \quad \beta, \quad \phi, \quad \ldots, \quad \phi \quad \} \\
&\vdots \\
\Gamma_n &= \{\ \Gamma_n[1], \Gamma_n[2], \Gamma_n[3], \ldots, \Gamma_n[S] \ \} \\
&\vdots \\
\Gamma_N &= \{\ \beta, \quad \beta, \quad \beta, \quad \ldots, \quad \beta \quad \}
\end{aligned}$$

and let their group specific priors be given as $P(\Gamma_n) = \prod_{s=1}^{S} P(\Gamma_n[s])$ with projected priors:

$$P(\phi) = P(F) \cdot P(S) \qquad \text{and} \qquad P(\beta) = 1 - P(\phi)$$

In the absence of occlusion a voxel is assigned to be part of a visual hull $\mathcal{H}$ if all sensors classify the voxel as skin-colored foreground, with prior probability $P(\mathcal{H})$. That is:

$$\mathcal{H} = \Gamma_1 \qquad \text{and} \qquad P(\mathcal{H}) = P(\Gamma_1) \tag{2}$$

$P(\mathcal{H})$ is defined as the occupancy ratio between the expected number of skin-colored foreground voxels and the total number of voxels. Projected skin priors can thus be derived from $P(\mathcal{H})$ as $P(S) = \frac{\sqrt[S]{P(\mathcal{H})}}{P(F)}$. Equivalently projected foreground priors can be derived from a visual hull of all foreground objects $\mathcal{H}_F$ with an expected volume occupancy ratio $P(\mathcal{H}_F)$ as $P(F) = \sqrt[S]{P(\mathcal{H}_F)}$. We have chosen $P(\mathcal{H}_F)$ and $P(\mathcal{H})$ statically from reference reconstructions which have been generated with a traditional SfS algorithm.
Cameras in SfS setups are usually mounted with wide stereo baselines leading to statistical independents between the camera views and Bayes theorem can be consulted to

estimate class probabilities.

$$P(\Gamma_n|\mathbf{c}_1,\ldots,\mathbf{c}_S) = P(\Gamma_n)\cdot\prod_{s=1}^{S}\frac{P(\mathbf{c}_s|\Gamma_n)}{P(\mathbf{c}_s)} \qquad n=1,\ldots,N \tag{3}$$

Here $P(\mathbf{c}_s|\Gamma_n) = P(\mathbf{c}_s|\Gamma_n[s])$ is the conditional probability of the observation in sensor s, given a certain super class in its view. Conditional probabilities can be rewritten in means of posterior probabilities to plug in per pixel MAP assignments:

$$P(\Gamma_n|\mathbf{c}_1,\ldots,\mathbf{c}_S) = P(\Gamma_n)\cdot\prod_{s=1}^{S}\frac{P(\Gamma_n[s]|\mathbf{c}_s)}{P(\Gamma_n[s])} \tag{4}$$

Here $P(\Gamma_n[s]|\mathbf{c}_s)$ conforms to the posterior probability of a certain superclass in sensor s given its observation $\mathbf{c}_s$. This is to say:

$$P(\Gamma_n[s]|\mathbf{c}_s) := \begin{cases} P'(F,S|\mathbf{c}_s) & \text{if} \quad \Gamma_n[s]=\phi \\ 1-P'(F,S|\mathbf{c}_s) & \text{if} \quad \Gamma_n[s]=\beta \end{cases}$$

Finally the partitioning of voxels to super classes is obtained by following Bayes rule for minimum error. Therefore a voxel is assigned to the most probable super class Γ_m:

$$\Gamma_m = \arg\max_{\Gamma_n} P(\Gamma_n)\cdot\prod_{s=1}^{S}\frac{P(\Gamma_n[s]|\mathbf{c}_s)}{P(\Gamma_n[s])}$$

As computation of all class posteriors becomes computational intensive with growing number of sensors, it has been recommended in [5] to limit computation to the foreground class and set a threshold on its posterior instead. Our results suggest the same as we have obtained equivalent reconstruction results for both algorithmic variants.
The algorithm introduced so far does not account for systematic errors given through occlusion or segmentation errors. The assignment of multiple foreground classes is a common approach to resolve this issue in SfS type algorithms. In pSfS this can be done in two ways. First, by assigning multiple super classes to the visual hull. If for example, the appearance of a single systematic error was to be allowed, equation (2) would become:

$$\mathcal{H} = \bigcup_{s=1}^{S+1}\Gamma_s \qquad \text{and} \qquad P(\mathcal{H}) = \sum_{s=1}^{S+1}P(\Gamma_s) \tag{5}$$

This approach has a serious disadvantage as all class posteriors now have to be computed. A more efficient procedure is given by assigning an active camera flag to each pixel and than limit the class computation to active projections. In the presence of occlusion masks these are the activity flags. If multiple foreground classes should be allowed, a given number of pixel projections with lowest foreground probability $P'(F,S|\mathbf{c})$ have to be disabled dynamically.

3 Mixture Particle Filtering

Detection of hand and face volumes within a reconstructed volume is usually a time consuming process which can be accelerated by incorporating temporal information through tracking. We assume independents of the movements of hands and head and therefore follow [7] and apply a 3D variant of mixture particle filtering for tracking. Here the joint distribution of object states is interpreted as a mixture in which each object is tracked with a dedicated particle filter. The prediction and update equations of the M-component mixture model are given with mixture weights $\sum_{m=1}^{M} \pi_{m,t} = 1$ as

$$\text{predict:} \quad p(\mathbf{x}_t|\mathbf{Y}_{t-1}) = \sum_{m=1}^{M} \pi_{m,t-1} \cdot p_m(\mathbf{x}_t|\mathbf{Y}_{t-1})$$

$$\text{update:} \quad p(\mathbf{x}_t|\mathbf{Y}_t) = \sum_{m=1}^{M} \left(\frac{\pi_{m,t-1} \cdot p_m(\mathbf{y}_t|\mathbf{Y}_{t-1})}{\sum_{n=1}^{M} \pi_{n,t-1} \cdot p_n(\mathbf{y}_t|\mathbf{Y}_{t-1})} \right) \cdot \frac{p(\mathbf{y}_t|\mathbf{x}_t) p_m(\mathbf{x}_t|\mathbf{Y}_{t-1})}{p_m(\mathbf{y}_t|\mathbf{Y}_{t-1})}$$

The first update term can be interpreted as the new mixture weight $\pi_{m,t}$ because the state $\mathbf{x}$ is not involved. Hence only the second term represents the component update. Component interaction is therefore limited to mixture weight computation which makes this particle filtering technique fast. Particle filters track 3D centroid positions of hand and face volumes. Hands and face are distinguished through their volume sizes. Particle states represent boxes in space with a fixed size, see right side of figure 1. We use the percentage of occupied skin-colored volume within these boxes as the source for weight evaluation. This average occupancy can be computed efficiently through utilization of a summed volume table for the reconstructed volume. The mixture particle formulation given above does not determine how a mixture is initialized or modified. In our setup initialization is done by spreading particles randomly until all expected objects are tracked. If a mode was found which is not already tracked, a new mixture is initialized on that mode. In cases in which object separation is impossible, a mixture update has to be enforced which provides merge and split operations. Here it is based on K-means analysis and similar to the one proposed in [7]. The difference is that we have to treat different particle types. Therefore we allow re-clustering only between mixtures of the same type, others are discared.

4 Results

We have implemented pSfS as well as tSfS and compared both with respect to reconstruction quality and performance. Achieved reconstruction results favor pSfS over tSfS, see figure 3 for a comparison. Both pSfS variants achieved more detailed reconstructions than tSfS. Explicit computation of fore-/background classes and limited evaluation by thresholding the foreground class resulted in similar reconstructions. The similarity between outputs of both pSfS algorithms can be explained by detailing the impact of background class evaluation. Explicit evaluation of background classes leads to a less false positive rate for foreground class assignment in presence of highly ambiguous voxels. These false positives are known to have a low foreground probability,

Table 1. Performance results of 3D reconstruction on a GPU

Reconstruction Type	Volume Resolution	Algo	Image eval.	GPU Readout	Total
tSfS	$64 \times 64 \times 48$ voxel	1.1ms	5.5ms	0.1ms	6.7ms
	$128 \times 128 \times 96$ voxel	5.7ms	5.5ms	0.8ms	12.0ms
	$256 \times 256 \times 192$ voxel	35.8ms	5.5ms	7.8ms	49.1ms
pSfS, foreground class	$64 \times 64 \times 48$ voxel	1.2ms	5.5ms	0.1ms	6.8ms
	$128 \times 128 \times 96$ voxel	5.9ms	5.5ms	0.8ms	12.2ms
	$256 \times 256 \times 192$ voxel	38.0ms	5.5ms	7.8ms	51.3ms
pSfS, all classes	$64 \times 64 \times 48$ voxel	4.3ms	5.5ms	0.1ms	9.9ms
	$128 \times 128 \times 96$ voxel	31.4ms	5.5ms	0.8ms	37.7ms
	$256 \times 256 \times 192$ voxel	241.8ms	5.5ms	7.8ms	255.1ms

as they would not be assigned to a background class otherwise. This implies that they can be equivalently eliminated by enforcing a threshold on posterior probabilities.
The presented SfS variants were implemented on a GPU with NVIDIA CUDA to permit interactive frame rates. Here performance results of three different volume resolutions are presented. The runtime values were measured on an Intel Q6600 running at 2.4GHz with a NVIDIA GeForce 8800 GTX graphics card and are listed in table 1.
The SfS performances vary between 1.1ms and 241.8ms, depending on the chosen algorithm and volume resolution. A performance comparison between tSfS and pSfS limited to foreground class evaluation resulted in similar runtimes. Both algorithms have a linear complexity $\mathcal{O}(S)$ where S is the number of sensors. In contrast explicit evaluation of fore-/background classes has an exponential complexity of $\mathcal{O}(2^S)$.
It is further essential to note how the presented algorithms behave in the presence of systematic errors like inter-object occlusion. tSfS and both pSfS variants cannot handle this and do not reconstruct partial occluded objects. The appearance of systematic errors therefore has to be explicitly modeled. As our particle filter is applied to low resolution volume reconstructions, we limit the following comparison to this type. Figure 3 depicts obtained reconstructions with 5 out of 6 cameras. Here. we compared tSfS and pSfS with explicit computation of multiple foreground classes and finally pSfS with the active camera concept. All algorithms achieve similar coarse reconstructions and can reconstruct the volume even in presence of occlusion.

5 Conclusion

We have presented a GPU based pSfS system which uses a cascade of classifiers for volumetric reconstruction of skin colored objects, i.e. to track hand and face volumes. Our GPU implementation makes the system suitable for the advanced HCI applications targeted, with a runtime of less than 15ms for coarse but reasonable volume resolutions.

Acknowledgments. We would like to thank Brendan McCane, Geoff Wyvill and Katrin Frank for their contributions. Part of this work has been funded by a University of Otago CALT research grant (JDLJ17400).

References

1. In *IEEE Recommended Practice for Electric Power Systems in Commercial Buildings*, page 388. 1990.
2. T. Caetano, S. Olabarriaga, and D. Barone. Performance evaluation of single and multiple-gaussian models for skin color modeling. *Computer Graphics and Image Processing, Proceedings. XV Brazilian Symposium on*, pages 275–282, 2002.
3. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proceedings of the International Conference on Pattern Recognition*, 1994.
4. J. Landabaso and M. Pardas. A unified framework for consistent 2-d/3-d foreground object detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(8):1040–1051, 2008.
5. J. L. Landabaso and M. Pardàs. Shape from Inconsistent Silhouette. *accepted for publication in Journal of Computer Vision and Image Understanding*, 2008.
6. T. Minka. The 'summation hack' as an outlier model. *"Unpublished manuscript available from http://research.microsoft.com/~minka"*, 2003.
7. J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. *ICCV 2003*, 02:1110, 2003.

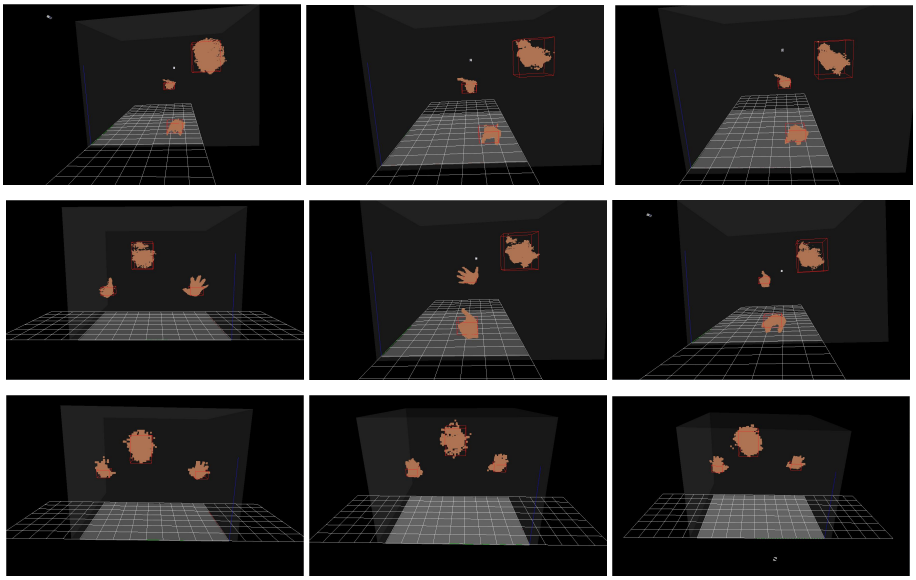

Fig. 3.

1. row: [left to right] tSfS, pSfS with fore-/background classes, pSfS with foreground class, all computations with 6 cameras in $128 \times 128 \times 96$ volume
2. row: pSfS, foreground thresholding with 6 cameras, $256 \times 256 \times 192$ voxels
3. row: [left to right] tSfS $P'(F, S|\mathbf{c})$ thresholded, pSfS multiple classes, pSfS active camera, all computations with 1 of 6 views occluded in $64 \times 64 \times 48$ volume

See also: www.mi.fh-wiesbaden.de/~cjohn/videos/psfsTracking.avi