Prolegomena for Robust Face Tracking

Microsoft Research Technical Report
MSR-TR-98-65
November 13, 1998

Kentaro Toyama
Vision Technology Group
Microsoft Research
Redmond, WA 98052
kentoy@microsoft.com

Paper presented at the Workshop on Automatic Facial Image Analysis and Recognition Technology (ECCV '98).

Abstract

Real-time 3D face tracking is a task with applications to animation, video teleconferencing, speechreading, and accessibility. In spite of advances in hardware and efficient vision algorithms, robust face tracking remains elusive for all of the reasons which make computer vision difficult: Variations in illumination, pose, expression, and visibility complicate the tracking process, especially under real-time constraints.

This paper considers the problem of robust face tracking. First, we survey recent work in face tracking, which helps us determine what is required for robust face tracking. In particular, we note that robust systems tend to possess some state-based architecture comprising heterogeneous algorithms, and that robust recovery from tracking failure requires several other facial image analysis tasks. Finally, we briefly describe one system for robust face tracking that uses Incremental Focus of Attention (IFA), a state-based architecture which allows fast recovery of lost targets within a unified framework.

Keywords: face tracking, real-time tracking, robust vision

1 Introduction

Amidst advances in hardware speed and greater demand for practical vision applications, *face tracking* has emerged as an area of research with exciting possibilities.

Face tracking allows hands-free human-computer interaction for novel user interfaces or game input devices. Graphical avatars can be puppeteered through face tracking. Low bit rate video teleconferencing will be possible with a combination of tracking, compression, and rendering. Finally, many applications which require further processing of the face – expression recognition, for example – often require face tracking.

We will focus on the task of tracking *facial pose*, where information about the position and orientation of the face is recovered. In particular, this paper will discuss issues of performing face tracking robustly.

2 Task Definition

Although there are some attempts to standardize the output of face tracking systems [21], researchers claiming to do "face tracking" track everything from the 2D image coordinates of the center of a head-like blob to complex expressions of a particular individual's face. For our purposes, we consider the following narrowly defined task:

Definition 1 *The* **3D Face-Pose Tracking Task** *is the real-time recovery of the six-degree-of-freedom (6-DOF) pose of a target structure that has a rigid geometric relationship to the skull of a particular individual, given a stream of live black-and-white or color images.*

There are several things to note in this definition.

The *pose* is a 6-vector, $[x\, y\, z\, \alpha\, \beta\, \gamma]^T$, where the parameters give the position and orientation of the target structure in some Euclidean coordinate system (usually a fixed frame relative to the camera).

Real-time tracking is important, because many applications require online tracking of a subject's face. As computing machinery becomes faster and faster, this constraint becomes less meaningful or, at least, easier to achieve. Efficient tracking algorithms will nevertheless remain of interest, however, because computation conserved during face tracking can always be allocated to further facial analysis tasks.

We define the task specifically for a single, unique individual, and not merely for any face that appears in view. Thus, when multiple faces are in the field of view, a tracking system should track only the face specified beforehand.

Aside from reasonable ambient lighting, we do not consider techniques which require significant structure or special lighting in the environment, whether such contrivances are visible to the human eye or not.

We are not concerned with the tracking of anything other than the pose of the target structure, although tracking of individual facial features may certainly play a part in pose estimation.

We distinguish the face tracking task from the related tasks of *face detection* and *face recognition*. Face detection answers the questions, "Is there a face in the image, and if so, where is it in the image?" The face recognition task asks, "Is the face in the given image familiar, and if so, which one is it?" We will discuss more about the relationship between tracking, detection, and recognition in Section 4.

Finally, we emphasize that the focus of this paper is on robustness and on architectural principles for robust face tracking. Many important dimensions of the task, such as precision of state estimates or generalizability of algorithms are not considered.

3 Survey of Existing Tracking Systems

In Figure 1, we tabulate systems which track facial pose and do so roughly within the constraints of Definition 1. We have striven to include all reported face tracking systems which purport to track face or head pose using vision alone, and have appeared in the literature prior to May, 1998. For brevity, only the most recent or most representative work by a particular group is tabulated. Because the table has been deliberately organized to identify trends across face tracking systems, the interesting aspects of the systems have been suppressed. Readers are urged to consult the original sources for details.

The table is split into three sections based on the dimensions of pose recovered by each system. The first section lists those systems which track position only. The second section includes systems which include some orientational information, but not all six degrees of freedom. The third section contains systems which track all six DOF. Listings within each section are alphabetical.

First Author, Year, Citation	Parameters	Actual Target	Top-Level Tracking	Recovery Method	Speed (msec)	H/W	Robust to
Birchfield 98 [3]	xyz	SCEB	C E	=	33	Pm 200	O R
Collobert 96 [7]	xy	AF	CM	CM T	40	DEC 250/4	O R
Crowley 97 [9]	xy	AF	F	CM F	40	150MHz	
Darrell 98 [10]	xy	SCB	C D T	=	83	2 SGIs	O R
Fieguth 97 [15]	xy	SCB	C	=	3.3	SGI R4600	O R
Isard 96 [20]	xy	HS	E	=	80	SGI R4400	O R
Raja 98 [31]	xyz	AF	C	=	67	Pm 200	O R
Sobottka 96 [33]	xy	SCB	E	C EF	NRT	SGI R4400	O R
Turk 96 [39]	xy	HS	E	=	33	Pm 166	O R
Chen 98 [5]	$\alpha\beta\gamma$	SCB	C	=	333	SGI Indigo	O R
de la Torre 98 [11]	xy $\alpha\beta\gamma$	PF	T	C F	NRT	Pm 200	O
Elagin 98 [14]	xyz β	AF	F	=	NRT	SGI Octane	O R
Niyogi 96 [25]	$\alpha\beta\gamma$	AF	T	=	90	SGI Indy	O R
Oliver 97 [26]	xyz γ	SCB	C	=	33	SGI R4400	O R
Pappu 98 [27]	xy $\alpha\beta\gamma$	PF	T		167	SGI	O R
Rae 98 [30]	xy $\alpha\beta$	AF	C F	=	NRT	SGI XZ400	O R
Xu 98 [41]	$\alpha\beta\gamma$	AF	F		NRT	–	
Bakic 98 [1]	xyz$\alpha\beta\gamma$	AF	C F	=	50	SGI Indy	
Basu 96 [2]	xyz$\alpha\beta\gamma$	3DTE	O		NRT	SGI	O R
Colmenarez 95 [8]	xyz$\alpha\beta\gamma$	AF	F		33	SGI Onyx	
DeCarlo 96 [12]	xyz$\alpha\beta\gamma$	PF	F O		NRT	SGI O2	O R
Edwards 98 [13]	xyz$\alpha\beta\gamma$	PF	T	=	NRT	–	O
Gee 96 [16]	xyz$\alpha\beta\gamma$	AF	F		8	Sun 10	O
Hager 96 [17]	xyz$\alpha\beta\gamma$	PF	T		33	SGI	O
Heinzmann 98 [18]	xyz$\alpha\beta\gamma$	PF	F	=	33	MEP	O
Horprasert 96 [19]	xyz$\alpha\beta\gamma$	AF	F		NRT	Sun	
Jebara 97 [22]	xyz$\alpha\beta\gamma$	(PF)	F	C F	40	SGI R5000	O
LaCascia 98 [4]	xyz$\alpha\beta\gamma$	PF	T		1000	SGI O2	O
Maurer 96 [23]	xyz$\alpha\beta\gamma$	PF	F		NRT	Sun 20	R
Petajan 96 [28]	xyz$\alpha\beta\gamma$	AF	F	=	33	Sun	
Potamianos 97 [29]	xyz$\alpha\beta\gamma$	AF	F	CM F	33	Pm 150	
Saulnier 95 [32]	xyz$\alpha\beta\gamma$	AF	F	=	100	SGI Indy	
Stiefelhagen 97 [34]	xyz$\alpha\beta\gamma$	AF	C F	=	33	HP-9000	
Toyama 98 [36]	xyz$\alpha\beta\gamma$	PF	F	C FT	33	Pm 266	
Wiles 97 [40]	xyz$\alpha\beta\gamma$	PF	F		500	Sun 10	O

Tracking & Recovery Algorithm Type			
C	color blob	E	edge (contour)
M	motion blob	F	feature
D	depth blob	T	template
		O	optic flow

Actual target		Speed and Hardware	
3DTE	3D textured ellipsoids	DC	Datacube w/MaxVideo250
AF	any face	HP	Hewlett-Packard
HS	head and shoulders shape	DEC	DEC Alpha
PF	particular face	MEP	Fujitsu MEP card
(PF)	particular face, any on recovery	NRT	more than 1 sec per cycle
SCB	skin-colored blob	Pm X	Pentium XMHz
SCEB	skin-colored elliptical blob	SGI	Silicon Graphics
		Sun X	Sun Sparc X

Robustness against			
O	partial occlusion	R	out-of-plane rotations

Figure 1: Face tracking systems (key at bottom).

The columns indicate various traits of each system. "Parameters" lists the degrees of freedom of pose recovered by the system, where x and y are horizontal and vertical axes, z represents the optical axis, and α, β, and γ represent rotations about x, y, and z, respectively. "Actual Target" indicates what the system is actually tracking (blob of skin-color, any face in view, a particular face, etc.), as listed in the key.

"Top-Level Tracking" shows the type of algorithm used when tracking under idealized conditions, and "Recovery Method" lists the kind of algorithms used to reinitialize a lost target. In both columns, the categories are restricted to *color*, *motion*, *depth*, *edge*, *feature*, *template*, and *optic flow* (this list is sorted in approximate order of increasing processing time). Systems labeled with "color," "depth," or "motion" track clusters of pixels classified accordingly. "Edge" indicates use of the contour of the head against the background for tracking. "Feature" means that specific landmarks or facial features are sought and tracked.

Features may be constrained geometrically. "Template" implies that a global template or set of templates of size approximately equal to the entire face is matched over the face. Note that throughout this article, any tracking methodology that aims to track specific landmarks on the face (including the use of small templates) will be considered a form of feature-based tracking. And, "optic flow" systems use optic flow. Finally, empty boxes suggest that tracking recovery is not implemented, and an equal sign, =, indicates that the same algorithm used for tracking is sufficient for recovery.

The next two columns show speed and hardware used.

Finally, the last column shows the performance of each system with respect to two types of visual disturbances: partial occlusions and out-of-plane rotations. Those marked with an `O` are able to continue tracking through small partial occlusions, though possibly with less precision. Those marked with an `R` are able to track faces in profile views. Note that this column considers only robustness in the sense of avoiding tracking failure; the ability to reinitialize tracking is not tallied.

The table immediately reveals several characteristics of the face tracking problem and possible solutions.

Simple Cues: First, many systems use variations on algorithms that use easily computed visual cues such as color or motion. Color, for example, is popular both for tracking as well as for recovery. In the table, we see that color appears frequently in position-tracking systems, but seldom in those systems that track orientation. Indeed, despite the variety in color models and clustering algorithms, overall performance of color tracking is similar – algorithms are fast and limited to tracking position only. Though not apparent from the table, color-based tracking also has difficulty with similarly colored distractions in the background. These limitations for tracking, however, do not prevent it from being a good cue for reinitialization, where speed is paramount, and precision, less important.

Feature vs. Template: In the bottom section of the table, we see that the dominant algorithm types for tracking orientation of faces are template-based and feature-based. Again, in spite of the variety in algorithms for template and feature tracking, the qualitative properties within each group are similar. Template and feature techniques can be considered the extreme ends of a continuum of matching techniques in which tradeoffs between match size and match density are made. Template-based tracking involves multiple templates or warping of templates to accommodate changing pose. The dense comparison of corresponding pixels between image and template is computationally expensive but it also provides more robust tracking than feature-based techniques. Feature-based tracking on the other hand, though more prone to individual outliers and mistracking, can be very efficiently implemented. Figure 1 shows two trends that bear out this analysis: Template-based methods are usually insensitive to small partial occlusions, but most frame rate implementations of 6-DOF face

tracking rely on tracking of features.

Difficulty of Recovery: Finally, we note that most 6-DOF systems are not able to recover from tracking failure, in contrast to systems which track fewer degrees of freedom. Among those systems which are able to recover from tracking failure, some rely heavily on the fact that there is only one blob of skin-color in the image, and others wait for a face to appear in a particular subregion of the image. Neither of these techniques guarantees recovery even when the target face is clearly visible in the image. More thorough recovery can be bought at the price of a brute-force approach, but computational costs are significant. That robustness and recoverability decrease with greater degrees-of-freedom tracked suggests that these issues are central to the problem of 3D tracking.

All of these trends taken together suggest that robust face tracking is a challenging testbed for vision-based tracking as a whole. We next consider the requirements for practical, robust, real-time 3D pose tracking of faces.

4 Requirements of Robust Tracking

What exactly are the requirements for robust real-time face tracking? Examination of the subgoals of this task helps to identify the requisite elements.

For *ante-failure robustness*, or the ability to avoid tracking failure [37], it is obvious that good motion tracking and pattern matching is necessary. In particular, tracking routines must be able to survive the simultaneous onslaught of standard computer vision problems, from pose changes to illumination variation.

Ante-failure robustness can go only so far, however, since circumstances such as prolonged full occlusion make pose estimation impossible. Thus, face tracking systems must necessarily incorporate mechanisms for *post-failure robustness* – the ability to recover from failure.

In the worst case, tracking failure implies that a system has zero information about the target face; all poses are equally probable. Recovery, then, becomes tantamount to tracking initialization, where face detection, followed by face recognition and possibly feature detection are necessary to localize and verify the target face before tracking can continue.

In order to perform these functions in real time, algorithms must make use of task constraints. In the case of faces, systems should take advantage of constraints offered by color, motion, face and feature symmetry, shape, trajectory continuity, and so on.

Most importantly, robust face tracking requires an architecture which dynamically swaps heterogeneous algorithmic components appropriate to the prevailing visual conditions and the state of knowledge of the system. Switching between algorithms allows a system to capitalize on the strengths of each component, while minimizing the effect of algorithmic weaknesses.

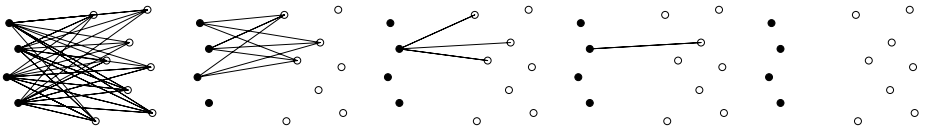


Figure 2: Schematic representation of search spaces.

5 Incremental Focus of Attention

Incremental Focus of Attention (IFA) [38] is a state-based framework for robust real-time tracking which satisfies, in part, the architectural desiderata of the previous section. We give a brief description of IFA below. For further details, see [38].

The key idea is to cast all tasks as *search space reduction*. Search space reduction is a powerful notion because it allows different operations on faces to be perceived as a single type of activity. The search space in face tracking is $\mathcal{C} \times \mathcal{O}$, the space of poses crossed with object classes, which we will call the space of *configurations*. Figure 2 schematically represents the types of search spaces that may arise during execution. In each of (a) through (e), the solid dots represent a single object class, the empty circles represent poses, and edges denote points in the search space. Thus,

- Tracking failure occurs when a search space is reduced to the empty set (This can be viewed as going from (c) to (e)).
- Face detection can be seen as search space reduction from (a) to (b).
- Face recognition is a search for a particular face ((b) to (c)).
- Tracking initialization occurs when the search is successful and pose is recovered ((a) to (b) to (c) to (d)).
- Face tracking is a repeated search in a restricted search space ((c) to (d)).

In IFA, algorithms are organized into *layers* and cast as consumers and producers of *configuration sets* (see Figure 3(left)). Layers take an input configuration set, in which the target configuration is expected (though not necessarily present) and return a smaller output configuration set, using the current image as a hint.

Layers come in two flavors: *trackers* guarantee that non-empty output configuration sets contain the actual target configuration if their input sets do, whereas *selectors* shrink configuration sets without such assurance (good selectors, however, will be likely to identify output sets containing the target configuration).

The layer structure is associated with a state transition graph (Figure 3(right)). When an IFA system assumes a particular state, the associated layer (horizontally to its left) is executed. Transitional links are followed based on the success

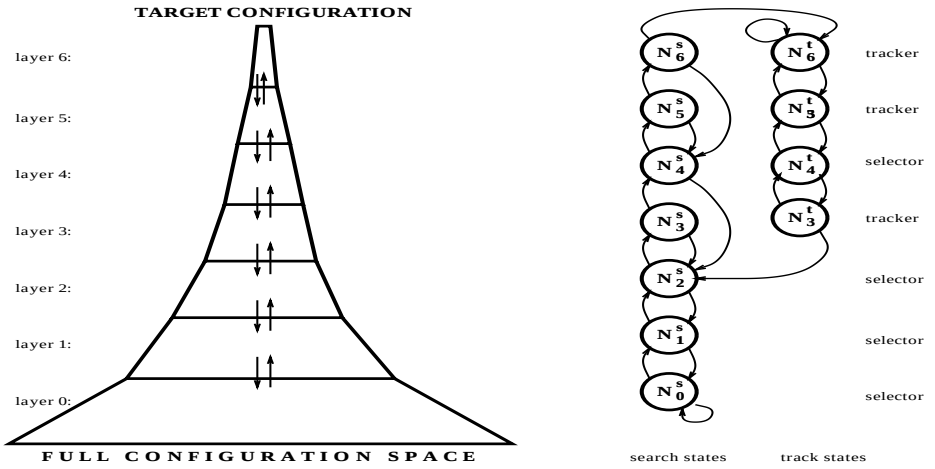
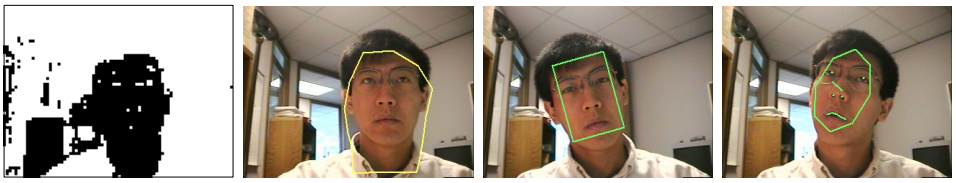


Figure 3: The Incremental Focus of Attention Framework.



(1) (3) (5) (6)

Figure 4: Face tracking at different layers.

(upward links) or failure (downward links) of the layer executed. Failure for trackers occurs when the output configuration set is empty, and failure for a selector occurs if control returns to the selector a specified number of times without reaching top-layer tracking (the number is set prior to execution for each selector state). Note also in Figure 3(right), that states are organized in two columns. Those in the left-hand column indicate `search` states in which no concrete target information is known; those in the right-hand column are `track` states where at least partial configuration is recovered.

6 Real-Time Implementation

Our current real-time implementation runs on a single-processor 266MHz Pentium II PC with a Matrox Meteor framegrabber.

The system uses six layers (see Figure 4 for four of the layers in action). In the following, all variables, k, indicate a constant threshold set empirically.

Layer 1 selects regions of the search space corresponding to pixels exhibiting

skin color. Skin color is defined to be those RGB values where

$$\begin{array}{ccccc} k_{rg}^{-} & < & r/g & < & k_{rg}^{+} \\ k_{rb}^{-} & < & r/b & < & k_{rb}^{+}. \end{array}$$

This color model describes a thin pyramidal wedge in RGB space which accepts skin colors of all races under approximately white light. Although probabilistic or adaptive color models are conceivable [26, 31], this layer is meant as an initial attention-focusing scheme only; a fast, liberal color model is preferred to a slower, more precise one which might miss actual skin under varying illumination. Any pixel not classified as skin-color immediately eliminates a range of configurations from the search space: facial configurations that would project skin-color to image pixels not classified as skin-color are thrown out.

Layer 2 is a color- and motion-based selector that is biased toward regions nearest the last observed position of the target. The search spirals outward and selects state sets consistent with those pixels which exhibit both skin color (as in Layer 1) and large motion,

$$\frac{dI(\mathbf{x})}{dt} > k_m.$$

$I(\mathbf{x})$ represents the image intensity at pixel $\mathbf{x}$. Search space reduction proceeds in a manner similar to Layer 1.

Layer 3 uses "radial spanning" to find the approximate size and shape of a single cluster of skin-colored pixels. Radial spanning is a fast cluster detection algorithm: Pixel probes are initialized at the center of the predicted position of the face and then extended radially outward until they find non-skin-colored pixels [35]. Probes are affected by forces as follows:

$$\begin{array}{lcll} F_i & = & F_i^{out} + F_i^{in} + F_i^{int}, & \text{where} \\ F_i^{out} & = & k_{out}, & \text{if pixel at } \mathbf{x}_i \text{ is skin color,} \\ F_i^{in} & = & k_{in}, & \text{if pixel at } \mathbf{x}_i \text{ is not skin color,} \\ F_i^{int} & = & k_{int} * \frac{(2\mathbf{x}_i - \mathbf{x}_{\mathbf{p}(i)} - \mathbf{x}_{\mathbf{s}(i)}) \cdot \mathbf{v}_i}{|2\mathbf{x}_i - \mathbf{x}_{\mathbf{p}(i)} - \mathbf{x}_{\mathbf{s}(i)}|} & \end{array}$$

where i indexes a predetermined number of probes (16 are used for face tracking), $\mathbf{x}_i$ is probe i's pixel location, $\mathbf{v}_i$ is probe i's expansion direction, and $\mathbf{p}$ and $\mathbf{s}$ return the predecessor and successor indices of i. The algorithm is similar to "draping" [39], but without the static background restriction, and to balloons [6], but with greater efficiency and resistance against spurious edges. The size and centroid of the resulting blob gives an approximate estimate of the center of the face in 3D. Thus, the search space is reduced considerably.

Layer 4 is the same as Layer 3, with an additional computation of the principal axis of the cluster; approximate position and in-plane orientation are tracked. This is the first step in which configurations corresponding to a range of orientations are eliminated from the search space.

Layer 5 performs linearized sum-of-squared-differences (SSD) tracking [17], which iteratively matches live images to a stored template acquired during a

manual initialization step:

$$\mathbf{X}_{t+1} = \mathbf{X}_t - (\mathbf{J}^T\mathbf{J})^{-1}\mathbf{J}^T(\mathbf{I}(\mathbf{X}_t, t+1) - \mathbf{I}(\mathbf{0}, t_0)),$$

where $\mathbf{X}$ is a vector $[x\, y\, \theta]^T$, $\mathbf{I}(\mathbf{X}, t)$ is a vector representing the image at time t translated and rotated according to $\mathbf{X}$, $\mathbf{J}$ is the is the Jacobian of $\mathbf{I}$ with respect to $\mathbf{X}$, and $\mathbf{I}(\mathbf{0}, t_0)$ is the original template. Fine position and in-plane orientation are tracked. This layer works only for approximately frontal poses. In those poses, by accepting only those matches which have a low SSD residual value, it ensures that the remaning configurations correspond to a particular individual, and not just any skin-colored object – face or otherwise. In addition, both in-plane position and rotation are more precisely localized, further thinning the search space.

Layer 6 tracks 5 point features of the face including the eyes, the nostrils, and the mouth. Features are initialized based on their expected relative locations with respect to the Layer 5 template. The eyes and nostrils are then tracked using small search windows with first moment computations on the inverse of pixel intensity to update feature and window positions (similar in spirit to the feature trackers used in [16]). The upper lip is tracked using a snake-like contour tracking algorithm attracted to intensity valleys in the image [24]. These features are then used to determine the 3D transformation from a face reference frame to the camera reference frame by finding a least-squares fit between an approximate geometric model of facial features and the tracked features under weak perspective:

$$\mathbf{T} = (\mathbf{P}^T\mathbf{P})^{-1}\mathbf{P}^T\mathbf{Q},$$

where $\mathbf{T}$ is the recovered transformation matrix, $\mathbf{P}$ is a 3x5 matrix of the concatenated image feature positions with the optical axis coordinate held constant, and $\mathbf{Q}$ is the 4x5 concatentation of the five model points in homogeneous coordinates. Simple algebra on $\mathbf{T}$, with the assumption that the face is oriented toward the camera allows full recovery of 6-DOF pose. At this point, the configuration space is shrunk to a small set of configurations corresponding to faces which appear very similar to the target face and which are in poses within a margin of error of ground truth.

Each of these layers performs one or more of the tracking, detection, and recognition functions. In particular, Layers 1-3 perform simple face detection. Layers 3-6 are all capable of tracking. Layers 5 and 6 perform trivial face recognition.

Results: Constructed as above, IFA-based face tracking allows for highly robust tracking at 30Hz. When visual conditions are ideal, tracking proceeds at the top layer, with recovery of complete 6-DOF pose (this is seen in Figure 5, where tracking takes place mostly at Layer 6). As conditions deteriorate (corresponding to dips in Figure 5), tracking continues at 30Hz, but with only partial pose estimation (for example, in Layer 3 the position in 2D is tracked without orientation information). When tracking fails, the system attempts a search for the target, using the same rapid search and track algorithms used to

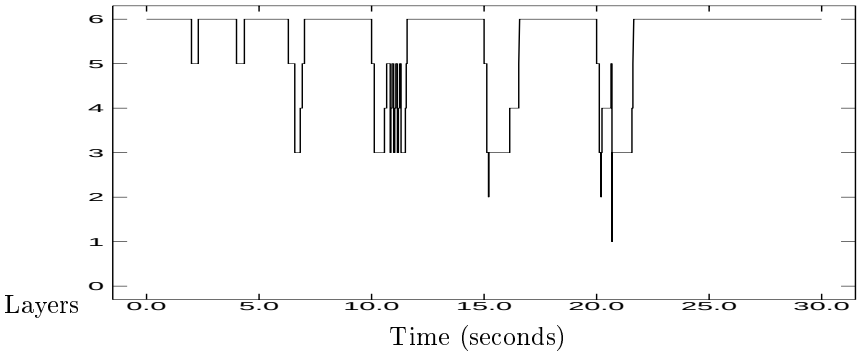


Figure 5: Tracking layer activity during various visual disturbances.

track the target (the rightmost dip to Layer 1 in Figure 5). Recovery typically takes between 66-200msec. The constant pressure of the system to move control upwards balanced with the downward force of a poor visual environment produce an equilibrium where the system applies visual and algorithmic resources to recover as much pose information as possible.

None of the layers of the system are particularly good at their respective tasks: Detection is based on blobs of color, obviously inadequate for accurate face detection, and recognition by straightforward template matching is known to be poor. However, the whole performs far better than the sum of the parts because the framework expects no one layer to perform a single function perfectly; all layers assist those above and below them in the framework. Of course, better detection and recognition algorithms would improve performance provided they are *fast*.

Finally, we mention that the layered structure automatically handles issues of resource management. For example, when the target is moving slowly and visual conditions are good, tracking recovers full pose. As the target moves more quickly and feature trackers are unable to keep up, control falls to lower layers which retain track of the target face using cheaper methods, but which recover only 2D information. More on this point is described in [38].

7 Conclusion

Face tracking is an emerging area within facial image analysis in which the goals of accuracy and robustness must be achieved under the strict constraints of real time. We have surveyed the current state of the art in face tracking, pointed out its successes and failures, and presented one possible implementation based on Incremental Focus of Attention.

Although tracking, *per se*, differs from the tasks of face detection and recognition, robust tracking requires efficient detection and recognition for recovery. By casting detection, recognition, and tracking as a single type of search, these tasks can be unified by a state-based framework such as IFA which dynamically

selects the appropriate algorithm to run based on internal and external state.

IFA is by no means the last word in architectures for robust tracking, and indeed, there are several issues that remain unaddressed by it. IFA itself may benefit from a probabilistic framework with a training stage to determine "softer" transitions, as opposed to the current switching paradigm. Also, there should be mechanisms for robust recovery under less-than-ideal visual conditions. Lastly, adaptation to the visual environment and to particular users could help to optimize the recovery process for a particular situation.

In face tracking overall, relatively little work has been done on tracking faces in views that deviate significantly from the frontal view. Associated problems include the handling of features appearing and disappearing from view, as well as problems of recognition from profiles. Also of interest are algorithms for acquiring and tracking features which vary from person to person (facial hair, glasses, etc.). And finally, as has been stressed throughout, face tracking always benefits from yet more accurate and efficient algorithms for face detection, recognition, feature localization, and feature tracking.

References

[1] V. Bakic and G. Stockman. Real-time tracking of face features. Technical report, Michigan State University, 1998. To appear (note title may differ).

[2] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Proc. Int'l Conf. on Patt. Recog.*, 1996.

[3] S. Birchfield and C. Tomasi. Elliptical head tracking using intensity gradients and color histograms. In *Proc. Computer Vision and Patt. Recog.*, 1998.

[4] M. La Cascia, J. Isidoro, and S. Sclaroff. Head tracking via robust registration in texture map images. In *Proc. Computer Vision and Patt. Recog.*, 1998.

[5] Q. Chen, H. Wu, T. Fukumoto, and M. Yachida. 3D head pose estimation without feature tracking. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 88–93, 1998.

[6] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Understanding*, 53(2):211–218, March 1991.

[7] M. Collobert, R. Feraud, G. Le Tourneur, O. Bernier, J. E. Viallet, Y Mahieux, and D. Collobert. LISTEN: A system for locating and tracking individual speakers. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 283–288, 1996.

[8] A. Colmenarez, R. Lopez, and T. S. Huang. 3D head pose computation from 2D images: Templates versus features. In *Proc. Int'l Conf. on Image Proc.*, 1995.

[9] J. Crowley and F. Berard. Multi-modal tracking of faces for video communication. In *Proc. Computer Vision and Patt. Recog.*, pages 640–645, 1997.

[10] T. Darrell, G. Gordon, J. Woodfill, and M. Harville. A virtual mirror interface using real-time robust face tracking. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 616–621, 1998.

[11] F. de la Torre, S. Gong, and S. McKenna. View alignment with dynamically updated affine tracking. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 510–515, 1998.

[12] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *Proc. Computer Vision and Patt. Recog.*, pages 231–238, 1996.

[13] G. J. Edwards, C. J. Taylor, and T. F. Cootes. Learning to identify and track faces in image sequences. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 260–265, 1998.

[14] E. Elagin, J. Steffens, and H. Neven. Automatic pose estimation system for human faces based on bunch graph matching technology. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 136–141, 1998.

[15] P. Fieguth and D. Terzopoulos. Color-based tracking of heads and other mobile objects at video frame rates. In *Proc. Computer Vision and Patt. Recog.*, pages 21–27, 1997.

[16] A. Gee and R. Cipolla. Fast visual tracking by temporal consensus. *Image and Vision Computing*, 14(2):105–114, March 1996.

[17] G. Hager and P.N. Belhumeur. Occlusion insensitive tracking of image regions with changes in geometry and illumination. Technical Report DCS-TR-1122, Yale University, 1996.

[18] J. Heinzmann and A. Zelinsky. Robust real-time face tracking and gesture recognition. In *Int'l Joint Conf. Artificial Intelligence 97*, pages 1525–1530, 1997.

[19] T. Horprasert, Y. Yacoob, and L. S. Davis. Computing 3-D head orientation from a monocular image sequence. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 242–247, 1996.

[20] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. on Computer Vision*, pages I:343–356, 1996.

[21] ISO MPEG-4 Committee. Coding of audio-visual objects. In *ISO/IEC JTC1/SC29/WG11*, Fribourg, October 1997.

[22] T. S. Jebara and A. Pentland. Parametrized structure from motion for 3D adaptive feedback tracking of faces. In *Proc. Computer Vision and Patt. Recog.*, 1997.

[23] T. Maurer and C. von der Malsburg. Tracking and learning graphs and pose on image sequences of faces. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 176–181, 1996.

[24] Y. Moses, D. Reynard, and A. Blake. Robust real time tracking and classification of facial expressions. In *Proc. Int'l Conf. on Computer Vision*, pages 296–301, 1995.

[25] S. Niyogi and W. T. Freeman. Example-based head tracking. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 374–377, 1996.

[26] N. Oliver, A. Pentland, and F. Berard. LAFTER: Lips and face real-time tracker. In *Proc. Computer Vision and Patt. Recog.*, pages 123–130, 1997.

[27] R. Pappu and P. Beardsley. A qualitative approach to classifying gaze direction. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 160–165, 1998.

[28] E. Petajan and H. P. Graf. Robust face feature analysis for automatic speechreading and character animation. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 357–362, 1996.

[29] G. Potamianos, E. Cosatto, H. P. Graf, and D. B. Roe. Speaker independent audio-visual database for bimodal ASR. In *Proc. Euro. Tutor. and Res. Wkshp. on AV Speech Proc.*, 1997.

[30] R. Rae and H. J. Ritter. Recognition of human head orientation based on artificial neural networks. *IEEE Trans. Neural Networks*, 9(2):257–265, 1998.

[31] Y. Raja, S. J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 228–233, 1998.

[32] A. Saulnier, M.-L. Viaud, and D. Geldreich. Real-time facial analysis and syntehsis chain. In *Proc. Int'l Wkshp on Autom. Face and Gesture Recog.*, pages 86–91, 1995.

[33] K. Sobottka and I. Pitas. Segmentation and tracking of faces in color images. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 236–241, 1996.

[34] R. Stiefelhagen, J. Yang, and A. Waibel. Tracking eyes and monitoring eye gaze. In *Proc. Wkshp on Perceptual UI*, Banff, Canada, 1997.

[35] K. Toyama. Radial spanning for fast blob detection. In *Proc. Int'l Conf. on Comp. Vision, Patt. Recog., and Image Proc.*, 1998.

[36] K. Toyama. Robust face pose tracking with Incremental Focus of Attention. In *Proc. Nimes 98: Complex Syst., Int. Syst., and Interfaces*, 1998.

[37] K. Toyama and G. Hager. If at first you don't succeed... In *Proc. AAAI*, pages 3–9, Providence, RI, 1997.

[38] K. Toyama and G. Hager. Incremental focus of attention for robust vision-based tracking. *Int'l J. of Computer Vision*, 1999.

[39] M. Turk. Visual interaction with lifelike characters. In *Proc. Automatic Face and Gesture Recognition*, 1996.

[40] C. S. Wiles, A. Maki, N. Matsuda, and M. Watanabe. Hyper-Patches for 3D model acquisition and tracking. In *Proc. Computer Vision and Patt. Recog.*, pages 1074–1080, 1997.

[41] M. Xu and T. Akatsuka. Detecting head pose from stereo image sequence for active face recognition. In *Proc. Int'l Conf. on Autom. Face and Gesture Recog.*, pages 82–87, 1998.