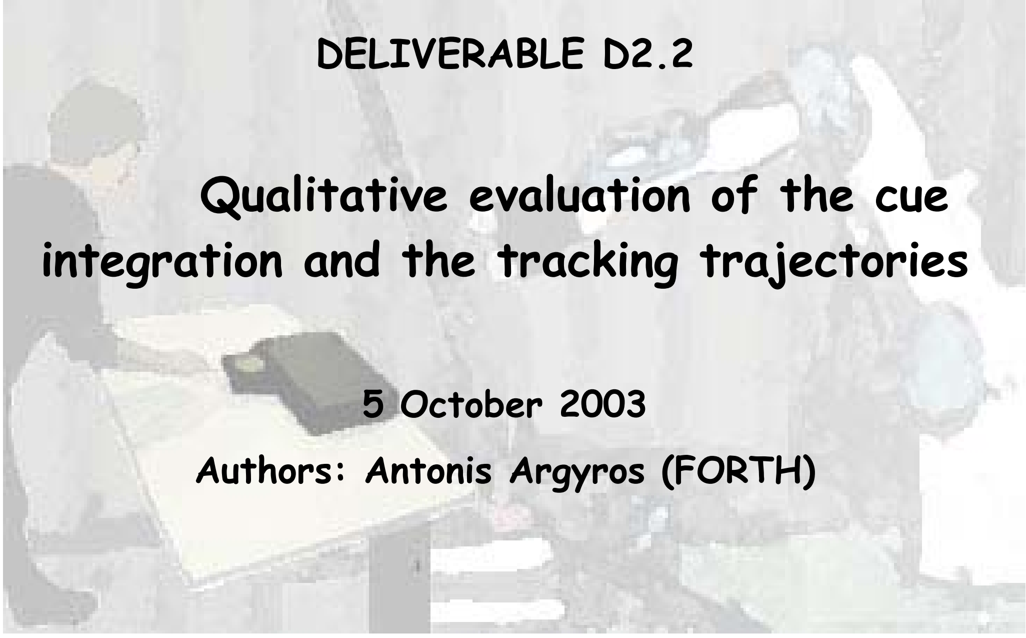


DELIVERABLE D2.2

Qualitative evaluation of the cue integration and the tracking trajectories

5 October 2003

Authors: Antonis Argyros (FORTH)

FOUNDATION FOR RESEARCH AND TECHNOLOGY -HELLAS

PROFACTOR Research for Success

The European Commission Community Research

information society technologies

TABLE OF CONTENTS

1. Summary

Deliverable D2.2 describes a system that performs 3D tracking of multiple skin-colored regions (SCRs) in images acquired by a calibrated, possibly moving stereoscopic rig. The proposed system consists of a collection of techniques that enable the modeling and detection of SCRs, the determination of their temporal association in monocular image sequences, the establishment of their correspondence between stereo images and the extraction of their 3D position in a world-centered coordinate system. The development of these techniques has been motivated by the need for robust, near real time performance. SCRs are detected using a Bayesian classifier whose training is performed with the aid of a novel technique. More specifically, the classifier is bootstrapped with a small set of training data. Then, as new images are being processed, an iterative training procedure is employed to refine the classifier. Furthermore, a technique is proposed for enabling the classifier to cope with illumination changes. Tracking of SCRs in time as well as association of SCRs in the images of the employed stereo rig is performed through computationally inexpensive and robust techniques. Association of SCRs in time is performed by a method that can cope with multiple SCRs, as they dynamically enter and exit the field of view of a moving observer while possibly occluding each other. One of the main characteristics of the developed Skin-Colored Regions Tracker (SCRT) is its ability to report the 3D position of SCRs in a world-centered coordinate system by employing a possibly moving stereo-rig with independently verging CCD cameras. The developed system operates on images of dimensions 640x480 at a rate of 13Hz on a conventional Pentium 4 @ 1.8 GHz processor. Representative experimental results from the application of the developed SCRT to image sequences are also provided.

D2.2 provides improvements on the SCR tracker that has been described in D2.1. The improvements are related mainly to the module that performs associations of SCRs in time.

2. Introduction

Humans have a remarkable ability to visually interpret the activities of other humans. Developing machines with similar perceptual and cognitive capabilities constitutes an ambitious research goal. The accomplishment of this goal will have far-reaching implications in a wide spectrum of applications such as human-machine interaction, gesture tracking for surveillance systems, development of tools for teaching by demonstration, etc. The importance and the difficulty of this problem justify the enormous volume of research efforts that have been devoted worldwide towards providing a robust solution.

A fundamental building block of any system able to interpret activities is one that permits the 3D tracking of a human operator as he performs a certain task. Several sensors and techniques have been developed to achieve this goal [1]. Nevertheless, vision-based methods are considered preferable because they are passive and not invasive, in the sense that they do not require modifications of the environment or any special equipment to be worn by the human operator. A fundamental issue in human tracking is related to the modelling of a human operator. The human body is a complex, non-rigid structure with many degrees of freedom. Therefore, the type and complexity of the proposed models vary dramatically [2,3], depending heavily on the requirements of the application domain under consideration. For example, tracking people in an indoors environment in the context of a surveillance application has completely different modelling requirements compared to tracking the fingers of a hand for vision-based recognition of a sign language. In this work, skin color is the fundamental visual cue employed to detect the presence of a human in a scene. Color offers many advantages over geometric models, such as robustness under occlusions, scale and resolution changes, as well as geometric transformations. Additionally, the computational requirements of color processing are considerably lower compared to those associated with the processing of complex geometric models. For the above reasons, color based models have been applied to a broad spectrum of applications such as content-based image retrieval, quality control, etc.

2.1 Previous work

Vision-based methods for tracking skin-colored regions in 3D need to provide answers to several questions, each of which constitutes an open research problem. How is skin-color modelled and how are instances of the employed model detected in an image? How are detected instances associated temporally in sequences of images? How is 3D position information attained from the inherently 2D observations of the tracked models?

What follows is a description of some representative methods that have been proposed towards solving the above problems. It is important to note that the available options for solving partial problems should be evaluated with respect to several criteria, such as the quality of their results, their robustness and their computational complexity.

Skin color modelling and detection

A major step towards providing a model of skin color is the selection of the color space to be employed. Several color spaces have been proposed including RGB [4], normalized RGB [5,6], HSV [7], YcrCb [8], YUV [9], etc. Color spaces efficiently separating the chrominance from the luminance components of color are typically considered preferable due to the fact that by employing only chrominance-dependent components of color, some degree of robustness to illumination changes can be achieved. The choice of such color spaces is also justified by the fact that different skin tones differ mostly on their chrominance and less on their intensity. Terrillon et al [10] review different skin chrominance models and evaluate their performance. A recent survey [17] includes a very interesting overview of the use of color for face (and, therefore skin-color) detection.

Having selected a suitable color space, the simplest methods define skin color by employing bounds in the coordinates of this space [8]. These bounds are typically selected empirically, i.e. by examining the distribution of skin colors in a pre-selected set of images. Another approach is to assume that the probabilities of skin colors follow a distribution that can be learned. This is achieved through an off-line procedure, although on-line iterative methods have also been suggested [7].

In contrast to the aforementioned non-parametric approaches, another paradigm related to methods making use of parametric models. These methods are based either on a unimodal Gaussian probability density function [11,5,9] or multimodal Gaussian mixtures [12,13,14,15] that model the probability distribution of skin color. In the case of a unimodal Gaussian density function, maximum likelihood estimation techniques are used to derive its parameters. The case of multi-modal Gaussian mixtures requires an Expectation-Maximization (EM) algorithm to be employed. According to Yang et al [16], a mixture of Gaussians is preferable compared to a single Gaussian distribution. Still, histogram models provide better accuracy and lower computational cost than mixture models for the detection of skin-colored areas in an image [6].

A few of the proposed methods perform some sort of adaptation to become insensitive to changes in the illumination conditions. For example, it has been suggested [12,13] to adapt a Gaussian mixture model that approximates the multi-modal distribution of the object's colors, based on a recent history of detected skin colored regions.

Skin color is an important cue towards detecting the presence of humans in a scene. However, it is often not enough to separate skin objects from non-skin objects that appear to be skin-colored. Therefore, skin color is often fused with other cues like motion, texture, shape and 3D structure information. A recent survey [17] gives an interesting overview of the use of color and other visual cues towards skin color detection.

Tracking

As soon as skin-colored regions have been modeled and can be detected in an image, another major problem relates to the temporal association of these observations in an image sequence. The traditional approach to solving this problem has been based on the original work of Kalman [18] and its extensions. If the observations and object dynamics are of Gaussian nature, Kalman filtering suffices to solve the tracking problem. However, in many cases the involved distributions are non-gaussian and thus the underlying assumptions of Kalman filtering are violated.

As reported in [22], recent research efforts that deal with object tracking can be classified into two categories, the ones that solve the tracking problem in a non-Bayesian framework [19,20,21,22,23,24] and those that tackle it in a Bayesian framework [25,26,27,28,29,30,31]. In many cases [25,26,27], the focus is on single object tracking. These single-object approaches usually rely upon sophisticated, powerful object models. In other cases [28,29,30,31] the problem of tracking several objects in parallel is addressed. Some of these methods solve the multi-object tracking problem by employing configurations of individual objects, thus reducing the multi-object tracking problem to several instances of the less-difficult single-object tracking problem. Other methods employ particle filtering-based algorithms, which track multiple objects simultaneously.

Despite the considerable amount of research devoted to tracking, an efficient and robust solution to the general formulation of the problem is still lacking, especially for the case of simultaneous tracking of multiple targets.

3D reconstruction

Tracking provides a mechanism for associating observations of models over time. Still, it involves 2D information regarding the location of the tracked model. To be able to provide 3D position information, at least two observations of the same object from different viewpoints are required. Although techniques based on multiple views acquired from more than two cameras have been proposed [32], most of the existing approaches [33] consider a single, calibrated stereoscopic system.

To the best of our knowledge, all existing approaches consider a static stereoscopic system. This is due to the fact that by employing a moving stereoscopic system the process of tracking is complicated considerably. If the stereoscopic system moves, everything changes in the fields of view of both cameras. Therefore, background subtraction (i.e. temporal change detection) cannot be used as a means of providing additional evidence regarding the presence of moving skin-colored regions (SCRs). In the particular case of cameras with independent pan and tilt control, the geometry of the stereoscopic system does not remain constant over time and, therefore, further complications related to 3D reconstruction are introduced. On the other hand, employing a moving stereoscopic system is often desirable because, in this case, cameras can be purposefully positioned in a way that facilitates the observation of a certain activity.

An important implication of employing a stereoscopic configuration towards computing 3D trajectories of the tracked objects is that model detection and tracking should be performed in both views, thus increasing the computational requirements of the tracking system. In addition, an extra computational procedure is required to associate the detected models between the images of a stereo pair. This is a crucial task because it permits the extraction of 3D information through standard 3D reconstruction techniques [34,35].

The proposed approach

In this paper, we present our approach to 3D tracking of multiple skin-colored regions (SCRs) observed by a moving stereoscopic system. This work has been carried out in the context of a more general research effort [36] towards developing a cognitive vision methodology that permits the interpretation of the activities of people handling tools. Research and development is focused on the active observation and interpretation of the activities, on the extraction of the essential activities and their functional dependence, and on organizing them into constituent behaviour elements. The approach is active in the sense that the system seeks to obtain views that facilitate the interpretation of the observed activities. Therefore, the ability to modify the viewpoint of observation of a certain activity is of utmost importance. Moreover, task and context knowledge is exploited as a means to constrain interpretation. Robust perception and interpretation of activities is the key to capturing the essential information that allows reproduction of task sequences from easy-to-understand representations.

The proposed system is able to track and report the 3D trajectories of all skin-colored regions (SCRs) present in the viewed scene. Compared to existing approaches, the proposed method for detecting SCRs has several attractive properties. A skin-color representation is learned through an off-line procedure. A new technique is proposed that permits the avoidance of much of the burden involved in the process of generating training data. Moreover, the proposed method adapts the skin-color model based on the recent history of tracked SCRs. Thus, without relying on complex models, the proposed approach is able to robustly and efficiently detect SCRs even in the case of changing illumination conditions.

The proposed system employs a moving stereoscopic system with cameras that have independent vergence control. To the best of our knowledge, this is the first method that is capable of tracking SCRs based on a moving stereoscopic system. Despite the motion of the cameras, the estimation of the 3D position of the detected and tracked SCRs is performed on a world-centered (i.e. extrinsic to the cameras) coordinate system. SCRs are tracked in time and associated between the images of each stereo pair by employing simple, computationally inexpensive techniques. Implemented in C, the developed tracking system can track multiple SCRs at a rate of 13Hz on a Pentium 4 processor running Linux; the employed stereo stream consists of images with dimensions 640x480.

The rest of the document is organized as follows. Section 3 describes the proposed SCR Tracker (SCRT). Section 4 studies the described SCRT system in the context of the cognitive vision framework of the ActIPret project. Section 5 provides sample results from the operation of the SCRT in both monocular and binocular image sequences. In section 6, issues related to the computational performance of the SCRT are discussed. Section 7 provides a list of extensions and ideas for improvements that are still under investigation. The main conclusions of this work are summarized in section 8.

3. Tracking of Skin-colored Regions

The developed Skin-Colored Regions Tracker (SCRT) is able to detect multiple SCRs and report their 3D position using images acquired by a moving stereoscopic head, such as the one shown in Fig. 1.

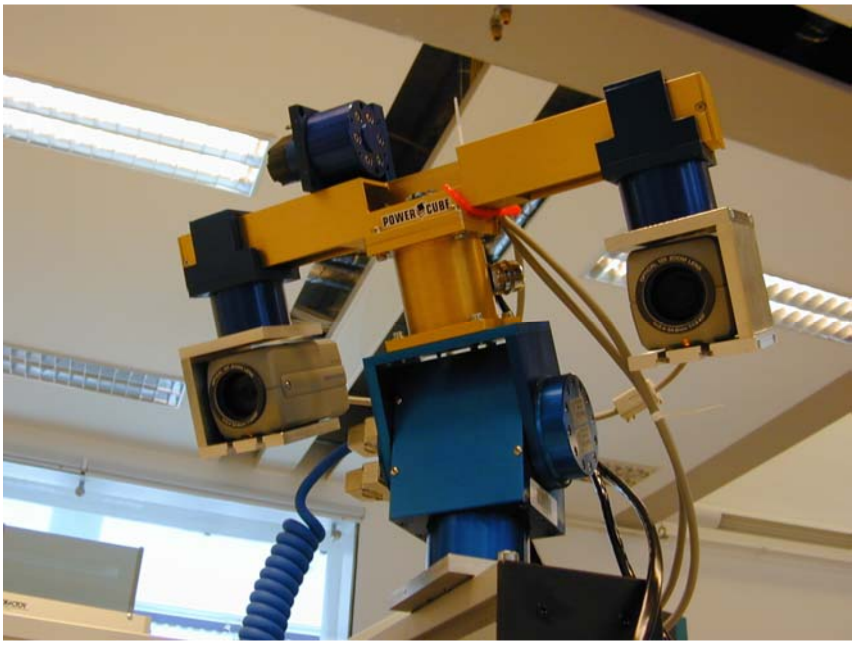


Figure 1: The stereoscopic head (courtesy Profactor GmbH) that is used to acquire stereo image pairs that are fed to the developed SCRT system.

Apart from providing raw stereo image streams, the stereoscopic head continuously delivers the position and orientation of each of the two CCD cameras with respect to a world-centered coordinate system. This is accomplished through the use of proprioceptive information provided by the motor encoders of the head. The developed SCRT exploits multiple cues towards achieving SCR tracking. These cues include color information, structure information as well as information regarding the camera positions and the epipolar geometry of the stereo system. The remainder of this section provides a brief outline of the SCRT system; more detailed descriptions of each of its functional modules are provided in subsequent sections.

At each time instance t, the stereoscopic system acquires a synchronized image stereo pair, IL(t) and IR(t). Each of the pairs' images is independently fed to a skin color detection (SCD) module. SCD involves four key operations, specifically (a) measurement of the probability of a pixel being skin colored, (b) hysteresis thresholding on the derived probabilities map, (c) connected components labeling to come up with SCRs and, (d) computation of statistical information for each SCR (up to 2nd order moments). The derived SCRs, together with the SCRs derived at the previous time instance t-1, are then associated in time (AT module). The aim of this module is (a) to assign a new, unique label to each new SCR (i.e. to an SCR that appears in the field of view for the first time) and (b) to propagate the labels of already detected SCRs in time. Then, the SCRs detected in the left and the right images along with the associated labels, are fed to a module that corresponds SCRs between the two images of the stereo pair (AS module). In fact, each SCR in the right image of the stereo pair is assigned the label of the corresponding SCR in the left image of the stereo pair, if such a corresponding SCR actually exists. Having completed this type of association, the centroids of the corresponding SCRs are refined using a correlation-based stereo matching technique, carried out by the centroids matching (CM) module. This ensures that these points correspond to the same 3D scene point. The refined matches are then fed to a 3D reconstruction (3DR) module which, taking into account the known geometry of the stereoscopic system as well as the intrinsic calibration parameters of the cameras, computes the 3D location of the centroid pertaining to each SCR. Finally, the 3D position that the system reports for each SCR is a weighted sum of 3D measurements in a sliding time window. The temporal smoothing (TS) module provides this type of functionality. A high-level block diagram providing an overview of the developed SCRT system is illustrated in Figure 2. In what follows, a more detailed description of each of the aforementioned modules is provided.

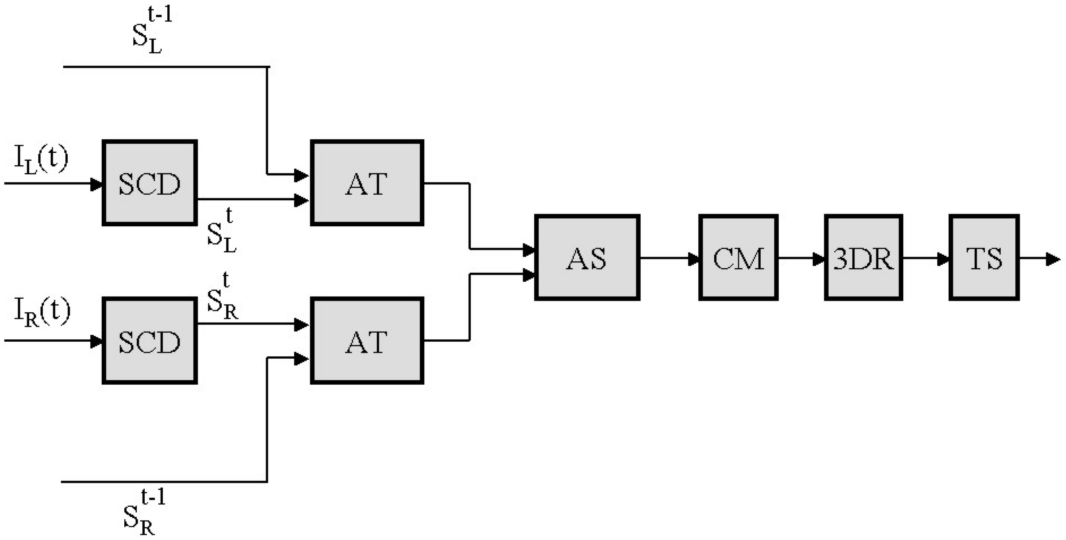


Figure 2: Block diagram of the proposed SCRT system.

3.1 Skin color detection (SCD module)

Skin color detection (SCD) is one of the fundamental building blocks of the developed SCRT system. The goal of the SCD module is to detect skin-colored regions in an image. SCD adopts a Bayesian approach, involving (a) an iterative training phase and (b) an adaptive detection phase.

Basic supervised training and detection mechanisms

A set of training input images is selected on which a human operator manually marks skin-colored regions. The color representation used in this process is YUV [37] 4:2:2 that directly encodes the images acquired by the cameras used in the stereoscopic system of Fig. 1. However, the Y-component of this representation is not employed for two reasons: (a) the Y-component corresponds to the illumination of an image point and therefore, by omitting it the developed classifier gains some illumination-independence characteristics, (b) by employing a 2D color representation (UV), as opposed to a 3D one (YUV), the dimensionality of the problem is reduced and the computational requirements of the overall system are lowered.

Assuming that image points *I(x, y)* have a color *c = c(x, y) = (u, v)*, the training set is used to compute the following information:

- The prior probability *P(s)* of having skin color in an image. This is the ratio of the skin-colored image points in the training set over the total number of image points.
- The prior probability *P(c)* of the occurrence of each color *c* in the training set. This is computed as the ratio of the number of occurrences of each color *c* over the total number of image points in the training set.
- The prior probability *P(c|s)* of a color *c* being a skin color. This is defined as the ratio of the number of occurrences of a color *c* within the skin-colored areas over the number of skin-colored image points in the training set.

Based on the information extracted in the training phase, the probability *P(s|c)* of a color *c* being a skin color can be computed by employing the Bayes rule [38]:

$$P(s \mid c) = \frac{P(c \mid s)P(s)}{P(c)} \tag{1}$$

Then, the probability of each image point *I(x, y)* being skin-colored can be determined with the aid of a look-up table, indexed with the point's color. The resulting probability map is subsequently thresholded and all image points with probability $P(s|c) > T_{max}$ are considered as skin-colored. These points constitute the seeds of potential SCRs. More specifically, image points with probability $P(s|c) > T_{min}$ where $T_{min} < T_{max}$, that are immediate neighbors of skin-colored image points are recursively added to the set of skin-colored points. The rationale behind this region growing operation is that an image point with relatively low probability of being skin-colored should be considered as such, in case that it is a neighbor of an image point with high probability of being skin-colored. This hysteresis thresholding type of operation has been very successfully applied to edge detection[39] and also proves extremely useful in the robust identification of SCRs. Indicative values for the thresholds T_{max} and T_{min} are 0.5 and 0.15, respectively.

A connected components labeling algorithm is then responsible for assigning different labels to the image points of different SCRs. Size filtering on the derived connected components is also performed to eliminate small, isolated blobs that are attributed to noise and do not correspond to interesting skin-colored regions. Thus, connected components that consist of less than T_{size}=500 image points are rejected from further consideration. Each of the remaining connected components corresponds to an SCR whose 2D image position is defined by its centroid.

Adaptability

The basic scheme for skin color detection described previously has two major drawbacks:

- **Training:** Training is an off-line procedure that does not affect the on-line performance of the SCRT system. Nevertheless, it is a very time-consuming process, in the sense that a human operator should manually mark all skin-colored pixels in the chosen training set. Moreover, in order to come up with a training set that is capable of supporting tracking of various skin tones in images acquired from different cameras, a large training set is required. Therefore, a method that automates the processing of training data is considered quite important.
- **Detection:** In the case of varying illumination conditions, the SCD module may produce poor results, despite the fact that the employed color representation has certain illumination-independent characteristics. Hence, a method is required that adapts the representation of skin-colored image points according to the recent history of detected skin-colored points.

To cope with the first problem, an adaptive training procedure has been developed. Training is performed on an initial, small set of images for which the human operator provides ground truth by defining skin-colored regions. Then, detection together with hysteresis thresholding is used to continuously update the prior probabilities $P(s)$, $P(c)$ and $P(c|s)$ in new images. The updated prior probabilities are then used to re-classify the full data set into skin-colored and non-skin colored image points. In cases where the classifier produces wrong results (false positives / false negatives), manual user intervention for correcting these errors is necessary; still, up to this point, the classifier can automatically complete much of the required work. The final training phase of the classifier is then performed based on the training set that results after user editing. This process for adapting the prior probabilities *P(s)*, *P(c)* and *P(c|s)* can either be disabled as soon as it is decided that the achieved training is sufficient for the purposes of the SCRT system or continue as more input images are fed to the system.

At this point, it is important to note that hysteresis thresholding is crucial for achieving the previously described adaptation of prior probabilities. If hysteresis thresholding is not used, colors with probability $P(s|c)<T_{max}$ will never have the chance of being considered as skin colors. Hysteresis thresholding with a threshold T_{min} considerably smaller than T_{max}, allows colors with low probability of representing skin to be considered as skin colors and permits the appropriate adaptation of their probabilities.

To solve the second problem, the SCD module maintains two sets of prior probabilities *P(s)*, *P(c), P(c|s),* corresponding to the training set and $P_w(s)$, $P_w(c)$, $P_w(c|s)$, corresponding to the evidence that the system gathers during the *w* most recent frames. Clearly, the second set better reflects the "recent" appearance of SCRs and is better adapted to the current illumination conditions. SCD is then performed based on

$$P_A(s\,|\,c) = aP(s\,|\,c) + (1-a)P_w(s\,|\,c) \tag{2}$$

where *P(s|c)* and $P_w(s|c)$ are both given by eq. (1), but involve prior probabilities that have been computed from the whole training set (for *P(s|c)*) and prior probabilities that have been computed from the detection results in the last *w* frames (for the case of $P_w(s|c)$). In eq. (2), a is a sensitivity parameter that controls the influence of the training set in the detection process ($0 < a \leq 1$). If $a=1$ then SCD takes into account only the training set and no adaptation takes place; if a is close to zero, then the SCD becomes very "reactive", relying strongly on the recent past for deriving a model of the immediate future. Values of $a=0.8$ and *w*=5 give very good results in the tests that have been carried out.

A basic advantage of the proposed scheme lies in its simplicity. Other methods for adaptation have been proposed in the literature [12,13]. However, these methods require much more complex modeling of the color characteristics of skin (i.e. modeling based on mixtures of Gaussians). An interesting study [6] has shown that, compared to mixture models, histogram models like the one proposed in this work provide better accuracy and lower computational cost for skin detection.

3.2 Associating SCRs in time (AT module)

As soon as an SCR is detected, it has to be tracked over time. This is a crucial functionality of the SCRT system since it provides the temporal continuity of SCR observations.

We assume that at time *t*, *M* blobs have been detected as described previously. Each blob b_j, $1\leq j \leq M$, corresponds to a set of connected skin-colored image points. Note that the correspondence among blobs and objects is not necessarily one-to-one. As an example, two crossing hands are two different skin-colored objects that appear as one blob at the time one occludes the other. In this work we assume that an object may correspond to either one blob or part of a blob. Symmetrically, one blob may correspond to one or many objects. We also assume that the spatial distribution of the pixels depicting a skin-colored object can be coarsely approximated by an ellipse. This assumption is valid for skin-colored objects like hand palms and faces. Let *N* be the number of skin-colored objects present in the viewed scene at time *t* and o_i, $1 \leq i \leq N$, be the set of skin pixels that image the *i*-th object. We also denote with $h_i = h\left(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \vartheta_i\right)$ the ellipse model of this object where $\left(c_{x_i}, c_{y_i}\right)$ is its center, α_i and β_i are the lengths of its major and minor axis, respectively, and ϑ_i is its orientation on the image plane. Finally, we use capital letters $B = \bigcup_{j=1}^{M} b_j$, $O = \bigcup_{i=1}^{N} o_i$ and $H = \bigcup_{i=1}^{N} h_i$ to denote the union of skin-colored pixels, object pixels and ellipses, respectively. Tracking amounts to determining the relation between object models (h_i) and observations (b_j) in time.

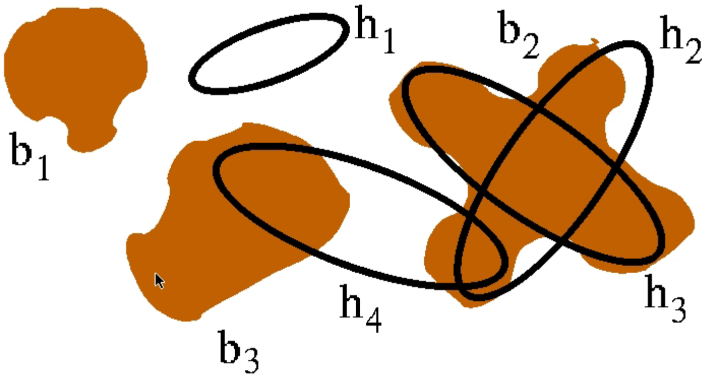


Figure 3: Various cases of the relation between skin-colored blobs and object hypotheses.

Figure 3 exemplifies the problem. In this particular example there are three blobs (b_1, b_2 and b_3) while there are four object hypotheses (h_1, h_2, h_3 and h_4) from the previous step.

What follows is an algorithm that can cope effectively with the data association problem. The proposed algorithm needs to address three different sub-problems: (a) object hypothesis generation (i.e. an object appears in the field of view for the first time) (b) object hypothesis tracking in the presence of multiple, potential occluding objects (i.e. previously detected objects move arbitrarily in the field of view) and (c) object model hypothesis removal (i.e. a tracked object disappears from the field of view).

Object hypothesis generation

We define the distance *D(p, h)* of a point *p=p(x, y)* from an ellipse $h\left(c_x, c_y, \alpha, \beta, \vartheta\right)$ as follows:

$$D(p,h)=\sqrt{\vec{u}\cdot\vec{u}} \tag{3}$$

where

$$\vec{u}=\begin{bmatrix}\cos(\vartheta) & -\sin(\vartheta)\\ \sin(\vartheta) & \cos(\vartheta)\end{bmatrix}\left(\frac{x-x_c}{\alpha},\frac{y-y_c}{\beta}\right)$$

From the definition of *D(p, h)* it turns out that the value of this metric is less than 1.0, equal to 1.0 or greater than 1.0 depending on whether point *p* is inside, on, or outside ellipse *h*, respectively. Consider now a model ellipse *h* and a point *p* belonging to a blob *b*. In the case that *D(p,h) < 1.0*, we conclude that the point *p* and the blob *b* support the existence of the object hypothesis *h* and that object hypothesis *h* predicts blob *b*. Consider now a blob *b* such that:

$$\forall p \in b, \min_{h\in H}\left\{D(p,h)\right\} > 1.0 \tag{4}$$

Equation 4 describes a blob with empty intersection with all ellipses of the existing object hypotheses. Blob b_1 in Fig. 3 is such a case. This implies that none of the existing object hypotheses predicts the existence of this blob. For each such blob, a new object hypothesis is generated. The parameters of the generated object hypothesis can be derived directly from the statistics of the distribution of points belonging to the blob. The center of the ellipse of the object hypothesis becomes equal to the centroid of the blob and the rest of the ellipse parameters can be computed from the covariance matrix of the bivariate distribution of the location of blob points. More specifically, it can be shown that if the covariance matrix Σ of the blob's points distribution is $\Sigma=\begin{bmatrix}\sigma_{xx} & \sigma_{xy}\\ \sigma_{xy} & \sigma_{yy}\end{bmatrix}$

then an ellipse can be defined with parameters:

$$\alpha=\sqrt{\lambda_1}, \beta=\sqrt{\lambda_2}, \vartheta=\left(\frac{-\sigma_{xy}}{\lambda_1-\sigma_{yy}}\right) \tag{5}$$

where $\lambda_1 = \frac{\sigma_{xx} + \sigma_{yy} + \Lambda}{2}$, $\lambda_2 = \frac{\sigma_{xx} + \sigma_{yy} - \Lambda}{2}$ and $\Lambda = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 - 4\sigma_{xy}{}^2}$. Algorithmically, at each time *t*, all detected blobs are tested against the criterion of eq. 4. For all qualifying blobs, an object hypothesis is formed and the corresponding ellipse parameters are determined based on eqs. (5). Moreover, all such blobs are excluded from further consideration in the subsequent steps of object tracking.

Object hypothesis tracking

After new object hypotheses have been formed as described in the previous section, all the remaining blobs must support the existence of past object hypotheses. The main task of the tracking algorithm amounts to associating blob pixels to object hypotheses. There are two rules governing this association:

- **Rule 1:** If a skin-colored pixel of a blob is located within the ellipse of some object hypothesis (i.e. supports the existence of the hypothesis) then this pixel is considered as belonging to this hypothesis.
- **Rule 2:** If a skin-colored pixel is outside all ellipses corresponding to the object hypotheses, then it is assigned to the object hypothesis that is closer to it, using the distance metric of eq. (3).

Formally, the set *o* of skin-colored pixels that are associated with an object hypothesis *h* is given by *o* = $R_1 \cup R_2$ where $R_1 = \{ p \in B : D(p,h) < 1.0 \}$ and $R_2 = \{ p \in B : D(p,h) = \min_{k \in H} D(p,k) \}$.

In the example of Fig. 3, two different object hypotheses (h_2 and h_3) are "competing" for the skin-colored area of blob b_2. According to the rule 1 above, all skin pixels within the ellipse of h_2 will be assigned to it. According to the same rule, the same will happen for skin pixels under the ellipse of h_3. Note that pixels in the intersection of these ellipses will be assigned to both hypotheses h_2 and h_3. According to rule 2, pixels of blob b_2 that are not within any of the ellipses, will be assigned to their closest ellipse which is determined by eq. (3).

Another interesting case is that of a hypothesis that is supported by more than one blobs (see for example hypothesis h_4 in Fig. 3). Such cases may arise when, for example, two objects are connected at the time they first appear in the scene and later split. To cope with situations where a hypothesis *h* receives support from several blobs, the following strategy is adopted. If there exists only one blob *b* that is predicted by *h* and, at the same time, not predicted by any other hypothesis, then *h* is assigned to *b*. Otherwise, *h* is assigned to the blob with which it shares the largest number of skin-colored points. In the example of Fig. 3, hypothesis h_4 gets support from blobs b_2 and b_3. Based on the above rule, it will be finally assigned to blob b_3.

After having assigned skin pixels to object hypotheses, the parameters of the object hypotheses h_i are re-estimated based on the statistics of pixels o_i that have been assigned to them.

Object hypothesis removal

An object hypothesis should be removed either when the object moves out of the camera's field of view, or when the object is occluded by another (non-skin colored) object in the scene. Thus, an object hypothesis h should be removed from further consideration whenever

$$\forall p \in b, \min_{h \in H} \left\{ D(p,h) \right\} > 1.0 \tag{6}$$

Equation (6) essentially describes hypotheses that are not supported by any skin-colored image points. Hypothesis h_1 in Fig. 3 is such a case. In practice, we permit an object hypothesis to “survive” for a certain amount of time, even in the absence of any support, so that we account for the case of possibly poor skin-color detection. In our implementation, this time interval has been set to half a second, which approximately amounts to fourteen image frames.

Prediction

In the processes of object hypothesis generation, tracking and removal that have been considered so far, data association is based on object hypotheses that have been formed at the previous time step. Therefore, there is a time lag between the definition of models and the acquisition of data these models need to represent. Assuming that the immediate past is a good prediction for the immediate future, a simple linear rule can be used to predict the location of object hypotheses at time *t*, based on their locations at time *t-2* and *t-1*. Therefore, instead of employing $h_i = h\left(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \vartheta_i\right)$ as the ellipses describing the object hypothesis *i*, we actually employ $\hat{h}_i = h\left(\hat{c}_{x_i}, \hat{c}_{y_i}, \alpha_i, \beta_i, \vartheta_i\right)$ where $\left(\hat{c}_{x_i}(t), \hat{c}_{y_i}(t)\right) = C_i(t) + \Delta C_i(t)$. In the last equation, $C_i(t)$ denotes $\left(c_{x_i}, c_{y_i}\right)$and \$ $\Delta C_i(t) = C_i(t-1) - C_i(t-2)$.

The above equations postulate that an object hypothesis will maintain the same direction and magnitude of translation on the image plane, without changing any of its other parameters. Experimental results have shown that this simple prediction mechanism performs surprisingly well in complex object motions, provided that processing is performed close to real-time.

3.3 Associating SCRs in a stereo pair (AS module)

In order to provide information regarding the 3D position of each SCR, the tracker should be able to associate SCRs between the images of the stereo pair. This purpose is served by the AS module. As it has already been stated in this paper, it is assumed that the position and orientation of each camera of the stereo pair is known with respect to a world-centered coordinate system. Based on this information, it is possible to compute the rotation matrix *R* and the translation vector *t* of the relative rigid motion between the coordinate systems of the cameras of the stereoscopic system. This, in turn, provides the means to analytically compute the fundamental matrix *F* that captures the underlying epipolar geometry of the stereo pair [40]:

$$F = \frac{1}{\det(A_R)}[e_R]_x H_\infty \tag{7}$$

where

$$e_R = A_R t \tag{8}$$

In the above equations, A_L and A_R are the intrinsic calibration matrices of the left and right cameras respectively, H_∞ is the homography of the plane at infinity and e_R is the epipole in the right image. The symbol $[e_R]_x$ denotes the skew-symmetric matrix associated with vector cross product, i.e. for each vector y, $[e_R]_x y = e_R \times y$. Assuming that m_L and m_R are two corresponding points in the left and the right images of the stereo pair respectively, then m_R is constrained to lie on the epipolar line l_R defined [35] as l_R=Fm_L. Similarly, m_L is constrained to lie on the epipolar line l_L defined as l_L=$F^T m_R$. (see Fig. 4). The AS module employs the epipolar constraint to associate SCRs between the images of a stereo pair. As in the case of the AT module, we denote with S_L^t and S_R^t the sets of SCRs that have been detected at time *t* in the left and right images of the stereo pair, respectively. Moreover, $S_L^t(i)$ and $S_R^t(j)$ denote specific SCRs with indices *i* and *j* detected at time *t*, $1 \le i \le N_L^t$, $1 \le j \le N_R^t$. A distance measure $D_S(S_L^t(i), S_R^t(j))$ is defined between two SCRs $S_L^t(i), S_R^t(j)$ as

$$D_S(S_L^t(i), S_R^t(j)) = \max\left\{d\left(Fm_L^t(i), m_R^t(j)\right), d\left(F^T m_R^t(j), m_L^t(i)\right)\right\} \tag{9}$$

In the above equation, $m_L^t(i)$ and $m_R^t(j)$ are the centroids of SCRs $S_L^t(i)$ and $S_R^t(j)$, respectively, and $d\left(l, p\right)$ denotes the Euclidean distance of point *p* from line *l*. An SCR $S_L^t(i)$ matches an SCR $S_R^t(j)$ where

$$j = \arg\min_{1 \le k \le N_R^t} \left\{ D_S(S_L^t(i), S_R^t(k)) \right\} \tag{10}$$

Symmetrically, an SCR $S_R^t(j)$ matches an SCR $S_L^t(i)$ where

$$i = \arg\min_{1 \le k \le N_L^t} \left\{ D_S(S_R^t(j), S_L^t(k)) \right\} \tag{11}$$

$$H_\infty = A_R R A_L^{-1} \tag{12}$$

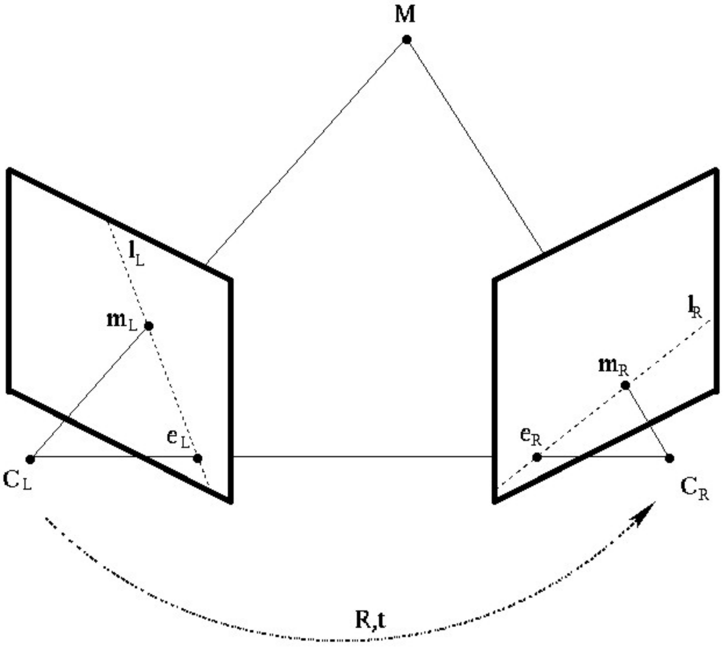


Figure 4: A graphical illustration of the epipolar geometry of a stereo pair. The epipolar plane *C_LMC_R* intersects the two image planes along the epipolar lines l_L and l_R.

Two SCRs $S_L^t(i)$ and $S_R^t(j)$ are assumed to correspond to each other if (a) $S_L^t(i)$ matches $S_R^t(j)$, (b) $S_R^t(j)$ matches $S_L^t(i)$ and (c) $D_S(S_L^t(i), S_R^t(j)) < T_S$, where T_S is a threshold depending on the accuracy of the estimated epipolar geometry. Note that by definition this distance definition is symmetric, i.e. $D_S(S_L^t(i), S_R^t(j)) = D_S(S_R^t(j), S_L^t(i))$.

For all corresponding SCRs, the label of the SCR in the left image is propagated to the corresponding SCR in the right stereo image. All SCRs that have not been corresponded are excluded from further consideration in the subsequent process of 3D position estimation. Such SCRs typically correspond to skin-colored objects that are visible only in one of the two cameras of the stereo pair.

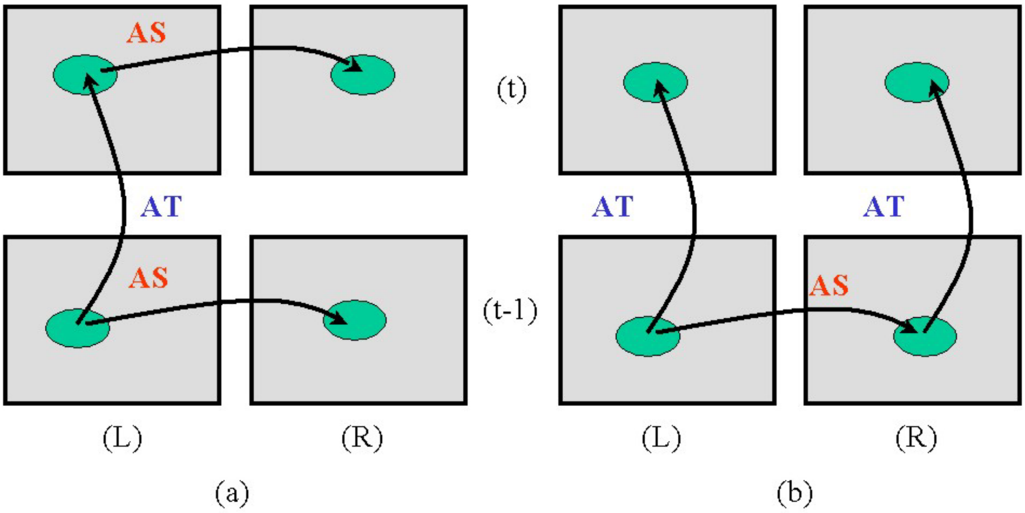


Figure 5: Two schemes for achieving the propagation of labels of SCRs both in time and between two stereo views. (a) The AS module is used to associate SCRs between the left and the right images of the stereo pair, at each point in time. The AT module is used to propagate labels in time, only in the image sequence of the left camera. (b) The AS module is used to associate SCRs only when a new SCR appears in the field of view. Two instances of the AT module are then used to propagate the SCR labels in time independently for the left and right image sequences. Since AT is typically more robust compared to AS, the second approach is adopted.

The above method for associating SCRs becomes unstable if the epipolar geometry is not accurately computed. In this case, the threshold T_S has to be set conservatively to a quite large value, which leaves room for errors in the association of SCRs. For this reason, 3D position information from previous time instances can be used, if available. More specifically, when computing distances $D_S(S_L^t(i), S_R^t(j))$, the 3D position of the SCR that results from the assumption that $S_L^t(i)$ actually corresponds to $S_R^t(j)$, is computed. If the resulting 3D position is invalid (in the sense that this position is either not plausible or it differs substantially from the SCR's 3D position in the previous time step), a penalty term is added in the corresponding distance measure to guarantee that $S_L^t(i)$ and $S_R^t(j)$ will not be considered as corresponding.

In general, the camera position and orientation information that is computed from the encoders of the stereoscopic system is not reliable enough to enable the accurate estimation of the epipolar geometry of the stereo pair. For instance, experiments carried out with prerecorded image sequences indicated that the average distance of image points from their estimated epipolar lines was in the order of 15 pixels. This in turn, affects the robustness of the AS module. To overcome this problem, AS is applied only to SCRs that appear in the field of view for the first time. As soon as this is achieved, the AT module which is more robust compared to AS, assumes the role of propagating the correct SCR labels in both images of the stereo pair. This is further exemplified in Fig. 5. It should be mentioned at this point that a more accurate, image-based estimation of the stereo system's epipolar geometry [41] (as opposed to the currently employed encoders-based estimation) will considerably improve the robustness of the AS module. Still, such methods incur a significant computational overhead that is undesirable in the context of the SCRT system.

3.4 Centroid matching (CM module)

The correspondence of SCRs between the left and the right images of the stereo pair leads to a rough correspondence between SCR centroids. This could be directly used for deriving the 3D position of SCRs. However, centroids are computed by the SCD module from the mean *x*- and *y*-coordinates of each SCR. Therefore, it is not guaranteed that the left and right centroids of an SCR correspond to the same 3D point. In order to refine this rough, initial correspondence, a correlation-based matching algorithm is employed. Let m_L and m_R denote the centroids of an SCR in the left and right images of the stereo pair, respectively. Then, a model template W_M around m_L and a search window W_S around m_R are defined. W_M is placed over all possible positions in W_S and a correlation measure Δ is computed. The location m'_R in W_S where the correlation measure Δ is maximized (Δ_{MAX}) is considered as the refined right centroid of the specific SCR. The process is repeated symmetrically, by defining a model template around m_R and a search window around m_L. If this search gives rise to a correlation score greater than Δ_{MAX} for some point m'_L in the left image, then we consider the (m'_L, m_R) pair of centroid correspondences instead of the (m_L, m'_R) pair. This centroid refinement process is repeated for all pairs of corresponding SCRs. Note that if epipolar geometry has been computed accurately enough, search bands defined along epipolar lines can be used instead of search regions. The correlation measure Δ used in the CM module is inspired by the work of Hirschmüller [42] on dense stereo matching. The model template W_M is divided into five overlapping sub-windows, a central (W_C), an upper-left (W_{UL}), an upper-right (W_{UR}), a bottom-left (W_{BL}) and a bottom-right (W_{BR}). These five windows overlap each other, as shown in Fig. 6. All sub-windows have dimensions of 3x3, resulting in a 5x5 model template. Since in the work of Hirschmüller [42] a rectified stereo pair is assumed, search templates are essentially one-dimensional. In our case, the dimensions of the search window are 25x25 pixels.

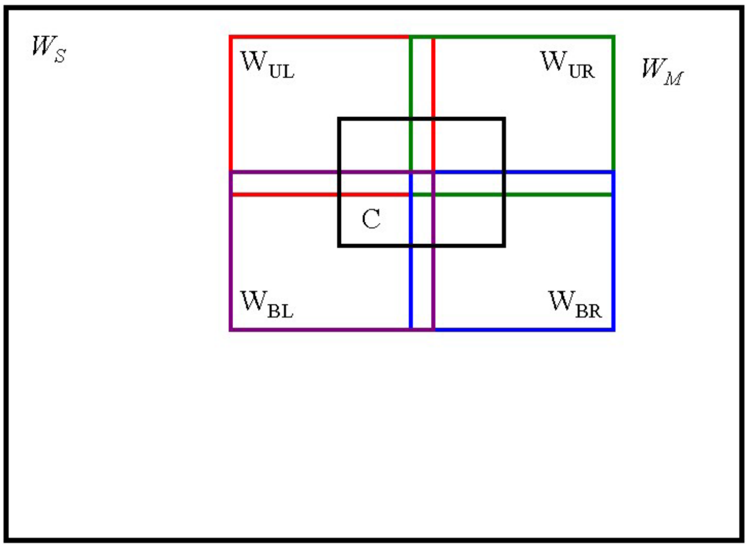


Figure 6: The configuration of overlapping windows used in the correlation method proposed by Hirschmüller [42].

At each placement of the model template W_M in the search window W_S, five correlation values Δ_C, Δ_{UL}, Δ_{UR}, Δ_{BL} and Δ_{BR}, are independently computed. These values measure the correlation of each sub-window with the corresponding part in the search window. Then, the correlation value Δ for this particular placement can be computed by adding the values of the two best surrounding correlation windows Δ_{max1} and Δ_{max2} to the middle one

$$\Delta = \Delta_C + \Delta_{max1} + \Delta_{max2} \tag{13}$$

In fact, this approach uses a small central window and supports the correlation decision by four nearby windows. This formulation enables the refinement process to cope very well with depth discontinuities and occluded/revealed regions that introduce errors when standard correlation is employed between the model window and the corresponding part of the search window. It should be noted that the cases of depth discontinuities and occlusions are very common in the particular SCR tracking scenario, where the SCRs are typically small image regions, closer to the cameras compared to their immediate surroundings.

3.5 3D reconstruction of the position of SCRs (3DR module)

The refined centroid correspondences of the SCRs are fed to a 3D reconstruction module (3DR), which computes the 3D position of each SCR. Two different reconstruction methods have been considered.

The first method [34] computes the 3D position *(X, Y, Z)* of a point *P*, given its projections m_L and m_R in the left and the right image of a stereo pair, as:

$$Z = -\frac{(m_R \times e_R)\cdot(m_R \times H_\infty m_L)}{\left\| m_R \times H_\infty m_L \right\|^2}$$

$$X = Z[A_L^{-1}(0)]m_L \tag{14}$$

$$Y = Z[A_L^{-1}(1)]m_L$$

In eqs. (14), m_L and m_R are homogeneous vectors, and e_R and H_∞ are defined as in eqs. (7) and (8), respectively. Moreover, $[A_L^{-1}(r)]$ denotes the vector that corresponds to the *r*-th row of the inverse of the intrinsic calibration parameters matrix of the left camera. Equation (14) gives the 3D position *(X, Y, Z)* of a point *P* with respect to the coordinate system of the left camera. The 3D position of this point with respect to the world-centered coordinate system can easily be computed through a rigid 3D transformation involving the position of the left camera with respect to the world-centered coordinate system.

The second method computes the 3D position of a point as the intersection of two 3D lines. More specifically, a 3D line is defined by 3D points C_L and M_L, where C_L is the optical center of the left camera and M_L is the 3D position of the centroid of an SCR on the left image. Similarly, a second 3D line is defined by 3D points C_R and M_R where C_R is the optical center of the right camera and M_R is the 3D position of the centroid of the corresponding SCR in the right image. In the case of perfect, noiseless measurements these two lines should intersect at the desired 3D point. However, noise in the corresponding image coordinates m_L and m_R as well as inaccuracies in the calibration parameters will almost certainly result in these two lines being skew. Then, the 3D location *P* of the SCR is [43]

$$P = \frac{1}{2}\left(C_L + \hat{v}_L s_L + C_R + \hat{v}_R s_R\right) \tag{15}$$

where

$$s_L = \frac{\det\left(M_R - M_L \quad \hat{v}_R \quad v_{LR}\right)}{\left|v_{LR}\right|^2} \qquad s_R = \frac{\det\left(M_R - M_L \quad \hat{v}_L \quad v_{LR}\right)}{\left|v_{LR}\right|^2} \tag{16}$$

and

$$\hat{v}_L = \frac{M_L - C_L}{\left|M_L - C_L\right|} \qquad \hat{v}_R = \frac{M_R - C_R}{\left|M_R - C_R\right|} \qquad v_{LR} = v_L \times v_R \tag{17}$$

If the 3D lines actually intersect, then *P* in eq. (15) is their point of intersection. If the 3D lines are skew, then *P* is the midpoint of the minimum-length line segment that connects the two 3D lines. Points C_L, M_L, C_R and M_R can easily be computed from the known 3D positions and orientations of the cameras with respect to the world centered coordinate system and the knowledge of the refined centroids of corresponding SCRs.

The first reconstruction method is based (a) on accurately computed epipolar geometry between the cameras of the stereo pair and (b) on the availability of point correspondences that satisfy this epipolar geometry. If the first condition is satisfied and the second is not, there exist methods [34] to refine the point matches so as to enforce that refined matches satisfy the prescribed epipolar geometry. If, however, the epipolar geometry has not been accurately computed, the 3D reconstruction is inaccurate even for perfect matches. In the context of the SCRT system, the second method provides more accurate 3D reconstruction results compared to the first reconstruction approach and, therefore, has been adopted in the 3DR module.

3.6 Temporal smoothing (TS)

The temporal smoothing (TS) module performs temporal filtering of the derived 3D position of each SCR, based on the assumption that the 3D trajectory of an SCR is smooth as a function of time. The current implementation considers 3D positions P_{t-2}, P_{t-1} and P_t of an SCR as they have been computed in the last three time instances *t-2, t-1* and *t*, and reports the 3D position *P* defined as a weighted average of these measurements

$$P = 0.6P_t + 0.3P_{t-1} + 0.1P_{t-2} \tag{17}$$

Weights are appropriately adapted whenever 3D position measurements in time instances *t-2* and/or *t-1* are not available.

4. SCRT in the context of the ActIPret framework

The SCRT system described in the previous sections has been initially implemented in standard C, as a stand-alone application that takes as input sequences of stereoscopic images. This permits extensive, off-line experimentation and testing. However, in the context of the ActIPret project, SCRT is used as a hand tracker component (HT) together with many other components in a cognitive vision framework. A schematic presentation of this framework, which shows the interaction of HT with the rest of the components, is shown in Fig. 7.

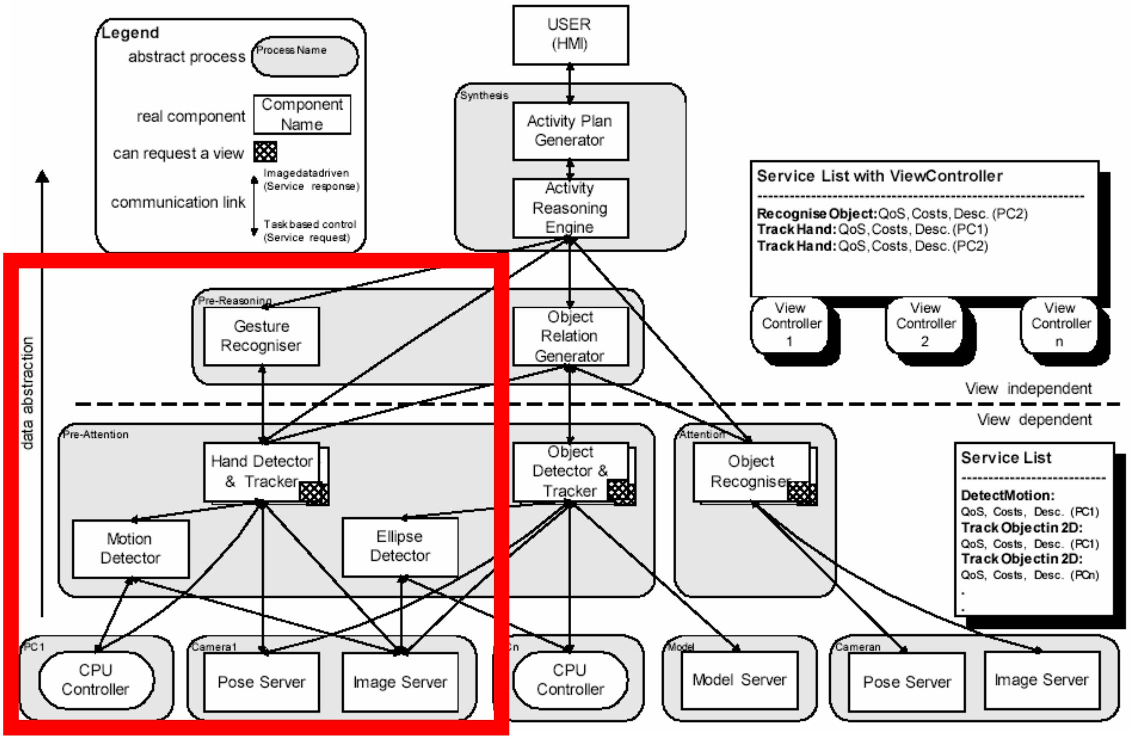


Figure 7: The HT component within the ActIPret framework. The HT component interacts with the Image Server and Pose Server components in order to acquire the required image, camera geometry and camera calibration data. Moreover, it has to communicate with the Gesture Recognizer component to deliver the necessary information regarding the position of the tracked hands.

As it is shown in Fig. 7, the HT component needs to communicate with the Image Server and Pose Server components in order to acquire the required image, camera geometry and camera intrinsic calibration data. Moreover, it has to communicate with the Gesture Recognizer component to provide the necessary information regarding the position of the tracked hands. The activation of the components is accomplished in a top-to-bottom manner. More specifically, the Gesture Recognizer activates the HT component, asking for possible hand hypotheses within a 3D space of interest (SOI). Then, the HT component activates the Image Server and Pose Server components requesting image stereo pairs, camera position information and camera calibration data. As soon as Image Server and Pose Server respond to the HT component with the requested data, detection and tracking of hand hypotheses is initiated and the HT component reports the 3D positions of the detected hand hypotheses to the Gesture Recognizer.

To support the necessary interaction of the HT component with the rest of the components in the ActIPret framework, the original, framework-independent, core HT functionality has been extended with a framework-dependent layer, which supports the necessary communication structures. The interaction of the HT component with the rest of the components in the ActIPret framework has been verified in several project integration meetings.

5. Sample Results

In this section, representative results are provided from the application of the SCRT system to sequences of images. All experiments reported here as well as several others that are not included due to space limitations, were conducted by employing the same set of parameters required by the different SCRT modules.

5.1 2D tracking

In this section, representative results related to the 2D version of the proposed tracker are provided. The reported experiment consists of a long (3825 frames) sequence that has been acquired and processed on-line and in real-time on a Pentium 4 laptop computer running MS Windows at 2.56 GHz. A web camera with an IEEE 1394 (FireWire) interface has been used for this experiment.

For the reported experiment, the initial, "seed" training set contained 20 images and was later refined in a semi-automatic manner using 80 additional images. The training set contains images of four different persons that have been acquired under various lighting conditions.

Figure 8 provides a few characteristic snapshots of the experiment. A demo video illustrating this experiment can be found at http://www.ics.forth.gr/cvrl/demos. For visualization purposes, skin-colored pixels appear in white. Moreover, the ellipse of each tracked object hypothesis is shown. Different colors correspond to different object hypotheses.

In the beginning of the experiment, the camera is still and the tracker correctly asserts that there are no skin-colored objects in the scene (Fig. 8(a)). Later, the hand of a person enters the field of view of the camera and starts moving at various depths, directions and speeds in front of it. At some point in time, the camera also starts moving in a very jerky way; the camera is mounted on the laptop's monitor, which is being moved back and forth. The person's second hand enters the field of view; hands now move in overlapping trajectories. Then, the person's face enters the field of view. Hands disappear and then reappear in the scene. All three objects move independently in disjoint trajectories and in varying speeds ((b)-(d)), ranging from slow to fast; at some point in time the person starts dancing, jumping and moving his hands very fast like an orchestra conductor. The experiment proceeds with hands moving in crossing trajectories. Initially hands cross each other slowly and then very fast ((e)-(g)). Later on, the person starts applauding which results in his hands touching but not crossing each other ((h)-(j)). Right after, the person starts crossing his hands like tying in knots ((k)-(o)). Next, the hands cross each other and stay like this for a considerable amount of time; then the person starts moving, still keeping his hands crossed ((p)-(r)). Then, the person waves and crosses his hands in front of his face ((s)-(u)). The experiments concludes with the person turning the light on and off ((v)-(x)), while greeting towards the camera (Fig. 8(x)).

As it can be verified from the snapshots and the accompanying video, the labelling of the object hypotheses is consistent throughout the whole sequence, which indicates that they are correctly tracked. Thus, the proposed tracker performs very well in all the above cases, some of which are challenging. Note also that no images of the person depicted in this experiment were contained in the training set. Several tests have been carried out aiming at assessing the computational performance of the developed tracker. As an indicative example, the 3825 frames sequence presented previously has been acquired and processed at an average frame rate of 28.45 fps (320x240 images).

5.2 3D Tracking

Experiments with two different stereoscopic sequences are reported here, each of which has been acquired by a different stereoscopic system. More specifically, the first sequence ("Sequence1_head") has been acquired by the stereo head shown in Fig. 1, while the second sequence ("Sequence2_arm") has been captured by a different set of cameras mounted on a robotic arm. Despite the fact that in both experiments there are four different cameras with different color response characteristics, a single training set has been established and skin color detection (SCD) has been based on the same set of probabilities derived through eq. (1), for the images of all four cameras. The initial, "seed" training set contained 40 images (10 from each camera) and was later refined in a semi-automatic way, using 160 additional images (40 from each camera).

The "Sequence1_head" sequence shows a human operator while manipulating a CD player (the operator opens the tray, picks-up a CD, places it in the tray and closes the tray). The stereoscopic system does not move in this experiment. The full sequence consists of 146 left and 146 right frames. Figure 9 (top to bottom, left to right) shows characteristic snapshots from the obtained tracking results. Every 10th frame is shown in this figure. For visualization purposes, each SCR appears as a color blob superimposed on the right image of the stereo pair. A cross marks the centroid of each SCR. Moreover, an ellipse derived from the statistics of each SCR is shown around each color blob. It can be seen that the system identifies three SCRs, namely (1) the head of the human operator, (2) the skin-colored arm of the armchair and (3) the hand of the human operator. It can also be verified that the labeling of the SCRs is consistent throughout the whole sequence, which means that SCRs are correctly tracked both in time and between the images of the stereo pair.

Figure 10, shows the 3D trajectories computed by the system for the three tracked SCRs. The upper facet of the CD player has also been reconstructed to serve as reference. The trajectory of the operator's hand appears qualitatively correct. In this particular sequence, the arm of the armchair does not move, while the head of the operator moves slightly. To measure the stability of the derived 3D coordinates, the 3D bounding box of all estimated 3D positions for each SCR has been computed. For the case of the static arm of the armchair, the dimensions of this bounding box are 1.2cm x 1.5cm x 1.6cm, which is close to zero, as expected.

In the "Sequence2_arm" sequence a human operator again manipulates a CD player. In this sequence cameras move in time. The sequence consists of 134 left and 134 right frames. Figure 11 (top to bottom, left to right) shows characteristic snapshots from the obtained tracking results. Every 10th frame is again shown in this figure. It can be seen that the system identifies two SCRs corresponding to (1) the head of the human operator and (2) the hand of the human operator. It can also be verified that the labeling of the SCRs is consistent throughout the whole experiment, which implies that SCRs are correctly tracked both in time and between the images of the stereo pair. Figure 12 shows the 3D trajectory computed by the system for the hand of the operator. Note that despite the motion of the cameras, the hand trajectory appears smooth, which serves as an indication that only small range errors are introduced due to SCD detection, centroid estimation and camera motion estimation processes.

Videos related to the above experimental results can be retrieved at http://www.ics.forth.gr/cvrl/demos.html

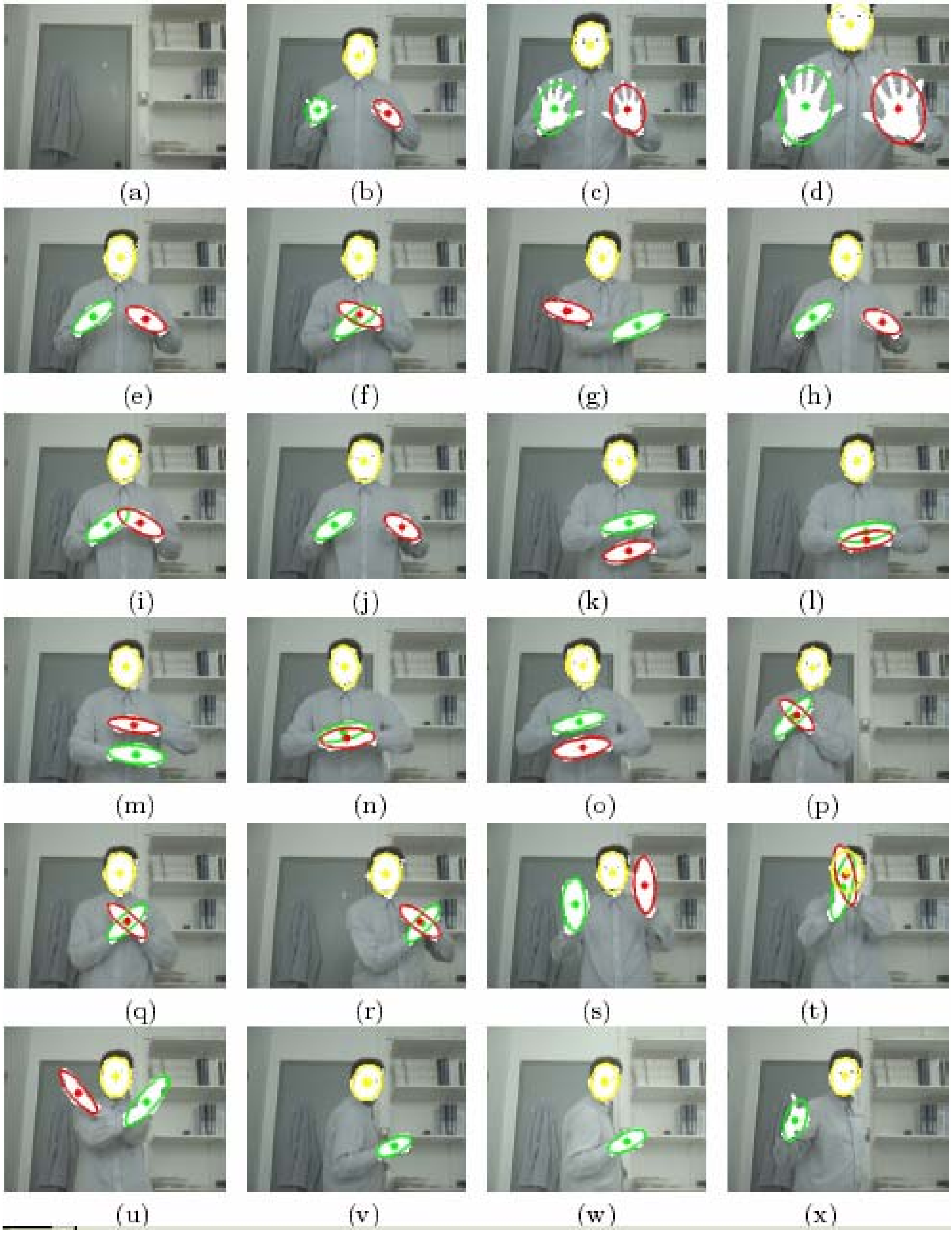


Figure 8: Characteristic snapshots of the on-line 2D tracking experiment

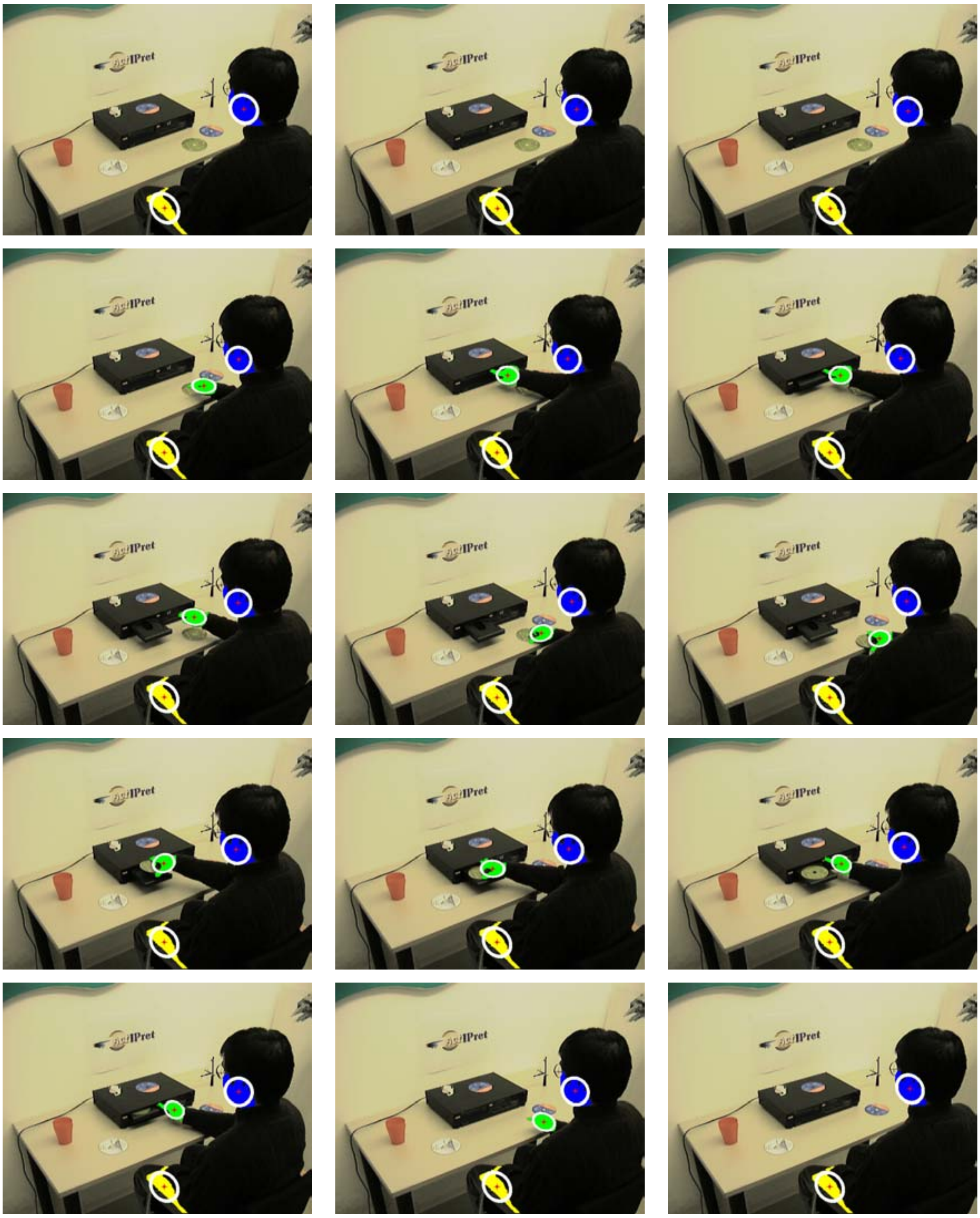


Figure 9: Tracking results for the “Sequence1_head” sequence. Each SCR appears as a color blob superimposed on the right image of the stereo pair.

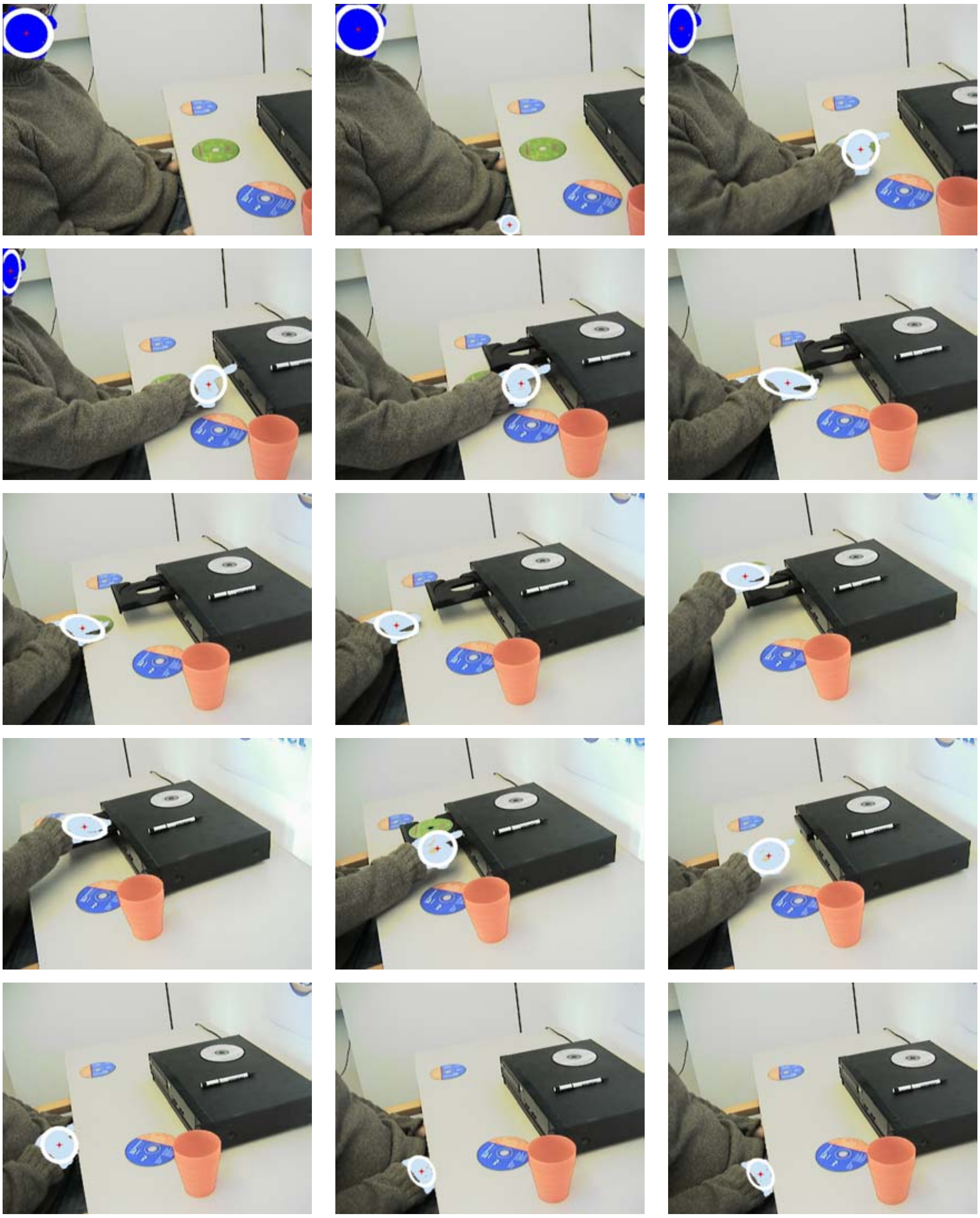


Figure 10: Tracking results for the "Sequence2_arm" sequence. Each SCR appears as a color blob superimposed on the right image of the stereo pair.

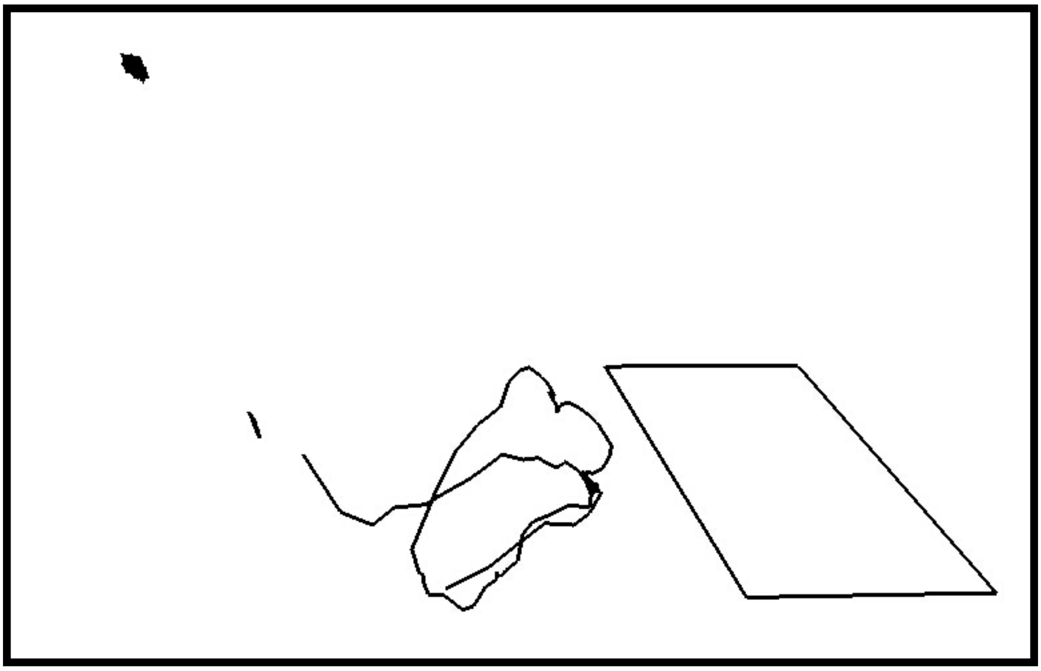


Figure 11: The 3D trajectories of the SCRs tracked in the experiment of Fig. 9. The top-left and middle-left isolated spots correspond to the motion of the operator's head and of the armchair's arm, respectively. The trajectory in the center of the image corresponds to the hand trajectory. The upper facet of the CD player has also been reconstructed, to serve as a reference.

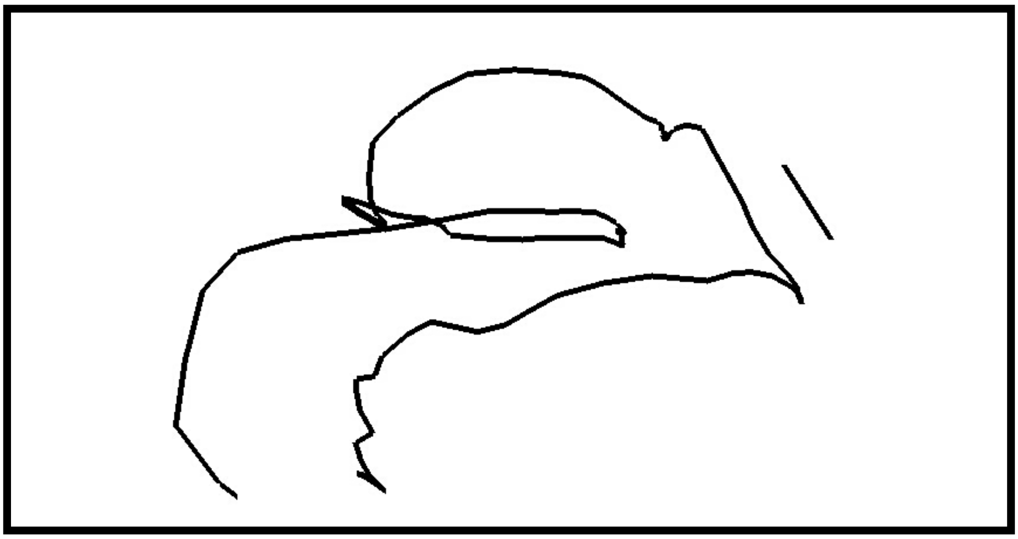


Figure 12: The 3D trajectory of the hand detected in the experiment of Fig. 10. The straight-line segment appearing in the right part of the figure corresponds to the tray of the CD player.

6. SCRT performance considerations

Several tests have been carried out aiming at assessing the functionality and the performance of a prototype implementation of the SCRT system. Both off-line experiments (involving pre-recorded image sequences) and on-line experiments have been conducted. It turns out that one-cycle of operations of the SCRT system takes approximately 75 milliseconds to process 640x480 images on an Intel P4@1.8 GHz running Linux. The cycle includes all SCRT system functionality plus reading a stereo pair of 640x480 images from the hard disk. Approximately 40% of the cycle time is spent on image I/O, 40% on SCD and the rest 20% on the remaining modules. The SCRT system may be modified to operate on sub-sampled versions of the original images. More precisely, if the input images are sub-sampled by a factor of two (i.e. 320x240 images are employed), then the SCRT cycle time drops to 35 milliseconds. The reason why the performance gain is not directly proportional to the input data reduction (i.e. a factor of four) is that SCRT always imports full-resolution images, which are then subsampled appropriately. Therefore, acquisition time is constant and independent of the operational image resolution. As an illustration of this claim, image I/O and sub-sampling were measured to account for almost 80% of the cycle time in the case of half image resolution.

An important observation is that the SCR trajectories computed in full image resolution are in close resemblance with the SCR trajectories computed at half resolution. To illustrate this, the SCRT system has been applied to the "Sequence1_head" sequence, both at full and half resolution and the average distance between the reconstructed hand 3D positions in these two cases was computed. This distance was in the order of 6 millimeters, thus demonstrating that a significant speedup can be obtained without sacrificing much of accuracy. Still, a rate of 13 Hz (in the case of full resolution images) or 28 Hz (in the case of half resolution images) is considered sufficient for the purposes of SCRT.

7. Extensions under consideration

3D tracking of multiple SCRs in scenes observed by a moving stereoscopic system is a difficult research problem in cases where a robust performance under general conditions is required. The HT component that has been developed within the ActIPret project has several attractive features. Nevertheless, there are still important improvements that can be made and certain research and development activities at FORTH aim towards realizing such improvements. As an example, currently, each hand hypothesis is represented as a point in 3D space. However, in the ActIPret cognitive vision framework, it would be also desirable to provide information regarding the 3D pose of each hand hypothesis as well as information regarding the 3D positions of fingertips. Towards this goal, an alternative model of a hand has being investigated which enables tracking of the fingertips of a hand. This provides, in turn, a coarse approximation of the shape of the hand and evidence on the hand pose.

Another observation that may lead to improvements is that in the current version of the HT component, the camera positions and orientations are provided by the Pose Server component, which relies on information derived from the encoders of the stereoscopic system. This information is not accurate enough, especially in the case of moving cameras. Research at FORTH has resulted in a camera-tracking system that employs point correspondences to robustly compute the 3D position and orientation of a moving camera. This novel method [4, 5] is briefly described in Appendix A and is expected, when employed, to improve substantially the camera position and orientation estimation processes, which will in turn, improve the accuracy in estimating the 3D position of hand hypotheses.

8. Discussion

In this document, the SCRT system has been described. SCRT is capable of detecting and tracking multiple skin-colored regions in scenes viewed by a moving stereoscopic system in which each camera can be moving independently. The computational performance SCRT system is near real-time when operating in full resolution 640x480 images and can be considerably improved by sub-sampling the input images by a factor of two. In this case, the HT component operates in real time, without a noticeable degradation of the quality of the computed 3D trajectories.

Currently, SCRs correspond to any skin-colored region in the employed images; moreover each SCR is represented as a point in 3D space. However, in many cases, it is desirable to focus attention on the activities of human hands and it would also be desirable to provide information regarding the 3D pose of each hand hypothesis and the 3D positions of fingertips. Another axis of research aims at providing specialized hand models that will turn the SCRT system into a 3D hand tracker.

Last but not least, current research and development activities are targeted towards integrating vision-based camera tracking techniques [44] with the SCRT system. It is expected that vision-based camera tracking will improve substantially the camera position and orientation estimation processes, which will in turn, improve the accuracy in estimating the 3D position of SCRs. Moreover, this will alleviate the current dependence of SCRT on the specialized equipment needed to continuously monitor the position and orientation of the two cameras of the stereo-rig.

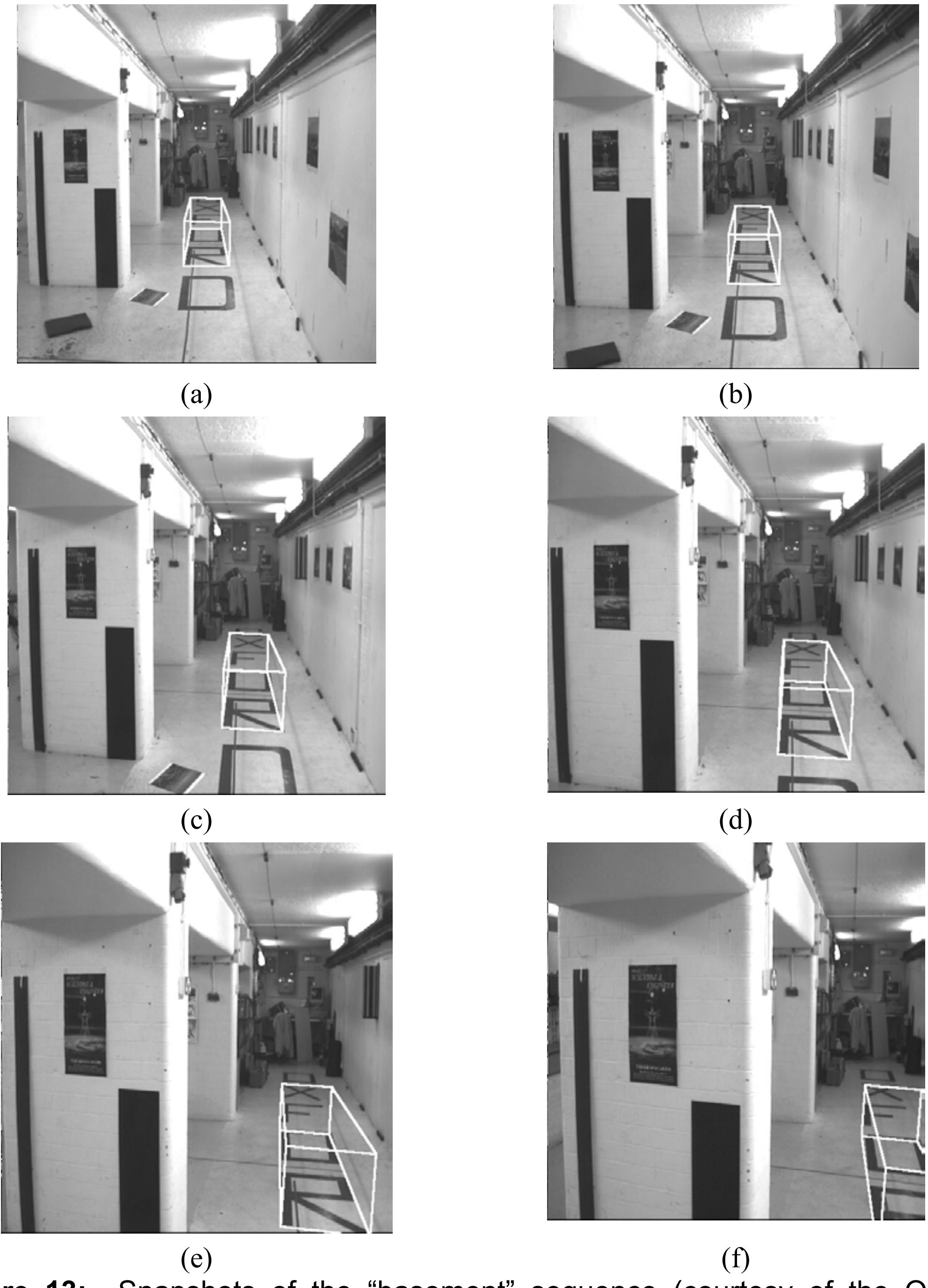
9. APPENDIX A: Image-based camera tracking

We address the problem of camera matchmoving over a sequence of images acquired by a freely moving observer, a task that has a broad spectrum of useful applications in domains such as augmented reality and creation of special effects. Camera matchmoving is an application involving synthesis of real scenes and artificial objects, in which the goal is to insert computer-generated graphical 3D objects into live-action footage depicting unmodeled, arbitrary scenes. Graphical objects should be inserted in a way so that they appear to move as if they were a part of the real scene. Seamless, convincing insertion of graphical objects calls for accurate 3D camera motion tracking (i.e. pose estimation), stable enough over extended sequences so as to avoid the problems of jitter and drift in the location and appearance of objects with respect to the real scene. Additionally, the placement of the objects with respect to the real scene often requires the extraction of limited 3D geometry information; for instance, accurate 3D reconstruction of a few guiding control points is in many cases sufficient. Matchmoving finds several important applications in augmented reality as well as the creation of special effects in the post-production industry. To provide the versatility required by such applications, very demanding camera tracking requirements, both in terms of accuracy and speed, are imposed.

At the core of the proposed approach lies a novel, feature-based 3D plane tracking technique. Given a triplet of consecutive images and a plane homography between the first two of them, the plane tracker is capable of estimating the homography induced by the same plane between the second and third images, without requiring the plane to be segmented from the rest of the scene. In other words, the proposed method operates by “chaining” (i.e. propagating) across frames the image-to-image homographies due to some 3D plane. The chaining operation represents projective space using a ``plane + parallax'' decomposition, which permits the combination of constraints arising from all available point matches, regardless of whether they actually lie on the tracked 3D plane or not. Being straightforward to extend over long image sequences, plane tracking permits the estimation for each image pair in the sequence of the homographies induced by the 3D plane. Knowledge of such homographies allows the corresponding projection matrices encoding camera motion to be expressed in a common projective frame and therefore to be recovered directly. Additional knowledge of intrinsic camera calibration can be used to upgrade projective reconstruction to a Euclidean one. In addition to camera motion, the proposed method can recover a rough representation of 3D structure. Finally, it is shown that the tracked plane can be a virtual one, thus raising the implicit assumption regarding the presence of at least one 3D plane in the viewed scene.

Figures 13 and 14 provide representative results from a conducted experiment that involves augmenting an image sequence with an artificial 3D object. The point features employed in this experiment have been extracted and matched automatically. The experiment was performed on the well-known Oxford “basement” image sequence. This sequence consists of 11 frames acquired by a camera mounted on a mobile robot as it approached the scene while turning left. The proposed method was applied to the “basement” sequence and the camera 3D locations for each image along with a 3D point model of the scene were recovered. Using a few of the recovered 3D points, a wire-frame rectangular parallelepiped was inserted into the scene. Specifically, aiming to give the impression of an object lying on the floor, the parallelepiped was inserted so that its bottom face extends from the top of the second “O” to the bottom of the letter “R” in the word “OXFORD”. In a real application, a more complex 3D model would have been inserted into the scene with the aid of a 3D graphics package. Figs. 13(a)-(f) are snapshots of the sequence resulting by augmenting the original one. A top view of the VRML 3D model that was recovered, showing also the location of the inserted parallelepiped as well as the camera locations and trajectory is illustrated in Fig. 14. As it is clear from the results, the accuracy of camera matchmoving using the proposed method is satisfactory.

The average running time of the proposed matchmoving method for each image triplet was 102 ms on an Intel P4@1.8 GHz running Linux. This time does not include the time required for matching points among frames; around 350 points were matched between every pair of successive frames.



(a) (b)

(c) (d)

(e) (f)

Figure 13: Snapshots of the “basement” sequence (courtesy of the Oxford Visual Geometry Group), corresponding to frames 0, 2, 4, 6, 8 and 10, resulting after augmenting the original sequence with a rectangular parallelepiped drawn in red.

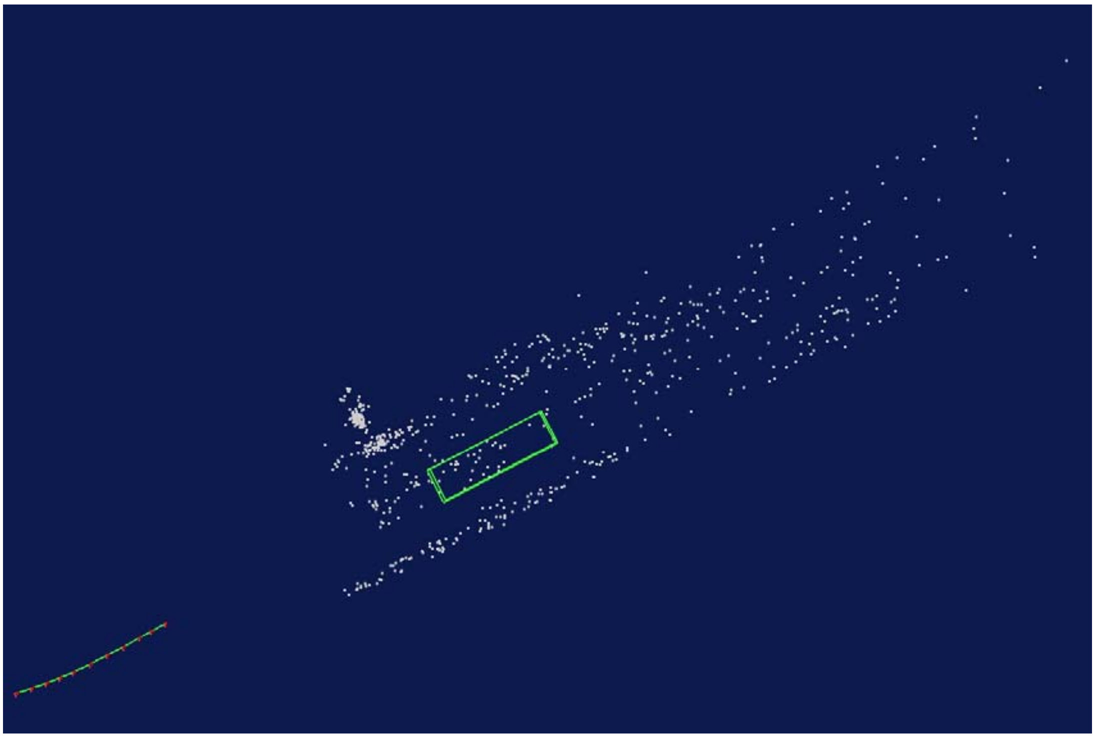


Figure 14: Top view of the VRML 3D reconstruction for the scene of Fig. 13, showing also the inserted object and the 3D camera locations; see text for explanation.

10. List of abbreviations

ABBREVIATION	**FULL NAME**
AS	Association of Hand Hypotheses between the images of a stereo-pair.
AT	Association of Hand Hypotheses in Time
CM	Centroid Matching
f.o.v	Field of view
HH	Hand Hypothesis
HT	Hand Tracker
SCD	Skin Color Detection
SOI	Space of Interest
TS	Temporal Smoothing
3DR	3D Reconstruction

11. References

[1] K. Meyer, H.L. Applewhite, F.A. Biocca, “A Survey of Position Trackers”, PRESENCE, special issue on Teleoperator and Virtual Environments, Vol. 1, No. 2, (MIT Press 1992), pp. 173-200.

[2] D.M. Gavrila, “The Visual Analysis of Human Movement: A Survey”, Computer Vision and Image Understanding, Academic Press, vol.73, no.1, (1999) pp. 82-98.

[3] Q. Delamarre, O. Faugeras, “3D Articulated Models and Multi-View Tracking with Physical Forces”, CVIU journal, vol. 81, (2001) pp. 328-357.

[4] T.S. Jebara and A. Pentland, “Parameterized Structure from Motion for 3D Adaptive Feedback Tracking of Faces”, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, (1997) pp. 144-150.

[5] S.H. Kim, N.K. Kim, S.C. Ahn and H.G. Kim, “Object oriented Face Detection Using Range and Color Information”, Proc. 3rd IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, (1998) pp. 76-81.

[6] M.J. Jones and J.M. Rehg, “Statistical Color Models with Application to Skin Detection,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, (1999) pp. 274-280.

[7] D. Saxe and R. Foulds, “Toward Robust Skin Identification in Video Images”, 2nd International Conference on Automatic Face and Gesture Recognition, Vermont, USA, (1996) pp. 379-384.

[8] D. Chai and K.N. Ngan, “Locating Facial Region of a Head-and-shoulders Color Image”, Proc. 3rd IEEE International Conf. on Automatic Face and Gesture Recognition, Nara, Japan, (1998) pp. 124-129.

[9] M.H. Yang and N. Ahuja “Detecting Human Faces in Color Images”, Proc. IEEE International Conference on Image Processing, Chicago, Illinois, USA, 1, (1998) pp. 127-130.

[10] J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu, “Comparative Performance of Different Skin Chrominance Models and Chrominance Spaces for the Automatic Detection of Human Faces in Color Images”, Proc. IEEE International Conference on Face and Gesture Recognition, (2000) pp. 54-61.

[11] J. Cai and A. Goshtasby A, “Detecting Human faces in Color Images”, Image and Vision Computing 18(1): (1999) pp. 63-75.

[12] S. McKenna, Y. Raja and S. Gong, “Tracking Color Objects Using Adaptive Mixture Models”, Image and Vision Computing 17(3-4): (1999) pp. 225-231.

[13] Y. Raja, S. McKenna and G. Gong, “Tracking and Segmenting People in Varying Lighting Conditions Using Color”, Proc. 3rd IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, pp. 228-233.

[14] T.S. Jebara and A. Pentland, “Parametrized Structure from Motion for 3D Adaptive Feedback Tracking of Faces”, Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, (1997) pp. 144-150.

[15] T.S. Jebara, K. Russel and A. Pentland, “Mixture of Eigenfeatures for Real-time Structure from Texture”, Proc. 6th International Conf. on Computer Vision, Bombay, India, (1998) pp. 128-135.

[16] M.H. Yang and N. Ahuja, “Face Detection and Gesture Recognition for Human-computer Interaction”, Kluwer Academic Publishers, New York, (2001).

[17] M.H. Yang, D.J. Kriegman and N. Ahuja, “Detecting Faces in Images: A Survey”, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 24, no.1, (2002) pp. 34-58.

[18] R.E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”, Transactons of the ACME-Journal of Basic Engineering, (1960) pp. 35-45.

[19] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time Tracking of Non-rigid Objects Using Mean Shift”, proc. IEEE Conference on Computer Vision and Pattern Recognition, (2000) pp. 142–151.

[20] O. Javed and M. Shah. “Tracking and Object Classification for Automated Surveillance”, In European Conference on Computer Vision, (2002) pp. 343–357.

[21] N.T. Siebel and S. Maybank”, Fusion of Multiple Tracking Algorithms for Robust People Tracking”, In European Conference on Computer Vision, (2002) pp. 373–387.

[22] M. Spengler and B. Schiele, “Towards Robust Multi-cue Integration for Visual Tracking”, In International Workshop on Computer Vision Systems, (2001) pp. 94–107.

[23] C. Stauffer and W. Grimson, “Adaptive Background Mixture Models for Real-time tracking”, In Proceedings IEEE Conf. On Computer Vision and Pattern Recognition, (1999) pp. 246–252.

[24] J. Triesch and C. von der Malsburg, “Democratic Integration: Self-organized Integration of Adaptive Cues”, Neural Computation, 13(9): (2001) pp. 2049–2074.

[25] R. Fablet and M. J. Black, “Automatic Detection and Tracking of Human Motion with a View-based Representation”, In European Conference on Computer Vision, (2002) pp. 476–491.

[26] M. Isard and A. Blake, “ICONDENSATION: Unifying Low-level and High-level Tracking in a Stochastic Framework”, Lecture Notes in Computer Science, 1406, (Springer, 1998) pp. 893-908.

[27] J. Vermaak, P. Perez, M. Gangnet, and A. Blake, “Towards Improved Observation Models for Visual Tracking: Selective Adaptation”, proc. European Conference on Computer Vision, (2002) pp. 645–660.

[28] C. Hue, J.-P. Le Cadre, and P. Pérez. “Sequential Monte Carlo Methods for Multiple Target Tracking and Data Fusion”, IEEE Trans. on Signal Processing, 50(2): (2002) pp. 309–325.

[29] M. Isard and J. MacCormick, “BraMBLe: A Bayesian Multiple-blob Tracker”, proc. International Conference on Computer Vision ICCV, (2001).

[30] E. Koller-Meier and F. Ade, “Tracking Multiple Objects Using the Condensation Algorithm”, Journal of Robotics and Autonomous Systems, 34:93–105, 2-3, (2001).

[31] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, “Color-based Probabilistic Tracking”, proc. European Conference on Computer Vision, (2002) pp. 661–675.

[32] Y. Li, A. Hilton, J. Illingworth, “A relaxation Algorithm for Real-time Multiple View 3D-tracking”, IVC(20), No. 12, (2002) pp. 841-859.

[33] T. Inaguma, K. Oomura, H. Saji and H. Nakatani, “Efficient Search Technique for Hand Gesture Tracking in Three Dimensions”, Spring-Verlag Berlin Heidelberg, (2000) pp. 594-601.

[34] R. Hartley and P. Sturm, “Triangulation”, CVIU, 68(2): (1997) pp. 146–157.

[35] O. Faugeras, Q.-T. Luong, and T. Papadopoulo, “The Geometry of Multiple Images”. MIT Press, (2001).

[36] S.O. Orphanoudakis, A.A. Argyros, M. Vincze “Towards a Cognitive Vision Methodology: Understanding and Interpreting Activities of Experts”, ERCIM News, No 53, Special Issue on “Cognitive Systems, April 2003. See also http://actipret.infa.tuwien.ac.at

[37] K. Jack, “Video Demystified: A Handbook for the Digital Engineer”, HighText Publications Inc. (1993).

[38] DA Forsyth and J Ponce, “Computer Vision: A Modern Approach”, Prentice Hall (2003).

[39] J.F. Canny, “A Computational Approach to Edge Detection”, IEEE Trans. on PAMI, 8(11): (1986) pp. 769-798.

[40] L.Robert, C. Zeller, O.D. Faugeras, M. Hebert, “Applications of Non-Metric Vision to Some Visually-Guided Robotic Tasks”, in Y. Aloimonos (ed.), Visual Navigation: From Biological Systems to Unmanned Ground Vehicles, ch.5, Lawrence Erlbaum Associates, (1997) pp.89-134.

[41] Z. Zhang, "Determining the Epipolar Geometry and its Uncertainty: A Review", IJCV, 27(2): (1998) pp. 161-195.

[42] Heiko Hirschmüller (2001), "Improvements in Real-Time Correlation-Based Stereo Vision", IEEE Workshop on Stereo and Multi-Baseline Vision at IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA, (2001) pp. 141-148.

[43] R. Goldman, “Intersection of Two Lines in Three-Space”, In Graphics Gems I (Ed. A. S. Glassner), San Diego, Academic Press, (1990) pp. 304.

[44] M.I.A. Lourakis, A.A. Argyros, “Efficient 3D Camera Matchmoving Using Markerless, Segmentation-Free Plane Tracking”, Technical Report, ICS/FORTH TR 324, Institute of Computer Science, FORTH, Sep. 2003.