Real-Time 3D Hand-Computer Interaction: Optimization and Complexity Reduction

Jonas Fredriksson, Sven Berg Ryen, Morten Fjeld
TableTop Interaction Lab (www.t2i.se), CSE
Chalmers University of Technology, Sweden
cryptic.aprx@gmail.com, svenryen@gmail.com, morten@fjeld.ch

ABSTRACT

This paper presents a low-cost method for enabling 3D hand-computer interaction. The method, accompanied by a system, uses the frame capturing functionality of a single consumer-grade webcam. Our recent work has been focused on examining and realizing a less complex system. The presented method reduces the tracking effort to only one reference marker: a color-coded bracelet that helps locate the part of the captured frame containing the user's hand. The located area contains all the information needed to extract hand rotation and finger angle data. To facilitate hand feature extraction, we have outfitted the user's hand with a specially coded glove. The glove is equipped with two square palm markers, a marker on either side of the hand, and five distinctly shaded finger sheaths. We believe that an approach that only tracks only one marker will be more efficient than similar methods that track each finger separately. The method is further simplified by using spatial properties, drawn from physiological characteristics of the human hand, to limit the areas considered by the algorithm. Some challenges regarding webcam limitations may arise when attempting to carry this method into effect, including problems related to image noise and limited image- and color-resolution. Overlapping hands and fingers, hand positioning outside the field of view, and interference by local light sources are other exigent factors to consider.

Categories and Subject Descriptors

I.4.8 [Scene Analysis]: *Object recognition, Tracking*.
H.5.1 [Multimedia Information Systems]: *Artificial, Augmented and Virtual Realities*

General Terms

Algorithms, Performance, Design, Experimentation, Human Factors

Keywords

Hand Tracking, 3D Hand-Computer Interaction, 3D Interaction, 3D Navigation, Gesture, Mixed Reality

1. INTRODUCTION

We present a low-cost method and system to enable 3D hand tracking and interaction based on the frame capturing functionality of a single consumer-grade webcam. Based on this method, a system has been realized as shown in the video accompanying this paper [1]. Our goal is to develop an algorithm capable of performance at a rate of several frames per second using minimal system resources in order to achieve real-time one- and two-handed 3D desktop input. We also want to provide a viable solution that could be used by average home-users, which rules out utilizing sophisticated cameras. To achieve this goal, the task complexity needs to be reduced while retaining essential information. We hope to accomplish this by reducing the hand model's total number of degrees of freedom (DOF). Our hand model has 16 DOFs (six for the palm and two for each finger). Combining a simplified hand model with a heuristic approach to coordinate determination, we expect to keep error rates relatively low while significantly reducing computation costs.
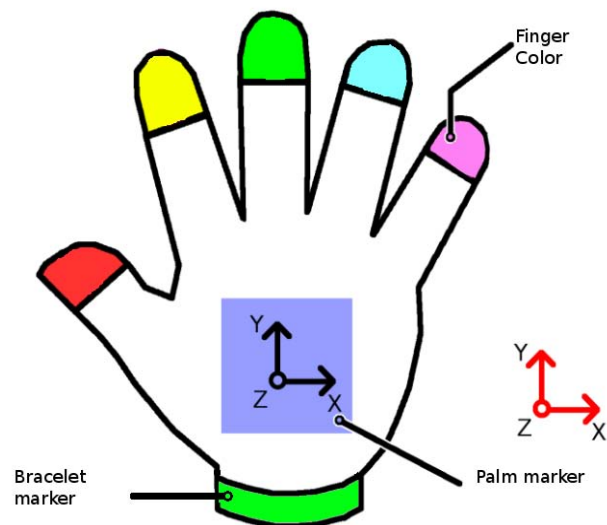

Figure 1. Finger colors are painted while bracelet and palm markers (back and front) are solid.

The user's palm and fingers are color-coded as shown in figure 1. A consumer-grade webcam typically sits on top of the computer screen looking down towards the user's hands. XYZ is the webcam coordinate system, where the X- and Y-axes are parallel to the camera view plane and the Z-axis runs along the camera's

---

[1] http://www.t2i.se/pub/media/3DHT_2008.wmw

line of sight. This coordinate system is drawn in red (figure 1, right). Using only one webcam makes it quite difficult to accurately calculate depth measurement (Z-axis). This limitation will affect the potential of depth-based techniques in interaction with 3D virtual environments.

Besides desktop screens, we conjecture that our approach may benefit interaction with large vertical and horizontal displays as well as collaborative and remote applications. Our related work as well as that of other researchers is presented in the following section. This is followed by sections on method optimization and complexity reduction, system design, web deployment, and future work. This paper is also offered as a color version [2].

2. RELATED WORK

We present a few works from the related area of particle filter algorithms, followed by some projects in the area of color tracking, and then summarize the initial steps of the project presented in this paper. While we are aware that data gloves are widely used to perform hand tracking [1], this is not the focus of our work.

2.1 Particle filter algorithms

Tracking articulated structures in a reasonable amount of time is a complex task due to the system's multiple dimensionalities. Several related research projects have previously introduced different particle filter algorithms [2][3][4][5][6]. During implementation, these algorithms have proven to be too slow for real-time tracking of the hand, which requires at least 20 seconds of processing time per frame. Better frame rates were obtained using multiple cameras like in Stenger, Mendonça, and Cipola's work [7][8], but the required processing power still prevents their results from being used in real-time applications. Also, de la Gorce et al. showed that vision-based tracking with monocular video stream provides the most natural, non-invasive form of hand motion capture [9][10]. Hence, they proposed efficient solutions for recovering 3D hand positioning from the information provided by the video stream of a single camera.

2.2 Color tracking algorithms

Rasmussen and Hager [11] did some early investigations of using color alone for tracking. They realized a technique which was successfully applied to such tasks as head and hand tracking. Later, Wu and Huang suggested a color tracking method employing so-called transductive learning [12]. Using a trained color classifier, they suggested an algorithm giving tight bounding boxes of the hand or face regions in video sequences. Finally, Simon et al. achieved real-time aspects and robustness in color tracking using a dynamically adjusted search window [13]. Besides tracking objects, the system could also adapt to their colors and sizes. Robustness was partly achieved through modeling of color appearance in dependence of the location using color grids. The ParadigmShift [14] project showed desktop interaction where color-coded hand-held objects were tracked and used for interaction with the desktop virtual environment. In the same project, up to two color-coded fingers were tracked and used

in simple interaction schemes. Finally, Smith, Piekarsky, and Wigley developed a hand tracking algorithm for low-powered mobile Augmented Reality (AR) user interfaces [15]. To realize AR outdoor pointing and selection they tested several colors and found one very suitable color. We assume that no such suitable color can be found for indoor use. Only tracking of the hand and one finger were realized.

2.3 Suggested method

The work presented here builds on preceding investigations of an algorithm detecting a user's hands utilizing palm and finger coding [16]. In previous investigations, fingers were color-coded and the palm was pattern- or color-coded. Related gesture recognition approaches have employed different means of hand pose classification in the webcam coordinate system. Our approach uses hand pose classification in the hand coordinate system that is rotationally independent of the webcam system. This requires less computation, low amounts of system training, and so, supports hand pose configuration in a more straightforward manner than previous approaches. Inconveniently, the use of only one webcam results in the existence of blind spots. That is, when the webcam's line of sight is included in the palm plane, the palm code is not visible to the webcam. Occlusion caused by clenched fingers or body parts other than the hand being tracked is another problem to be addressed.

3. METHOD OPTIMIZATION AND COMPLEXITY REDUCTION

This section discusses issues related to pattern and color tracking, coding choices, and the chosen tracking algorithm.

3.1 Pattern and color tracking

Firstly, the bracelet is used to determine the bounding box containing the hand. Secondly, the 3D position of the hand is determined by combining bracelet and palm marker information. Thirdly, for each finger two angles are tracked, the grip angle and the lateral tilt angle.

3.1.1 Analysis

Based on the bracelet, we can determine the area occupied by the hand. We establish the bracelet's tangent (principle) vector by calculating its slope based on the formula shown in (1). In (1), the slope for a set of points (x_i, y_i), $i = [1 \ldots N]$ is sampled along one finger. We create the base line for our hand area by clipping the line based on that tangent vector to the bounding box (smallest upright rectangle) containing all bracelet pixels. This line is extended a bit in each direction in order to encompass the full width of the hand. By using this line and its normal to sample the captured frame, we get a localized image of the hand without nearly any rotation in the image plane. This step, shown in figure 2, highly simplifies further analysis of the hand properties.

$$slope = \frac{(\sum_{i=1}^{N} x_i) * (\sum_{i=1}^{N} y_i) - N \sum_{i=1}^{N}(x_i * y_i)}{(\sum_{i=1}^{N} x_i)^2 - N \sum_{i=1}^{N}(x_i{}^2)} \qquad (1)$$

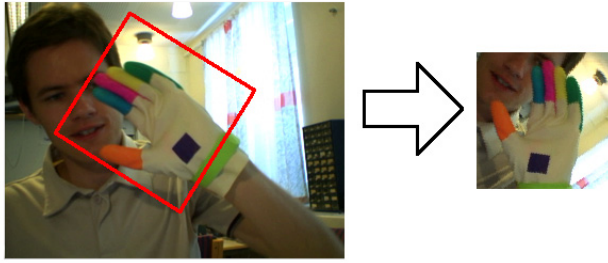[2] http://www.t2i.se/pub/papers/3DHT_2008.pdf

Figure 2. Sampling hand from captured frame.

3.1.2 3D position and rotation of the hand

The position of the hand can be found using only the position of the bracelet. However, using the palm marker and relating it to the current position of the hand area in the captured frame often helps attain a more exact and consistent result. The rotation of the hand is also calculated based on the palm marker in a way similar to that presented by Kato et al. [17]. Because of the physical characteristics of the hand (i.e. the fingers and palm are always next to each other), we base the calculations only on pixels in the area just above the bracelet, thus reducing the computational cost of the algorithm.

3.1.3 Finger angles

We track two different kinds of finger angles. One is the grip angle which we define as the curling of the finger towards the palm. This is actually the combination of two angles at two different parts of the finger, but we choose to assume that for most interaction purposes it suffices to only measure the amount of "gripping" a finger does. The second angle we define as the lateral tilt angle which measures the spread of the fingers. We use two different methods, one for each of the two types, to calculate the angle. Both methods are quite effective but are restricted to a certain rotation of the hand as explained in the discussion section. For the tilt angle we use the same formula as we did to calculate the bracelet slope. To get the grip angle we use a slightly different approach. By comparing the density of finger pixels around an estimated knuckle line, we can estimate the finger's Y position around an origin position at the knuckle's base (figure 3). This value can then be transformed into an angle value using an inverse tangent operation.

The method put forth does not yet consider thumbs. The reason is that a thumb rotates differently than the other fingers and as such requires a different version of the algorithm.
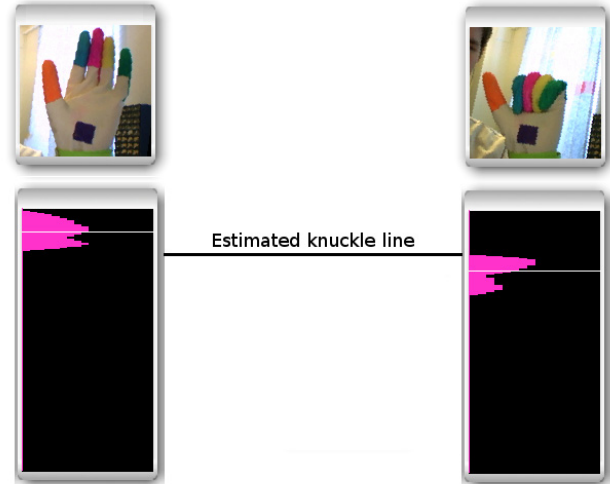

Figure 3. Determining grip angle from pixel density. The lower part of the image shows the pixel density per line. With mean value marked with a white line.

3.2 Coding choices

In this paper we focus on two different methods of marker-based tracking: pattern-based and color-based markers. The properties of the two approaches are summarized in table 1. Color-based tracking is computationally inexpensive and can be applied to almost any surface. Yet, it has the disadvantage of being very sensitive to color interference by local light sources. Also, it is hard to extract any 3D information like depth, position, and rotation using this method. These features are usually easier to track by using a pattern-based approach. However, some disadvantages of using patterns are that more pixels are required and that more expensive calculations are needed. In the color calculations performed here we use the Hue Saturation Value (HSV) space.

	Coding scheme	
	Pattern	Color
Works on non-planar surfaces	**No**	**Yes**
Amount of information	**High**	**Low**
Number of pixels required	**High**	**Low**

Table 1. Coding scheme comparison.

For our purposes, we find it useful to consider the human hand as consisting of smaller and larger parts. A method that is optimal for one part of the hand will probably not be so for other parts. We therefore try to find a method that achieves a proper balance between the two different ways of coding a marker. We have chosen to consider the palm of the hand as large, rigid, and flat. We felt that we could make use of this feature by employing a solid, pattern-based marker in the form of a single square that

would be used to calculate the depth and rotation of the hand. In contrast, the fingers are relatively small and curved. We therefore use color markers on all the fingers, which are used only to extract the relative angles of each finger. A third kind of marker is used to find the area within the captured field that is occupied by the hand. For this we use a color-based bracelet marker that functions well in establishing the 2D position and orientation of the wrist.

3.3 Tracking algorithm

The algorithm can be divided into two main parts as shown in figure 4. First, the hand area is found using the bracelet tracker. This information is then used to calculate the hand data from the pixels found in that area. A flow diagram showing elements of tracking and image analysis is shown in figure 5.
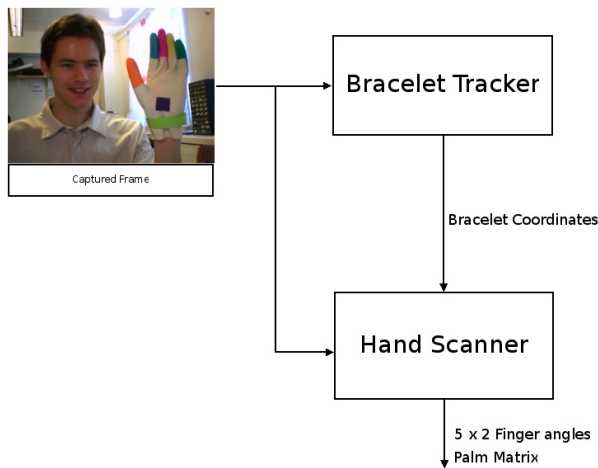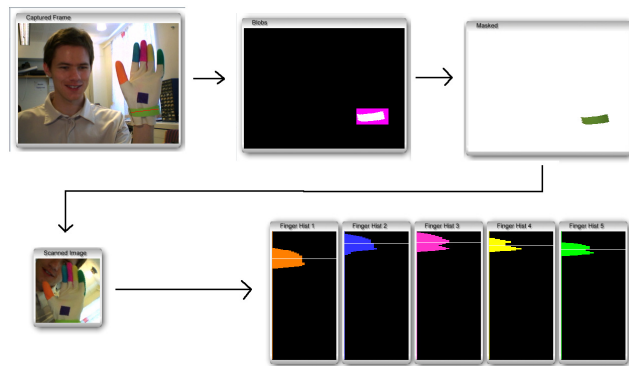

Figure 4. Algorithm overview.

Figure 5. Elements of tracking and image analysis (arrows indicate flow): Webcam image, blob tracking of the bracelet, extracted bracelet pixels based on color calibration, hand image, and for each finger a row pixel density histogram with white line showing mean row pixel density.

4. SYSTEM DESIGN

This section offers a description of how the method was realized as a proof-of-concept system. The first design issue presented is that of glove material and colors, followed by marker material, measurement of finger color variation, system calibration, and error handling.

4.1 Glove material and colors

The material of the glove is of little significance for the purpose of developing and testing the hand/gesture tracking algorithm. As long as it can be represented in the captured image as a solid color, any matte material could be used. We found the inside of a common, rubber washing glove suitable for use in our algorithm development. And though its material is not suitable for wearing as a finalized product, the rubber glove enabled us to quickly begin testing.

The glove has several colored regions, each of which serves a specific purpose for the tracking algorithm. We used felt tip pens to colorize the trackable regions of the glove. Our pens were supplied by a regular hardware store, but any pens that produce suitable colors can be used. The colors used are shown in table 2. Color values are sampled using an Apple iSight camera under a fluorescent lamp.

Color	(#)	(H, S, V) color value	Usage
Orange	**(1)**	H: 10 S: 80 V: 100	**Thumb**
Yellow	**(2)**	H: 40 S: 65 V: 85	**Ring finger**
Light green	**(3)**	H: 85 S:60 V:60	**Bracelet**
Green	**(4)**	H: 160 S: 85 V: 35	**Little finger**
Blue	**(5)**	H: 200 S: 70 V: 50	**Index finger**
Violet	**(6)**	H: 270 S:70 V:35	**Marker**
Pink	**(7)**	H: 320 S: 95 V: 95	**Middle finger**

Table 2. Colors names, values, and usage.
Color numbers refer to hue values in figure 8 and table 3.

4.2 Marker material

Figure 6 shows a visual comparison between a solid and a painted palm marker. The solid marker retains its straight edges even when the hand is folded. Our algorithm is not concerned with the shape of the finger markers. Also, the bracelet is a solid marker attached to the glove that will retain its geometrical features as the hand is rotated and moved. With the painted palm marker, the glove material is however subject to wrinkling, making for calculated rotation values that are less precise than with a solid marker.

Figure 6. Comparison of solid and painted marker.

4.3 Finger color measurements

The colors will be affected by shadows and highlights. As the hand is rotated and/or moved, the color values sampled by the camera will change.

We ran a test by visually inspecting 64 different images captured by the same camera under identical office light conditions. In the sample collection of images, the hand was rotated around all axes as shown in figure 7 (large in Appendix 1). The visual inspection was performed using a color picker tool and read-out values for hue and saturation in standard desktop image editing software. The results are presented in figure 8 where the dark regions represent the range of color hues sampled from an image of the hand in the calibration position. The lighter, expanded regions show the hue ranges that were found in the entire sample set.
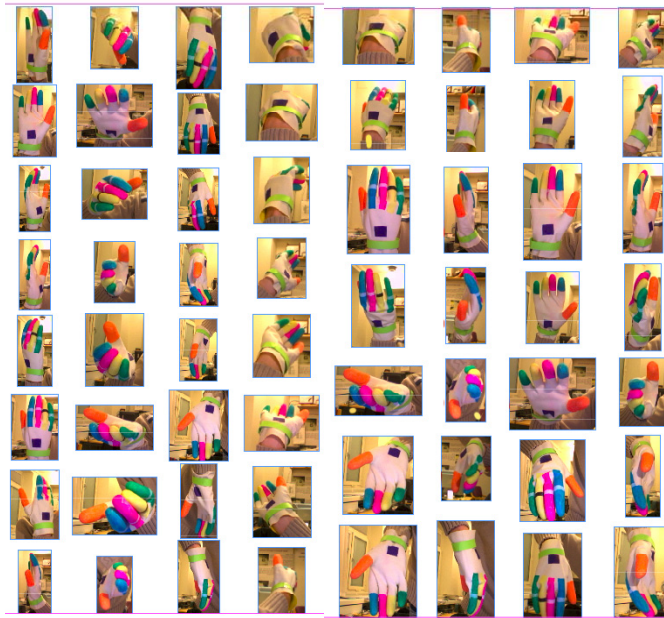

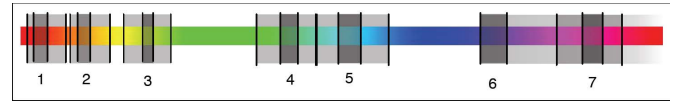**Figure 7. Sample collection of hand positions.**

Figure 8. Hue variation for each of the sampled colors: dark intervals when holding the hand in so-called neutral position; grey intervals for all other hand orientation.

Table 3 further illustrates these findings by comparing the hue and saturation values, seen in the image with the hand in the calibration position (columns 2 and 3), to the full sample set of images (columns 4 and 5). Column 6 indicates the number of images where the hue and/or saturation values did not fall within the ranges shown in columns 4 and 5. The images with hue and/or saturation values that differed significantly from the values seen in the rest of the images were all affected by shadows or reflections. These problems might be addressable by adjusting the light conditions in accordance with the material of the glove.

		Initial hand position		All 64 hand positions		
Color	(#)	H range	S range	H range	S range	#
Orange	(1)	8–15	83–86	4–26	60–100	2
Yellow	(2)	31–38	65–77	27–50	40–85	2
Light green	(3)	69–84	58–64	57–95	50–100	0
Green	(4)	154–167	100	142–175	100	4
Blue	(5)	188–203	100	176–216	70–100	5
Violet	(6)	268–282	75–90	268–365	30–100	0
Pink	(7)	326–336	100	310–346	85–100	0

Table 3. Hue and saturation variation for sampled colors.

4.4 System calibration

Upon application startup, calibration is required to take into account camera hardware and current light conditions. Also, during operation the need for re-calibration may arise due to changing light conditions. Fixed light-conditions are ideal, allowing for a quick and easy calibration stage.

In the calibration phase, the user must first hold their hand inside a square on the screen, so that the program can learn the color of the bracelet. Due to the differences in color sensing between different cameras (figure 9) under various light conditions, color values cannot be hard coded into the application. After the bracelet color has been detected, the system instructs the user to remove the hand from the image frame and remove anything with the same color as the bracelet from the image captured by the camera.

Figure 9. Colors captured by two different cameras under identical light conditions.

The program will not continue until all objects that may interfere with the tracking algorithm are either moved or covered. The bracelet is purposefully given a bright color that is not likely to appear on major objects in a room. Once all interfering objects are moved or covered, the program continues and a checkmark appears on the screen together with instructions for the user to hold up their hand again. At startup, the palm of the hand must face the camera with the fingers fully extended, pointing upwards.

Next, the program uses the bracelet color to discover the square marker directly above the bracelet. The program takes advantage of the contrast in color between the dark marker and the white glove in order to easily identify it. It reads the color value shift between the bracelet, the white space above it, and the marker when the camera tracks the colors upwards from the center of the bracelet. Once the marker is found, the application determines its dimensions when the hand is in the open-palm position. The size and geometry of the marker will allow the application to determine rotation and distance of the hand from the camera.

After finding the marker, the application will scan upwards on the glove to the white area surrounding the marker. The rows containing only white pixels will have a profile with one long run of consecutive colors. The width of the hand is estimated by the length of these runs. After reading a number of white rows the application reaches the fingers where the color values are no longer neutral. These rows will have several consecutive runs of similar color value with a width typically between ¼ and 1/5 of the total hand width. The application will keep track of the colors sampled from the fingers, but the three pixels closest to the fingers' edge will be dropped. To see which pixels belong to a finger, and not the background, the application will first sample the third row that matches the typical profile of five adjacent fingers. The color of the center pixel in each of the five color runs (one for each finger) will be used by the application when processing the subsequent scan lines upwards toward the fingertip.

The application will determine the size of the fully extended fingers, which can then be used to determine the angle of the finger when bent. As the user bends their finger, the application

perceives as being shorter. When the marker and bracelet keep their sizes while the finger shrinks, this is interpreted as bending the finger.

At a certain vertical distance from the palm marker, each finger color will have a specific width represented by continuous stretches of similar hue values. Based on this fact, the program learns each finger's distinct color. The application will function with essentially any combination of finger colors as long as there is sufficient space between the hue values to allow for color variance due to shadows and highlights. This feature will be useful once the algorithm is developed and there may be a need to adjust the glove's colors for maximum recognition or to benchmark the suitability of certain combinations.

When reading the colors, the application takes the average hue found horizontally across the glove. It then creates an interval in the hue spectrum on both sides of the captured color. For saturation and color value, the application will also expand the range/interval to include more colors. This expansion will be larger than that for capturing hue to allow for different shadow/highlight conditions.

4.5 Error handling

In use, the application constantly keeps track of the bracelet position. If it loses the bracelet, the tracking algorithm cannot work and instructions will be given for the user to make the bracelet visible again. As a precaution, the sides of the screen will turn slightly red when the bracelet approaches the edge of the screen. This follows the example of scroll acceleration that is seen in much computer software. As the user moves the pointer closer to the edge of the window, scrolling speed increases more and more. Here, in the same manner, the screen will turn redder and redder as the bracelet gets closer to the edge of the screen. This will train the user to not move the bracelet out of range.

5. WEB DEPLOYMENT

Because our algorithm consumes a low amount of CPU resources, it is possible to implement it for applications running in a web browser. Previously, web deployed real-time hand tracking has been difficult to implement and limited to very basic motion tracking or simple pre-coded gestures.

Flash and Java are among the common technologies for Rich Internet Applications (RIA) in which the user can interact with the content without requiring processing on the server-side. Flash is available for Mac, Windows, and Linux. A Millward-Brown study [18] conducted in March 2007 concludes that 98% of Internet-enabled desktops in mature markets can view applications developed in Flash. In the same study, Java had an 87.6% spread in the same market.

Most recently released as a beta version, a Flash Player Update of Flash 9 further enhances graphic performance and introduces hardware scaling for improved video performance and quality, better utilization of multi-core CPUs, and better communication between Flash and other web programming languages, such as JavaScript [19][20][21].

Applications developed with Flash can be used on any computer that has a network connection. No software installation is needed.

It is currently the only web browser plug-in that can access the video stream from webcams. Theoretically, camera access could be implemented through a signed Java applet, but the wider deployment of Flash and the emerging support for hardware rendering makes Flash a more ideal choice for 3D interaction in web applications.

6. FUTURE WORK

The method we have suggested has still not been finalized. It will probably benefit from further investigation into color tracking in general. Issues regarding color tuning, choice of glove material, application development, Application Programming Interface (API), and the investigation of performance and usability challenges will be of interest in the project's future.

6.1 Color tuning

The values in table 3 show that some colors are more affected than others by highlights and shadows resulting from light reflections. The square pattern is most affected, and we believe this is due to the dark color used in our prototype. It may be necessary to further refine our color choice so that the tracking works well under challenging conditions.

6.2 Suitable glove material

While the rubber glove prototype works well for tracking purposes, it is less suitable for wearing over an extended period of time. The material does not allow the skin to breath and the glove is typically very loose around the fingers. Suggested materials are cotton, nylon, or mixed fibers. The cotton gloves used by antique collectors and the nylon gloves used by brass band musicians will be worth testing.

6.3 Applications and API

We believe that the presented algorithm opens possibilities for a range of potential, innovative applications such as free-form shaping, tabletop interaction, edutainment, and remote collaboration. To support such uses, we plan to offer an API.

6.4 User studies

An empirical evaluation of actual system usage is required. Some of the first functions we plan to evaluate are instructions given at start-up, instructions given during use, help when the system loses sight of the bracelet, and other error handling. The system calibration also needs to be evaluated.

6.5 Performance issues

When color tuning and material issues have been resolved, we plan to examine how robust and accurately the system performs finger bend recognition. There is also a need to examine robustness when the user's hand is not perpendicular to the camera or hands and/or fingers overlap. More generally, we plan to carry out quantitative studies to investigate the soundness and accuracy of the approach presented. With one or a few applications available, we would also like to carry out formal user studies.

REFERENCES

[1] Heumer, G., Ben Amor, H., Weber, M., and Jung, B. (2007): Grasp Recognition with Uncalibrated Data Gloves - A Comparison of Classification Methods. In Proceedings of IEEE Virtual Reality 2007.

[2] Bray, M. Koller-Meier, E., and Van Gool, L. (2007): Smart Particle Filtering for High-Dimensional Tracking. Computer Vision and Image Understanding, Springer LNCS.

[3] Bray, M. Koller-Meier, E. Schraudolph, N. N., and Van Gool, L. (2006): Fast stochastic optimization for articulated structure tracking. Image and Vision Computing, pp. 351-363.

[4] Bray, M., Koller-Meier, E., Mueller, P., Van Gool, L., and Schraudolph, N.N. (2004): 3D Hand Tracking by Rapid Stochastic Gradient Descent Using a Skinning Model. 1st European Conference on Visual Media Production (CVMP).

[5] Sidenbladh, H., Black, M. J., and Fleet, D.J. (2000): Stochastic tracking of 3D human Figures using 2D image motion. In ECCV, pp. 702-718.

[6] Isard, M. and Blake, A. (1998): Condensation – conditional density propagation for visual tracking, International Journal on Computer Vision, 29(1), pp. 5-28.

[7] Stenger, B., Mendonça, P. R. S., and Cipola, R. (2001): Model-Based 3D Tracking of an Articulated Hand. In Proceedings of the CVPR, Vol. II, pp. 310-315, Kauai, USA.

[8] Stenger, B., Mendonça, P. R. S., and Cipola, R. (2001): Model-Based Hand Tracking Using an Unscented Kalman Filter. In Proceedings of the BMVC, Vol. I, pp. 63-72, Manchester, UK.

[9] de la Gorce, M. and Paragios, N. (2006): Monocular Hand Pose Estimation Using Variable Metric Gradient-Descent 17th British Machine Vision Conference, (BMVC).

[10] de la Gorce, M., Paragios, N., and Fleet, D. (2008): Model-Based Hand Tracking with Texture, Shading and Self-occlusions. In Proceedings of the IEEE Conference in Computer Vision and Pattern Recognition (CVPR).

[11] Rasmussen, C. and Hager, G. (1996): Tracking Objects by Color Alone. Technical Report DCS-RR-1114, Yale University.

[12] Wu, Y. and Huang, T. S. (2000): Color tracking by transductive learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, pp. 133-138.

[13] Simon, M., Behnke, S., and Rojas, R. (2001): Robust Real Time Color Tracking. In Robocup 2000: Robot Soccer World Cup IV P. Stone, T. R. Balch, and G. K. Kraetzschmar, Eds. Lecture Notes In Computer Science, vol. 2019. Springer-Verlag, London, pp. 239-248.

[14] CamTrax Technologies: Paradigm Shift: any game, any webcam
http://www.camspace.com
(Last visited: September 1st 2008)

[15] Smith, R., Piekarski, W., and Wigley, G (2005): Hand tracking for low powered mobile AR user interfaces. In Proceedings of the Sixth Australasian Conference on User interface - Volume 40. ACM International Conference Proceeding Series, vol. 104. Australian Computer Society, Darlinghurst, Australia, pp. 7-16.

[16] Duca, F., Fredriksson, J., and Fjeld, M. (2007): Real-Time 3D Hand Interaction: Single Webcam Low-Cost Approach. In Proceedings of the Workshop at the IEEE Virtual Reality 2007 Conference; Trends and Issues in Tracking for Virtual Environments, pp. 1-5.

[17] Kato, H. Billinghurst, M., Poupyrev, I., Imamoto, K., and Tachibana, K. (2000): Virtual Object Manipulation on a Table-Top AR Environment. In Proceedings of the International Symposium on Augmented Reality (ISAR), pp. 111-119.

[18] Adobe Systems Incorporated (2007): Adobe - Flash Player Statistics; http://www.adobe.com/products/player_census/flashplayer/ (Last visited: September 1st 2008)

[19] Adobe Systems Incorporated (2007): Adobe - Flash Player: Adobe Flash Player 9 Release Notes; http://www.adobe.com/support/documentation/en/flashplayer /9/releasenotes.html (Last visited: September 1st 2008)

[20] Adobe Systems Incorporated (2007): Adobe - Flash Player Version Penetration; http://www.adobe.com/products/player_census/flashplayer/v ersion_penetration.html (Last visited: September 1st 2008)

[21] Adobe Systems Incorporated (2007): Adobe Labs - Flash Player 9 Update Beta Release Notes; http://labs.adobe.com/technologies/flashplayer9/releasenotes. html (Last visited: September 1st 2008)

Appendix 1: Sample collection of hand positions (scaled-up version of figure 7).