Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2011)
A. Bargteil and M. van de Panne (Editors)

Practical Color-Based Motion Capture

Robert Wang[1,2] Sylvain Paris[3] Jovan Popović[3,4]

[1]MIT CSAIL [2]3Gear Systems [3]Adobe Systems [4]University of Washington

Figure 1: *We describe a lightweight color-based motion capture system that uses one or two commodity webcams and a color shirt to track the upper body. Our system can be used in a variety of natural lighting environments such as this squash court, a basketball court or outdoors.*

Abstract

Motion capture systems have been widely used for high quality content creation and virtual reality but are rarely used in consumer applications due to their price and setup cost. In this paper, we propose a motion capture system built from commodity components that can be deployed in a matter of minutes. Our approach uses one or more webcams and a color shirt to track the upper-body at interactive rates. We describe a robust color calibration system that enables our color-based tracking to work against cluttered backgrounds and under multiple illuminants. We demonstrate our system in several real-world indoor and outdoor settings.

1. Introduction

Motion capture data has revolutionized feature films and video games. However, the price and complexity of existing motion capture systems have restricted their use to research universities and well-funded movie and game studios. Typically, mocap systems are setup in a dedicated room and are difficult and time-consuming to relocate. In this paper, we propose a simple mocap system consisting of a laptop and one or more webcams. The system can be setup and calibrated within minutes. It can be moved into an office, a gym or outdoors to capture motions in their natural environments.

Our system uses a robust color calibration technique and a database-driven pose estimation algorithm to track a multi-colored object. Color-based tracking has been used before for garment capture [SSK*05] and hand-tracking [WP09]. However these techniques are typically limited to studio settings due to their sensitivity to the lighting environment. Working outside a carefully controlled setting raises two major issues. The color of the incident light may change, thereby altering the apparent color of the garment in non-trivial ways. One may also have to work in dimly lit scenes that require slower shutter speeds. However, longer exposure times increase motion blur, which perturbs tracking. We contribute a method that continuously compensates for light color variations and is robust to motion blur. We demonstrate in the results section that this enables our system to track activities in real-world settings that are challenging for existing garment-based techniques.

Our tracking approach is complementary to more sophisticated setups such as those that use infrared cameras and markers. Without achieving the same accuracy, our system is sufficiently precise to perform tasks such as motion anal-

ysis and contact detection, which makes it usable for augmented reality and human-computer interaction. Its low cost and ease of deployment make it affordable to the masses and we believe that it can help spread mocap as an input device for games and virtual reality applications. Furthermore, since our system enables tracking at interactive rates, it enables instant feedback for previsualization purposes.

2. Related work

A variety of motion capture technologies have been introduced in the last two decades. We refer to the survey of Welch and Foxlin [WF02] for a comprehensive overview. In this section, we focus on the approaches most related to ours.

White et al. [WCF07] and Scholz et al. [SSK*05] propose methods to track garments in motion using high-frequency color patterns. To handle occlusions, White uses many cameras and Scholz relies on user intervention. In comparison, we use low-frequency patterns that are less likely to be fully occluded, which allows our method to work automatically from only one or a few cameras. We share this property with the technique of Wang and Popović [WP09] for tracking hands in an office environment. In comparison, we aim for a broader range of conditions such as outdoors, sport centers, and casual living rooms. The main challenge stemming from these environments is their uncontrolled lighting, which is often dim and non-uniform in intensity and color. Our approach explicitly addresses these difficulties and is able to produce accurate estimates whereas Wang and Popović often yield gross errors under such conditions as shown in the result section.

Recent wearable systems have enabled motion capture in almost any environment, e.g. [MJKM04, VAV*07]. However, they do not provide the absolute position of the subject and require off-line processing of the data. Our approach proposes a complementary trade-off. While the subject has to remain within the visible range of the cameras, we provide absolute tracking and interactive feedback.

The accuracy of markerless motion capture systems typically depend on the number of cameras used. Monocular or stereo systems are portable but less accurate and limited by the complexity of the motion [MHK06].

Commercial systems such as Microsoft Kinect [SFC*11] and iMocap [iMo07] also aim for on-site and easy-to-deploy capture. The Kinect active illumination system is limited to indoor use and subject to the occlusion limitations of a single viewpoint, while iMocap is marker-based which probably requires a careful setup. Beyond these differences, our work and these methods share the same motivations of developing mocap for interactive applications such as games [IWZL09], augmented reality, and on-site previsualization.

Our work is also related to image analysis techniques that rely on colors. Comaniciu et al. [CRM03] track an object in image space by locally searching for a specific color histogram. In comparison, we locate the shirt without assuming a specific histogram, which make our approach robust to illumination changes. Furthermore, our algorithm is sufficiently fast to perform a global search. It does not rely on temporal smoothness and can handle large motions. Dynamic color models have been proposed to cope with illumination changes, e.g. [MRG99, KMB07, SS07]. The strong occlusions that appear with our shirt would be challenging for these models because one or several color patches can disappear for long periods of time. In comparison, we update the white balance using the a priori knowledge of the shirt color. We can do so even if only a subset of the patches is visible, which makes the process robust to occlusions. Generic approaches have been proposed for estimating the white balance, e.g. [GRB*08], but these are too slow to be used in our context. Our algorithm is more efficient with the help of a color shirt as a reference.

3. Overview

We propose a system for upper body motion capture that uses one or more cameras and a color shirt. Our system is designed with low cost and fast setup in mind. The cameras are USB webcams that generate 640×480 frames at 30 Hz. They are geometrically calibrated using a standard computer vision technique [Zha00] and color calibrated by scribbling on a single frame from each camera. The entire setup time typically takes less than five minutes.

Our processing pipeline consists of several steps. First, we locate and crop the multi-colored shirt from the frame by searching for an image region with the appropriate color histogram (§ 4). Our histogram search technique allows us to pick out the shirt with only a coarse background subtraction and white balance estimate. Once the object has been located, we perform a robust pose estimation process that iteratively refines both the color and pose estimate of the shirt region (§ 5).

We demonstrate the robustness of our system under changing illumination and a dynamic background (§ 6). Processing is performed at interactive rates on a laptop connected to two webcams, making our system a portable and inexpensive approach to motion capture suitable for virtual and augmented reality.

4. Histogram search

Our first task is to locate the multi-colored shirt. The shirt is composed of 20 patches colored with a set of 10 distinct colors, each of which appears twice. Our shirt is distinctive from real-world objects in that it is particularly colorful, and we take advantage of this property to locate it. Our procedure is robust enough to cope with a dynamic background and inaccurate white balance. It is discriminative enough to start from scratch at each frame, thereby avoiding any assumption of temporal coherence.

To locate the shirt, we analyze the local distribution of chrominance values in the image. We define the chrominance of an (r, g, b) pixel as its normalized counterpart $(r, g, b)/(r + g + b)$. We define $h(x, y, s)$ as the normalized

chrominance histogram of the $s \times s$ region centered at (x, y). In practice, we sample histograms with 100 bins. Colorful regions likely to contain our shirt correspond to more uniform histograms whereas other areas tend to be dominated by only a few colors, which produces peaky histograms (Fig. 2). We estimate the uniformity of a histogram by summing its bins while limiting the contribution of the peaks. That is, we compute $u(h) = \sum_i \min(h_i, \tau)$, setting $\tau = 0.1$. With this metric, a single-peak histogram has $u \approx \tau$ and a uniform one $u \approx 1$. Other metrics such as histogram entropy perform similarly. The colorful shirt region registers a particularly high value of u. However, choosing the pixel and scale (x', y', s') corresponding to the maximum uniformity $u_{\max} = \max u(x, y, s)$ proved to be unreliable. Instead, we use a weighted average favoring the largest values:

$$(x', y', s') = \frac{1}{\sum_{x,y,s} w(x, y, s)} \sum_{x,y,s} (x, y, s)\ w(x, y, s)$$

where $w(x, y, s) = \exp\left(-\frac{[u(h(x,y,s)) - u_{\max}]^2}{u_\sigma^2}\right)$ and $u_\sigma = \frac{1}{10} u_{\max}$.

The shirt usually occupies a significant portion of the screen, and we do not require precise localization. This allows us to sample histograms at every sixth pixel and search over six discrete scales. We build an integral histogram to accelerate histogram evaluation [Por05].

While the histogram search process does not require background subtraction, it can be accelerated by additionally restricting the processing to a roughly segmented foreground region. In practice, we use background subtraction [SG99] for both the histogram search and to suppress background pixels in the color classification (§ 5).

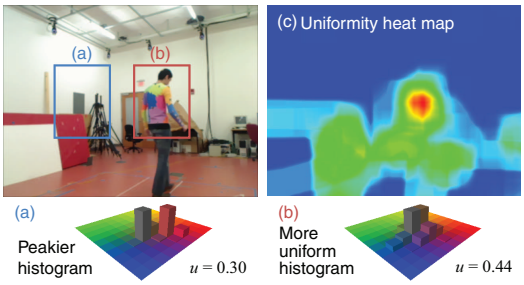

Figure 2: *The colorful shirt has a more uniform chromaticity histogram (b) with many non-zero entries whereas most other regions have a peakier histogram (a) dominated by one or two colors. We visualize our uniformity measure $u(h(x, y, s))$ with scale $s = 80$ as a heat map.*

5. Color and pose estimation

After the shirt has been located, we perform color classification on the shirt region and estimate the 3-D pose. In our context, the former is particularly challenging because lighting may change in color and intensity. For instance, a yellow patch may appear bright orange in one frame and dark brown in another (Fig. 3). In this section, we describe a continuous color classification process that adapts to changing lighting and variations in shading. First, we describe the offline process of modeling the shirt colors. The online component estimates an approximate color classification and 3D pose before refining both to obtain the final pose. In addition to the final pose, we compute an estimate of the current illumination as a white balance matrix and maintain a list of *reference illuminations* that we use to recover from abrupt lighting changes.

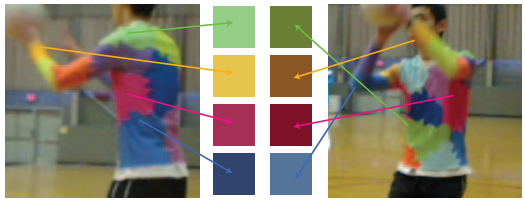

Figure 3: *The measured values of the color patches can shift considerably from frame to frame. Each row shows the measured value of two identically colored patches in two frames from the same capture session.*

5.1. Color model and initialization

We model each of the $k = 10$ distinct shirt colors as a Gaussian $N(\mu_k, \Sigma_k)$ in RGB space. We build this model ahead of time by manually labeling five white-balanced images of our shirt (Fig. 5).

At the beginning of a capture session, we estimate the illumination by asking the user to manually label an image of the shirt taken in the current environment by coarsely scribbling on each of the visible patches. We solve for a 3×3 white balance matrix W that maps the mean patch colors μ'_k from the reference image to the mean colors μ_k of the Gaussian color model, that is, $W = \text{argmin}_W \sum_k \|W\mu_k - \mu'_k\|^2$. The white balance matrix W is used to bootstrap our color and pose tracking, and we also use it to initialize our list of reference illuminations.

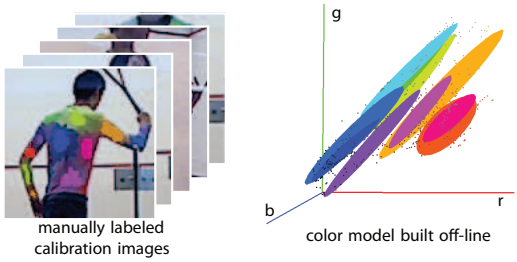

Figure 5: *Ahead of time, we build a color model of our shirt by scribbling on 5 white-balanced images. We model each color with a Gaussian distribution in RGB space.*

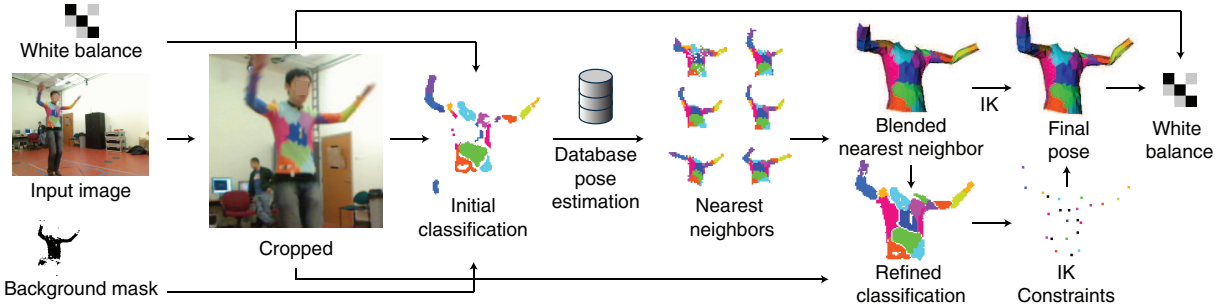

Figure 4: *After cropping the image to the shirt region, we white balance and classify the image colors. The classified image is used to estimate the upper-body pose by querying a precomputed pose database. We take the pose estimate to be a weighted blend of these nearest neighbors in the database. The estimated pose can be used to refine our color classification, which is converted into a set of patch centroids. These centroids drive the inverse kinematics (IK) process to refine our pose. Lastly, the final pose is used to estimate the white balance matrix for the next frame.*

5.2. Online analysis

The online analysis takes as input the colorful cropped region corresponding to the shirt (§ 4). We roughly classify the colors of this region using our color model and white balance estimate. The classified result is used to estimate the pose from a database. Next, we use the pose estimate to refine our color classification, which is used in turn to refine the pose. Lastly, we update our current estimate of the white balance of the image (Fig. 4).

Step 1: Color classification We white balance the image pixels I_{xy} using a 3×3 matrix W. In general, W is estimated from the previous frame, which we will explain in Step 5. For the first frame in a sequence, we use the user-labeled initialization (§ 5.1). After white balancing, we classify the colors according to the Gaussian color models $\{(\mu_k, \Sigma_k)\}_k$. We produce an id map $\mathbf{r}_{xy}$ defined by:

$$\mathbf{r}_{xy} = \begin{cases} \underset{k}{\operatorname{argmin}} \|WI_{xy} - \mu_k\|_{\Sigma_k} & \\ & \text{if } \|WI_{xy} - \mu_k\|_{\Sigma_k} < T \\ \textit{background} & \text{otherwise} \end{cases}$$

where $\|\cdot\|_\Sigma$ is the Mahalanobis distance with covariance Σ, that is: $\|X\|_\Sigma = \sqrt{X\,\Sigma^{-1}X}$, and T is a threshold that controls the tolerance of the classifier. We found that $T = 3$ performs well in practice, that is, we consider that a pixel belongs to a Gaussian if it is closer than three standard deviations to its mean. In addition we use a background subtraction mask to suppress false-positives in the classification.

Most of the time, the above white balance and classification approach suffices. However, during a sudden change of illumination our white balance estimate from the previous frame may no longer be valid. We detect this case when less than 40% of the supposedly foreground pixels are classified. To overcome these situations, we maintain a list of previously encountered reference illuminations expressed as a set of white balance matrices $W \in \mathcal{W}$. When we detect a poor classification, we search among these reference matrices $\mathcal{W}$ for the one that best matches the current illumination. That is, we re-classify the image with each matrix and keep the one that classifies the most foreground pixels.

Step 2: Pose estimation Once the colors have been correctly identified as color ids, we can estimate the pose with a data-driven approach [WP09]. We precompute a database of $80,000$ upper-body poses that are selected by uniformly sampling a large database spanning a variety of upper-body configurations and 3-D orientations. The poses of the database were taken from linear blends of generic action poses from the Poser 7 software and do not include any squash or basketball-specific sequences. We rasterize each pose as a tiny id map $\mathbf{r}^i$. At run time, we search our database for the ten nearest neighbors $\bar{\mathbf{r}}^i$ of our classified shirt region, resized as a tiny 40×40 id map. We take our pose estimate to be a weighted blend of the poses corresponding to these neighbors $\mathbf{q}^i$ and rasterize the blended pose $\mathbf{q}^b$ to obtain an id map $\mathbf{r}^b$. This id map is used in Step 3 to refine the classification and Step 4 to compute inverse kinematics (IK) constraints.

The blended pose $\mathbf{q}^b$ expresses an approximate configuration of the upper body, but does not account for the global pose. To obtain the global position and orientation of the subject, we associate 2D projection constraints to the centroid of each color patch of the rasterized id map $\mathbf{r}^b$ and transform these constraints to the original query image space of each image. We solve a 6-DOF inverse kinematics problem to obtain the global position and orientation that best matches the projection constraints from both cameras.

Step 3: Color classification refinement Our initial color classification (Step 1) relies on a global white balance. We further improve this classification by leveraging the rasterized pose estimate $\mathbf{r}^b$ computed in Step 2. This makes our approach robust to local variations of illumination.

We use the id map of the blended pose $\mathbf{r}^b$ as a prior in our classification. We analyze the image pixels by taking

into account their measured color I_{xy} as before and also the id predicted by the rasterized 3-D pose $\mathbf{r}^b$. To express this new prior, we introduce $d_{xy}(\mathbf{r}, k)$, the minimum distance between (x, y) and a pixel (u, v) of the rasterized predicted prior with color id k:

$$d_{xy}(\mathbf{r}, k) = \min_{(u,v)\in\mathcal{S}_k} \|(u, v) - (x, y)\| \quad \text{with } \mathcal{S}_k = \{(u, v) \mid \mathbf{r}_{uv} = k\}$$

With this distance, we define the refined id map $\hat{\mathbf{r}}$:

$$\hat{\mathbf{r}}_{xy} = \begin{cases} \underset{k}{\operatorname{argmin}} \|WI_{xy} - \mu_k\|_{\Sigma_k} + C\ d(\mathbf{r}^b, k) & \\ \qquad \text{if } \|WI_{xy} - \mu_k\|_{\Sigma_k} + C\ d(\mathbf{r}^b, k) < T & \\ \textit{background} \quad \text{otherwise} & \end{cases}$$

We set the influence of the prior term C to $6/s$ where s is the scale of the cropped shirt region. The classifier threshold T is set to five.

We compared the strength of our pose-assisted color classifier with the Gaussian color classifier by varying the classification thresholds and plotting correct classification versus incorrect classifications (Fig. 6). This additional information significantly improves the accuracy of our classification by removing impossible or highly improbable color classification given the pose estimate.

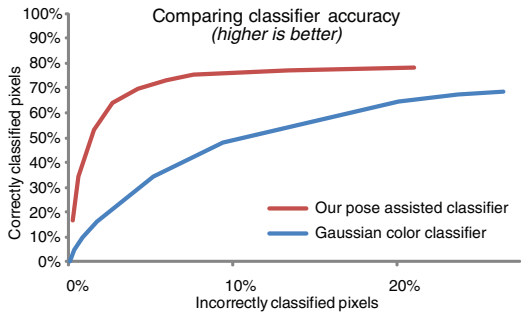

Figure 6: *Our pose-assisted classifier classifies more correct pixels at a lower false-positive rate than the baseline Gaussian classifier discussed in Step 1.*

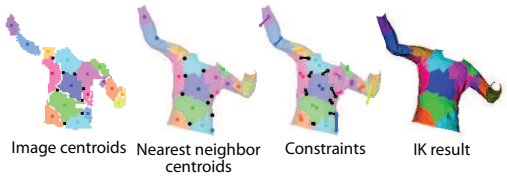

Figure 7: *For each camera, we compute the centroids of the color-classified id map $\hat{\mathbf{r}}^i$ and correspond them to centroids of the blended nearest neighbor to establish inverse kinematics constraints.*

Step 4: Pose refinement with inverse kinematics We extract point constraints from the newly computed id map $\hat{\mathbf{r}}$ to refine our initial pose estimate $\mathbf{q}^b$ using inverse kinematics (IK). We also take into account the pose $\mathbf{q}^h$ at the previous frame.

For each camera i, we compute the centroids $\mathbf{c}^{ki}$ of each patch k in our color-classified id map $\hat{\mathbf{r}}^i$. We also render the pose estimate $\mathbf{q}^b$ as an id map and establish correspondences between the rendered centroids of our estimate and the image centroids. We seek a new pose $\mathbf{q}^\star$ such that the centroids $\mathbf{c}^{\star i}$ of its id map $\mathbf{r}^{\star i}$ coincide with the image centroids $\mathbf{c}^{ki}$ (See 7). We also want $\mathbf{q}^\star$ to be close to our initial guess $\mathbf{q}^b$ and to the previous pose $\mathbf{q}^h$. We formulate these goals as an energy:

$$\mathbf{q}^\star = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_{i,k} \|\mathbf{c}^{\star i}(\mathbf{q}) - \mathbf{c}^{ki}\|^2_{\Sigma_c} + \|\mathbf{q} - \mathbf{q}^b\|^2_{\Sigma_b} + \|\mathbf{q} - \mathbf{q}^h\|^2_{\Sigma_p}$$

where the covariances matrices Σ_c, Σ_b, and Σ_h are trained off-line on ground-truth data similarly to Wang and Popović [WP09]. That is, for each term in the above equation, we replace $\mathbf{q}$ by the ground-truth pose and $\mathbf{q}^h$ by the ground-truth pose at the previous frame, and compute the covariance of each term over the ground-truth sequence.

Step 5: Estimating the white balance for the next frame As a last step, we refine our current estimate of the white balance matrix W and optionally cache it for later use in case of a sudden illumination change (Step 1). We create an id map from our final pose $\mathbf{q}^\star$ and compute a refined $W^\star$ matrix using the same technique as in Section 5.1. We use $W^\star$ as initial guess for the next frame. We also add $W^\star$ to the set $\mathcal{W}$ of reference illuminations if the minimum difference to each existing transformation in the set is greater then 0.5, that is, if: $\min_{W\in\mathcal{W}} \|W^\star - W\|_F > 0.5$ where $\|\cdot\|_F$ is the Frobenius norm.

6. Results

We evaluated our two-camera system in several real-world indoor and outdoor environments for a variety of activities and lighting conditions. We captured footage in a dimly lit indoor basketball court, through a glass panel of a squash court, at a typical office setting, and outdoors (Fig. 8). In each case, we were able to setup our system within minutes and capture without the use of additional lights or equipment.

To stress test our white balance and color classification process, we captured a sequence in the presence of a mixture of several fluorescent ceiling lights and a tungsten floor lamp. As the subject walked around this scene, the color and intensity of the incident lighting on the shirt varied significantly depending on his proximity to the floor lamp. Despite this, our system robustly classifies the patches of the shirt, even in the event when the tungsten lamp is suddenly turned off. Unlike other garment tracking techniques [SSK*05, WCF07, WP09], our method dynamically white-balances the images, which makes it robust to these lighting variations. We show in the companion video that this procedure is critical to the success of our approach.

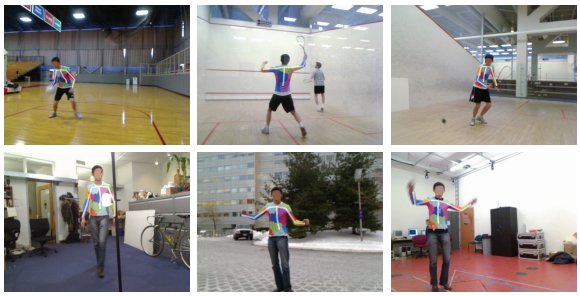

Figure 8: *We demonstrate motion capture in a basketball court, inside and outside of a squash court, at the office, outdoors and while using another (Vicon) motion capture system. The skeleton overlay is more visible in the accompanying video.*

Our system runs at interactive rates. On an Intel 2.4 GHz Core 2 Quad core processor, our Java implementation processes each frame in 120 ms, split roughly evenly between histogram search, pose estimation, color classification, and IK.

We evaluated the accuracy of our system by simultaneously capturing a sequence containing a variety of movements with a 16 camera Vicon motion capture system and our two-camera system. We applied a standard correction step to the Vicon data to fill gaps, smooth trajectories, and manually correct marker mislabelings due to occlusions. We also compared our results to the method of Wang and Popović [WP09], which we adapted to handle the upper body, but which lacks the pose prior and white balancing steps of our approach. The data from our method and from the Wang and Popović approach are left unprocessed to avoid any bias. On simple sequences without occlusions, both methods perform well. However on faster motions and in presence of occlusions, our algorithm can be twice as accurate (Fig. 9). On average, the Wang and Popović method RMS error is 5.1 cm and ours is 4.0 cm, that is, about 20% better. Because RMS is often an insufficient measure of visual quality [Ari06], we provide the plot of the shoulder joint angle that confirms that our method is closer to the ground-truth data (Fig. 10), as well as a video of the corresponding captured motions. A visual comparison shows that our approach faithfully reproduces the ground truth motion whereas the Wang and Popović technique exhibits significant jittering. We also compared the two methods on a jumping jacks sequence in which the arms are moving quickly. Whereas the Wang and Popović technique loses track of the arms because of the motion blur, our method correctly handles this sequence (Fig. 11 and companion video).

We demonstrate possible uses of our approach on two sample applications (Fig. 12 and companion video). The "squash analysis" software tracks a squash player; it enables replay from arbitrary viewpoints and provides statistics on the player's motion such as the speed and acceleration of the arm. The "goalkeeper" game sends balls at the player who

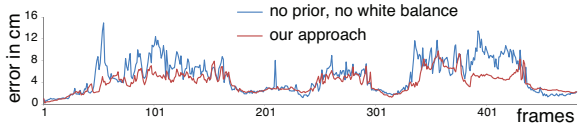

Figure 9: *We compare the accuracy (RMS of all mesh vertices) between a simple system without pose prior nor adaptive white balance akin to [WP09] and our approach. In absence of occlusion, both methods perform equivalently but on more complex sequences with faster motion and occlusions, our approach can be nearly twice more precise. On average, our method performs 20% better.*

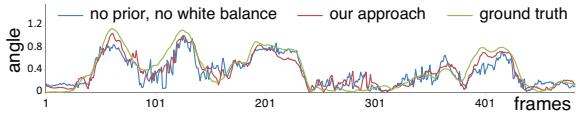

Figure 10: *We plot the angle of a shoulder joint for ground-truth data captured with a Vicon system, our method, and our method without the pose prior or white balance steps (akin to [WP09]). Our results are globally closer to the ground truth and less jittery than those of the simple method. This is better seen in the companion video.*

has to block them. This game is interactive and players move according to what they see on the control screen. These two proof-of-concept applications demonstrate that our approach is usable for a variety of tasks, it is sufficiently accurate to provide useful statistics to an athlete and is effective as a virtual reality input device.

Discussion As with all camera-based methods, our approach requires a line of sight to the subject. In the squash scenes, both cameras are placed behind the players, leading to several moments where an arm is occluded by the body for one of the cameras. Likewise, the basketball sequences include frames where the ball occludes the body. During these momentary occlusions, or when the subject exits the camera view frustum our approach is less accurate, but we are always able to recover (Fig.13). We have also experimented with using the Kalman filter for IK, which copes with occlusions better for slow movements. However, because it also degrades faster motion and is computationally more expensive, we chose not to use the Kalman filter.

While we have tested our system on several users, our database assumes a generic upper body shape of the subject. Subjects that differ significantly from the torso shape used are tracked less well. We hope to explore generating a variety of database reflecting different body shapes in future work.

We can localize the shirt even with several color objects in the background, such as in the Vicon studio scene. In both the basketball and squash scenes, we show interaction with another subject and a dynamic background. Our system handles motion blur well (Fig. 11), although in very dark en-

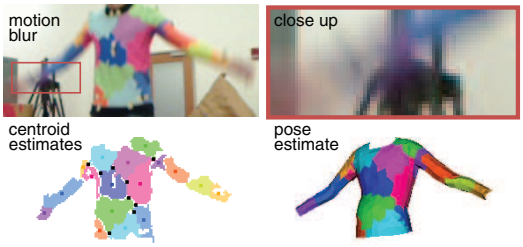

Figure 11: *Despite the significant motion blur in this frame, we are still able to estimate the patch centroids and the upper-body pose.*

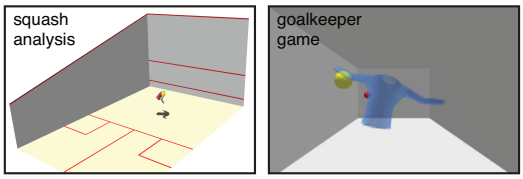

Figure 12: *We demonstrate possible applications of our approach on two proof-of-concept applications: a motion analytics tool for sports and a game. See the text and the companion video for detail.*

vironments the color classification can be overwhelmed by camera noise.

7. Conclusion

We have demonstrated a lightweight practical motion capture system consisting of one or more cameras and a color shirt. The system is portable enough to be carried in a gym bag and typically takes less than five minutes to setup. Our robust color and pose estimation algorithm allows our system to be used in a variety of natural lighting environments such as an indoor basketball court and an outdoor courtyard. While we use background subtraction, we do not rely on it and can handle cluttered or dynamic backgrounds.

Finally our system runs at interactive rates, making it suitable for use in virtual or augmented reality applications. We hope that our low-cost and portable system will spur the development of novel interactive motion-based interfaces and provide an inexpensive motion capture solution for the masses.

References

[Ari06] ARIKAN O.: Compression of motion capture databases. *ACM Trans. Graph. 25*, 3 (2006).

[CRM03] COMANICIU D., RAMESH V., MEER P.: Kernel-based object tracking. *IEEE Trans. Pattern Analysis Machine Intell. 25*, 5 (2003), 564–575.

[GRB*08] GEHLER P. V., ROTHER C., BLAKE A., SHARP T., MINKA T.: Bayesian color constancy revisited. In *IEEE Conf. Computer Vision and Pattern Recognition* (2008).

[iMo07] iMocap, 2007. http://tinyurl.com/6gxoum7.

[IWZL09] ISHIGAKI S., WHITE T., ZORDAN V. B., LIU C. K.: Performance-based control interface for character animation. *ACM Trans. Graphics 28*, 3 (2009), 1–8.

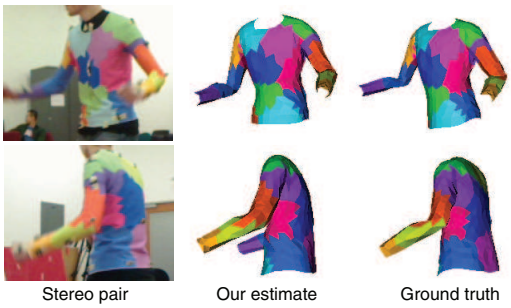

Figure 13: *In this frame, the right arm is occluded by the body in the second camera. Our estimate is less accurate during this occlusion due to depth ambiguity.*

[KMB07] KAKUMANU P., MAKROGIANNIS S., BOURBAKIS N.: A survey of skin-color modeling and detection methods. *Pattern Recognition 40*, 3 (2007), 1106–1122.

[MHK06] MOESLUND T. B., HILTON A., KRÜGER V.: A survey of advances in vision-based human motion capture and analysis. *Int. Journal of Computer Vision and Image Understanding 104*, 2-3 (2006).

[MJKM04] MILLER N., JENKINS O. C., KALLMANN M., MATRIĆ M. J.: Motion capture from inertial sensing for untethered humanoid teleoperation. In *Int. Conf. of Humanoid Robotics* (Nov 2004), pp. 547–565.

[MRG99] MCKENNA S. J., RAJA Y., GONG S.: Tracking colour objects using adaptive mixture models. *Image Vision Comput. 17*, 3-4 (1999), 225–231.

[Por05] PORIKLI F.: Integral histogram: a fast way to extract histograms in cartesian spaces. In *IEEE Conf. Computer Vision and Pattern Recognition* (2005), vol. 1, pp. 829–836.

[SFC*11] SHOTTON J., FITZGIBBON A., COOK M., SHARP T., FINOCCHIO M., MOORE R., KIPMAN A., BLAKE A.: Real-Time Human Pose Recognition in Parts from Single Depth Images. In *IEEE Conf. Computer Vision and Pattern Recognition* (2011).

[SG99] STAUFFER C., GRIMSON W.: Adaptive background mixture models for real-time tracking. In *IEEE Conf. Computer Vision and Pattern Recognition* (1999), vol. 2, pp. 246–252.

[SS07] SRIDHARAN M., STONE P.: Color learning on a mobile robot: Towards full autonomy under changing illumination. In *Int. Joint Conference on Artificial Intelligence* (2007), pp. 2212–2217.

[SSK*05] SCHOLZ V., STICH T., KECKEISEN M., WACKER M., MAGNOR M. A.: Garment motion capture using color-coded patterns. *Computer Graphics Forum 24*, 3 (2005), 439–447.

[VAV*07] VLASIC D., ADELSBERGER R., VANNUCCI G., BARNWELL J., GROSS M., MATUSIK W., POPOVIĆ J.: Practical motion capture in everyday surroundings. *ACM Trans. Graphics 26*, 3 (2007), 35.

[WCF07] WHITE R., CRANE K., FORSYTH D. A.: Capturing and animating occluded cloth. *ACM Trans. Graphics 26*, 3 (2007).

[WF02] WELCH G., FOXLIN E.: Motion tracking: no silver bullet, but a respectable arsenal. *Computer Graphics and Applications 22*, 6 (Nov./Dec. 2002), 24–38.

[WP09] WANG R. Y., POPOVIĆ J.: Real-time hand-tacking with a color glove. *ACM Trans. Graphics 28*, 3 (2009), 1–8.

[Zha00] ZHANG Z.: A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence 22*, 11 (2000).