Detection and Tracking of Faces in Real Environments

R. Herpers[1,2], G. Verghese[1], K. Derpanis[1], R. McCready[1], J. MacLean[1],
A. Levin[1], D. Topalovic[1], L. Wood[1], A. Jepson[1], J. K. Tsotsos[1]

[1] Department of Computer Science, University of Toronto
6 King's College Road, Toronto, Ontario M5S 3G4, Canada
[2] Department of Applied Computer Science, University of Applied Sciences
Rathausallee 10, 53757 Sankt Augustin, Germany

Abstract

*A **S**tereo **A**ctive **V**ision **I**nterface is introduced which detects frontal faces in real world environments and performs particular active control tasks dependent on changes in the visual field. Firstly, connected skin colour regions in the visual scene are detected by applying a radial scanline algorithm. Secondly, facial features are searched for in the most salient skin colour region while the blob is tracked by the camera system. The facial features are evaluated and, based on the obtained results and the current state of the system, particular actions are performed. The SAVI system is thought of as a smart user interface for teleconferencing, telemedicine, and distance learning. The system is designed as a Perception-Action-Cycle (PAC), processing sensory data of different kinds and qualities. Both the vision module and the head motion control module work at frame rate. Hence, the system is able to react instantaneously to changing conditions in the visual scene.*

1 Introduction

Recent increased demands on teleconferencing systems have been caused by the need to communicate quickly and efficiently over large distances. Currently available teleconferencing systems are mostly restricted to the transmission of video and audio data based on standard hardware and software solutions. There are no advanced or sophisticated features available that provide active control functions. But in most real world vision applications, dynamic scenes require fast and appropriate reactions to changing conditions. For that reason, a system is designed following the paradigm of a *Perception-Action-Cycle* (PAC), in which different sensory data are processed within a closed loop, while the system is reacting instantaneously to changing conditions.

SAVI is based on a robotically controlled binocular head called TRISH-2 (fig. 1), a new prototype of the stereo vision head TRISH-1 [10]. TRISH-2 consists of two 24-bit colour CCD-cameras with 4 optical degrees of freedom (zoom, focus, iris, exposure/shutter for each camera). In addition, there are 4 mechanical degrees of freedom (common tilt and rotation as well as independent pan) which have to be handled. The design of the head motion controller is based predominantly on research done on the preceding robot system PLAYBOT[12]. In general, both cameras can be controlled independently or in master/slave mode. In master/slave mode one camera, in the following called the attentive camera, attends, tracks, or zooms in on an object of interest while the other camera may provide an overview of the visual scene and/or stereo information.

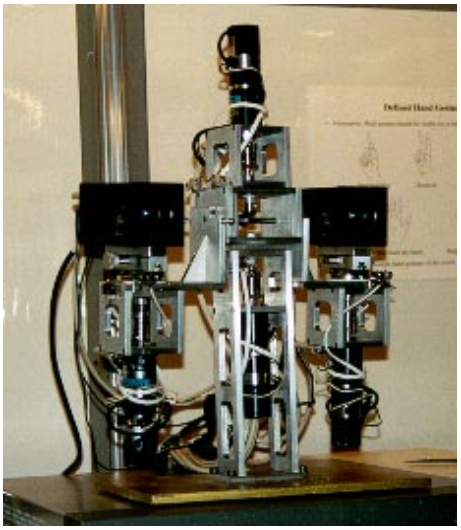


Figure 1. TRISH-2, the binocular head of SAVI.

The presented work is related mostly to the work done in the area of intelligent environments [15, 9], multimodal user interfaces, and advanced multi-media systems [3, 11]. Most of these systems do not include any active control. They are based on a fixed camera setup with a wide angle view and do not provide any possibility of changing their internal settings with respect to the above mentioned degrees of freedom.

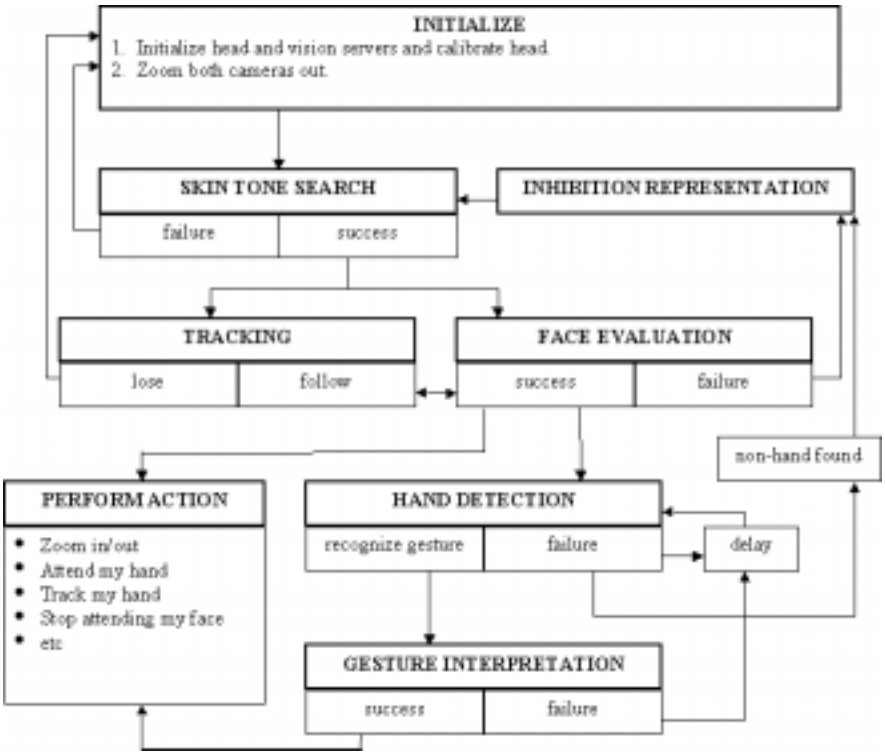


Figure 2. Control flow of SAVI's supervisor.

It has to be mentioned that the SAVI system is still under development. This contribution will therefore concentrate on a detailed presentation of the general system design (section 2), the skin tone search (section 3), the face detection (section 4), and the tracking (section 5). Some details concerning hardware specifications and conclusions are given at the end of this contribution.

2 System design

SAVI's control system is realized by a so-called *supervisor* which is defined by a finite set of states and related transitions (fig. 2). To be complete, the entire system design of SAVI, including the hand gesture detection modules, will be introduced here. However, a detailed description and discussion of the hand gesture detection module is beyond the scope of this contribution. This contribution will therefore concentrate on a comprehensive discussion of the skin tone search, the face evaluation, and the tracking of the moving object of interest.

Skin Tone Search During *Skin Tone Search*, the visual input of both cameras is evaluated to find the most salient skin tone blob (fig. 3 and fig. 4). For this task both cameras are zoomed out. The size, shape, and some other simple features of the connected skin tone blobs are verified and regions a priori known as uninteresting are excluded. An ordered list of the remaining skin tone blobs is computed or updated based on previous evaluation steps. Subsequently, the most salient skin tone region is selected and both cameras are focussed on it.

Tracking During *Tracking*, the most salient skin tone blob is kept fixated and actively followed (tracked) with both cameras. When the object stops moving or only little motion is present, SAVI begins evaluating the skin tone blob.

Face Evaluation During *Face Evaluation*, facial features are searched for in the skin tone blob (fig. 8). If successful, the system may perform particular actions or may search for a hand gesture. Otherwise, the rejected blob is suspended from further processing by applying an inhibition function and *Skin Tone Search* is called to find the next most salient skin tone region.

Hand Detection During *Hand Detection*, a skin tone blob to the right of the attended face is evaluated for an initial hand gesture. The attentive camera is zoomed in and the detected face-hand area remains fixated upon. If successful, the skin blob area is passed to the 'Gesture Interpretation' module. Otherwise, an inhibition representation is computed.

Gesture Interpretation During *Gesture Interpretation*, the right hand is fixated upon and tracked with the leading camera, while the other camera is zoomed out and keeps the corresponding face in the visual field. When the hand stops moving, SAVI evaluates the hand blob's shape for identifiable gestures. If successful, a corresponding action is per-

formed. Otherwise, the *'hand detection'* module is called to find the initial hand gesture again.

During the operation of the supervisor, there are several possible failures that may occur (e.g. losing track of the current blob, or a head controller or vision server failure), which require particular error management. All of these control tasks are intergated within the supervisor as well.

3 Visual search for skin colour blobs

To reduce the search space, those areas which are known to be unsuitable for face detection should be excluded in advance. Therefore, the computation begins by searching for meaningful connected skin tone areas which satisfy particular model constraints. The coherent skin tone regions will be considered in more detail during subsequent processing steps to confirm the presence of a face.

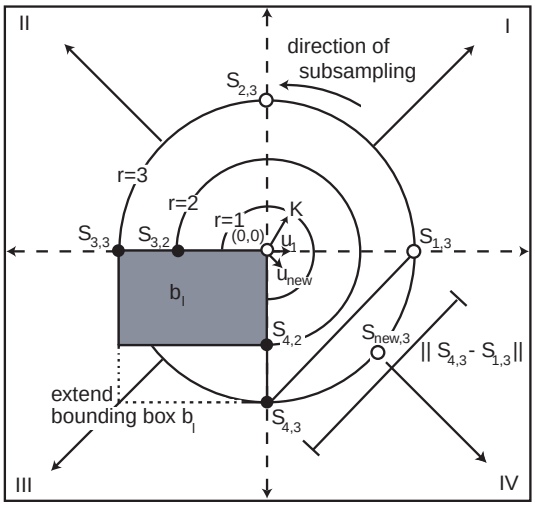


Figure 3. Method of the radial scanline algorithm. In quadrant I and II the notation is presented. u_1 refers to the unit vector of scanline s_1, $S_{1,3}$ refers to the scanline position of scanline s_1 at circle number $r = 3$ and K denotes the step width. In quadrant III the extension of an existing bounding box is explained by inclusion of scanline point $S_{4,3}$ to the skin colour blob b_l and in quadrant IV the insertion of a new scanline s_{new} starting at position $S_{new,3}$ between $S_{4,3}$ and $S_{1,3}$ is shown; the associated unit vector is u_{new}.

The computation of the skin colour detection module is based on HSV colour values derived from the 24-bit RGB video signals. The use of the HSV colour space is advantageous because skin colours of different races and with different brightness values form a connected volume in the HSV colour space [8].

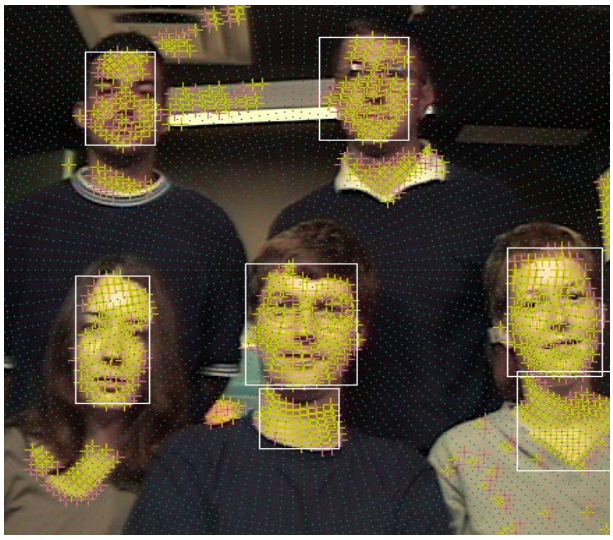


Figure 4. Computation result of the radial scanline algorithm. Detected skin colour pixels are marked by yellow and pink plus signs dependent on two different thresholds. Connected skin colour regions marked with a white bounding box have already passed consistency checks and may be processed further. All "yellow" and "pink" pixels not included in bounding boxes are assigned to be noise or of lower interest.

For efficient real-time skin colour blob detection, a radial scanline detection method has been developed. Starting at the center of a particular rectangular image region (region of interest, ROI), it scans radially outward along approximately concentric circles with a particular step width K (fig. 3 and fig. 4). A scanline s is defined by its unit vector u and the circle number r, which is multiplied by the constant factor K mentioned previously. The unit vector u is defined by its origin $O = (0, 0)$ and a point $P_{unit} = (x, y)$ on the unit circle, with $\|u\| = 1$. The inital set of scanlines is given by an ordered set (counter clockwise) of unit vectors $U := \{u_1, \ldots, u_h\}$ with $h := 2^a, a \geq 2$. The head position $S_{u_i,r}(x, y)$ or for simplicity $S_{i,r}(x, y)$ of a particular scanline s_i is calculated by:

$$S_{i,r}(x, y) = u_i(x, y) \times r \times K \tag{1}$$

Figure 3 shows the main principles of the scanline algorithm. Fundamental to the radial scanline algorithm is the construction and continuous update of a number of bounding box representations, b_j, which cover the underlying connected skin tone regions. For the computations of the bounding box representations, some additional functions are necessary:

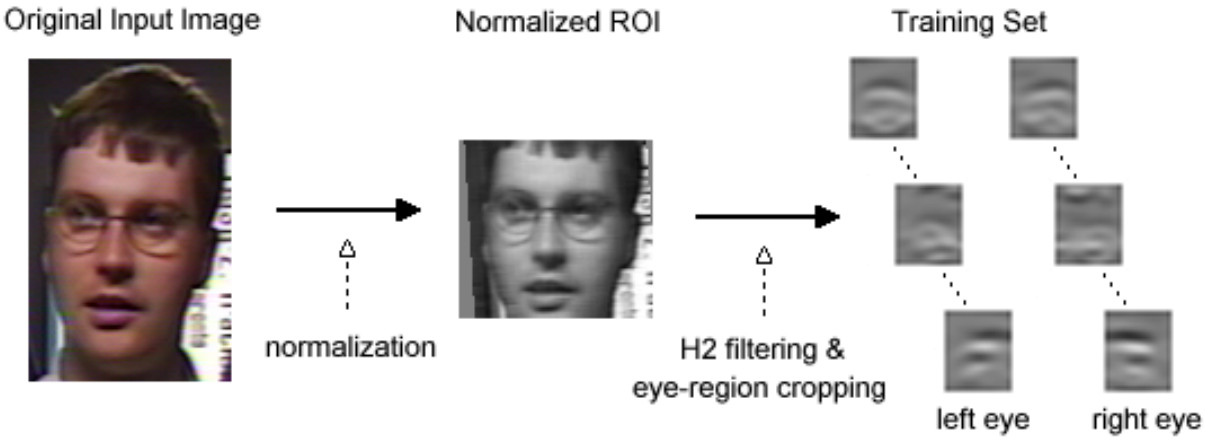


Figure 5. Processing to obtain a training set. Computation of the eye region representation (training).

- A boolean function

$$skin(x) := \begin{cases} 1 \ ; & \text{if colour}(x) = \text{ skin tone,} \\ 0 \ ; & \text{otherwise.} \end{cases}$$

- A neighbourhood function $N(s_i) := \{S_{i,r-1}, S_{i-1,r}\}$ for $i > 1$. For $i = 1$ the neighbourhood N is reduced to $N(s_1) := \{S_{1,r-1}\}$. Only at the beginning of the scan, the bounding box representations b_j are initialized so that in this case no neighbourhood exists.
- A blob association function $bl(x) := b_j$ with b_j blob number of the blob list B, $B := \{null, b_1, \ldots, b_m\}$, $m \in N$. A skin tone blob is defined as $b := (TL, BR, \Omega)$, where $TL := (x, y)$ refers to the top left corner, $BR := (x, y)$ refers to the bottom right corner of the associated bounding box, and Ω is a set of scanline positions associated with that blob, $\Omega_j := \{S_{i,j}; S_{i,j}$ are associated to $b_j\}$. For the blob association of a scanline position $S_{t,r}$ with $t \in$ Indexset of U several cases have to be considered:

$$bl(S_{t,r}) = \begin{cases} b_{new}; & \text{if } skin(s_{t,r}) = 1 \wedge \\ & sk(N(S_{t,r})) = 0, \\ bl(N(S_{t,r})); & \text{if } skin(S_{t,r}) = 1 \wedge \\ & sk(N(S_{t,r})) = 1 \vee \\ & (sk(N(S_{t,r}) = 2 \wedge \\ & bl(S_{t,r-1}) = bl(S_{t-1,r})), \\ M(b_i, b_j); & \text{if } skin(S_{t,r}) = 1 \wedge \\ & sk(N(S_{t,r})) = 2 \wedge \\ & bl(S_{t,r-1}) \neq bl(S_{t-1,r}), \\ null; & otherwise. \end{cases}$$

where b_{new} denotes a new blob representation to be included in B and $sk(X) := \sum_{i=1}^{\|X\|} skin(x_i)$ where X is a set of positions $X := \{x_1, \ldots x_n\}$. To ensure that also the neighbourhood relationship between the first $S_{1,r}$ and the last $S_{h,r}$ scanline heads with respect to a particular circle r is considered, these two scanlines are checked if they are both skin colour after each complete scanning iteration. If so and they do not point to the same blob b_l, the merge function $M(b_1, b_h)$ is called. Therefore, to be precise the just mentioned blob association function has to be expanded by the following special case: $M(b_1, b_h)$; if $skin(S_{1,r}) = skin(S_{h,r}) = 1 \wedge bl(S_{1,r}) \neq bl(S_{h,r})$.

In quadrant III of fig. 3 the extension of an existing bounding box according to this scheme is explained by a sample scanline point $S_{4,3}$ to be included in the skin colour blob b_l.

- A merge function $M(b_i, b_j) = M((TL_i, BR_i, \Omega_i), (TL_j, BR_j, \Omega_j)) := b_{new}$ if b_i and b_j are adjacent, that means they share at least one direct neighbour in terms of the neighbourhood definition of N.

$$M(b_i, b_j) = \begin{cases} TL_{new}(x) = \begin{cases} TL_i(x); & \text{if } TL_i(x) \leq \\ & TL_j(x), \\ TL_j(x); & \text{otherwise.} \end{cases} \\ TL_{new}(y) = \begin{cases} TL_i(y); & \text{if } TL_i(y) \geq \\ & TL_j(y), \\ TL_j(y); & \text{otherwise.} \end{cases} \\ BR_{new}(x) = \begin{cases} BR_i(x); & \text{if } BR_i(x) \geq \\ & BR_j(x), \\ BR_j(x); & \text{otherwise.} \end{cases} \\ BR_{new}(y) = \begin{cases} BR_i(y); & \text{if } BR_i(y) \leq \\ & BR_j(y), \\ BR_j(y); & \text{otherwise.} \end{cases} \\ \Omega_{new} = \Omega_i \cup \Omega_j. \end{cases}$$

b_j has to be removed from B while b_i will be replaced by the new merged blob b_{new}.

If the distance between the heads of two successive scanlines exceeds a predefined threshold $\|S_{i,r_f} - S_{i+1,r_f}\| > D$

then a new scanline is spawned and inserted between them with intermediate orientation (fig. 3 quadrant IV). The new scanline s_{new} is defined only for circle numbers $r \geq r_f$. The first head position is calculated using the following equation:

$$S_{new,r_f} = \frac{S_{i,r_f} + S_{i+1,r_f}}{2} \tag{2}$$

while the corresponding unit vector u_{new} is given by:

$$u_{new} = \frac{S_{new,r_f}}{\|S_{new,r_f}\|} \tag{3}$$

The new unit vector u_{new} must be inserted into U, hence, the elements of U must be reindexed appropriately to maintain the ordered set. In quadrant IV of fig. 3 the insertion of a new scanline s_{new} starting at position $S_{new,3}$ is shown.

The processing of the scan line algorithm presented ensures that the entire area of interest is considered, by starting with a high density in the center and reducing slightly the scan-density in the periphery (fig. 4). Subsequently, the most salient skin colour blob satisfying particular constraints, e.g. with respect to the size and position, is processed further.

4 Face evaluation

After the skin colour blob detection, a verification of the most salient skin colour blob is performed. Typical facial features must be detected in the bounding box area to ensure that a face is present. The most characteristic and distinctive facial features are the eye regions. Therefore, an *eigenimage* based approach to detect and recognize eye regions has been developed. In contrast to the classical eigenimage approach published by Turk and Pentland [13], here only spatially limited facial regions are considered and some other fundamental changes in the calculation have been made. Eye regions, which include the eyebrow, show very prominent horizontally oriented edge and line structures. Accordingly, the eye representation developed here is based on the edge information computed by a horizontally oriented H2 edge detection filter [1]. In the following, we refer to the filtered image as the *'input image'* or simply as *'the image'* because all further processing is done with respect to the H2 filtered image.

$$H2(x,y,\sigma) = \left(\frac{x}{\sigma}0.3458\left(\frac{x}{\sigma}\right)^2 - 1.559\right) e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

Figure 5 shows a diagram of the processing steps required to obtain a training set for the eye representation. First, the red channel of the 24-bit colour image is scaled, aligned, and normalized with respect to the iris centers. Then, the H2 edge detection is performed. To avoid special border considerations during the edge filtering, the entire facial region is convolved first and subsequently, the eye regions are cropped, taking a spatially fixed area surrounding the iris centers. The cropped and H_2-filtered eye regions $F_{H_2}(I_{eye})$ of both eyes from a gallery of 60 randomly chosen faces served as training set Γ. The data matrix A of training set Γ is defined as follows:

$$A = \begin{bmatrix} P_1^T \\ P_2^T \\ \vdots \\ P_n^T \end{bmatrix} \tag{5}$$

where the $P_i = F_{H_2}(I_{eye,i})$ and each image P is considered an image vector of size $m \times 1$. Using singular value decomposition (SVD), A is decomposed into $A = U\Sigma V^T$ where U is an $n \times n$, V is an $m \times m$ and Σ is an $n \times m$ matrix. Σ is a diagonal matrix which contains the singular values S_i of A. It has at most $p = \min(m,n)$ non-zero diagonal elements. U and V are both orthogonal matrices. All m columns of V are eigenvectors of $A^T A$ corresponding to S. Each eigenvector V_i is understood as an image vector, which is referred to as an *'eigenimage'*, or in our case as an *'eigeneye'* of the training set. The SAVI system uses the first three eigeneyes as a representation or model for eye regions (fig. 8c). The underlying assumption is that these three eigeneyes carry enough information to reliably detect an eye region when a skin tone region is given.

For detecting an eye position, a skin tone blob under consideration is scaled appropriately using the size parameters of the skin colour blob as initial estimates (fig. 8a). In the next step, the first eigenimage V_1 (fig. 8c top row) is applied to the H2 filtered image $F_{H_2}(I_t)$ (fig. 8b) and the coefficient map C_1 is computed.

$$c_j = \sum_{i=1}^{m} f_{H_2}(x_i) V_j(y_i) \tag{6}$$

where c_j refers to the jth-coefficient ($j = 1, \ldots, 3$) at a particular image position x and $f_{H_2}(x_i)$ denotes the value of the filtered image at position x_i.

All image positions with $c_1 \leq 0$ are excluded from further considerations because these are known to be uninteresting in advance. Subsequently, the second and third coefficients are computed for the remaining image positions. The residual R is computed using the following equations for each image position x:

$$r = \frac{p - \sum_{i=1}^{3} c_i^2}{p} \tag{7}$$

where p denotes the power at image position x_i:

$$p = \sum_{i=1}^{m} \left(f_{H_2}(x_i)\right)^2 \tag{8}$$

Figure 8(e) shows a sample residual image, in which all excluded areas have been assigned to a white colour value.

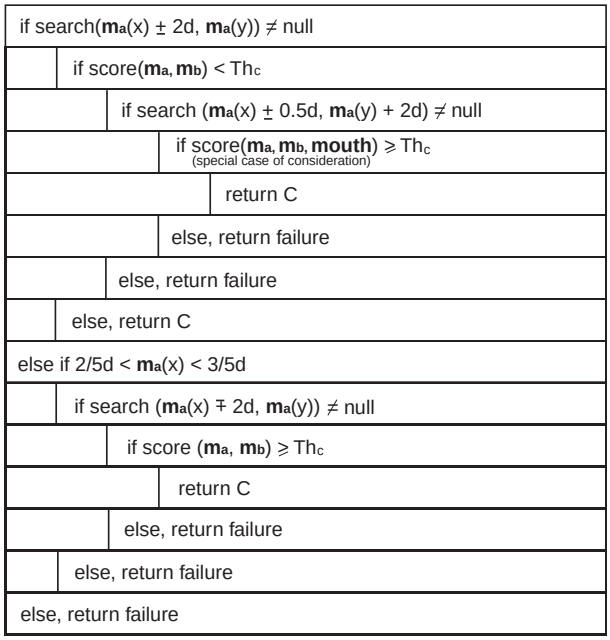


Figure 6. Application of model knowledge.

Each of the remaining image positions carry a particular grey value. By making these exclusions, time consuming computations for the second and third eigenimage coefficients, as well as for the residual image, are avoided. The values are only computed, on average, for half of the image positions in the considered region.

The absolute minimum and a second minimum, which is spatially located at the expected second eye position, are searched for during the evaluation of the residual image (fig. 6). The presence of both eyes in an approximately horizontal position, also satisfying additional model knowledge assumptions, is taken as a characteristic and reliable feature for a human face (fig. 7).

While the position of the absolute minimum m_a can be computed in parallel with the calculation of the residual map, the position of a second local minimum m_b has to be determined relative to the absolute one in a separate process where $m_a \neq m_b$. Furthermore, model knowledge about the facial relations, FK, must be applied in order to find the expected position for a reliable minimum. For that task a gradient descent-like algorithm called $search(x, FK)$ is applied. The search algorithm is defined iteratively as (for simplicity, in the following the model knowledge assumptions FK have been waived from the function):

$$search(m_{b,t+1}) = min(r(N_i(m_{b,t}))),$$
$$i := \{0, \ldots, 3\} \quad \text{until } m_{b,t+1} = m_{b,t}, \quad \text{with}$$
$$search(m_{b,0}) = (m_a(x) \pm 2\hat{d}, m_a(y)),$$

('$\pm$' depending on whether m_a is associated to the left or the right eye), where $N_i(x) = ((2i+1) \times (2i+1)$ pixel neighbourhood surrounding $x) - \sum_{k=i}^{1} N_{k-1}(x)$, for $1 \leq i \leq 3$, with $N_0(x) = x$, and $r(x)$ refers to the residual of x (equa. 7). The search for $m_{b,t+1}$ is cancelled at the smallest calculated neighbourhood if $r(m_{b,t}) > min(r(N_i(m_{b,t})))$, for $1 \leq i \leq 3$.

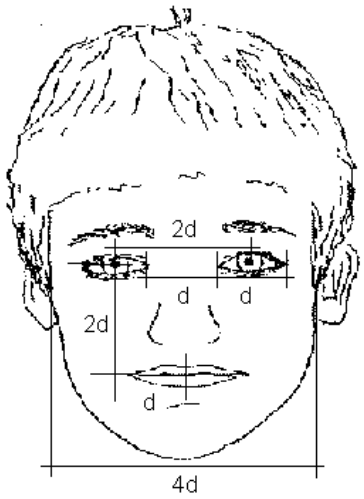


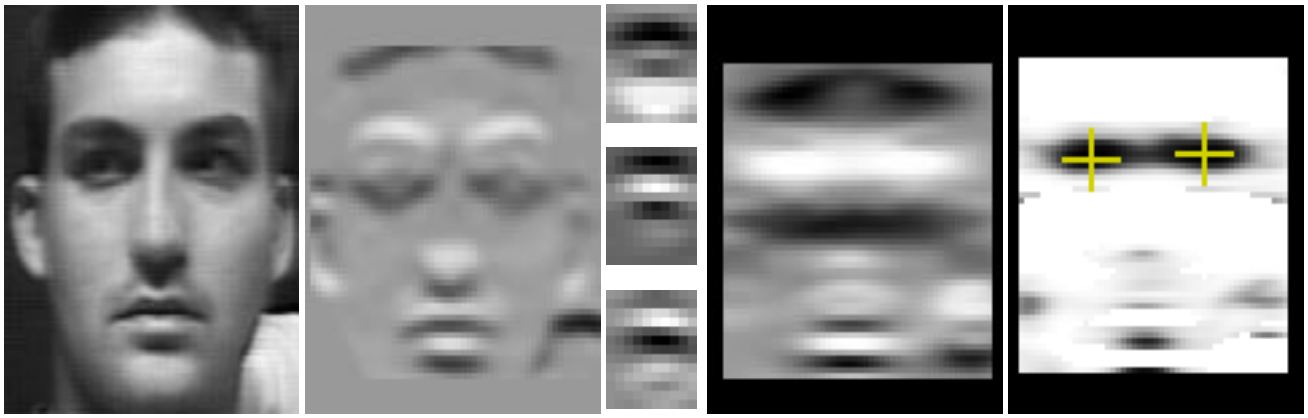
Figure 7. Ideal facial proportions which are relative to the reference facial distance d.

$\hat{d}$ denotes the estimate of the 'reference facial distance' d, which represents an ideal proportion and appears several times in the measurement of distances between facial features [14, 6]. Figure 7 shows several examples of distances of facial features relative to the reference facial distance d. In the current implementation of SAVI, $\hat{d}$ is calculated by the width of the face, which is $4d$ in the ideal case.

The evaluation of the detected minima positions is performed by computing a confidence score $score(m_a, m_b)$ of the absolute minimum m_a and the second minimum m_b.

$$score(m_a, m_b) := \sum_{i=0}^{N} w_i \; c_i$$

where the c_i's are particular constraint confidence scores, $0 \leq c_i \leq 1$, for $i \in \{1, \ldots, N\}$, and w_i are weighting factors associated with each constraint c_i , $0 \leq w_i \leq 1$ for $i \in \{1, \ldots, N\}$ and $\sum_{i=0}^{N} w_i = 1$. Typical number of constraints is $N = 5$. There are a number of different quantization steps during the calculation of constraint confidence scores c_i, which are referred to by an additional running index j, $j \in \{1, \ldots, N\}$. Each quantization step j is associated with an empirically established, constant confidence value $k_{i,j}$, where $0 \leq k_{i,j} \leq 1$, and a threshold $th_{i,j}$, $th_{i,j} \in \Re$. Most thresholds $th_{i,j}(d)$ are functions of d, e.g. for the confidence rating c_2. In this case, the confidence value depends on the difference between the currently measured distance between the pupil centers and the ideal distance $2d$ and is set relative to d.



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

Figure 8. Evaluation of a possible facial region by detecting both eye regions and checking model knowledge assumptions. (a) Red channel of an appropriately scaled bounding box region of a test image; (b) filtering result using a horizontal oriented H2 filter; (c) first three eigenimages of the eye region representation (from top to bottom); (d) coefficient map of the application of first (top) eigenimage to the filtered image (all bright areas show positive coefficient values); (e) residual map after applying all three eigenimages (dark values show low values or low errors). Only positions which have positive values in the first coefficient map (d) are considered during the evaluation step of the residual image. The white areas in (e) indicate the excluded image positions while the plus signs show the positions where the eyes were detected.

- Evaluation of the depth of both minima:

$$c_1(m_a, m_b) := c_{1,h}(r(m_a)) \times c_{1,h}(r(m_b))$$

$$\text{where } c_{1,h}(r(x_l)) := \begin{cases} k_{1,j,l}; & \text{if } r(x_l) < th_{1,j,l} \\ 0; & \text{otherwise,} \end{cases}$$

$$\text{and } x_l \in \{m_a, m_b\} \text{ and } l := \{1, 2\}.$$

- Distance between pupil centers:

$$c_2(m_a, m_b) := \begin{cases} k_{2,j}; & \text{if } \|2d - \|m_a - m_b\|\| < \\ & th_{2,j}(d) \\ 0; & \text{otherwise.} \end{cases}$$

- Horizontal displacement of the pupil centers:

$$c_3(m_a, m_b) := \begin{cases} k_{3,j}; & \text{if } \|\frac{\text{face width}}{2} - \frac{m_a(x)+m_b(x)}{2}\| \\ & < th_{3,j}(d) \\ 0; & \text{otherwise.} \end{cases}$$

- Vertical displacement of the pupil centers:

$$c_4(m_a, m_b) := \begin{cases} k_{4,j}; & \text{if } \|\frac{\text{face height}}{3} - \frac{m_a(y)+m_b(y)}{2}\| \\ & < th_{4,j}(d) \\ 0; & \text{otherwise.} \end{cases}$$

- Tilt of the eyes:

$$c_5(m_a, m_b) := \begin{cases} k_{5,j}; & \text{if } \|\frac{\arctan(\|m_a(y)-m_b(y)\|)}{\|m_a(x)-m_b(x)\|}\| < \\ & th_{5,j}(d) \\ 0; & \text{otherwise.} \end{cases}$$

During a possible postprocessing step, several additional facial features of a more precise value can be examined in more detail (see for instance [4, 5]). To enable such a processing, the facial region has to be zoomed in and tracked appropriately. The tracking of the face will be continued until either it disappears from the visual field or a particular command indicates a new action.

5 Tracking

Based on a modified version of the radial scan algorithm introduced previously, an efficient tracking method has been developed. The dynamic change in position(s) of the bounding box(es) surrounding the skin tone blob in each frame of the video sequence is considered and motion vectors with respect to the centroid(s) of the bounding box(es) are calculated. A history of past centroid positions is maintained for each bounding box. Error management ensures that possible misleading positions are excluded. Only a fixed fraction of centroid positions fitting a special homogeneity criterion are kept. A streak of bright plus signs (approximately 30 tail positions of the centroid) in figure 9 indicates the past positions of the bounding box centroids, which are stored and/or updated in the location history. During tracking, the size of the bounding box of the attended skin colour object is dynamically adjusted by continuously applying a simplified version of the radial scan algorithm. Furthermore, another error management process ensures that image collisions that occur, while the supervisor is attending multiple moving objects, can also be handled.

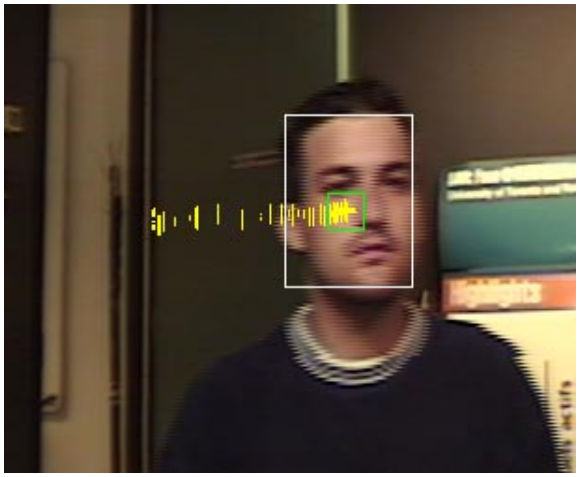


Figure 9. Tracking of a face. History (yellow plus signs) and prediction (green square) of the bounding box's centroid of a moving skin colour target.

To predict the position of the blob's centroid with respect to a particular future time step, a finite set of past motion steps is evaluated. The green square in figure 9 indicates the predicted centroid position in the next time step. The computation of a time dependent prediction is particularly important in maintaining the tracking of a skin tone blob because tracking cannot be performed continuously without any break. Breaks may occur if processing power is used for other tasks or if multiple targets are tracked simultaneously. It is clear that confidence will drop rapidly with increasing time delays, but based on a history of past locations for a number of frames of the sequence, such a prediction may be useful for a small time range.

6 Hardware specifications

The hardware system configuration consists of a Tyan Tiger motherboard with dual Pentium II 400MHz MMX CPUs, 512 KB L2 cache per CPU, 128 MB RAM, two Imaging Technologies IC4-COMP S-Video frame grabbers, one Precision Motion Control DCX-PC 100 motion controller with 4 DCX-MC110 modules. Two SONY EVI-310 cameras, each having a resolution of 512×480 pixels, together with IF-51 serial controllers are connected to the frame grabbers and the motherboard's serial ports. Four servo actuators are connected to the DCX-110 modules for controlling pan and tilt of the head, and pan of each camera. There are 3 main threads in our parallel program. One thread runs the supervisor, the second and third threads control the display of the left and right cameras, respectively. The head motor controller is designed as a separate process that communicates with the vision system via TCP/IP sockets. In this manner the head is controllable from anywhere on the Internet.

7 Conclusions and discussion

One fundamental aspect of SAVI's design is the integration of model knowledge at different stages of processing. This integration is absolutely necessary to reduce the search space and to pay attention to only those aspects of the visible scene which are relevant. To find reliable and robust model knowledge for real world applications, which also satisfies performance requirements, is not an easy task. In addition, the drawback of the selection of particular model knowledge is that it may cause failures in situations which are not covered by the model knowledge assumptions. But there is no easy resolution to this problem because without a minimum set of constraints and restrictions a real-time system on currently modest hardware may be impossible. Therefore, we know that our system will not suit all possible circumstances because it is not developed as a general purpose machine. But it works reliably and provides meaningful results in comparison to static systems with no active control. Two animated GIF sequences showing how SAVI works can be found at URL: http://www.cs.toronto.edu/~herpers/projects.html.

SAVI is still under construction and changes may be necessary to improve its performance. But the general underlying idea of SAVI will remain the same. Static vision systems, which do not enable any kind of reaction to changing conditions, will be replaced by more sophisticated solutions incorporating a particular set of active behaviors needed to cope with more specific and/or extensive applications. The set of actions provided by these new systems is then mainly dependent on the task or application and the environment for which they are designed.

Acknowledgements

We thank the many students and staff members at the University of Toronto and York University who have contributed: Chakra Chennubhotla, Laura Hopkins, Caroline Pantofaru, Nancy Yuen. Development of SAVI has been funded by IRIS (Institute for Robotics and Intelligent Systems, a Government of Canada Network of Centres of Excellence), NSERC (the Natural Science and Engineering Council of Canada) and IBM Canada, Centre of Advanced Studies. R. Herpers acknowledges the support of the Deutsche Forschungsgemeinschaft (DFG), Grant: He-2967/1-1.

References

[1] W.T. Freeman and E.H. Adelson, "The design and use of steerable filters for image analysis", *IEEE Trans. PAMI*, Vol.**13**, 891-906, 1991.

[2] W. T. Freeman and C. D. Weissman, "Television control by hand gestures", *Int. Workshop on Autom. Face- and Gesture-Rec.*, M. Bichsel (edt.), Zurich, Switzerland, pp. 179-183, 1995.

[3] J. Cai, A. Goshtasby, and C. Yu, "Detecting human faces in color images", *Int. Workshop on Multi-Media Database Management Systems*, to be published, 1998.

[4] R. Herpers, H. Kattner, H. Rodax, and G. Sommer, "GAZE: An attentional processing strategy to detect and analyze the prominent facial regions", *Int. Workshop on Autom. Face- and Gesture-Rec.*, M. Bichsel (edt.), Zurich, Switzerland, pp. 214-220, 1995.

[5] R. Herpers, M. Michaelis, K.H. Lichtenauer, and G. Sommer, "Edge and keypoint detection in facial regions", *Second Int. Conf. on Automatic Face and Gesture Recognition*, Killington, Vermont, IEEE Computer Society Press, pp. 212-217, 1996.

[6] R. Herpers, "GAZE: A common attentive processing strategy for the detection and investigation of salient image regions", Tech-Rep. No. 9714, University of Kiel, Germany, 1997.

[7] R. Herpers and G. Sommer, "An attentive processing strategy for the analysis of facial features", *Face Recognition: From Theory to Applications*, H. Wechsler et al. (eds), NATO ASI Series F, Springer-Verlag, Vol. 163, pp. 457-468, 1998.

[8] R. Kjeldsen and J. Kender, "Finding skin in colour images", *Second Int. Conf. on Automatic Face and Gesture Recognition*, Killington, Vermont, IEEE Computer Society, pp. 312-317, 1996.

[9] M. Lucente, G.-J. Zwart, and A. D. George, "Visualization space: A testbed for deviceless multimodal user interfaces", *Intelligent Environments Symposium '98*, AAAI Spring Symposium, 1998.

[10] E. Milios, M. Jenkin and J. K. Tsotsos, "Design and performance of TRISH, a binocular robot head with torsional eye movements", *Int. J. of Pattern Rec. and Artificial Intelligence*, Vol. 7(1), pp. 51-68, 1993.

[11] Y. Raja, S. J. Mckenna, and S. Gong, "Tracking and segmenting people in varying lighting conditions using colour", *Third Int. Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, IEEE Computer Society Press, pp. 228-233, 1998.

[12] J. K. Tsotsos, G. Verghese, S. Dickinson, M. Jenkin, A. Jepson, E. Milios, S. Stevenson, M. Black, D. Metaxas, S. Culhane, Y. Ye, and R. Mann, "PLAYBOT A visually-guided robot for physically disabled children", *Image and Vision Computing*, Vol. 16, pp. 275-292, 1998.

[13] M. Turk and A. Pentland, "Eigenfaces for Recognition", *J. Cogn. Neurosci.*, Vol.3, pp. 71-86, 1991.

[14] L. da Vinci, *Treatise on Painting*, Vol.1, Princeton, New Jersey, Princteon University Press, 1550.

[15] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland "Pfinder: Real-time tracking of the human body", *IEEE Trans. PAMI*, Vol. 19, No. 7, pp. 780-785, 1997.