Name-It: Naming and Detecting Faces in News Videos

Shin'ichi Satoh
National Center for Science Information Systems

Yuichi Nakamura
University of Tsukuba

Takeo Kanade
Carnegie Mellon University

We developed Name-It, a system that associates faces and names in news videos. It processes information from the videos and can infer possible name candidates for a given face or locate a face in news videos by name. To accomplish this task, the system takes a multimodal video analysis approach: face sequence extraction and similarity evaluation from videos, name extraction from transcripts, and video-caption recognition.

The Name-It system[1,2] associates names and faces in news videos. Assume that we're watching a TV news program. When persons we don't know appear in the news video, we can eventually identify most of them by watching only the video. To do this, we detect faces from a news video, locate names in the sound track, and then associate each face to the correct name. For face-name association, we use as many hints as possible based on structure, context, and meaning of the news video. We don't need any additional knowledge such as newspapers containing descriptions of the persons or biographical dictionaries with pictures. Similarly, Name-It can associate faces in news videos with their right names without using an a priori face-name association set. In other words, Name-It extracts face-name correspondences only from news videos.

Name-It takes a multimodal approach to accomplish this task. For example, it uses several information sources available from news videos—image sequences, transcripts, and video captions. Name-It detects face sequences from image sequences and extracts name candidates from transcripts. It's possible to obtain transcripts from audio tracks by using the proper speech recognition technique with an allowance for recognition errors. However, most news broadcasts in the US already have closed captions. (In the near future, the worldwide trend will be for broadcasts to feature closed captions.) Thus we use closed-caption texts as transcripts for news videos. In addition, we employ video-caption detection and recognition. We used "CNN Headline News" as our primary source of news for our experiments.

Given image sequences, transcripts, and video captions as information sources, Name-It associates extracted faces with extracted name candidates using the correlation of their timing information and face similarity information. Video captions are also taken into account as supplementary information. To associate faces and names, Name-It integrates several advanced image processing and natural-language processing techniques—face sequence extraction and similarity evaluation from videos, name extraction from transcripts, and video-caption recognition. Although these technologies aren't always highly accurate, integrating these results will help the system achieve more accurate output.

With respect to face-name association, the Piction system[3] works similarly to Name-It. Piction identifies faces within a given captioned newspaper photograph by extracting faces from the photograph and analyzing the caption to obtain geometric constraints among faces. The system then labels each face with a name. A drawback of Piction is that face location information is assumed to be described in captions—for example, "top row, from left, are Michael, Brian ...." On the other hand, Name-It doesn't assume such a description. Instead, while Piction deals with one photograph and caption at a time, Name-It processes many videos—including many news topics—to collect a fraction of a hint from each video fragment to infer face-name association. In doing this, Name-It uses face similarity while Piction doesn't.

To realize Name-It, further analysis of video semantic content proves necessary. State-of-the-art video analysis technologies automatically extract video structure information.[4,5] Typically, once a video is given as a target, it's decomposed into segments or shots. These shots are then classified based on the video's structure. This process employs several techniques, such as cut detection, color histogram calculation and comparison, camera motion analysis, motion segmentation, and so on. Among them, cut detection and color his-

1070-986X/99/$10.00 © 1999 IEEE

togram calculation and comparison are incorporated into Name-It to provide hints for video content analysis.

Since Name-It primarily handles face information, face detection and face similarity evaluation play essential roles. Much research has targeted face detection and matching (for an extensive survey, see Chellappa, Wilson, and Sirohey[6]). It's noteworthy to contrast face identification and Name-It. In face identification, a face (category) set for comparison with given faces is given a priori. Although Name-It primarily associates faces and names in videos, it automatically generates a face-name association set from given videos that may even be used for face identification.

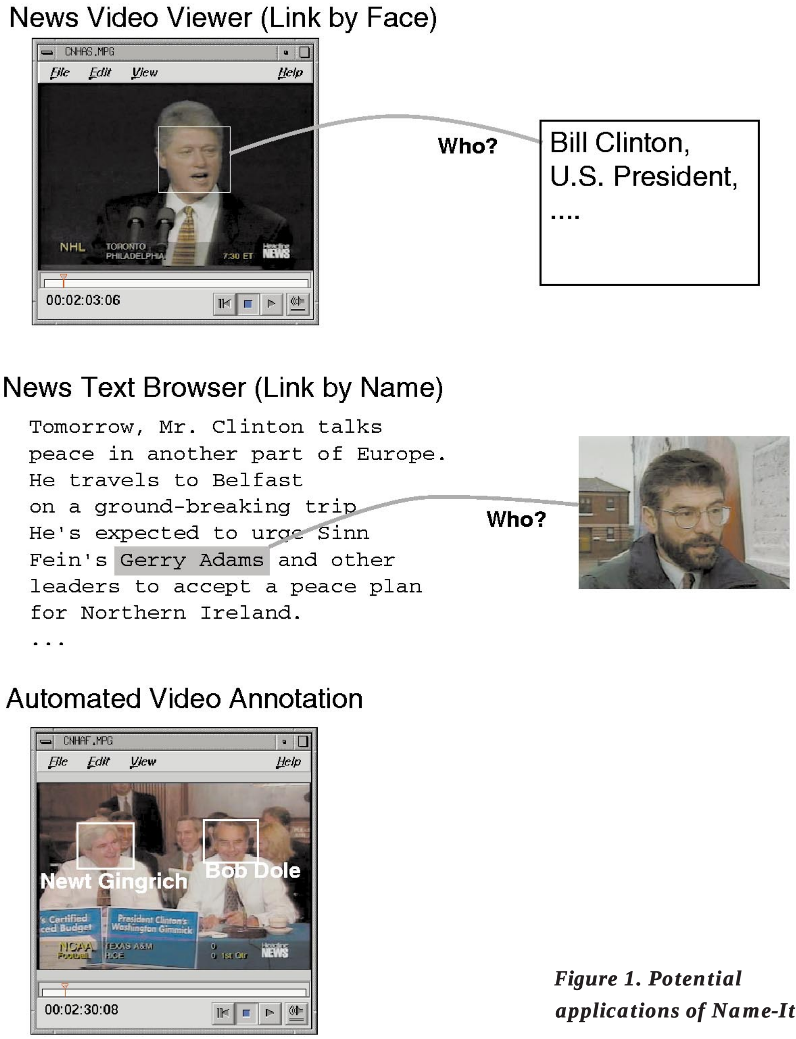
By providing face-name association, Name-It performs "individual detection" rather than mere "face detection," because associated faces and names correspond to certain individuals who are of interest in news video topics. As a result, Name-It enables several potential applications (see Figure 1), including

- a news video viewer that interactively provides a personal description of the displayed face,
- a news text browser that gives facial information in response to names, and
- an automated video annotation generator for faces.

Overview of Name-It

Typical news video consists of several topics, each having a corresponding person or persons of interest. Figure 2 shows the typical structure of a news topic in which US President Bill Clinton is the person of interest.

We set the primary goal of Name-It as associating faces and names of persons of interest in news video topics. To achieve this goal, we employ the process shown in Figure 3, next page. Name-It must extract faces from image sequences and names from transcripts—both the faces and names correspond to persons of interest. However, these tasks are hard to accomplish. Faces of persons of interest tend to appear under several conditions such as frontal views or close-ups, or they're on screen for a long duration. But faces

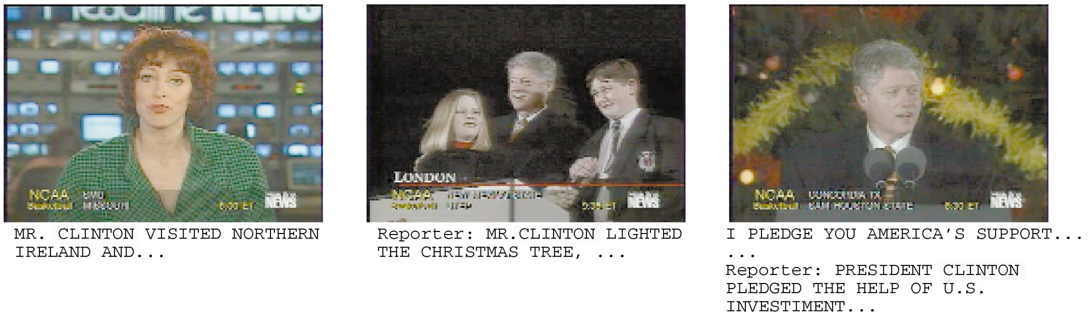

News Video Viewer (Link by Face)

News Text Browser (Link by Name)

```
Tomorrow, Mr. Clinton talks
peace in another part of Europe.
He travels to Belfast
on a ground-breaking trip
He's expected to urge Sinn
Fein's Gerry Adams and other
leaders to accept a peace plan
for Northern Ireland.
...
```

Automated Video Annotation

Figure 1. Potential applications of Name-It.


MR. CLINTON VISITED NORTHERN IRELAND AND...

Reporter: MR.CLINTON LIGHTED THE CHRISTMAS TREE, ...

I PLEDGE YOU AMERICA'S SUPPORT... ... Reporter: PRESIDENT CLINTON PLEDGED THE HELP OF U.S. INVESTIMENT...

Figure 2. Typical composition of a news topic.

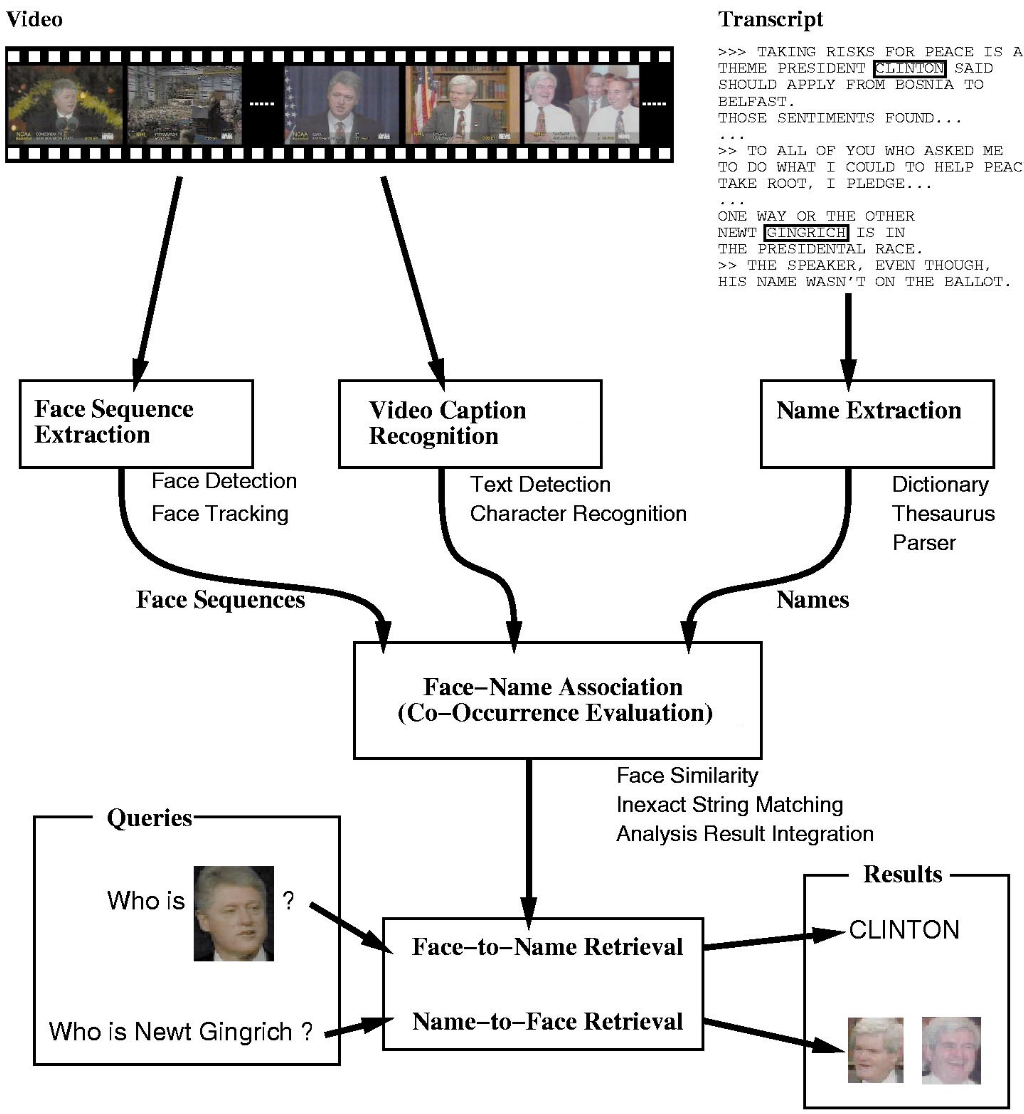

Video
Transcript
>>> TAKING RISKS FOR PEACE IS A THEME PRESIDENT CLINTON SAID SHOULD APPLY FROM BOSNIA TO BELFAST.
THOSE SENTIMENTS FOUND...
...
>> TO ALL OF YOU WHO ASKED ME TO DO WHAT I COULD TO HELP PEACE TAKE ROOT, I PLEDGE...
...
ONE WAY OR THE OTHER NEWT GINGRICH IS IN THE PRESIDENTAL RACE.
>> THE SPEAKER, EVEN THOUGH, HIS NAME WASN'T ON THE BALLOT.
Face Sequence Extraction
Face Detection
Face Tracking
Video Caption Recognition
Text Detection
Character Recognition
Name Extraction
Dictionary
Thesaurus
Parser
Face Sequences
Names
Face–Name Association (Co–Occurrence Evaluation)
Face Similarity
Inexact String Matching
Analysis Result Integration
Queries
Who is ?
Who is Newt Gingrich ?
Face–to–Name Retrieval
Name–to–Face Retrieval
Results
CLINTON

Figure 3. The Name-It process. First, Name-It extracts faces from video and names from transcripts. Then it associates the names and faces.

that meet these conditions don't always correspond to persons of interest. That is, there's no perfect method to extract faces of persons of interest by image-sequence analysis alone. Meanwhile, extracting names of persons of interest requires an in-depth semantic analysis of the transcript. Several studies reported at the Message Understanding Conference[7] achieved sufficient accuracy in selecting all names from text. However, selecting names of only persons of interest proved a much harder problem. Therefore, we extract faces and names likely to correspond to persons of interest. The system employs face detection and tracking to extract face sequences and natural-language processing techniques using a dictionary, thesaurus, and parser to locate names in transcripts.

Given extracted faces and names, Name-It associates those that correspond to one another. Since transcripts don't necessarily give explanations of videos, no straightforward method exists for associating faces in videos and names in transcripts. However, by observing the typical news video composition given in Figure 2, we can assume that a corresponding face and name are likely to coincide and may be an associated face-name pair. However, potential difficulties exist in associating faces and names: the lack of necessary faces or names and possible multiple correspondences of faces and names. For example, even if the system successfully extracts a person of interest's face in a topic, it might not find the correct name that coincides with that face.

As an example of multiple correspondence,

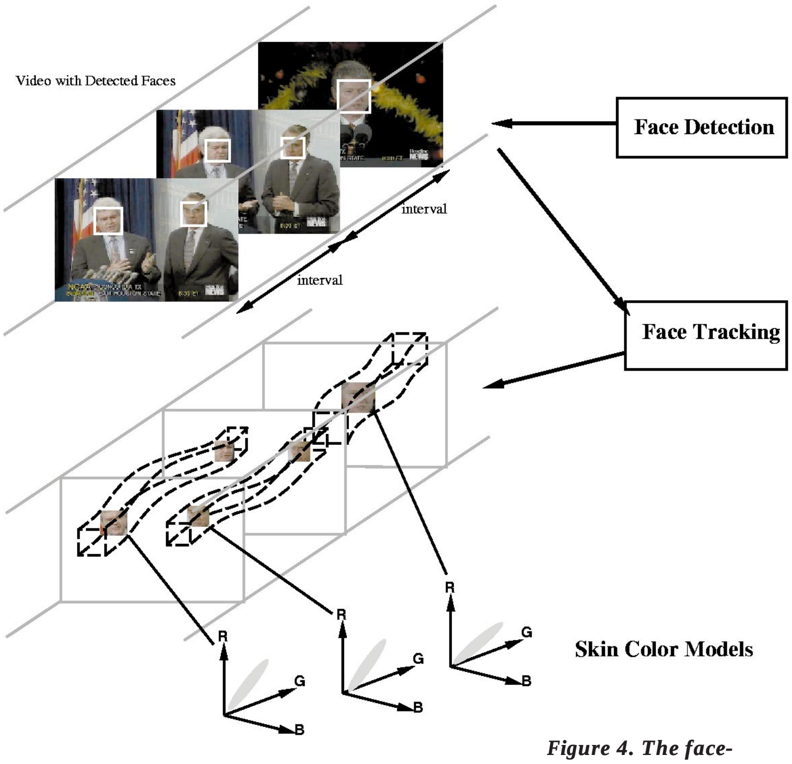


Figure 4. The face-tracking process, which involves face detection, skin-color model extraction, and skin-color region tracking.

assume that topic A is about Clinton and former US Senator Robert Dole, and topic B is about Clinton and former House Speaker Newt Gingrich. The system can't decide whether Clinton's face shown in topic A corresponds to name "Clinton" or "Dole" or whether Clinton's face shown in topic B corresponds to name "Clinton" or "Gingrich." To compensate for this drawback, Name-It gives priority to a face-name pair that coincides more frequently and outputs the pair as an associated face-name pair. Obviously, face similarity is required to evaluate face-name association (for example, to match the faces in topic A and topic B). Thus the system regards these faces as identical and can infer that the face coincides in more topics with the name "Clinton" than with others ("Dole" or "Gingrich"). Evaluating face similarity may also resolve the problem of lack of faces or names. Even if a face doesn't coincide with the correct name, we expect other faces identical to this face in other topics to coincide with the correct name.

The system also employs video-caption recognition to obtain face-name association. Video captions are superimposed text on video frames, therefore representing literal information. The captions are directly attached to image sequences and give an explanation of the video. In many cases, a video caption is attached to a face and usually represents a person's name. Thus, video-caption recognition provides rich information for face-name association. However, because video captions don't necessarily appear for all faces of persons of interest, Name-It uses the video captions as supplements to the transcripts. For example, some faces appear like persons of interest in a news program, but aren't mentioned in the transcripts. Instead, their names often show up in the video captions. To achieve video-caption recognition, we use text detection and character-recognition techniques (see Figure 3).

Finally, results obtained by these techniques should be integrated to provide face-name association. As a unified measurement integrating multimodal analysis, we use a co-occurrence factor, which represents a likelihood factor that a face and a name correspond to each other. This integration should give better face-name association results, even though the results of analyses are imperfect. As shown in Figure 3, to compensate for the problems of the lack of faces or names and multiple correspondence of faces and names, the system employs face similarity to evaluate the co-occurrence factor. Since character recognition for video captions can't be perfect because of the poor quality of video images, it's compensated for by an inexact string match method. As a result, although each analysis may not discriminate faces or names of persons of interest in topics, association results may eventually correspond to face-name pairs of persons of interest in topics.

Face information extraction

Here we describe extraction of those faces that might correspond to persons of interest in topics. We first employ face detection and tracking to detect face sequences in videos. Then, using an eigenface-based method, we evaluate face similarity. To enhance the face similarity evaluation, we select the most frontal view of a detected face sequence and use the selected face for the eigenface method. Finally, given videos as input, the system outputs a two-tuple list: timing information (start~end frame) and face identification information.

Face tracking

Face tracking consists of three components—face detection, skin-color model extraction, and skin-color region tracking (see Figure 4). We describe the face tracking components in the following subsections.

Face detection. First, Name-It applies face detection to every frame within a certain interval of frames. This interval should be small enough so that the detector doesn't miss important face sequences, yet large enough to ensure reasonable processing time. Optimally, we apply the face detector at intervals of 10 frames. The neural network-based face detector[8] employs a neural network arbitration method and bootstrap algorithm to detect mostly frontal faces of various sizes and at various locations. The system outputs the detected face as a rectangular region that includes most of the skin, but excludes the hair and the background. The face detector can also detect eyes. To ensure that the faces are frontal and close up, we use only faces in which eyes are detected successfully. A detected face is tracked bidirectionally in time to obtain a face sequence.

Skin-color model extraction and tracking. Once the system detects a face, it extracts the skin-color model. In several cases, researchers have used the Gaussian model in (r, g) space ($r = R/(R + G + B)$, $g = G/(R + G + B)$) as a general skin-color model for face tracking.[9,10] However, for our research we used the Gaussian model in (R, G, B) space because this model is more sensitive to the skin color's brightness, and thus much more suitable for the model tailored for each face sequence.

Let F be the detected face region and $I(x, y)$ be color intensities $[R\ G\ B]^t$ at (x, y). A skin-color model consists of a covariance matrix C, a mean M, and a distance d:

$$M = \frac{1}{N} \sum_{(x,\ y) \in F} I(x,\ y) \qquad \textbf{(1)}$$

$$C = \frac{1}{N} \sum_{(x,\ y) \in F} \big(I(x, y) - M\big)\big(I(x, y) - M\big)^t \qquad \textbf{(2)}$$

where N is the number of pixels in F. We used a constant for d. The system extracts a model for each detected face and uses it to extract skin candidate pixels in the subsequent frames. (A pixel $I(x, y)$ is a skin candidate pixel if $(I(x, y) - M)^t\, C^{-1}(I(x, y) - M) < d^2$.) Then the system composes a binary image of the skin candidate pixels. It applies noise reduction with region enlarging or shrinking and contour tracing of regions to obtain skin candidate regions. The overlap between each of these regions and each of the face regions of the previous frame is evaluated to decide whether one of the skin candidate regions is the succeeding face region. In addition, the system applies the scene-change detection method based on the subregion color histogram matching.[11] Face-region tracking continues until the system encounters a scene change or until it can't find a succeeding face region.

Face similarity evaluation

To evaluate face similarity, we employ a face similarity measurement based on the eigenface method. Since this method is very sensitive to face orientation, we prefer using frontal faces for evaluation. However, faces detected by the method described above aren't necessarily frontal enough. On the other hand, since we have face sequences, we can choose any face from the sequence. Therefore, we first select the best frontal view of a face—that is, the most frontal face from each face sequence. Then we apply the eigenface method to the selected faces for similarity evaluation of face sequences.

The most frontal face selection. To choose the most frontal face from all detected faces, the system first applies a face-skin region clustering method. For each detected face, cheek regions—which we presume have skin color—are located by using the eye locations the face detector obtained. Using the cheek regions as initial samples, the system employs the region growing method in the (R, G, B, x, y) space to obtain the face-skin region. We assume a Gaussian model in (R, G, B, x, y) space; (R, G, B) contributes by making the region have skin color, and (x, y) contributes by keeping the region almost circular. Then, the system calculates the face-skin region's center of gravity (x_f, y_f). Let the locations of the right and left eyes of the face be (x_r, y_r), (x_l, y_l), respectively. We assume that the most frontal face has the smallest difference between x_f and $(x_l + x_r)/2$ and the smallest difference between y_l and y_r. To evaluate these conditions, we calculate the frontal factor Fr for every detected face

$$w_f = \frac{5}{3}(x_l - x_r) \qquad \textbf{(3)}$$

$$Fr = \left(1 - \frac{\left|2x_f - x_r - x_l\right|}{w_f}\right) + \frac{1}{2}\left(1 - \frac{\left|y_l - y_r\right|}{w_f}\right) \qquad \textbf{(4)}$$

where w_f is the normalized face-region size. This size is determined so that a square $w_f \times w_f$ overlaps with the eyes, nose, and mouth, but barely overlaps with the background in most cases. The factor for an ideal frontal face is 1.5. The first term of

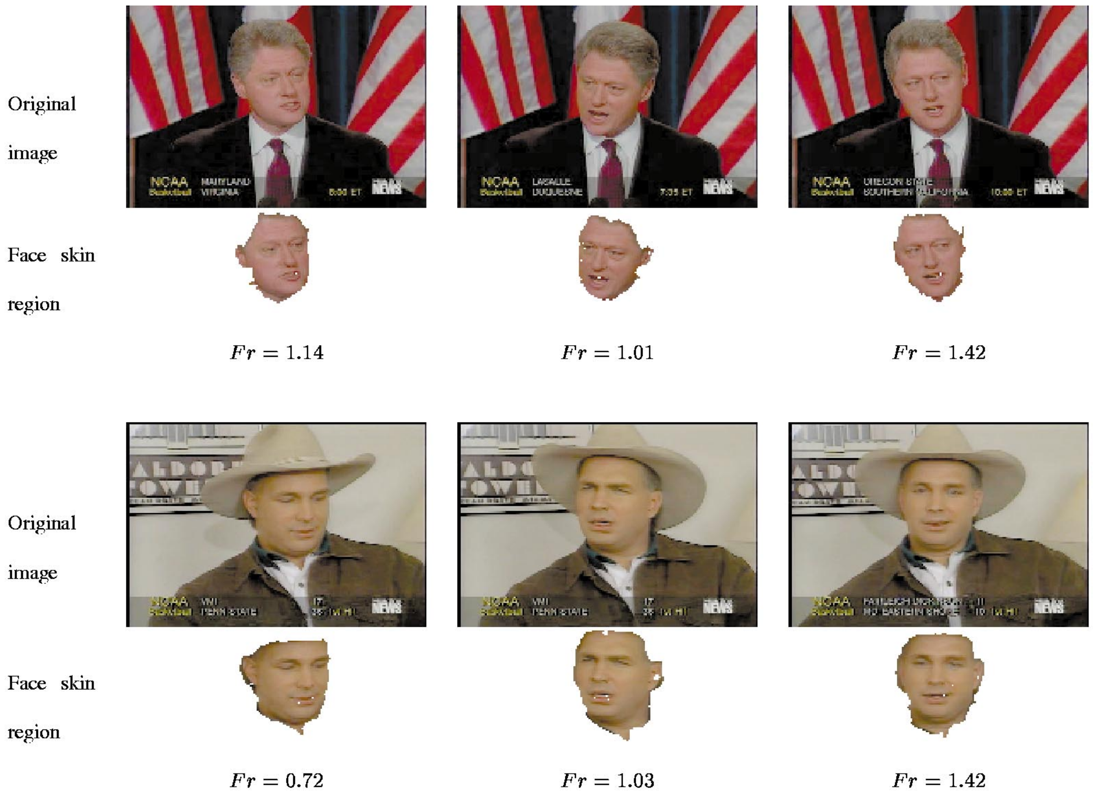


Figure 5. Frontal face selections showing example faces, extracted face-skin regions, and frontal factors.

Equation 4 becomes 1 iff x_f equals $(x_l + x_r)/2$, less than 1 otherwise. The second term of Equation 4 becomes 1/2 iff y_l equals y_r, less than 1/2 otherwise. The system chooses the face having the largest Fr as the most frontal face of the face sequence. Figure 5 shows example faces, extracted face skin regions, and frontal factors.

Eigenface-based face similarity evaluation. To evaluate face similarity, we employ the eigenface-based method.[12] Although it doesn't necessarily achieve the best performance, we chose this method because it's less restrictive to input face images (that is, it doesn't require face features detection, such as eyes, nose, mouth corners, and so on). Plus, it's compatible with face-similarity values. Each of the most frontal faces is normalized into a 64 × 64 image using the eye positions, then converted to a point in the 16-dimensional eigenface space. As we'll describe later, we processed five hours of news videos, and the system extracted 556 face sequences. To train eigenfaces, we used all 556 faces (that is, the training set equals the evaluation set). Since we fixed the video corpus to five-hour news videos, we can still take this approach. However, to incrementally process news videos, for example, we need to fix the training face set in advance and apply trained eigenfaces to faces other than the training set. Face similarity can be evaluated as the face distance—the Euclidean distance between two corresponding points in the eigenface space. Let $d_f(F_i, F_j)$ be the face distance of faces F_i and F_j. We define the similarity $S_f(F_i, F_j)$ as

$$S_f\left(F_i, F_j\right) = e^{-\frac{d_f^2\left(F_i, F_j\right)}{2\sigma_f^2}} \tag{5}$$

where σ_f is a standard deviation of the Gaussian filter in the eigenface space. The range of similarity is from 0 to 1, where similarity of the same face is 1.

Evaluation

Figure 6 (next page) shows the start and end frames of face sequences and the selected frontal face frames using the face extraction method. In Figure 6a, although the faces appearing in the start and end frames aren't frontal, the system successfully selected the frontal face. Figure 6b shows that the system can handle face sequences having scene changes by using special effects (such as wiping) in the sequence's start and end frames. A 30-minute video takes roughly 30 hours to process on a Silicon Graphics 200-MHz R4400 worksta-

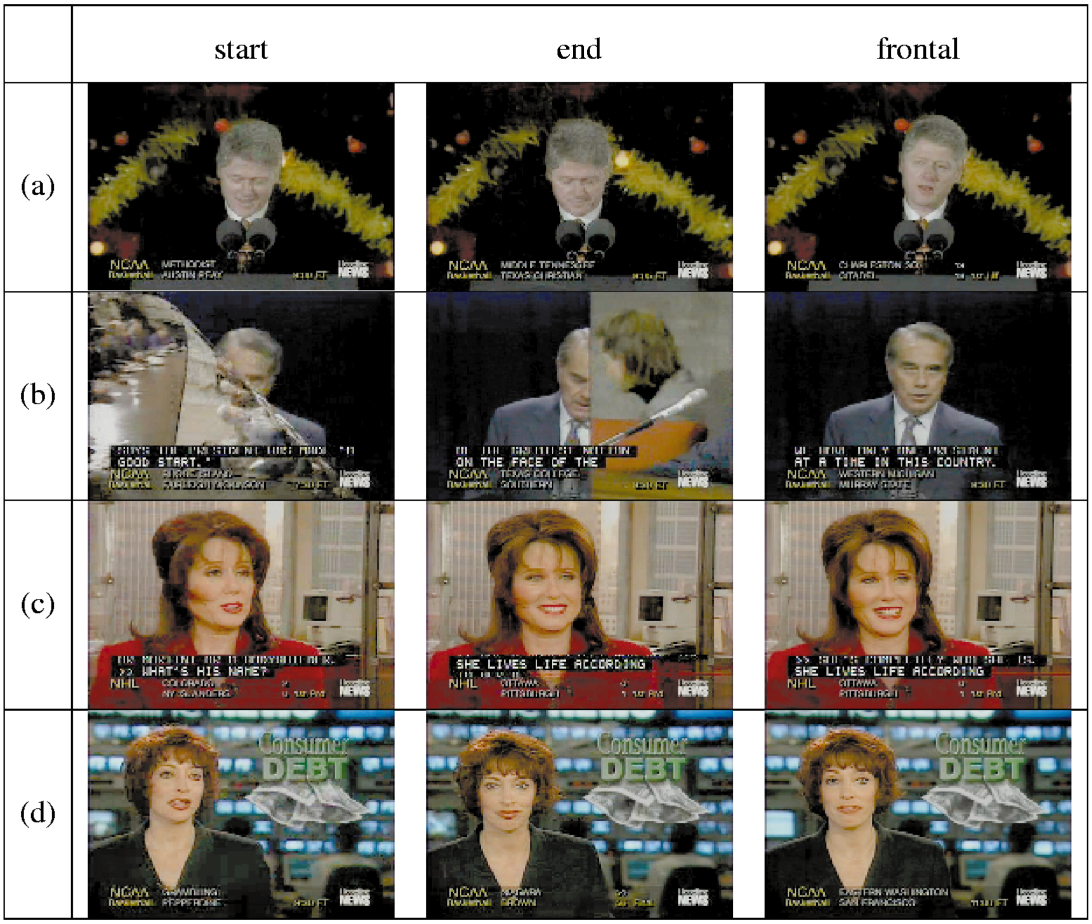


Figure 6. Face extraction results. (a, c, and d) Successful selection of a frontal face. (b) Even with scene changes using special effects, the system can handle face sequences.

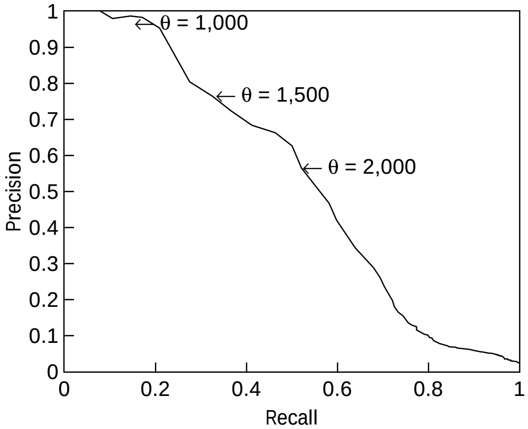


Figure 7. Precision-recall graph of an identical face-pair selection.

tion. Most of the time goes to face detection.

To evaluate face-sequence detection, we examined the face-sequence extraction results of a half-hour news video. The system extracted 65 face sequences and missed four sequences due to face-detection failure (two cases had specular reflection on eye glasses and two cases had shade on faces). The system output one nonface sequence due to a face-detection error and two sequences composed of two face sequences merged into one sequence. In one case, the system failed to detect the scene change because the scene dissolved between two face sequences. In another case, the system failed to track the face because the video segment was monochrome. This video was one of the most difficult for face-sequence extraction, yet the system extracted more than 90 percent of actual face sequences with only one false extraction.

To examine face-similarity evaluation results, we manually named each face sequence. Among 556 face sequences taken from five hours of news videos, we manually named 308 sequences and left 248 unknown. Then we examined every pair of face sequences to obtain distances $d_f(F_i, F_j)$. We labeled pairs whose distance fell below a certain threshold ($d_f(F_i, F_j) < \theta$) as expected identical pairs. On the other hand, we called pairs having the same name real identical pairs. To evaluate the appropriateness of face distance, we used a precision-recall graph comparing expected and real identical pairs while varying threshold θ (see Figure 7). Although precision is 99 percent and recall is 14 percent for $\theta = 1{,}000$, if we increase θ, precision decreases very rapidly. To preserve precision as high as possible, we used 1,000 for σ_f in Equation 5. The graph depicts that the defined face distance doesn't achieve good separation between identical and nonidentical face pairs. However, we will show in our experimental results that this face similarity evaluation method still works fairly well when integrated into the Name-It system.

Name information extraction

Here we describe the extraction of names that might correspond to persons of interest in topics. The system uses advanced natural-language processing to extract name candidates from transcripts. We'll also describe how the name candidate extraction uses lexical and grammatical analysis and knowledge of a topic's structure in news videos. The system outputs a three-tuple list: a name candidate, timing information, and a score representing the likelihood of that name belonging to a person of interest.

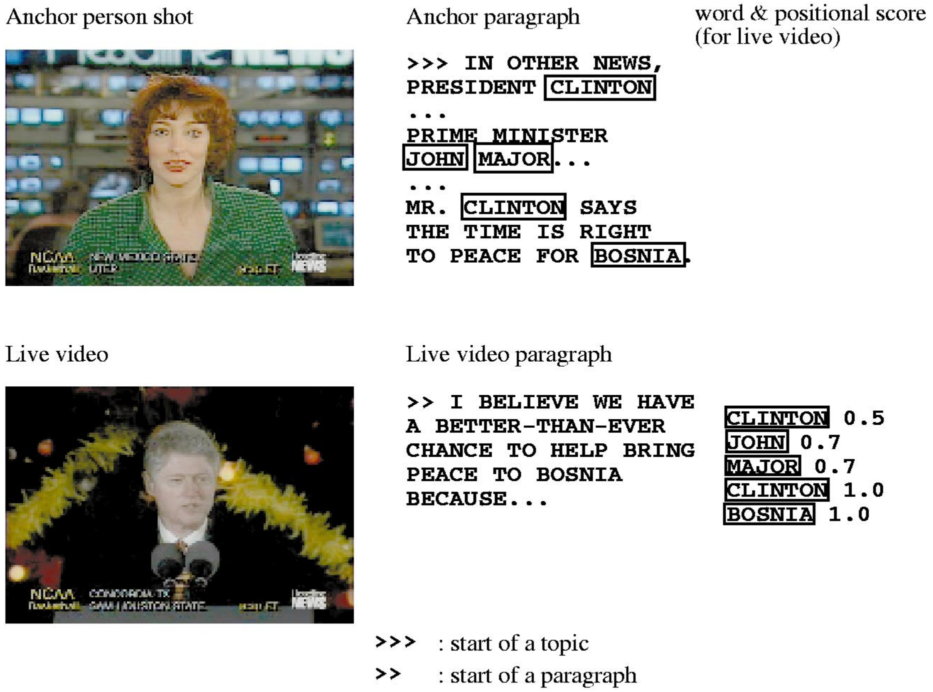

Anchor person shot
Anchor paragraph
word & positional score (for live video)
>>> IN OTHER NEWS, PRESIDENT CLINTON
...
PRIME MINISTER JOHN MAJOR...
...
MR. CLINTON SAYS THE TIME IS RIGHT TO PEACE FOR BOSNIA.
Live video
Live video paragraph
>> I BELIEVE WE HAVE A BETTER-THAN-EVER CHANCE TO HELP BRING PEACE TO BOSNIA BECAUSE...
CLINTON 0.5
JOHN 0.7
MAJOR 0.7
CLINTON 1.0
BOSNIA 1.0
>>> : start of a topic
>> : start of a paragraph

Figure 8. An anchor person shot with accompanying anchor paragraph and a live video feed with its corresponding paragraph. Positional scores for the live video are shown on the right-hand side.

Typical structure of news videos

The highest component in news video is an individual topic. Each topic contains one or more paragraphs, which roughly correspond to scenes. In closed-caption texts of "CNN Headline News," the components can be easily distinguished—a topic is preceded by >>> and a paragraph by >> (see Figure 8). We use this literal information to discriminate between an anchor paragraph and a live or file video shot from videos. For other news programs, we could use news video parsing techniques.[4,5]

An anchor paragraph typically appears at the beginning of the topic, in which an anchor person gives an overview of the topic. Live or file video paragraphs—actual videos related to the topic or speeches by a person of interest—typically appear after an anchor paragraph. A live or file video paragraph, especially one containing close-up scenes of a person of interest, proves quite useful for Name-It. In some cases, such a paragraph includes the narrator's explanations of the person in the close-up. Since the face coincides with its name in corresponding transcripts, Name-It simply evaluates the coincidence of extracted name candidates with face sequences in order to obtain face-name association. However, in other cases, the live or file video paragraph consists of the speech of the person in the close-up. Since the person rarely mentions his or her own name in the speech, corresponding transcripts may not contain the desired name. This situation requires extra care. In such cases, the system offsets time lags in name information. (We provide detailed descriptions in the following sections.) Finally, each name candidate is output with the score that represents the likelihood that the name corresponds to a person of interest.

Conditions of name candidates

Each name candidate should satisfy some of the following conditions:

1. The candidate should be a noun that represents a person's name or that describes a person (president, fireman, and so on).

2. The candidate should preferably be an agent of an act—especially an act of speech, attendance at a meeting, or a visit. For example, a speaker is usually centered in the speech scene, while other people aren't always shown in videos even if they're mentioned.

3. The candidate tends to be mentioned earlier than others in the topic in transcripts. (In a news video, important information that might have corresponding images is usually mentioned earlier rather than later.)

4. The candidate tends to be mentioned just before a live video is shown. The person appearing in a live video rarely mentions his or her own name. Instead, just before the live video, an anchor person tends to introduce the candidate (see Figure 8).

The system evaluates these conditions for each word in the transcripts by using a dictionary (the *Oxford Advanced Learner's Dictionary of Current*

English[13]), a thesaurus (WordNet[14]), and a parser (Link Parser[15]). Then the system outputs the three-tuple list: a word, timing information (frame), and a normalized score reflecting the above conditions.

Score calculation

Referring to the dictionaries and parsing results, the system calculates the score for each word in the transcripts. The score is normalized so that a score close to 1 corresponds to a word that most likely corresponds to a person of interest. We define the score calculation as follows:

- *Grammatical score*: After consulting the dictionary, the system gives 1 to proper nouns, 0.8 to common nouns, and 0 to other words. By consulting the parsing results, the system gives 1 to nouns and 0.5 to other words. If the system fails to parse, it gives 0.5 to all words. The net grammatical score equals the product of the two scores.

- *Lexical score*: After consulting the thesaurus, the system gives 1 to persons, 0.8 to social groups, and 0.3 to other words.

- *Situational score*: The act corresponding to the word is represented by the verb in the sentence that includes the word. After consulting the thesaurus, the system gives 1 to speech, 0.8 to attendance at meetings, and 0.3 otherwise.

- *Positional score*: The system gives 1 to words that appear in the first sentence of a topic, 0.5 to words that appear in the last sentence of a paragraph, and a linearly interpolated score to other words according to the position of the sentence in which the word appears. For live or file video paragraphs, the system also outputs the same tuples as those of the paragraph that appears before the live or file video paragraphs (possibly the anchor paragraph), replacing the timing information with that of the live or file video (see Figure 8). In addition, the system replaces the positional score according to the position of the sentence in the anchor paragraph: 1 for the sentence just before the live video, 0.5 for the first sentence of the topic, and a linearly interpolated score otherwise.

Finally, the system calculates the net score as the product of all four scores. The execution time for a 30-minute news video is approximately 1.5 hours on an SGI 200-MHz R4400 workstation. Most of that time goes to parsing. We determined several parameters used in score calculation empirically. Although we had an impression that these parameters wouldn't be very sensitive to face-name association results, there's still room for an in-depth study in score calculation for name candidates.

Evaluation

We examined one 30-minute news video and manually extracted 105 name words from a transcript containing 3,462 words. While the system automatically extracted 752 words as name candidates, only 94 of them were correct (it missed 9 and 658 were false alarms, that is, precision was 13 percent and recall 91 percent). This excessive name-candidate extraction resulted from the system extracting words that were proper nouns or nouns used as agents, in order not to miss any "name." But even with this poor name-candidate extraction, the overall system achieved good performance in face-name association, as we'll show in the experimental results section.

Video-caption recognition

Attached directly to image sequences, video captions provide text information. In many cases, they're attached to faces and usually represent a person's name. Thus, video-caption recognition provides rich information for face-name association, although not necessarily attached to all faces of persons of interest. We briefly describe video-caption recognition in this section. (See Sato et al.[16] for further information.)

Figure 9 shows a typical frame with video captions. Since we use "CNN Headline News" for target news videos, captions appear in bright color, superimposed directly onto the background images. To achieve video-caption recognition, the system first detects text regions from video frames. Several filters, including differential filters and smoothing filters, help achieve this task. Clusters with bounding regions that satisfy several size constraints are selected as text regions. The detected text regions are preprocessed to enhance video-caption image quality. First, the system applies the filter that minimizes intensities among frames. This filter suppresses complicated and moving backgrounds, yet enhances characters because they're placed at the exact position for a sequence of frames. Next, the system applies the linear interpolation filter to quadruple the resolution. Then it applies template-based character recognition. The current system can recognize only uppercase letters, but it has achieved a 76 percent character-recognition rate.

Since character recognition results aren't perfect, inexact matching between the results and character strings is essential for face-name association. To cope with this problem, we extended the edit-distance method.[17] Assume that C is the character-recognition result and N is a word. Define the distance $d_c(C, N)$ to represent how much C differs from N. When C and N are the same, the distance is 0, whereas when C and N differ (that is, C and N don't share any characters), the distance is 1. The system calculates the distance by using a dynamic programming algorithm.

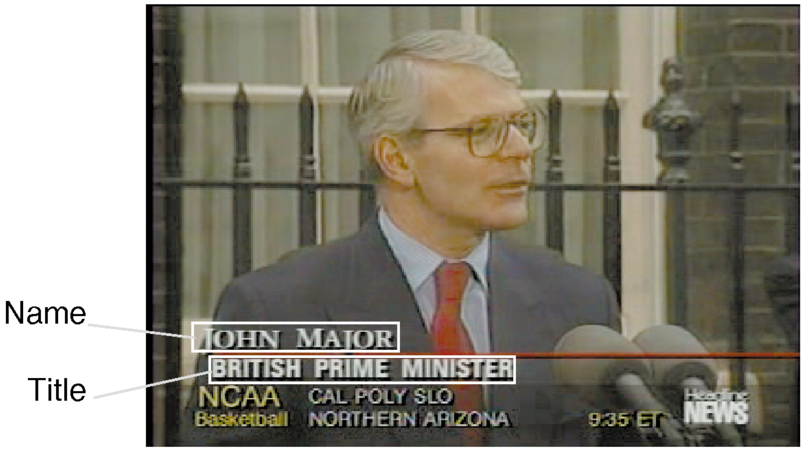


Figure 9. A typical video caption.

Face-name information association

Here we'll describe the algorithm and co-occurrence factor that work together to associate face-name information. The system calculates the co-occurrence factor taking into account analysis results of face-sequence extraction, face matching, name-candidate extraction, and video-caption recognition. Inaccuracy of each technology is compensated in this process.

Algorithm

In this section, we describe the algorithm for retrieving face candidates by a given name. We use the co-occurrence factors that take advantage of face extraction and similarity evaluation, name extraction, and video-caption recognition. Let N and F be a name and a face, respectively. The co-occurrence factor $C(N, F)$ measures the degree to which face F matches name N. Think of the names N_a, N_b, …, and the faces F_p, F_q, …, where N_a corresponds to F_p. Then $C(N_a, F_p)$ should have the largest value among co-occurrence factors of any combination of N_a and the other faces (such as $C(N_a, F_q)$ and so on) or of the other names and F_p (such as $C(N_b, F_p)$ and so on). To retrieve face candidates by a given name or name candidates by a given face, we use the co-occurrence factor:

1. Calculate the co-occurrences of combinations of all face candidates with a given name or vice versa (name candidates with a given face).

2. Sort the co-occurrences.

3. Output faces (or names) that correspond to the N largest co-occurrences.

Co-occurrence factor

Here we define the co-occurrence factor $C(N, F)$ of a face F and a name N. Extracted face sequences are obtained as a two-tuple list (timing, face identification): $\{(t_{F_i}, F_i)\}=\{(t_{F1}, F_1), (t_{F2}, F_2),\ldots\}$, where $t_{F_i}=t_{F_i}^{\text{start}}\sim t_{F_i}^{\text{end}}$. We can define the duration of a face sequence by the function $\text{dur}(t_{F_i})=t_{F_i}^{\text{end}}-t_{F_i}^{\text{start}}$. Name extraction results are given as a three-tuple list (word, timing, score):

$$\begin{aligned}&\{(N_j, t_k^{N_j}, s_k^{N_j})\}\\&=\{(N_1, t_1^{N_1}, s_1^{N_1}), (N_1, t_2^{N_1}, s_2^{N_1}), \ldots\\&\quad (N_2, t_1^{N_2}, s_1^{N_2}), \ldots\}\end{aligned}$$

Note that a name N_j may occur several times in a video, so each occurrence is indexed by k.

Timing similarity, $S_t(t_F, t_N)$, between the timing of a face F and a name N is defined as follows:

$$S_t\left(t_F, t_N\right)=\begin{cases} e^{-\frac{\left|t_N-t_F^{\text{start}}\right|^2}{2\sigma_t^2}} & \left(t_N<t_F^{\text{start}}\right)\\ 1 & \left(t_F^{\text{start}}\le t_N<t_F^{\text{end}}\right)\\ e^{-\frac{\left|t_F^{\text{end}}-t_N\right|^2}{2\sigma_t^2}} & \left(t_F^{\text{end}}\le t_N\right)\end{cases} \qquad \textbf{(6)}$$

This is basically a step function having 1 if t_N falls in the range between t_F^{start} and t_F^{end}, but its edges are dispersed using a Gaussian filter with standard deviation σ_t. The Gaussian filter should compensate for the time delay between the video and transcript.

The caption recognition results are obtained as a two-tuple list (timing, recognition result):

$$\left\{\left(t_{C_i}, C_i\right)\right\} = \left\{\left(t_{C_1}, C_1\right), \left(t_{C_2}, C_2\right), \ldots\right\}$$

where $t_{C_i} = t_{C_i}^{\text{start}} \sim t_{C_i}^{\text{end}}$ because each caption has a duration. First, the system chronologically compares the caption-recognition result with a face. We simply define the timing similarity, $S_t'(t_C, t_F)$, of timing of caption C and face F, as follows:

$$S_t'\left(t_C, t_F\right) = \begin{cases} 0 & \left(t_F^{\text{end}} < t_C^{\text{start}} \text{ or } t_C^{\text{end}} < t_F^{\text{start}}\right) \\ 1 & \left(\text{otherwise}\right) \end{cases} \quad \textbf{(7)}$$

Next, the similarity between a caption recognition result C and a name N is defined using the distance $d_c(C, N)$. To take into account only pairs of a caption and name that match well—that is, pairs of a caption and name whose distance is very small—we define the similarity $S_c(C, N)$ of a caption C and a name N as

$$S_c\left(C, N\right) = \begin{cases} 0 & \left(d_c\left(C, N\right) > \theta_c\right) \\ 1 & \left(\text{otherwise}\right) \end{cases} \quad \textbf{(8)}$$

where θ_c is the threshold value for the distance between captions and names. We set θ_c to 0.2 for our experimental system.

Finally, we define the co-occurrence factor $C(N, F)$ of the name N and the face F as

$$C\left(N, F\right) = \frac{\sum_i S_f\left(F_i, F\right)\left(\sum_k s_k^N S_t\left(t_{F_i}, t_k^N\right) + *\right)}{\sqrt{\sum_i S_f^2\left(F_i, F\right) \operatorname{dur}\left(t_{F_i}\right) \sum_k \left(s_k^N\right)^2}} \quad \textbf{(9)}$$

$$* = w_c \sum_j S_t'\left(t_{C_j}, t_{F_i}\right) S_C\left(C_j, N\right) \quad \textbf{(10)}$$

where w_c is the weight for caption-recognition results. Roughly speaking, when a name and a caption match and the caption and a face match at the same time, the face equivalently coincides with w_c occurrences of that name. We use 1 for the value of w_c. Figure 10 depicts the calculation process for the main portion of the numerator in Equation 9. Intuitively, the numerator represents the number of occurrences of the name N that coincide with face F, taking face similarities and name scores into account. That number is then normalized with the denominator to prevent the "anchor person problem." An anchor person coincides with almost any name. A face or name that coincides with any name or face should correspond to no name or face. The more names an anchor person coincides with, the larger the denominator becomes by summing for each coincident name. Thus the resulting co-occurrence factor becomes small for anchor persons.

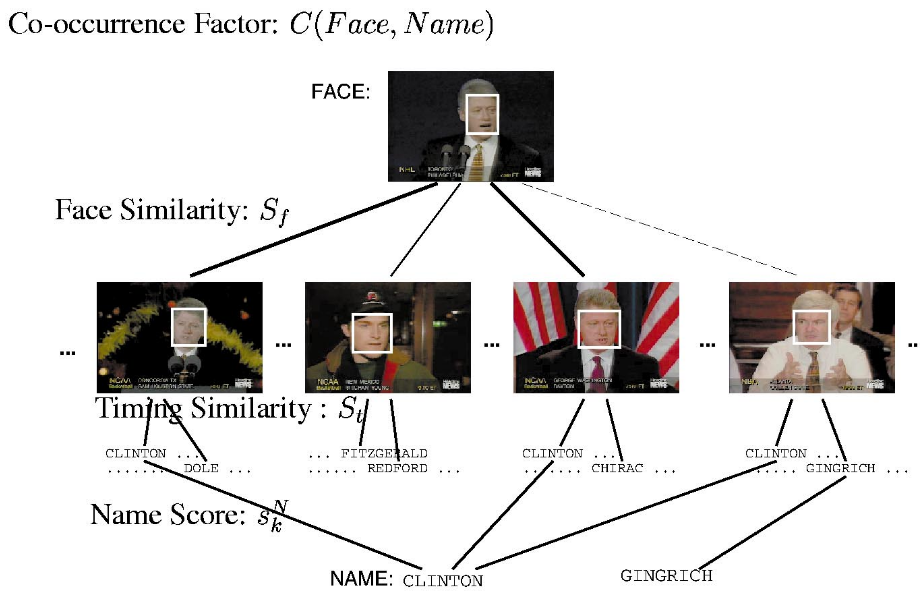
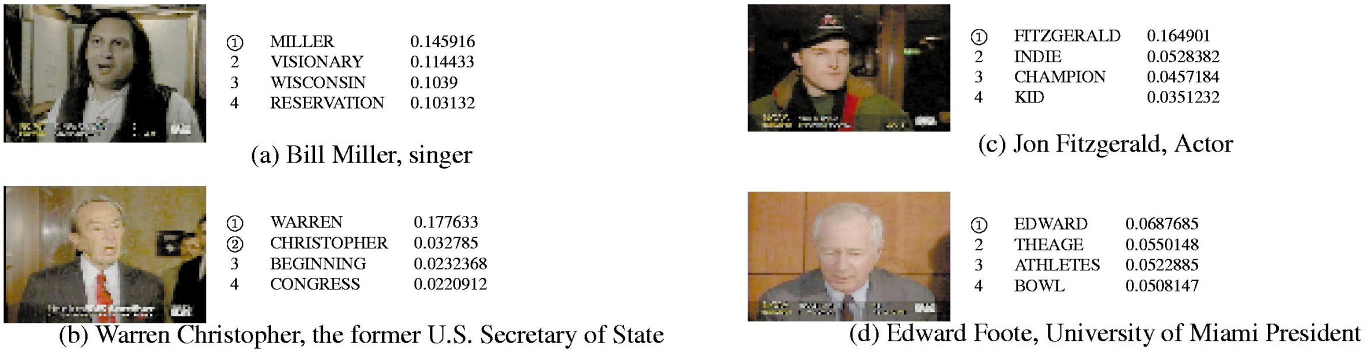


Figure 10. Calculating the co-occurrence factor. This figure depicts the calculation of co-occurrence between the face of Clinton (top) and the name "Clinton" (bottom). Edges from Clinton's face to other faces represent face similarity $S_f(F_i, F_{Clinton})$; edges from faces to names represent timing similarity S_t (t_{F_i}, t_k^N); and edges from names to "Clinton" represent name scores s_k^N in the numerator of Equation 9.

Experiments

We implemented the Name-It system on an SGI workstation. We processed 10 "CNN Headline News" videos (30 minutes each) for a total of five hours of video. The system extracted 556 face sequences from the videos.

Name-It performs name candidate retrieval from a given face and face candidate retrieval from a given name. In face-to-name retrieval, the system is given a face, then outputs name candidates with co-occurrence



①	MILLER	0.145916
2	VISIONARY	0.114433
3	WISCONSIN	0.1039
4	RESERVATION	0.103132

(a) Bill Miller, singer

①	WARREN	0.177633
②	CHRISTOPHER	0.032785
3	BEGINNING	0.0232368
4	CONGRESS	0.0220912

(b) Warren Christopher, the former U.S. Secretary of State

①	FITZGERALD	0.164901
2	INDIE	0.0528382
3	CHAMPION	0.0457184
4	KID	0.0351232

(c) Jon Fitzgerald, Actor

①	EDWARD	0.0687685
2	THEAGE	0.0550148
3	ATHLETES	0.0522885
4	BOWL	0.0508147

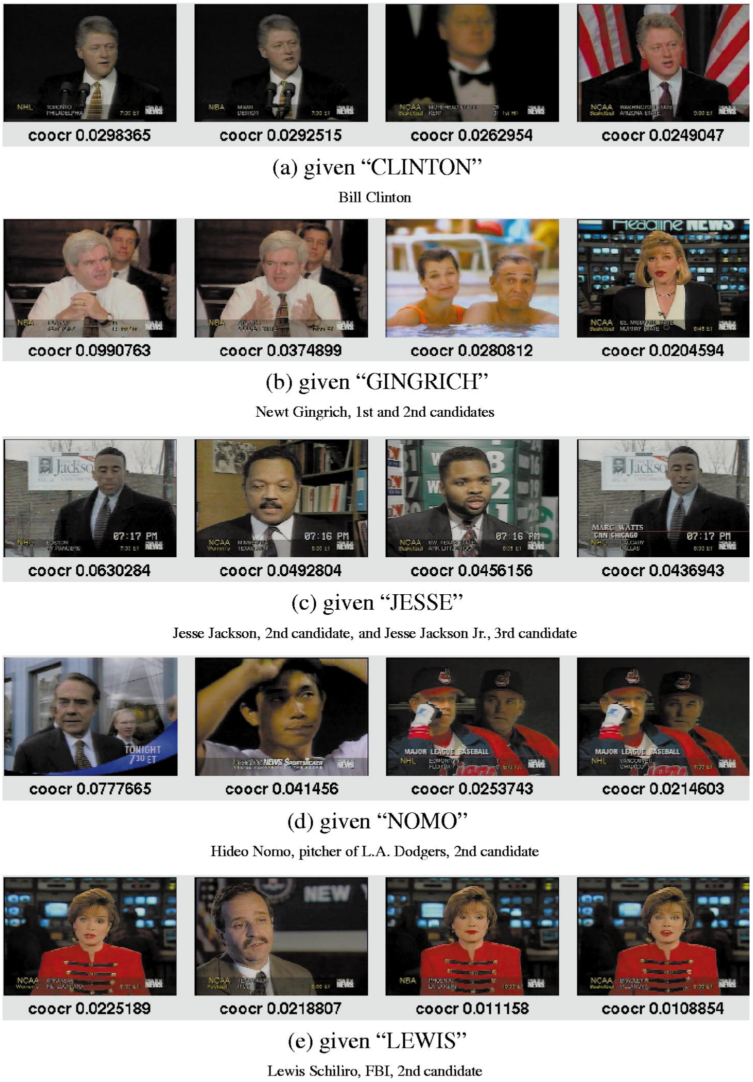
(d) Edward Foote, University of Miami President

Figure 11. Face-to-name retrieval results.

factors in descending order. Likewise, in name-to-face retrieval, the system outputs face candidates of a given name with co-occurrence factors in descending order.

Figure 11 shows the results of face-to-name retrieval. Each result shows an image of a given face and ranked name candidates associated with co-occurrence factors. Correct answers are denoted by the circled number rankings. Figure 12 shows the results of name-to-face retrieval. The top four face candidates are shown in order from left to right with corresponding co-occurrence factors. Although the correct answers acquire higher ranking, the results might be recognized as imperfect due to many incorrect candidates within the top four results. However, when we recall that Name-It extracts face and name information and combines these unreliable sets of information to obtain face-name association, inevitably the results contain unnecessary candidates. Thus, the results demonstrate good performance in face-to-name and name-to-face retrieval.

After the experiments, we evaluated the Name-It system in terms of accuracy. We used 308 man-



(a) given "CLINTON"

Bill Clinton

(b) given "GINGRICH"

Newt Gingrich, 1st and 2nd candidates

(c) given "JESSE"

Jesse Jackson, 2nd candidate, and Jesse Jackson Jr., 3rd candidate

(d) given "NOMO"

Hideo Nomo, pitcher of L.A. Dodgers, 2nd candidate

(e) given "LEWIS"

Lewis Schiliro, FBI, 2nd candidate

Figure 12. Name-to-face retrieval results.

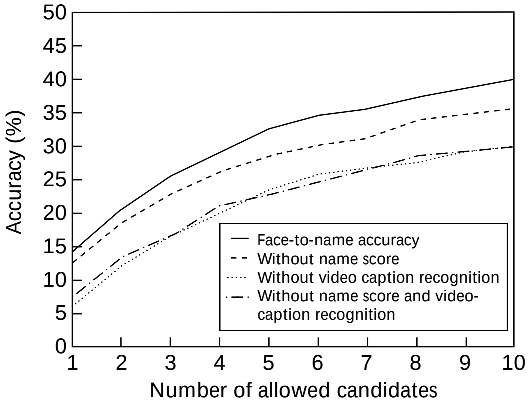


Figure 13. Accuracy of face-to-name retrieval.

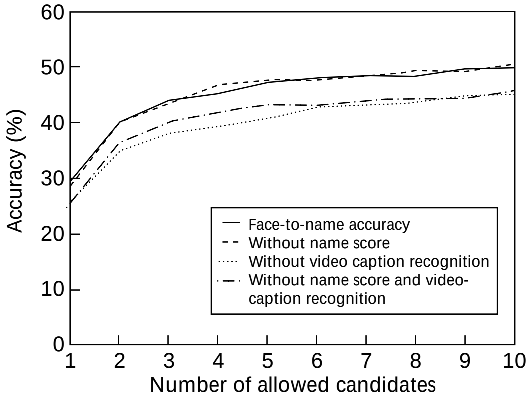


Figure 14. Accuracy of name-to-face retrieval.

ually named face sequences as the correct answer. Figures 13 and 14 depict the accuracy of face-to-name and name-to-face retrieval. In this accuracy evaluation, if the correct answer is output in the top *N* candidates, we regard this output as correct. (The output is correct with *N* allowed candidates. Note that a name may correspond to several identical faces, and a face may correspond to both the given name and the family name.) Thus these graphs represent relationships between accuracy and the number of allowed candidates. They also show results using both name scores and video-caption recognition, results without name scores (set all scores to 1.0), results without video-caption recognition (set w_c to 0), and results without either name scores or video-caption recognition.

By comparing the results using both name scores and video captions to the results without video captions for both graphs, we can say that video-caption recognition contributes to higher accuracy. Actually, some faces aren't mentioned in the transcripts, but described in video captions. These faces can be named only by incorporating video-caption recognition (such as Figure 11d and Figure 12e). Figure 13 shows that name score evaluation proves effective for face-to-name retrieval. However, according to Figure 14, it doesn't cause any major difference in accuracy for name-to-face retrieval. This result indicates that name scores properly reflect whether each word corresponds to a person of interest in topics (in face-to-name retrieval). By contrast, name scores cannot represent which occurrence of a certain word coincides with a face sequence of the person of the name in name-to-face retrieval. In other words, name scores succeed in inferring which word is likely to correspond to a person of interest. However, they fail to infer which word actually coincides with the face sequence. The main reason for this is the fact that transcripts don't explain videos directly. To overcome this problem, the system may need in-depth transcript recognition, as well as in-depth scene understanding, and a proper way to integrate these analysis results. The graphs also disclose that Name-It achieves an accuracy of 33 percent in face-to-name retrieval and 46 percent in name-to-face retrieval with five candidates allowed.

Conclusions

Name-It associates faces and names in news videos by integrating face-sequence extraction and similarity evaluation, name extraction, and video-caption recognition into a unified factor: co-occurrence. The successful experimental results demonstrate the effectiveness of a multimodal approach in video content extraction. Although the performance of each individual technology is not always high, our experiments demonstrate that Name-It achieves good face-name association. Further research will aim to enhance each technique, as well as analyze and improve the integration method. MM

Acknowledgment

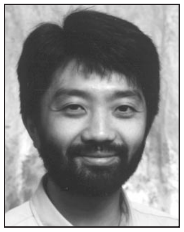
This material is based on work supported by the National Science Foundation under Cooperative Agreement No. IRI-9411299. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

1. S. Satoh and T. Kanade, "Name-It: Association of Face and Name in Video," in *Proc. of Computer Vision and Pattern Recognition*, IEEE Computer Society Press, Los Alamitos, Calif., 1997, pp. 368-373.
2. S. Satoh, Y. Nakamura, and T. Kanade, "Name-It: Naming and Detecting Faces in Video by the Integration of Image and Natural-Language Processing," in *Proc. of Int'l Joint Conf. on Artificial*

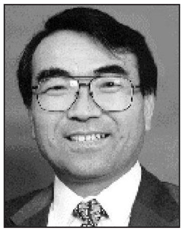
Intelligence, Morgan Kaufmann, San Francisco, 1997, pp. 1488-1493.

3. R. Chopra and R.K. Srihari, "Control Structures for Incorporating Picture-Specific Context in Image Interpretation," *Proc. Int'l Joint Conf. on Artificial Intelligence*, Morgan Kaufmann, San Francisco, 1995.
4. D. Swanberg, C.F. Shu, and R. Jain, "Knowledge Guided Parsing in Video Database," *Proc. Symp. on Electric Imaging, Science, and Technology*, SPIE Press, Bellingham, Wash., Vol. 1908, 1993, pp. 13-24.
5. S.W. Smoliar and H. Zhang, "Content-based Video Indexing and Retrieval," *IEEE MultiMedia*, Vol. 1, No. 2, April-June (Summer) 1994, pp. 62–72.
6. R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and Machine Recognition of Faces: A Survey," *Proc. IEEE*, Vol. 83, No. 5, 1995, pp. 705-740.
7. *Proc. Sixth Message Understanding Conference*, Morgan Kaufmann, San Francisco, 1995.
8. H.A. Rowley, S. Baluja, and T. Kanade, "Neural Network-based Face Detection," *IEEE Trans. on PAMI*, Vol. 20, No. 1, 1998, pp. 23-38.
9. J. Yang and A. Waibel, "Tracking Human Faces in Real Time," Tech. Rep. CMU-CS-95-210, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1995.
10. H.M. Hunke, "Locating and Tracking of Human Faces with Neural Networks," Tech. Rep. CMU-CS-94-155, School of Computer Science, Carnegie Mellon University, Pittsburgh, 1994.
11. M. Smith and T. Kanade, "Video Skimming and Characterization through the Combination of Image and Language Understanding Techniques," *Proc. Computer Vision and Pattern Recognition*, IEEE CS Press, Los Alamitos, Calif., 1997, pp. 775-781.
12. M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71-86.
13. *Oxford Advanced Learner's Dictionary of Current English* (computer usable version), R. Mitton, ed., The Oxford Text Archive, Oxford, UK, June 1992, http://ota.ox.ac.uk/.
14. G. Miller et al., "Introduction to WordNet: An Online Lexical Database," *Int'l J. Lexicography*, Vol. 3, No. 4, 1990, pp. 235-244.
15. D. Sleator, "Parsing English with a Link Grammar," *Proc. Third Int'l Workshop on Parsing Technologies*, Assoc. for Computational Linguistics, 1993.
16. T. Sato et al., "Video OCR for Digital News Archives," *Proc. IEEE Workshop on Content-based Access of Image and Video Databases (*ICCV 98), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 52-60.
17. P.A.V. Hall and G.R. Dowling, "Approximate String Matching," *ACM Computing Surveys*, Vol. 12, No. 4, 1980, pp. 381-402.



Shin'ichi Satoh is an associate professor at the National Center for Science Information Systems (NACSIS), Japan. He received a BE in 1987 and ME and PhD degrees in 1989 and 1992, respectively, from the University of Tokyo. His research interests include video analysis and multimedia databases. He was a visiting scientist at the Robotics Institute, Carnegie Mellon University, Pittsburgh from 1995 to 1997.

Yuichi Nakamura is an assistant professor at the University of Tsukuba, Japan. He received a BE in 1985 and ME and PhD degrees in electronical engineering from Kyoto University in 1987 and 1992, respectively. His research interests and activities include video analysis and video utilization for knowledge sources. He was a visiting scientist at the Robotics Institute, Carnegie Mellon University, Pittsburgh, in 1996.



Takeo Kanade is currently the U.A. Helen Whitaker University Professor of computer science and Director of the Robotics Institute at Carnegie Mellon University. He received a PhD in electrical engineering from Kyoto University, Japan in 1974. He has written more than ten patents and worked in multiple areas of robotics—computer vision, manipulators, autonomous mobile robots, and sensors. He has been elected to the National Academy of Engineering and is a Fellow of the IEEE, a Fellow of ACM, a Founding Fellow of the American Association of Artificial Intelligence, and the founding editor of *International Journal of Computer Vision*. He has received several awards including the Joseph Engelberger Award, Japan Robot Association Award, Otto Franc Award, Yokogawa Prize, and Marr Prize Award.

Readers may contact Satoh at the National Center for Science Information Systems, 3-29-1 Otsuka, Bunkyo-ku, Tokyo 112-8640, Japan, e-mail satoh@rd.nacsis.ac.jp.