Interactive adaptive movie annotation

Vendrig, J.; Worring, M.

DOI
10.1109/MMUL.2003.1218254
Publication date
2003

Published in
IEEE Multimedia

Interactive Adaptive Movie Annotation

Jeroen Vendrig and Marcel Worring
MediaMill and the University of Amsterdam

Effectively labeling the visual content of movies is essential for annotation. We present the interactive and adaptive i-Notation system, which describes actors' names, automatically processes multimodal information sources, and deals with available sources' varying quality. It provides the basis for intelligent interaction and demonstrates significant improvements in annotation efficiency.

With the advance of digital video, movie viewers gain more control over what they see. We expect much more interactivity in films produced for DVD systems and online entertainment-on-demand systems.[1] A likely application is nonlinear video browsing (see Figure 1), letting viewers jump to their favorite scenes, actors, jokes, and so on. Rather than ask for a predefined subject, viewers may want to describe their interests. Hence, future interactive movie systems need to deal with a wide variety of requests. Our goal is to assist creators of interactive movie applications by enriching the video data with semantic metadata.

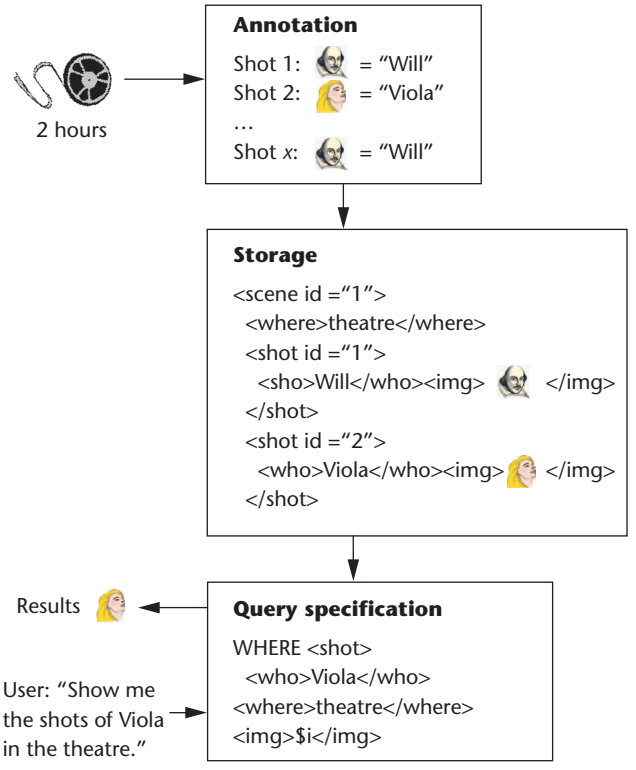

Figure 1. Example processes in an application where a viewer queries the video content.

Describing and answering

Consumers demand high-quality answers based on semantic queries, and in a perfect world they'd have these queries automatically answered. Although it's not currently possible, we're taking a step in this direction. For now, we're more specifically concerned with answering viewers' questions by annotating content. Within the large field of video content annotation—as described further in the "Video Annotation" sidebar—we focus on computer-assisted annotation. Our interactive, adaptive tool i-Notation assists users in shot grouping and in label finding. This tool automatically processes visual information, speech, and scripts and makes suggestions based on previous user decisions. The tool lets annotators lay the foundation for innovative retrievals by answering four questions:

- Where?
- When?
- What?
- Who?

For movies, Where? and When? are related because they both determine the scene (that is, the sequence of shots with the same time and locale). Automatic scene segmentation (as evaluated elsewhere)[2] performs well enough to handle Where? and When? manually at the scene level. Generally an answer to What? is interesting only in the context of the persons performing the action, so that Who? must be resolved first. Furthermore, viewers generally prefer seeing people, and consequently shots of people dominate most movies. Therefore, we focus on assisting annotators in answering the Who? question.

1070-986X/03/$17.00 © 2003 IEEE

Published by the IEEE Computer Society

Answering Who? requires attaching a label to each shot, describing the visible characters with their proper names. We annotate at the shot level since it's "the finest level of descriptive granularity" for movies.[3] A character is a person with a speaking part, excluding extras, such as people walking on the street. The definition of character still includes a wide range of people. In the movie *Shakespeare in Love* (which we use for examples throughout this article), 45 characters appear.

Labeling each shot is a tedious and time-consuming process. Even slow-paced movies contain 2,000 shots. The movie industry needs more effective methods for labeling shots with character names.

Generally the trivial approach to annotation is sequential, where we annotate shots one by one. With an effective annotation method, however, we can label shots simultaneously. Building an effective system involves two major components: *shot* and *label selection*. Shot selection groups shots that have the same label. In label selection, annotators identify characters for the shot set. This can take more time than shot selection because identifying unknown actors with a small part is time-consuming. A video annotation tool, though, can efficiently assist the annotator in this process.

Information sources

For movies, various internal and external information sources are available for automatic processing. Internal information sources are encoded in the movie visually, aurally, and textually—the latter in the form of closed captions. As external information sources, we use textual movie production scripts containing information about the movie content. Figure 2 (next page) shows how the information sources are segmented and related. In addition, movie encyclopedias provide visual information and structured textual information about actors.

Sometimes it takes more than one source to find out who's in a shot. The different channels provide overlapping and complementary information. At a certain point in time, the visual information shows who appears and the audio signal discloses what's said. Speech content from the audio signal equals closed caption content, but the latter has a smaller error rate.[4] Closed captions are time coded, but they lack character names. The script text describes what's said and by whom, but it isn't time coded. Thus, scripts and closed captions are supplementary.

Video Annotation

We can divide research on video content annotation into three fields:

- annotation formatting,
- environments for manual annotation, and
- computer-assisted annotation.

Annotation formatting tries to enable efficient retrieval and easy exchange. Examples are MPEG-7[1] and Algebraic Video.[2] Annotation formatting specifies and structures how annotations should be written but doesn't tell what the annotations are.

Manual annotation environments deal with managing the process of user interaction, accessing the data for visualization only. That is, how can we transfer annotations from the user's mind to the information storage system? Examples include MediaStreams[3] and the two microphones recording system.[4] Manual annotation helps the user specify the annotation, but again it doesn't assist in determining annotation values.

Computer-assisted annotation assigns labels to video content through a system's data analysis. Ideally such a system operates without human interaction during the process. However, often the automatic labeling quality is insufficient, resulting in semiautomatic annotation. The same techniques then assist the annotator to provide a starting point for annotation. Examples of automatic annotation systems are Name-It[5] and the video extension to FourEyes.[6] Name-It requires an explicit link between the visual appearance of a person and its label, such as a video caption or a reporter mentioning the name. FourEyes for video is geared toward a TV series with a small cast and an accurate script. Use of short videos allows for an image-based approach as done in the original FourEyes system.

References

1. F. Nack and A.T. Lindsay, "Everything You Wanted to Know about MPEG-7: Part 1," *IEEE MultiMedia*, vol. 6, no. 3, July–Sept. 1999, pp. 65-77.
2. R. Weiss, A. Duda, and D.K. Gifford, "Composition and Search with a Video Algebra," *IEEE MultiMedia*, vol. 2, no. 1, Spring 1995, pp. 12-25.
3. M. Davis, "Media Streams: An Iconic Visual Language for Video Representation," *Readings in Human–Computer Interaction: Toward the Year 2000*, 2nd ed., Morgan Kaufmann, 1995, pp. 854-866.
4. R. Lienhart, "A System for Effortless Content Annotation to Unfold the Semantics in Videos," *Proc. IEEE Int'l Workshop on Content-Based Access of Image and Video Databases*, IEEE CS Press, 2000, pp. 45-49.
5. S. Satoh, Y. Nakamura, and T. Kanade, "Name-It: Naming and Detecting Faces in News Videos," *IEEE MultiMedia*, vol. 6, no. 1, Jan.–Mar. 1999, pp. 22-35.
6. J.S. Wachman and R.W. Picard, "Tools for Browsing a TV Situation Comedy Based on Content Specific Attributes," *Multimedia Tools and Applications*, vol. 13, no. 3, 2001, pp. 255-284.

Obviously, faces are important visual information. Movie encyclopedias provide a priori information on the faces of famous actors. Face

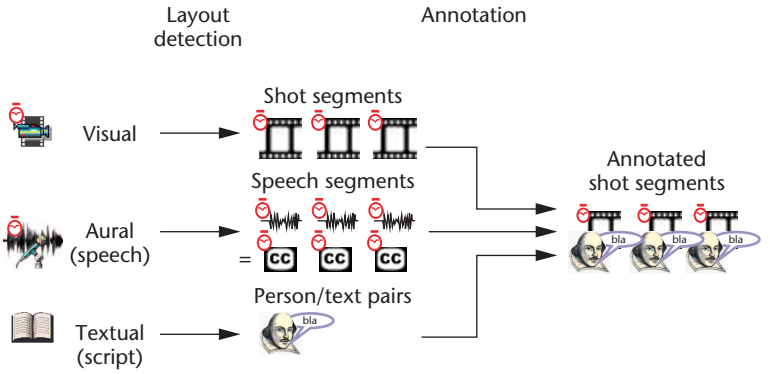

Figure 2. The available information sources are structured and aligned so that users can annotate the persons in a shot.

detection (as used by Carnegie Mellon's Name-It for news videos) could be used in movies as well. However, current face detection systems aren't fully robust under variations in lighting and orientation.[5] In movies, such variations are common, so that although face detection is useful in distinguishing shots that contain people, we can't use it to classify shots with full certainty.

Visual information is also important to define structure in movies. Narrative movies are divided into parts with semantic coherence. Visually, the coherence translates to repeating similar shots. For example, in *Shakespeare in Love*, the camera alternates between the faces of Will and Viola in a dialogue. Even when taken from various viewpoints, all shots show either Will or Viola. We've found that we can exploit semantic coherence to detect movie scenes.[2] The coherence supports effective annotation in a similar way. Because a character generally remains in the same scene, we can relate shots with the same character via the background. Hence, we can derive semantic labels from visual similarity based on a general feature.

We can also identify the speaker using aural information. However, current techniques in speech processing can't sufficiently work on an untrained data set, especially when background music and noise interfere. We propose employing textual information sources to alternatively detect the speaker for synchronization with the visual content.

Recall that three sources contain information about what's said:

- speech,
- closed captions, and
- script.

We employ closed captions as a substitute for speech from the audio signal. A script contains information about the speakers, but it's not time coded. We can't link the speakers to the visual content directly. Indirectly, we can synchronize scripts using closed captions, which carry a time code and have the same modality as a script. Because the actual video content may differ from the production script, a sentence from a closed caption can be found in the script by doing a fuzzy search.

Shot selection

As we mentioned previously, shot selection helps facilitate the annotation process. For shot selection, the system analyzes user interactions as an additional information source. This leads to interactive *adaptive* shot selection, where i-Notation presents shots to the user based on previous selections.

A label then describes all persons visible in the shot. An example label is "Will AND Viola AND Ralph." We consider "Will AND Ralph" a different label, regardless of the overlap. If no people are visible, the label is empty (for example, when a landscape is shown). The "unidentified people" label means people are visible but we can't recognize any characters (such as in blurry shots or in long distance shots).

The goal of adaptive shot selection is to present the unlabeled shots most likely to have the target label. We choose the target label as the previously selected label for a continuous annotation process.

Interaction information comprises both positive and negative information. A user gives positive information by selecting a label and associated shots, such as "these shots contain Viola." As a consequence, the user gives negative information for the remaining shots, because they therefore aren't associated with the "Viola" label.

Based on the various information sources, the i-Notation system ranks shots according to similarity to the target label. In the following sections, the individual similarity scores contribute to the overall similarity score—with all resulting in a value between 0 and 1.

Visual similarity

By *visual similarity*, we mean the similarity between already labeled and unlabeled shots. The system bases visual similarity on positive feedback from users using shot repetition.

Although the goal of annotation is to identify the person in the shot, we can use the background to compare the shots based on a global feature. We

use a combination of the hue-saturation histogram for the chromatic part of the color space and the intensity histogram for the achromatic part.[2]

The system compares a shot to all shots already labeled with the target label. It uses the average score for the most similar labeled shots as the final score.

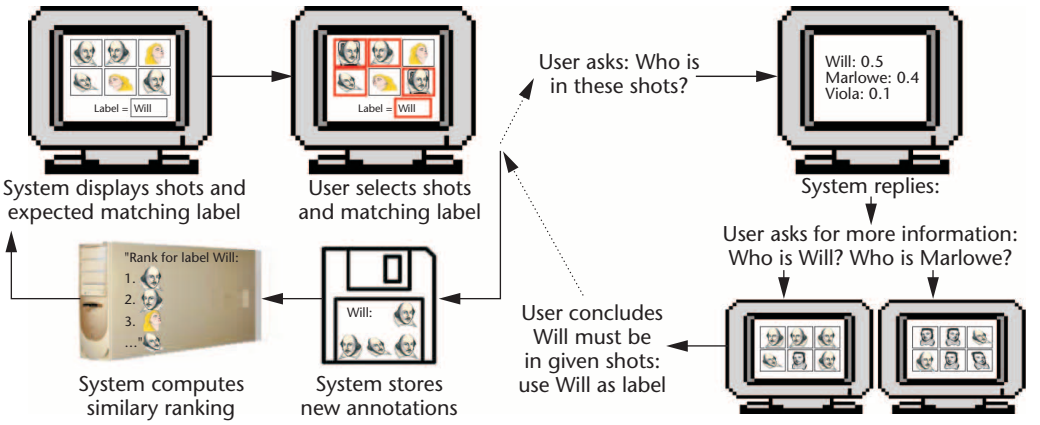

Figure 3. Interactive annotation process. The dotted arrows refer to the optional WhoIsWho function (see the "Label selection" section) for finding out the names of the characters in the shots.

Visual dissimilarity

We base *visual dissimilarity* on negative feedback from users between shots not having the target label and an unlabeled shot. Visual dissimilarity uses shot repetition to avoid selection of similar shots. We assume that shots not selected during the last user interaction don't have the target label. These shot/label combinations form a blacklist. The blacklist simply excludes the shots if the label matches the target label. However, we use the blacklist as negative feedback for visual dissimilarity for two reasons. First, users might make an error and miss a shot. Second, using the blacklist as a dissimilarity lets us extend the blacklist with other shots. For example, we can use visual similarity to rule out shots similar to the already blacklisted shots. In a deterministic blacklist, the impact of such an approach would be too high.

Label similarity

Here we refer to the similarity between the target and expected label for the unlabeled shot. *Label similarity* measures the correspondence between the character names in the target label and the names of the speaking characters in the shot. Since the expected labels are based on speech, they're usually not precisely synchronized with a shot's visual content. Hence, our system uses a similarity value. For each shot, the system determines an expected label and compares it to the target label. It considers the labels similar if they have at least one name in common. The similarity value is proportional to the number of lines during the shot.[6]

Person presence similarity

Here again we refer to similarity between the target label and unlabeled shot. The *person presence similarity*, however, measures whether the number of people visible corresponds to the target label. If the target label is the special "no people" tag, the unlabeled shot shouldn't contain any people. The similarity score is the percentage of the shot for which the number of faces found by the system matches the number of persons in the label.

Temporal similarity

Here we search for similarity between the unlabeled shot and shots known to have the target label. *Temporal similarity* exploits the tendency in movies where characters are more likely to reappear in close-by shots, implicitly making use of movie structure. We employ the temporal attraction measurement successfully used in logical story unit[7] segmentation.

Overall score explanation

We normalize the five similarity scores with the similarity score distribution,[8] resulting in one overall similarity measure between the given label and the unlabeled shot. Srihari[9] tackles a similar problem for annotating images in the Piction system by adding various scores after multiplying the individual values with a weighing factor. We can determine the weighing factors empirically using application knowledge. However, to avoid fine-tuning we set all weights to be equal. Based on the combined similarity score the system ranks the shots.

The annotator sees the top-ranked shots using keyframes for shot representation. The initial display targets the most-frequent names in the script. The number of shots shown during each interaction depends on the display size. The annotator selects a label and the matching shots. Next, the system computes a new ranking and the process iterates. Figure 3 depicts this process.

Label selection

Since there are more unknown actors in a movie than Hollywood stars, it's often hard to determine the label value. The annotator must find out the labels for these actors based on limited information. We propose using i-Notation's WhoIsWho function. The function works both ways in associating shots and labels. Its most common use is finding a label for selected shots. We can also use it as verification in case the system's answer doesn't convince the annotator.

The WhoIsWho function for pictures assumes that the requested label hasn't yet been used in the annotation process. Furthermore, WhoIsWho targets finding a label for one person only. If a shot contains more than one unknown person, the system must call the function several times.

WhoIsWho tells which character names are related to given shots based on the script. Selecting the correct character name is left to the user. The function ranks names according to their appearance frequency in the script in the context of the given shots. If the top-ranked name doesn't stand out, users can investigate further by asking what other shots contain the top-ranked names. If the users still aren't convinced, they can inspect the video and script in detail. Manual inspection is a tedious and time-consuming process, so an effective WhoIsWho function is crucial.

Evaluating i-Notation

We evaluate i-Notation's shot and label selection with user modeling. The user model specifies the user's choices, and the user's a priori knowledge. Hence, we have full control over experiment parameters, resulting in consistent and objective evaluation. For this purpose we first define a ground truth annotation for the movie.

Ground truth

Defining a ground truth is far from trivial. For example, actors Ben Affleck and Joseph Fiennes aren't easily confused close up. From a distance, however, filmed in an action scene and both dressed in blue suits, it's hard to tell them apart. Users then base recognition on assumptions and interpretations, which is undesirable for a ground truth.

To minimize subjectivity, we annotate a person only if the head appears in the keyframe and if the face is visible at some time during the shot. We focus on faces, as they're usually the only identifiable body parts. Note that a person's head doesn't necessarily equal the face, as he can be filmed from the back. We differentiate between shot and keyframes to avoid debating whether a person is recognizable, such as when a person is in the process of turning the head. Hence, we aren't restricted to a keyframe for annotating a ground truth label for an entire shot.

To focus evaluation of the WhoIsWho function on unknown characters, we have a list of celebrities to whom the function doesn't apply. As an objective and practical definition, a celebrity is an actor whose picture is published in the Internet Movie Database biography (http://www.imdb.com).

Shot selection evaluation

The user model and evaluation criterion for shot selection are straightforward. For each evaluated strategy the end result is exactly the same. Therefore, the evaluation criterion should measure the effort for labeling cost only and not the quality of the result. For comparison of shot-selection methods, we measure user effort by counting the total number of interactions—that is, the number of times a user made a shot selection for a label.

Although annotations consist of more than just selecting shots and labels, other efforts are independent of the shot-selection method. The most costly (in terms of time and money) other efforts in annotation are judging a shot's visual content and adding new labels to the list. Such efforts are independent of a specific method.

For interactive annotation, we assume that the annotator evaluates each shot before labeling it, resulting in a maximum and a minimum number of interactions. The maximum number is the number of shots—that is, each shot labeled individually. The minimum number is counted by following an ideal case scenario in which as many shots as possible are labeled by one interaction. The resulting number depends on the number of shots shown on screen and on the movie's label distribution. For example, if nine shots are shown during each interaction, in an ideal scenario the system labels nine shots by one interaction. The label "Will," applying to 389 shots, can be used 43 times in an interaction displaying nine shots of the same label (totaling 387 shots). Then the two remaining shots need an additional, nonoptimal interaction—in the sense that the full capacity of the display space can't be used.

For measuring the interactive annotation performance gain G for the actual number of inter-

actions f, the maximum number of interactions m is a reference point:

$$G = \frac{m - f}{m} \cdot 100 \text{ percent}$$

Here G ranges from 0 to 100 percent. The worst-case scenario, namely when m is reached, results in zero gain. Even the ideal case scenario won't reach 100 percent. Only a fully automatic, completely faultless annotation system would reach 100 percent. In practice, the upper bound for the criterion value equals the performance gain in the case of a minimal number of interactions. As noted before, the upper bound varies per movie.

As we base the maximum number of interactions on the default situation of having no video annotation tools, the evaluation criterion reflects the economic impact of tools assisting annotators.

Evaluating label selection

Label selection should help users form their opinion. As it's hard to measure to what extent proposed labels influence the user, we measure how often the advice was correct regardless of acceptance of the advice. We model the users, having them first select shots with the same label. Next, for each unknown person in the shots, users activate the WhoIsWho function and evaluate the names proposed. Celebrities in the movie as well as characters already labeled are dismissed from the names list. The remaining list yields a quantitative and qualitative evaluation measure for the WhoIsWho function.

The quantitative evaluation measures how often we can use the WhoIsWho function and whether the answer is correct. Note that if the list of names is empty, the function can't assist the user. The number of times the system returns a nonempty list should be high enough so that a user is willing to invest time in employing the function. The quantitative evaluation measure W_u counts the percentage of cases where the function returns one or more character names:

$$W_u = \frac{\text{number of advices}}{\text{number of calls}} \cdot 100 \text{ percent}$$

Thus, the measure expresses how often we can use the WhoIsWho function on average.

The qualitative evaluation criterion is concerned with the quality of the found names. Since we perform the WhoIsWho function on a collection of shots, the same name can appear several times. The system selects the character name that occurs most often. In case of a tie, we considered the advice ambiguous and therefore incorrect. Next, we compare the selected character name with the ground truth to determine whether the advice is correct. The qualitative evaluation criterion W_c measures the percentage of correct advices:

$$W_c = \frac{\text{number of correct advices}}{\text{number of calls}} \cdot 100 \text{ percent}$$

We measure WhoIsWho's success by comparing it to a random selection of a name from the set of yet unidentified character names. We use the average probability W_r that the system selected the correct name, measured for the same cases as for W_c.

Table 1. Annotation performance gain for three movies in various scenarios, in the case of nine shots displayed.

Movie	Sequential (%)	Adaptive (%)	Ideal (%)
Shakespeare in Love	42	60	83
Sneakers	33	66	85
LA Confidential	34	69	86

Results

We defined the ground truth for the full-length movies *Shakespeare in Love*, *Sneakers,* and *LA Confidential*. We additionally evaluated the first half hours of the movies *The Matrix*, *The Fugitive*, and *Being John Malkovich*. Automatic shot segmentation resulted in 7,522 shots in total for the 7.5 hours of movie playtime.

For evaluating shot selection performance, we used sequential and adaptive annotation strategies. Value m in the worst-case scenario corresponds to the number of shots evaluated. In the ideal-case scenario, a faultless system is simulated using a priori available ground truth information.

We experimentally determined that a nine-shot display (3×3) is the maximum number maintaining good visibility of the visual content for the average movie.

Table 1 shows the annotation performance gains for the three full-length movies. Figure 4 (next page) shows a typical example of performance progress. In the ideal case scenario the performance is maximal initially. In the end, performance decreases because there are few shots with the same label. The opposite effect is seen in the case of the sequential strategy as the end credits are annotated, all having the same special label "no people."

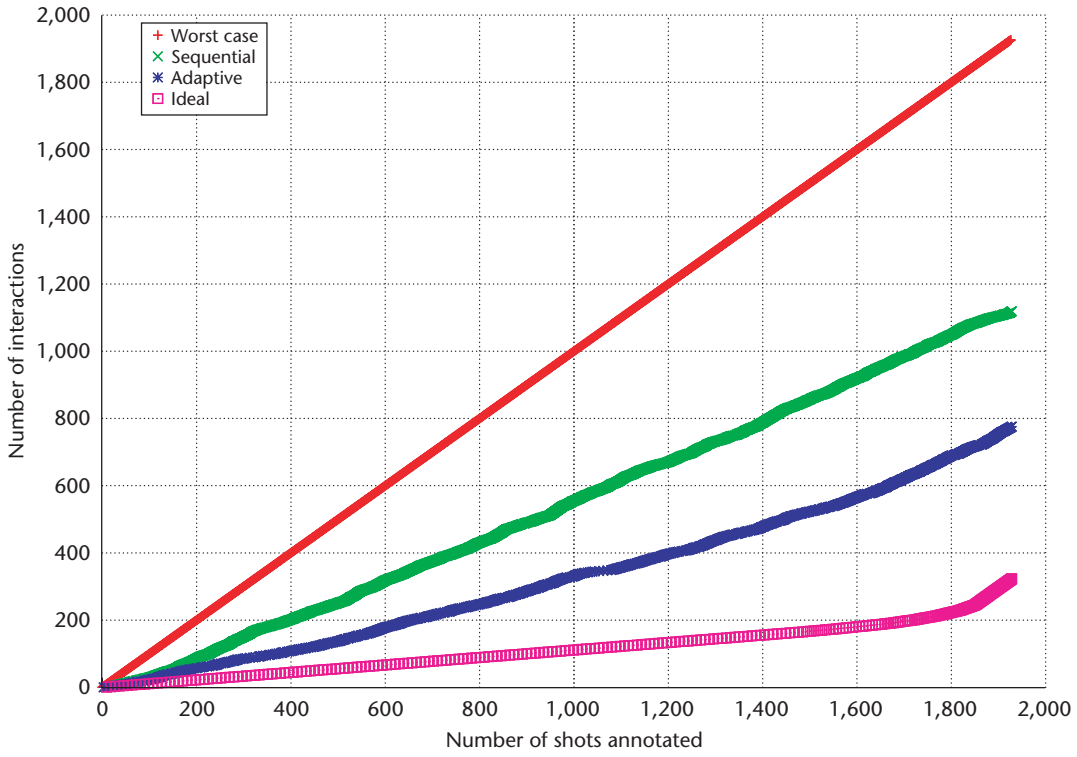

Figure 4. Interactive annotation performance for Shakespeare In Love, *with nine shots shown on the screen. We show results for the sequential and adaptive approach, as well as results for the worst case and ideal case.*

Results for the validation set are similar. For both sets, the performance gain for the adaptive strategy after half an hour is approximately 70 percent. The ideal performance gain is 86 percent on average. For the sequential strategy, results are less consistent. The movies *The Matrix* (57 percent) and *Being John Malkovich* (62 percent) perform significantly better than the other four movies. Inspection of the ground truth shows that within a nine-shot window there's less variety in labels for the two movies. Still, the adaptive strategy performs at least 20 percent better than the sequential strategy, confirming the positive results found for full-length movies.

We measured the effect of using multimodal information for annotation by comparing the use of either visual or label similarity only. For all movies we found similar results. For the full-length movies, the use of visual similarity costs 6 to 14 percent more interactions. Using textual similarity only costs 11 to 13 percent extra interactions. For *Shakespeare in Love* and *Sneakers*, visual similarity performs significantly better than textual similarity. An important reason for the relatively weak performance of visual similarity for *LA Confidential* is the selection of pseudohomogeneous shots. These are shots in the same setting with the same speaker, but with different characters visible. An example is a dialogue in the same setting showing two persons, both individually and together. Especially if just one person is talking, shots from the scene will be similar for all features, reducing annotation efficiency.

We evaluated label selection for the three full-length movies, resulting in 112 calls to the WhoIsWho function in total. Table 2 shows the results for label selection. The movie *Sneakers* profits from its lower complexity—that is, a smaller number of characters than the other movies. We confirmed the lower complexity with the val-

Table 2. Evaluation results for label selection for the adaptive and sequential approaches.

Movie	Number of Celebrities	Number of Unknown Characters	Quantitative Success Rate W_u (%)		Qualitative Success Rate W_c (%)	
			Adaptive	Sequential	Adaptive	Sequential
Shakespeare in Love	7	35	63	51	73	78
Sneakers	7	22	77	64	94	93
LA Confidential	10	40	55	48	73	63

ues for random selection measurement W_r. The value for *Sneakers* is 8 percent, while the value for the other movies is 4 percent. The WhoIsWho function outperforms the random selection.

Conclusion and future directions

The multimodal adaptive approach pays off for interactive character annotation, costing 33 to 50 percent fewer user interactions than the sequential approach. Considering that in practice annotating a movie consumes one day, annotators save a significant amount of time using the i-Notation tool.

The similarity-based shot selection procedure is transparent. There's no need to set thresholds or other magic numbers. Changing settings is limited to the underlying features, such as the number of bins in a histogram, which impacts end results in extreme cases only. Our system is applicable to other movies without modifying any configuration setting. The annotation process results help an application answer any viewer query relating to the characters in a movie.

For finding out who a specific character is, we implemented the label selection technique, WhoIsWho, in the i-Notation system. The function is useful in a relatively small number of cases only, because many characters seldomly speak in the movie, although they appear onscreen frequently. However, in cases where the function provides a name, it's reliable. In addition, WhoIsWho reduces the complexity of the label selection for the remaining character names. In conclusion, the WhoIsWho function proves powerful for label selection.

Our future research will focus on better use of movie structure. Preliminary results show that we can solve the problem of selecting pseudohomogeneous shots by dividing shots into groups with the same label based on the same visual feature used in the current system. The necessary additional step compares shot differences rather than similarities, resulting in two questions for future research: How can the selection of a group of pseudohomogeneous shots be detected? How can the additional comparison step be incorporated into the i-Notation system? **MM**

References

1. J. Korris and M. Macedonia, "The End of Celluloid: Digital Cinema Emerges," *Computer*, vol. 35, no. 4, Apr. 2002, pp. 96-98.
2. J. Vendrig and M. Worring, "Systematic Evaluation of Logical Story Unit Segmentation," *IEEE Trans. Multimedia*, vol. 4, no. 4, Dec. 2002, pp. 492-499.
3. G. Davenport, T. Aguierre Smith, and N. Pincever, "Cinematic Principles for Multimedia," *IEEE Computer Graphics and Applications*, vol. 11, no. 4, July 1991, pp. 67-74.
4. P.J. Jang and A.G. Hauptmann, "Learning to Recognize Speech by Watching Television," *IEEE Intelligent Systems*, vol. 14, no. 5, Sept./Oct. 1999, pp. 51-58.
5. M.-H. Yang, D.J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, Jan. 2002, pp. 34-58.
6. J. Vendrig and M. Worring, *Multimodal Person Identification*, LNCS 2383, Springer Verlag, 2002, pp. 175-185.
7. Y. Rui, T.S. Huang, and S. Mehrotra, "Constructing Table-of-Content for Videos," *Multimedia Systems*, vol. 7, no. 5, Sept. 1999, pp. 359-368.
8. J. Vendrig, M. Worring, and A.W.M. Smeulders, "Filter Image Browsing: Interactive Image Retrieval by Using Database Overviews," *Multimedia Tools and Applications*, vol. 15, no. 1, Sept. 2001, pp. 83-103.
9. R.K. Srihari, "Automatic Indexing and Content-Based Retrieval of Captioned Images," *Computer*, vol. 28, no. 9, Sept. 1995, pp. 49-56.

Jeroen Vendrig is a senior researcher at MediaMill, a University of Amsterdam spin-off in conjunction with the Netherlands Organization for Applied Scientific Research (TNO-TPD) that develops multimedia indexing tools. His research focuses on interactive video segmentation and visualization of video content and retrieval. Vendrig has an MS in business information systems and a PhD in computer science from the University of Amsterdam.

Marcel Worring is a cofounder of MediaMill and an associate professor of computer science at the University of Amsterdam. His main research interests are automatic structuring and indexing of multimedia content for content-based access, exploration, and presentation. Worring has an MS (honors) in computer science from the Free University Amsterdam and a PhD from the University of Amsterdam.

Readers may contact Jeroen Vendrig at vendrig@science.uva.nl.