Automatic Tracking of Face Sequences in MPEG Video

Yunlong Zhao and Tat-Seng Chua
School of Computing
National University of Singapore, Singapore 117543
E-mail: {zhaoyl, chuats}@comp.nus.edu.sg

Abstract

Human faces are commonly found in video streams and provide useful information for video content analysis. This paper presents a robust face tracking system to extract multiple face sequences from MPEG video without human intervention. Specifically, a view-based DCT-domain face detection algorithm is first applied periodically to capture mostly frontal and slight slanting faces of variable sizes and locations. The face tracker then searches the target faces in local areas across frames in both the forward and backward directions. The tracking combines color histogram matching and skin-color adaptation to provide robust tracking. This paper focuses on developing effective techniques that exploits the features in DCT domain and the characteristics of video compression standards like MPEGs. The effectiveness of the algorithm is demonstrated using a range of videos obtained from multiple sources like the news and movies.

1 Introduction

We are faced with an increasing amount of video data. In order to support the efficient management and retrieval of video information, we need to model and index these video data properly based on their characteristics and contents. One important content element is the human face which can be used to derive most essential semantics of video. A sequence of frames with the face of a person can help to interpret his/her behavior in the video. Usually, they are more meaningful than other low level visual features, such as the color histograms. Furthermore, human faces are well defined objects with distinct visual and structural features. Thus it is possible to develop effective algorithm to detect and track faces automatically in video. A possible application is to support the strata-based digital video modeling and retrieval system [6].

Although human faces have distinct visual and structural features, detecting and tracking faces in general video is still quite difficult as compared to applications for human-computer interactions or surveillance. The difficulties arise from the unrestricted nature of faces and the environments in which they occur in video. The difficulties include:

- There is no restriction on the appearances, sizes, locations and poses of faces in video.
- New faces may enter or leave the scene, or they may be occluded.
- The background is often cluttered and complex.
- The face, camera and background can be in motion simultaneously.
- Multiple faces close to each other will cause ambiguities in tracking.

These difficulties give problems to those methods that depend on specific visual or motion cues for face detection and tracking. In addition, in order to process the large amount of video data, efficient algorithms are needed to save computation cost.

Previous works on object (face) detection and tracking mostly work on uncompressed video. Recently, methods have been proposed to fulfill the tasks in compressed video [9, 11]. Schonfeld and Lelescu [9] utilized the motion vectors to approximate region motion, which roughly corresponds to the object motion. Wang et al. [11] performed face detection in I-frames, and approximated the face locations in P-frames by projecting the motion vectors back into the face regions in the previous I- or P-frames. All the locations are treated as noisy observations and fed into the Kalman filter. Normally, approximating the object motion using motion vector is inadequate and prone to error. The power of the systems tends to be limited by relying on motion vectors for object tracking.

Taking all the above issues into considerations, this paper focuses on developing an efficient and robust algorithm to detect and track multiple moving faces directly from compressed video with minimal decompression. We do not assume any prior knowledge of scenes

Proceedings of the Computer Graphics International (CGI'03)
1530-1052/03 $17.00 © 2003 IEEE

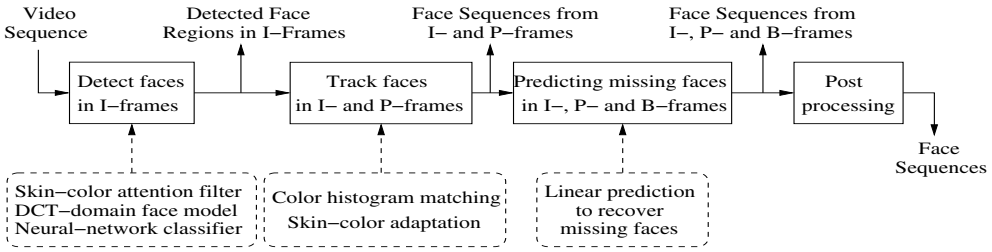


Figure 1. System diagram of the extraction and tracking of face sequences.

and camera positions. The paper is organized as follows. Section 2 gives an overview of our approach. Section 3 introduces our method to detect human faces in video, which is used to initialize the face tracking algorithm. Section 4 describes the framework for extracting multiple face sequences from MPEG video by tracking and interpolation. Section 5 presents experimental results. Finally, we conclude the paper in Section 6.

2 Overview of Our Approach

Given a MPEG video clip as input, we perform the face detection and tracking selectively on the I-, P- and B-frames according to their frame types. Figure 1 gives the overview of our approach. It consists of multiple stages, including face detection, face tracking, face region prediction and post-processing.

First, an attention filter based on multiple Gaussian skin-color model and a DCT-domain face detector [3] is employed to detect face regions in the I-frames.

Next, for each located face region, we track it forward and backward in the I- and P-frames without considering the in-between B frames. Besides the efficiency considerations, this is to compensate for the variations of face patterns due to changes of pose, position and scale. The tracking process is accomplished by a hypothesize-and-test procedure [4, 1] based on the combination of color histogram matching and skin-color adaptation. The color histogram matching is performed at variable scales and locations in local areas. For each face sequence, a specific Gaussian model is trained and updated to adapt to possible changes in skin colors. In the tracking process, the reference face is updated adaptively according to a confidence measure based on the skin-color model, which decides if the tracking result should be accepted or rejected.

The above two steps produce a series of spatio-temporal positions of face regions in the I- and P-frames. However, some corresponding faces in certain I- and P-frames may be missing because of the possible changes in face pose and occlusions. We use those detected faces as keypoints and perform linear prediction or interpolation to estimate the parameters of missing faces. The corresponding faces in the B-frames are also interpolated from the detected faces in the I- and P-frames. This approach is reasonable because of the high coherence between video frames. Finally, we link partial face sequences with similar faces together to form the final face sequences.

3 Face Detection in DCT Domain

In the first stage, we employ a method to detect frontal and slightly slanting faces directly in MPEG video. It performs the two-step process consists of a skin-color classification, and gradient energy representation with neural-network classification [3]. Related techniques are also developed to ensure that all the processes are done in compressed domain.

3.1 Skin Color Representation

Skin colors are often employed for face detection and tracking as they provide valuable hints to the presence of human faces in video. In contrast to some geometric features, they are insensitive to face orientation, occlusion, or even the presence of glasses and sideburns [8]. With proper selection of the color space and normalization, skin color differences among people can be reduced and the distribution can be characterized by a multivariate normal distribution under a certain lighting condition [12, 8].

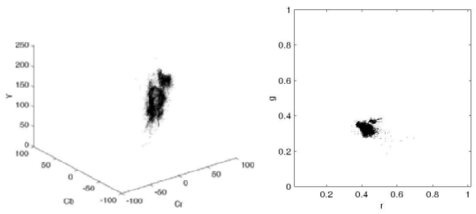
We choose the normalized (r, g) to represent a color pixel where $r = R/(R+G+B)$, $g = G/(R+G+B)$. The sample skin colors are shown in $YCbCr$ space and rg plane respectively in Figure 2. We perform the clustering on the rg samples using K-means algorithm. The skin-color model is a mixture of Gaussians with mean μ_i and covariance matrix $\mathbf{K}_i$, where $i = 1, 2, \cdots, N$ [10]. For simplicity, we use multiple Mahalanobis classifiers to label a color $\mathbf{x} = (r, g)^T$ as skin or nonskin color. It is defined as follows,

$$d_i(\mathbf{x}) = (\mathbf{x} - \mu_i)^T \mathbf{K}_i^{-1} (\mathbf{x} - \mu_i) \qquad (1)$$

For $\mathbf{x}$, if at least one $d_i(\mathbf{x}) < H_i$ is satisfied, it is classified as a skin color. Otherwise, it is classified as a non-skin color. H_i is the respective threshold.

3.2 Face Region Detection in DCT Domain

The face detection consists of two stages. First, we use skin-color classification to segment the possible face



(a) (b)

Figure 2. Distribution of sample skin colors. (a) in YCbCr space and (b) in normalized rg plane.

(a) (b) (c)

Figure 3. Example of face region detection in DCT domain.

regions. This is to quickly narrow down the regions for further analysis. In each I frame, we classify the blocks into two categories, with skin colors or otherwise. The DC coefficient of each block is used as the representative color for the whole block and Equation 1 is used for the evaluation with the predefined multiple Gaussian model. We group the skin blocks into large skin regions by connected component analysis. Finally, we get a number of nonoverlapping skin-color regions bounded by rectangles. These rectangles are candidate face regions.

It is well known that color information alone is not robust enough for detecting face in video. Other objects and background may possess colors similar to human skin. Thus additional information is needed to remove the false detections from the first step. In this system, we employ a model-based face detector to remove the false detection and decide the exact face size and location in the candidate region. The face detector is based on DCT-domain representation and neural-network classification. More details about the DCT-domain face detection algorithm can be found in [3].

An example of face region detection is shown in Figure 3. Figure 3(a) shows the original frame. Figure 3(b) gives the potential face regions detected by skin color classification. The final result after applying DCT-domain face model is shown in Figure 3(c).

4 Face Tracking in MPEG Video

Once a face region is detected in an I-frame, we start to track it forward and backward in adjacent frames. There are at least two reasons to perform face tracking. First, face tracking helps to recover those faces missed by the view-based face detector introduced above. Second, face tracking determines the correspondence between the face regions in different frames. Here, we do not care about the 3D structure of the face. Our goal is to use the 2D patch matching and frame coherence to capture the location and scale of the face region in new frames.

In the process of face detection and tracking, there are two possible cases for the correspondences between the face regions detected. First, a face may be picked up simultaneously by the detection module and the tracking module, as we perform both face detection and face tracking in the I-frames. We check the relative positions and overlapping status of the two results to decide if they correspond to the same face. Second, the face that has been lost during tracking for a number of frames is detected again. The newly detected face region will be link to the interrupted face sequence by color histogram match with the stored face templates. Cues like the duration of loss tracking, sizes and positions of the faces are also evaluated to prune errors.

In the following discussions, we represent a face sequence in spatio-temporal format as:

$$F = \{B_t^i, H_t^i, G_t^i\} \;\; t = 0, 1, \cdots, T, \;\; i = 0, 1, \cdot, N(t) \quad (2)$$

$B_t^i = (x_t^i, y_t^i, w_t^i, h_t^i)$, where (x_t^i, y_t^i) and (w_t^i, h_t^i) denotes the center and size of the rectangle box bounding the face region i at time t, respectively.
H_t^i is color histogram that describes the color distribution of the face region.
G_t^i is a Gaussian representing the skin-color distribution in the region.
$N(t)$ is the number of face regions present at time t.

4.1 Searching for the best match

The face tracking task is to update the state by finding a region in the candidate frame, whose location, size and visual features best match that of the target face. To accomplish this, a hypothesize-and-test procedure is employed [4, 1]. The best estimation for the position and size of the face corresponds to a point $\mathbf{s}^*$ in the search space S, which maximizes $d(\mathbf{s})$:

$$\mathbf{s}^* = arg \max_{\mathbf{s} \in S} d(\mathbf{s}) \quad (3)$$

We employ the color histogram in the Luv color space to model color characteristic within the face region [2]. It is stable and insensitive to scale changes. The matching score between the face template H_{t-1} and the candidate region H_t is defined as the color histogram intersection,

$$d(\mathbf{s}) = \sum_{i=1}^{N} \{1 - \frac{|H_{t-1}(i) - H_t(i)|}{max[H_{t-1}(i), H_t(i)]}\} * H_{t-1}(i) \quad (4)$$

where $N = 176$ is the number of color bins used.

Besides measures to determine and verify the goodness of the match, the success of the tracking task is largely dependent on the scale of the search space, especially the changing face sizes.

Given two previous reference face regions B_{t-1} and B_{t-2} in a face sequence, we first predict a search region B_t^p at time t, in which we look for the best match to B_{t-1}. Due to the temporal continuity of the face region, the prediction of the search region can be restricted to a local area. For simplicity, we employ linear prediction to estimate the position of the search region as follows,

$$x_t^p = x_{t-1} + \Delta x_{t-1}, y_t^p = y_{t-1} + \Delta y_{t-1} \quad (5)$$

where $\Delta x_{t-1} = x_{t-1} - x_{t-2}$, and $\Delta y_{t-1} = y_{t-1} - y_{t-2}$. In particular, if there is only one reference face region B_{t-1} available at time $(t-1)$ as the face tracker is just initialized, we simply set $x_t^p = x_{t-1}$ and $y_t^p = y_{t-1}$.

As for the size of the search region, no prediction is performed. Instead, we assume that face changes position and scale only moderately within the time interval ($< 1/8$ second). Thus we simply set the size of the search region to be 50% larger than the reference face region B_{t-1}: $w_t^s = 1.5w_{t-1}$ and $h_t^s = 1.5h_{t-1}$.

The search space S is then defined as a number of candidate regions denoted by $\mathbf{s}$ with coordinates (x, y, w, h) around the predicted location (x_t^p, y_t^p) within the search region.

$$\begin{aligned} S \quad = \quad & \{\mathbf{s} : |x - x_t^p| \leq \Delta x \cdot m_x, |y - y_t^p| \leq \Delta y \cdot m_y, \\ & |w - w_t^p| \leq \Delta w \cdot n_w, |h - h_t^p| \leq \Delta h \cdot n_h\} \end{aligned} \quad (6)$$

where Δw and Δh control the size changes of the candidate regions. In our implementation, we set $\Delta w = 0.05w_{t-1}$, $\Delta h = 0.05h_{t-1}$, $n_w = 2$ and $n_h = 2$. This implies that the size of the candidate region is changing from $0.9w_{t-1} \times 0.9h_{t-1}$ to $1.1w_{t-1} \times 1.1h_{t-1}$. Although assuming fixed face size can provide satisfactory results in applications like human computer interactions [4], it is not satisfactory for general video. The above search process aims to accommodate and constrain the face size changes. Δx and Δy are the step size between the adjacent candidate regions.

4.2 Verification of the matching result

During the tracking of a face across frames, the 3D motion of human and the changing illumination conditions will give rise to changes in the 2D image pattern of target. It is hard for a tracker with only a fixed face template to handle this as the residual of matching may increase sharply. A common solution to this problem is to update the face template accordingly to accommodate the latest tracking results. However, because of the possible error accumulated in the face template in the tracking process, the face tracker may lose the targets gradually. Meanwhile, the tracking may be failed due to occlusion or disappearance of the face region. Thus, given a matching score for a face template, we need a confidence measurement to decide if the match should be accept or rejected. To this end, we have designed a measure based on the skin color information pertaining to a specific face sequence.

The measurement, $L(t)$, of the region tracked at time t is given by:

$$L(t) = \frac{1}{N} \sum_{\mathbf{x} \in R(t)} (\mathbf{x} - \mu_t)^T \mathbf{K}_t^{-1} (\mathbf{x} - \mu_t) \quad (7)$$

where N is the total number of pixels in the region, and μ_t and $\mathbf{K}$ are the mean and covariance matrix of the skin-color model of the face sequence respectively.

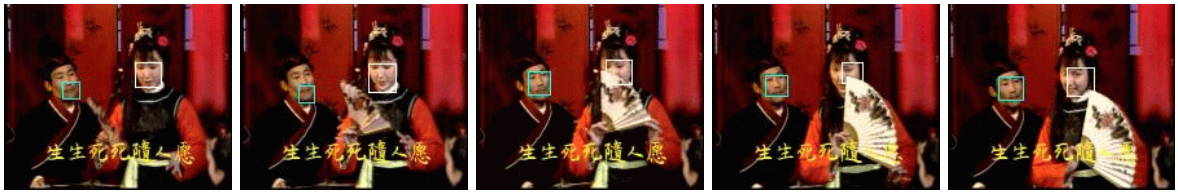
If the face tracker loses the face, or the face is occluded, there is often a sudden drop in the value of $L(t)$. Meanwhile, if the tracker suffers from the errors accumulated across frames and moves away from the actual face region, the value of $L(t)$ will also drop gradually. To deal with the above situations, we need a dynamic threshold. Here we adapt the temporal filter as introduced in [8]. The mean (m) and standard deviation (δ) of $L(t)$ are computed for the most recent T successful trackings. A threshold $(m - k\delta)$ is set to detect the failure of the tracking. If $L(t) > (m - k\delta)$, the match will be rejected and the face is considered lost. When a face is lost, we suspend the adaptation and preserve the old template, until the face is re-captured with sufficiently high confidence. We also employ heuristics to help determine whether the face region is occluded or disappear. One example is to select the body part as reference. In our experiments, $T = 15$ and $k = 2.2$.

4.3 Modeling the skin colors of a face sequence

Other than occlusion of faces and errors in the actual tracking process, the other source of error is that the skin-color distribution of the face changes during tracking because of environmental reasons. To tackle this problem, we build a specific Gaussian model to capture the skin-color distribution for the faces during tracking. Although the number of skin-color samples is limited in a single frame, the parameter estimation can be continuously improved as more samples incoming along the progress of tracking. The estimations gained in the previous frames are carried to the estimations in the following frames [10]. Thus, the model adapts to the characteristics of a particular face sequence, and changes of the skin colors. This helps to refine the predefined skin-color model and ensure the stability of the tracking process.

Frame #165 Frame #178 Frame #182 Frame #185 Frame #193

Figure 4. Sample frames for video 1 with face detection and tracking results.



Frame #910 Frame #915 Frame #925 Frame #945 Frame #965

Figure 5. Sample frames for video 2 with face detection and tracking results.

Frame #216 Frame #231 Frame #256 Frame #266 Frame #277

Figure 6. Sample frames for video 3 with face detection and tracking results.

The learning samples $\mathbf{x}$ is a sequence of skin-color vectors extracted at different time t, $X^{(t)} = \{\mathbf{x}(i)\}, i = 1, 2, \cdots, N^{(t)}$. We can estimate the mean $\mu^{(t)}$ and covariance matrix $\mathbf{K}^{(t)}$ of the set of pixels, $\mathbf{X}^{(t)}$, at time t [10].

Taking the previous estimations into account, the overall mean μ_t and covariance matrix $\mathbf{K}_t$ of the samples up to time t can be derived from the corresponding parameters of the samples up to time $t-1$ and the present estimation $\mu^{(t)}$ and $\mathbf{K}^{(t)}$ at time t:

$$\mu_t = \frac{1}{\sum_{j=1}^{t} N^{(j)}} \sum_{j=1}^{t} \sum_{\mathbf{x} \in X^{(j)}} \mathbf{x} = \mu_{t-1} + \alpha_t[\mu^{(t)} - \mu_{t-1}]$$

$$\begin{aligned} \mathbf{K}_t &= \frac{1}{\sum_{j=1}^{t} N^{(j)}} \sum_{j=1}^{t} \sum_{\mathbf{x} \in X^{(j)}} \mathbf{x}\mathbf{x}^T - \mu_t \mu_t^T \\ &= \mathbf{K}_{t-1} + \alpha_t[\mathbf{K}^{(t)} - \mathbf{K}_{t-1}] \\ &\quad + \alpha_t(1-\alpha_t)\Delta\mu\Delta\mu^T \end{aligned}$$

where $\Delta\mu = \mu^{(t)} - \mu_{t-1}$. Empirically, we set $\alpha = 0.5$ and the latest estimation has the strongest influence.

4.4 Recovery of the misses in I-, P- and B-frames

After face detection and tracking, we have a set of successive spatio-temporal positions of the face regions in the I- and P-frames. These face regions can be used as keypoints to recover the positions and sizes of the missing faces in certain I- and P-frames by prediction or interpolation. We assume that the trajectory of the face regions following a first- or second-order piecewise approximation along time [1, 5]. Normally, the prediction error tends to be small with the assumption that the face will not move too far away as the video frame rate is around 30 frames/second. For simplicity, linear prediction and interpolation are used.

Similarly, the face regions in I- and P-frames are used as keypoints to recover the missing parameters of faces in B-frames by interpolation, as no face detection or tracking operations has been performed in these frames.

5 Experimental Results

The proposed face detection and tracking algorithms were tested on a PIII Dell PC (Linux installed) with video streams from multiple sources like news and movies. We selected three examples with different characteristics to demonstrate the effectiveness of our method. They are all from movies in MPEG-1 format with frame size of 352×288 pixels. We used the MPEG developing classes designed by Li and Sethi [7] for partially decoding the video into frames of DCT blocks.

The first video has 418 frames with 27 I-frames, 108 P-frames and 270 B-frame. A single face is present

from the beginning to the end. It takes about 45 seconds to detect and extract the face sequence. This is a bit slow as many faces in the clip are rather big (around 145 pixels). Results of several frames from the video clip are shown in Figure 4. The face turns away from facing the camera from frame 178 to frame 193. But the system successfully tracks it by prediction and interpolation. The template and color model adaptation is suspended during this period.

The second video has 828 frames with 70 I-frames, 206 P-frames and 552 B-frames. Two faces are present from the beginning to the end. It takes about 52 seconds to detect and extract the face sequences. Sample results are shown in Figure 5. Bounding boxes with different colors are used to discriminate multiple face sequences. One face is occluded in some frames. But the system still can track it steadily.

The third video has 454 frames with 38 I-frames, 114 P-frames and 302 B-frames. The number of face present is in changing, with someone entering and other leaving the scene. Occlusions also occur in some frames. The maximum number of faces present is three. It takes about 50 seconds to detect and extract the face sequences. Sample results are shown in Figure 6. The situation is quite complex in this video clip. But the system is still able to link the corresponding faces and presents satisfactory results.

The above examples demonstrate the effectiveness of the face detection and tracking algorithms for video clips from multiple sources. It can handle the situations like occlusions, misses in detection and out of plane rotation, etc. The periodical face detection ensures the recovery from errors and tracking of multiple faces. The backward tracking further enhances the ability of the system to recover from errors. The scheme of performing linear prediction for the I- and P-frames and interpolation for the B-frames are also found to be effective.

The speed of processing is variable and dependent on factors such as the number and size of faces present in the scene. Because the pre-defined skin color model is not optimized with respect to a particular video source, the candidate region achieved by it tends to be larger than the real size. This causes more computation in the succeeding operations for face detection. Nevertheless, the system is able to run at around 25 frames/second for simple cases such as the video clips of anchor persons from the MPEG-7 test data set.

6 Conclusion and Future Work

This paper presents a system to extract, track and group face sequences in MPEG video automatically. It can be used for digital video modeling, retrieval and browsing. Various possible situations regarding face detection and tracking in video are considered to provide efficient and feasible solutions to the problem. Experiments on videos from multiple sources are used to demonstrate the effectiveness of the algorithms. Future work will focus on testing on more video data, enhancing the scheme of face sequence extraction, and optimizing the skin color modeling. In addition to visual features, cues like speech, closed caption, text caption in video will be investigated to improve the face related applications.

References

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 232–237, 1998.

[2] T.-S. Chua and C. Chu. Color-based pseudo-object for image retrieval with relevance feedback. In *Intl. Conf. on Advanced Multimedia Content Processing*, pages 148–162, 1998.

[3] T.-S. Chua, Y. Zhao, and M. Kankanhalli. Detection of human faces in a compressed domain for video stratification. *The Visual Computer*, 18(2):121–133, 2002.

[4] P. Fieguth and D. Terzopoulos. Color-based tracking of heads and other mobile objects at video frame rates. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 21–27, 1997.

[5] S. Jeannin and A. Divakaran. Mpeg-7 visual motion descriptors. *IEEE Trans. on Circuits and Systems for Video Technology*, 11(6):720–724, June 2001.

[6] M. S. Kankanhalli and T.-S. Chua. Video modeling using strata-based annotation. *IEEE Multimedia*, 7(1):68–74, Jan. 2000.

[7] D. Li and I. K. Sethi. Mdc: A software tool for developing MPEG applications. In *Proc. of IEEE Intl. Conf. Multimedia Computing and Systems*, volume 1, pages 445–450, 1999.

[8] Y. Raja, S. J. McKenna, and S. Gong. Colour model selection and adaptation in dynamic scenes. In *Proc. of European Conf. on Computer Vision*, pages 460–474, 1998.

[9] D. Schonfeld and D. Lelescu. Vortex: Video retrieval and tracking from compressed multimedia databases. In *Proc. of IEEE Intl. Conf. on Image Processing*, volume 3, pages 123–127, 1998.

[10] J. Schürmann. *Pattern classification: a unified view of statistical and neural approaches*. Jonh Wiley & Sons, INC., 1996.

[11] H. Wang, H. S. Stone, and S.-F. Chang. Facetrack: tracking and summarizing faces from compressed video. In *SPIE Multimedia Storage and Archiving System IV*, pages 19–22, 1999.

[12] J. Yang, W. Lu, and A. Waibel. Skin-color modeling and adaptation. Technical Report CMU-CS-97-146, School of Computer Science, Carnegie Mellon University, May 1997.