Image and Vision Computing xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Taking the bite out of automated naming of characters in TV video

Mark Everingham *,[1], Josef Sivic [2], Andrew Zisserman

Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, UK

ARTICLE INFO

Article history:
Received 26 January 2007
Received in revised form 1 October 2007
Accepted 24 April 2008
Available online xxxx

Keywords:
Video indexing
Automatic annotation
Face recognition

ABSTRACT

We investigate the problem of automatically labelling appearances of characters in TV or film material with their names. This is tremendously challenging due to the huge variation in imaged appearance of each character and the weakness and ambiguity of available annotation. However, we demonstrate that high precision can be achieved by combining multiple sources of information, both visual and textual. The principal novelties that we introduce are: (i) automatic generation of time stamped character annotation by aligning subtitles and transcripts; (ii) strengthening the supervisory information by identifying when characters are speaking. In addition, we incorporate complementary cues of face matching and clothing matching to propose common annotations for face tracks, and consider choices of classifier which can potentially correct errors made in the automatic extraction of training data from the weak textual annotation. Results are presented on episodes of the TV series "Buffy the Vampire Slayer".

1. Introduction

The objective of this work is to label television or movie footage with the names of the people present in each frame of the video. As has previously been noted [1,2] such material is extremely challenging visually as characters exhibit significant variation in their imaged appearance due to changes in scale, pose, lighting, expressions, hair style, etc. There are additional problems of poor image quality and motion blur.

We build on previous approaches which have matched frontal faces in order to "discover cast lists" in movies [3] or retrieve shots in a video containing a particular character [1,4] based on image queries. The main novelty we bring is to employ readily available textual annotation for TV and movie footage, in the form of subtitles and transcripts, to *automatically* assign the correct name to each face image.

Alone, neither the script nor the subtitles contain the required information to label the identity of the people in the video – the subtitles record *what* is said, but not by *whom*, whereas the script records who says *what*, but not when. However, by automatic alignment of the two sources, it is possible to extract *who* says what and *when*. Knowledge that a character is speaking then gives a very weak cue that the person may be visible in the video. A key to the success of our method is the novel use of visual speaker detection to leverage cues from the text – *visually* detecting which (if any) character in the video corresponds to the speaker. This gives us sufficient annotated data from which to learn to recognize the other instances of the character.

In addition to effective exploitation of cues from textual annotation, success depends on robust computer vision methods for face processing in video. We propose extensions to our method for connecting faces in video [4], which provides robust face tracks, and a novel extension of the "pictorial structure" method [5] which gives reliable localization of facial features in presence of significant pose variations. This paper is an extended version of [11].

1.1. Related work

Previous work on the recognition of characters in TV or movies has often ignored the availability of textual annotation. In the "cast list discovery" problem [3,6], faces are clustered by appearance, aiming to collect all faces of a particular character into a few pure clusters (ideally one), which must then be assigned a name manually. It remains a challenging task to obtain a small number of clusters per character without merging multiple characters into a single cluster. Other work [2] has addressed finding particular characters specified *a priori* by building a model of a character's appearance from user-provided training data, and efficient retrieval of characters based on example face images [4].

Assigning names given a combination of faces and textual annotation has similarities to the "Faces in the News" labelling of [7]. In that work, faces appearing in images accompanying news stories are tagged with names by making use of the names appearing in the news story text. A clustering approach is taken, initialized by cases for which the news story contains a single name and the accompanying image contains a single (detected) face. Here we are also faced with similar problems in establishing the correspondence between text and faces: ambiguity can arise from deficien-

* Corresponding author. Tel.: +44 113 3435370.
E-mail address: me@comp.leeds.ac.uk (M. Everingham).
[1] Present address: School of Computing, University of Leeds, Leeds LS2 9JT, UK.
[2] Present address: INRIA, WILLOW project-team, Laboratoire d'Informatique de l'Ecole Normale Superieure, CNRS/ENS/INRIA UMR 8548.

0262-8856/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2008.04.018

cies in the face detection, e.g., there may be several characters in a frame but not all their faces are detected, or there may be false positive detections; ambiguity can also arise from the annotation, e.g., in a reaction shot the person speaking (and therefore generating a subtitle) may not be shown.

The combination of face detection and text has also been applied previously to face recognition in video. In [8], transcripts (spoken text without the identity of the speaker) and video of news footage were combined to recognize faces. Much attention was directed at how to predict from a name appearing in the transcript (typically spoken by a news anchor-person) *when* (relatively) the person referred to might appear in the video; addition of a standard face recognition method to this information gave small improvements in accuracy. A recent related approach [9] explicitly restricts the search region of video using the occurrence of a name in the transcript, then applies a clustering approach to find the most-frequently occurring face in that region. A limitation of this approach is that it cannot find a person in parts of the video where their name is not mentioned. A method similar in spirit [10] applies multiple-instance learning instead of a clustering approach. That work also requires that the correct name be among candidates for any particular clip of video, and is further restricted to "monologue" news clips containing a single face.

1.2. Outline

Our method comprises three threads:

(i) Section 2 describes the processing of subtitles and script to obtain proposals for the names of the characters in the video. Mining useful information from each source requires the alignment of the two texts, achieved using a dynamic time warping algorithm.
(ii) Section 3 describes the processing of the video to extract face tracks and accompanying descriptors of face and clothing. As in some previous work in this area [1,3,4] we maintain multiple examples of a person's appearance to cover changes in, e.g., expression and clothing. Robustness to pose, lighting and expression variation in the description of the facial appearance is obtained by localizing facial features and using a parts-based descriptor extracted around the features. We also describe the visual speaker detection method which is pivotal in improving the strength of the supervisory information available from the text.
(iii) Section 4 describes the combination of the textual and visual information to assign names to detected faces in the video. Two classification approaches are considered: a "nearest neighbour" approach [11] which bases classification directly on exemplars extracted by speaker detection, and a support vector machine (SVM) classifier which can potentially correct errors made in speaker detection and prune unhelpful exemplars with poor appearance. Results of the method are reported in Section 5, and further discussion presented in Section 6. Section 7 offers conclusions and proposes directions for future research.

The method is illustrated on three 40 minute episodes of the TV serial "Buffy the Vampire Slayer". The episodes are "Real Me" (season 5, episode 2), "No Place Like Home" (season 5, episode 5), and "Blood Ties" (season 5, episode 13). In all cases there is a principal cast of 12 characters and various others, including vampires (who *are* detected by the face detector).

2. Subtitle and script processing

In order to associate names with characters detected in the video, we use two sources of textual annotation of the video which are easily obtained without further manual intervention: (i) subtitles associated with the video intended for hearing-impaired viewers; (ii) a transcript of the spoken lines in the video. Our aim here is to extract an initial prediction of *who* appears in the video, and *when*.

2.1. Subtitle extraction

The source video used in the experiments reported here was obtained in DVD format, which includes subtitles stored as bitmap images with lossless compression, and corresponding timing information. The subtitle text and time-stamps (Fig. 1) were extracted using the publicly available "SubRip" program [12] which uses a simple table lookup OCR method. Typically the extracted text contains some errors, mainly due to (i) incorrect word segmentation caused by variable length spacing between characters, and (ii) characters indistinguishable in the sans-serif font used without the use of context – primarily "l" and "I". An off-the-shelf spelling correction program was used to reduce the number of such errors.

Although the video used here was obtained in DVD format, subtitles can also be extracted in the same way from digital TV transmissions, which encode the subtitles using a similar lossless bitmap format.

2.2. Script processing

Scripts for the video were obtained from a fan web-site [13]. For the "Buffy the Vampire Slayer" footage used here, there are a number of such fan sites which contain scripts. We stress that for almost any movie or TV series it is possible to find the script on the web, and we expect the text and video processing methods here to generalize well to other genres of video. Straightforward text processing was used to extract the identity of the speaker and corresponding spoken lines from the HTML scripts, by identifying the HTML tags enclosing each script component, for example the speaker names are identified by bold text.

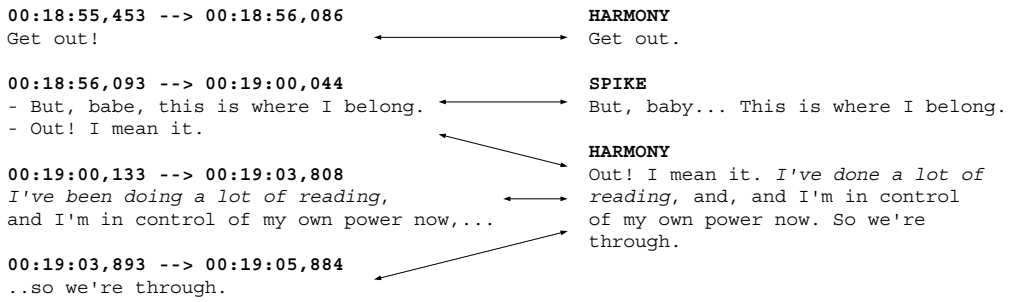


Fig. 1. Alignment of the subtitles (left) and script (right). The subtitles contain spoken lines and exact timing information but no identity. The script contains spoken lines and speaker identity but no timing information. Alignment of the spoken text allows subtitles to be tagged with speaker identity. Note that single script lines may be split across subtitles, and lines spoken by several characters merged into a single subtitle. The transcribed text also differs considerably – note the example shown in italics.

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

While the script contains the spoken lines and the corresponding identity of the speaker (Fig. 1), it contains *no* timing information other than the sequence of spoken lines. For example, in Fig. 1 it is known from the script that the character Harmony speaks, then Spike, but it is not known to which range of frames in the video these events correspond. The processed script thus gives us one of the pieces of information we require: *who* is speaking; the knowledge that someone is speaking will be used as a cue that they may be visible in the video. However, it lacks information of *when* they are speaking. By aligning the script and subtitles on the basis of the spoken lines, the two sources of information can be fused.

2.3. Subtitle and script alignment

Fig. 1 illustrates the alignment of subtitles and script. Note that the transcription of the spoken lines differs somewhat between the two sources. Examples include punctuation, e.g., "Get out!" vs. "Get out." and choices or errors made by the transcriber, e.g., "I've been doing a lot of reading" vs. "I've done a lot of reading". In addition, for the purposes of convenient on-screen viewing, single script lines may have been split across multiple subtitles, or lines spoken by different characters merged into a single subtitle. In order to align the two sources, matching of the spoken lines must allow for these inconsistencies.

A "dynamic time warping" [14] algorithm was used to align the script and subtitles. The two texts are converted into a string of fixed-case, un-punctuated words to reduce the effect of inconsistent casing or punctuation. Writing the subtitle text vertically, and the script text horizontally, the task is to find a path from top-left to bottom-right which moves only forward through either text (since sequence is preserved in the script), and makes as few moves as possible through unequal words. The globally optimal alignment, in terms of the number of mismatched words, is found efficiently using a dynamic programming algorithm. Given such an alignment between *words* of the subtitle and script strings, the task remains of transferring the alignment to the individual elements of each data source – the subtitle lines, and the script lines. A straightforward voting approach was used: the script line corresponding to a subtitle line is defined as the line for which the number of words in correspondence, according to the path found by dynamic time warping, is maximum.

The result of the alignment between subtitles and script is that each script line can be tagged with timing information from the subtitles. For example, in Fig. 1 it is now known from the alignment that the character Harmony speaks from approximately 18 min, 55.5 s to 18 min, 56 s in the video, and the knowledge that she is speaking for this time gives some clue that she *might* also be visible in the corresponding frames of video. Note however, that there will remain some implicit ambiguities in the alignment due to ambiguity in the two texts. An example appears in the second subtitle shown in Fig. 1; here, the person producing the subtitles has merged two spoken lines for convenient on-screen formatting. Although the alignment algorithm correctly assigns the two lines to the characters Spike and Harmony, it is not possible to establish at what time the first line finishes and the second line begins, since this information is lost by the merging of the lines into a single subtitle. Possibilities for resolving such ambiguities are discussed in Section 7.

It transpires that, while knowing that a particular person is speaking at a given time gives some cue that they may be visible in the video, this is at best a *weak* cue. Discussion of the possible *visual* ambiguities is deferred to Section 3.5, where a solution is proposed.

3. Video processing

This section describes the video processing component of our method. The aim here is to find people in the video and extract descriptors of their appearance which can be used to match the same person across different shots of the video. The task of assigning *names* to each person found is described in Section 4.

3.1. Face detection and tracking

The method proposed here uses face detection as the first stage of processing. A frontal face detector [15] is run on every frame of the video, and to achieve a low false positive rate, a conservative threshold on detection confidence is used. The output is a set of bounding boxes of detected faces for each frame. Example detections can be seen in Figs. 3a and 12. The use of a frontal face detector restricts the video content we can label to frontal faces, but typically gives much greater reliability of detection than is currently obtainable using multi-view face detection [16]. Methods for "person" detection have also been proposed [15,17,18] but are typically poorly applicable to TV and movie footage since many shots contain only close-ups or "head and shoulders" views, whereas person detection has concentrated on views of the whole body, for example pedestrians.

A typical episode of a TV series contains around 25,000 detected faces but these arise from just a few hundred "tracks" of a particular character each in a single shot. A face track [4] represents the appearance of a single character across multiple, not necessarily contiguous, frames of the video. Basing the learning and recognition of people on these tracks rather than individual faces offers two advantages: (i) the volume of data to be classified is reduced; (ii) stronger appearance models of a character can be built, since a single track provides multiple examples of the person's appearance. Consequently, face tracks are used from here on and define the granularity of the labelling problem.

Obtaining face tracks requires establishing that two faces in different frames of a shot correspond to the same character. Because a face track is restricted to a single shot this is a much simpler problem than the general task of establishing that two face images arise from the same person, since *motion* can be used to establish the correspondence. Face tracks are obtained as follows: first, for each shot, the Kanade–Lucas–Tomasi (KLT) tracker [19] is applied. This algorithm detects interest points in the first frame of the shot and propagates them to succeeding frames based on local appearance matching. Points which cannot reliably be propagated from one frame to the next are discarded and replaced with new points. The output is a set of point tracks starting at some frame in the shot and continuing until some later frame. For a given pair of faces A and B, in different frames (since faces in a single frame are assumed not to belong to the same character), the relevant point tracks can be assigned to one of three classes: (a) track intersects both A and B; (b) track intersects A but not B; (c) track intersects B but not A. Intersection of a point track and a face is defined by the point lying within the face bounding box in the corresponding frame. A confidence measure that the two faces A and B belong to the same character is then defined as the number of type (a) tracks divided by the total number of type (b) and (c) tracks – this is the ratio of tracks linking the faces to tracks which intersect only one face. Using this confidence measure, defined between every pair of face detections in the shot, faces are merged into face tracks by applying a standard agglomerative clustering algorithm. A threshold on the proportion of intersecting tracks is set to prevent the clustering algorithm merging unconnected faces; in all experiments this was set to 0.5. Fig. 2 shows examples of face tracks ob-

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

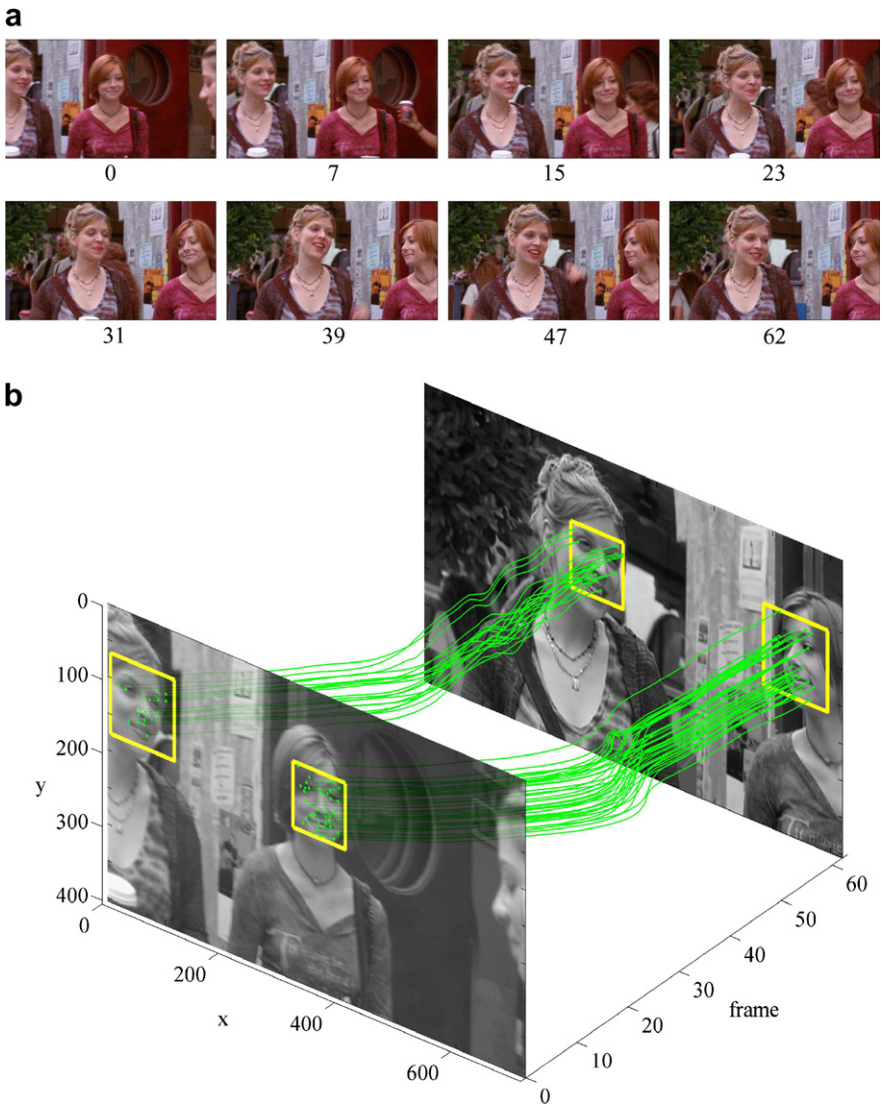


Fig. 2. Face tracking by point tracking. (a) Eight frames from a sequence of 63 frames where the camera first moves left (frames 0–30) and then stays still (frames 31–62). Corresponding frame numbers are shown below each frame. Note the changing facial expression of the actor on the left (frames 31–62) and the changing head pose of the actor on the right (around frame 31). (b) Trajectories of points tracked on the actors' faces shown as curves in the video volume between the first and last frame. Additional tracks which do not intersect the faces are omitted for clarity.

tained for a shot containing significant camera motion and variation in head pose and facial expression.

This simple tracking procedure is extremely robust. Compared to an approach of tracking the face directly using some face-specific or general appearance-based method the point feature-based approach has two advantages: (i) the method can establish matches between faces where the face has not been continuously detected due to pose variation or expression change. This is challenging for most tracking methods which do not reliably recover from occlusion; (ii) the method does not suffer from the "drift" common in object trackers, where the appearance model maintained by the tracker drifts onto another object in the video. In the proposed method, points are tracked in an "unbiased" manner without reference to the face detections such that there is no tendency to "hallucinate" by failing to terminate a track. It is worth noting that we applied a variant of the tracking method used here with success in previous work on face matching [4]. In that work the basic point tracker used affine covariant regions to provide more robust matching of features between frames. While the affine invariant method can potentially obtain longer tracks through more severe rotation or deformation of the face, its computational expense is considerably greater than that of the KLT method used here.

By tracking, the initial set of face detections is reduced to the order of 500 tracks, and short tracks (less than 10 frames, equivalent to 400 ms), which are most often due to false positive face detections, are discarded.

3.1.1. Shot change detection

As noted, the face tracking method is applied to individual shots of the video. Shot changes were automatically detected using a simple method of thresholding the distance between colour histograms computed for consecutive frames of the video. The shot change detection method gives some false positive detections, e.g., when a shot contains fast motion, and potentially might miss "fade" shot changes, although none appear in the Buffy video used here. However, the accuracy of shot detection is not at all critical to the overall performance of our method: (i) false positive shot changes merely cause splitting of face tracks, which typically can be "repaired" by matching the face appearance across the illusory

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

shot change; (ii) false negative shot changes are resolved by the point tracker, which typically will correctly fail to track points across a (missed) shot change.

3.2. Facial feature localization

The output of the face detector gives an approximate location and scale of the face. Extracting descriptors directly from this output would result in an unstable descriptor, due both to the approximate nature of the face detector output, for example the estimated scale fluctuates with variation in head pose, and the imaged face implicitly varies with changes in pose. A more stable description of the face appearance is obtained by basing it on the position of the facial features in the image. Nine facial features are located, see Fig. 3b – the left and right corners of each eye, the two nostrils and the tip of the nose, and the left and right corners of the mouth. Additional features corresponding to the centres of the eyes, a point between the eyes, and the centre of the mouth, are defined relative to the located features.

To locate the features, a model combining a generative representation of the feature positions with a discriminative representation of the feature appearance is applied.

3.2.1. Model of feature position and appearance

A variant of the probabilistic parts-based "pictorial structure" model [5] is used to model the joint position (shape) and appearance of the facial features. To simplify the model, two assumptions are made: (i) the appearance of each feature is assumed independent of the appearance of other features; (ii) the appearance of a feature is independent of its position. Under these assumptions, the confidence in an assignment F of positions to each facial feature can be written as a likelihood ratio

$$P(F|\mathbf{p}_1,\ldots,\mathbf{p}_n) \propto p(\mathbf{p}_1,\ldots,\mathbf{p}_n|F)\prod_{i=1}^{n}\frac{p(\mathbf{a}_i|F)}{p(\mathbf{a}_i|\overline{F})} \quad (1)$$

where $\mathbf{p}_i$ denotes the position of feature i in the detected face region and $\mathbf{a}_i$ denotes the image appearance about that point.

The joint position of the features $p(\mathbf{p}_1,\ldots,\mathbf{p}_n|F)$ is modelled as a mixture of Gaussian trees. The likelihood-ratio of the appearance terms is modelled using a discriminative classifier.

3.2.2. Model of appearance

For each facial feature, for example the corner of an eye, a feature/non-feature classifier was trained using a multiple-instance variant of the AdaBoost learning algorithm, which produces a strong classifier as a linear combination of "weak" classifiers. The multiple-instance variant iteratively updates labels on the training data, compensating for small localization errors in the training images. The features used as weak classifiers are the "Haar-like" features proposed by Viola and Jones [20] which can be computed efficiently using the integral image. The classifier is applied to the output of the face detector in a sliding window fashion, and the classifier output can be considered an approximate log-likelihood ratio which can be directly substituted into Eq. (1).

3.2.3. Model of position

The joint position of the facial features is modelled using a mixture of Gaussian trees, a Gaussian mixture model in which the covariance of each component of the mixture model is restricted to form a tree structure with each variable dependent on a single "parent" variable [21]. The model is an extension of the single tree proposed in [5], which was applied to facial feature localization using simple generative appearance models, and the recent combination of a single tree with a discriminative appearance model [22]. The use of a mixture of trees improves the ability of the model to capture pose variation; three mixture components were used, and found to correspond approximately to frontal views and views facing somewhat to the left or right. At training time, the model is fitted using an Expectation Maximization algorithm [21]. At testing time, efficient search for the feature positions using distance transform methods [5] is enabled by the use of tree-structured covariance in each mixture component.

A collection of annotated consumer photographs of faces [23], disjoint to the video data reported here, was used to fit the parameters of the position model and train the facial feature classifiers. The confidence in the feature localization (Eq. (1)) proves to be an effective measure for determining whether the face detector output is actually a face or a false positive detection, and is thresholded to prune false positive face detections.

Fig. 3 shows examples of the face detection and feature localization. Note that the "frontal" face detector also detects some faces with significant out-of-plane rotation. The facial features can be located with high reliability in the faces despite variation in scale, pose, lighting, and facial expression.

3.3. Representing face appearance

A representation of the face appearance is extracted by computing descriptors of the local appearance of the face around each of the located facial features. Extracting descriptors based on the feature locations [1,4] gives robustness to pose variation, lighting, and partial occlusion compared to a global face descriptor [24,25]. Errors may be introduced by incorrect localization of the features, which become more difficult to localize in extremely non-frontal poses, but using a frontal face detector restricts this possibility.

Before extracting descriptors, the face region proposed by the face detector is further geometrically normalized to reduce the scale uncertainty in the detector output and the effect of pose variation, e.g., in-plane rotation. An affine transformation is estimated which transforms the located facial feature points to a canonical set of feature positions (roughly those of a frontal vertical face). Appearance descriptors are computed around each facial feature within a circular support region in the canonical reference frame. Under the affine transformation each circle in the canonical frame corresponds to an ellipse in the original frame. A simple pixel-wise descriptor of the local appearance around a facial feature is extracted by taking the vector of pixels in the elliptical region and normalizing (so that the intensity has zero mean and unit variance) to obtain local photometric invariance. The descriptor for the face

(a) Face detections in original frames (b) Localized facial features

Fig. 3. Face detection and facial feature localization. Note the low resolution, non-frontal pose and challenging lighting in the example on the right.

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

is then formed by concatenating the descriptors for each facial feature. The distance between a pair of face descriptors is computed using Euclidean distance. Fig. 4 shows examples of the elliptical regions from which the descriptor is extracted, and the corresponding normalized image regions.

It is natural to consider the use of more established image representations commonly used in face recognition, for example so-called Eigenfaces [26] or Fisherfaces [27], or alternative local feature representations such as SIFT [28] which have successfully been used in feature-matching tasks including face matching [4], especially considering the simplicity of the descriptor proposed here. In classical face recognition work, two aspects differ from the situation here: (i) changes in pose, expression and lighting are typically assumed small; (ii) while multiple images of various people may be available for training (e.g., for learning a PCA basis), typically only a *single* "gallery" image is available to model a particular person [29]. Eigenface methods offer some invariance to very small changes in pose due to the empirically band-pass nature of the basis, but cannot cope with large variations in pose; Fisherface methods are typically very unstable in the presence of pose variation due to the empirically high-pass nature of the basis. The second point, however, is key: the use of a *single* image as the model for a person. This requires that the descriptor generalizes far from that single image if success is to be obtained for variations in pose and expression. However, in the domain considered here, as described in Sections 3.5 and 4, *multiple* exemplars are extracted as the model of the person. This requires less generalization from the descriptor, and excessive generalization will degrade performance. We return to this point in Section 6.

3.4. Representing clothing appearance

In some cases, matching the appearance of the face is extremely challenging because of different expression, pose, lighting or motion blur. Additional cues to matching identity can be derived by representing the appearance of the clothing [30–33]. We use a simple model of clothing location relative to the face and represent colour alone here [30,31]. Some recent work has also accounted explicitly for varying pose of the person in locating the clothing [32] and incorporated texture features [33].

As shown in Fig. 5, for each face detection a bounding box which is expected to contain the clothing of the corresponding character is predicted. The size and position of the box are fixed relative to the position and scale of the face detection. Within the predicted clothing box a colour histogram is computed as a descriptor of the clothing. We used the YCbCr colour space which has some advantage over RGB in de-correlating the colour components. The histograms had 16 bins per colour channel. The distance between a pair of clothing descriptors was computed using the chi-squared measure [34]. Fig. 5 shows examples which are challenging to match based on face appearance alone, but which can be matched correctly using clothing.

Of course, while the face of a character can be considered something unique to that character and in some sense constant (though note that characters in this TV series who are vampires change their facial appearance considerably), a character may, and does, change their clothing within an episode. This means that while similar clothing appearance suggests the same character, observing different clothing does not necessarily imply a different character. As described in Section 5, we found that a straightforward weighting of the clothing appearance relative to the face appearance proved effective here.

3.5. Speaker detection

The aligned subtitle and script annotation (Section 2.3) proposes one or more possible speaker names for each frame of the video containing some speech. Note that this annotation says nothing about *where* in the frame the speaker appears, or indeed whether they are in fact visible at all. With respect to the faces in the video, the annotation derived from text alone proves to be extremely ambiguous. There are three main forms of ambiguity, illustrated in Fig. 6: (i) there might be several detected faces present in the frame – the script does not specify which one corresponds to the speaker. Fig. 6a shows such a case, where the script tells us that Tara is speaking, but two faces are visible in the frame – which (if any) is Tara? (ii) even in the case of a single face detection in the frame the actual speaker might be undetected by the frontal face detector. Fig. 6b shows an example, where Buffy is speaking but is undetected because of the profile pose. Assuming that the single detected face (Willow) corresponds to the speaker would be an error in this case; (iii) the frame may be part of a "reaction shot" where the speaker is not present in the frame at all. Fig. 6b shows an example, where we see Willow and Buffy's reaction to what is said by Tara, who is off-screen "behind the camera".

The goal here is to enhance the annotation provided by the script, resolving these ambiguities by identifying the speaker using *visual* information. By confirming visually that a particular face in the image is that of someone speaking, the correspondence between that face and the name of the speaker given by the script is established.

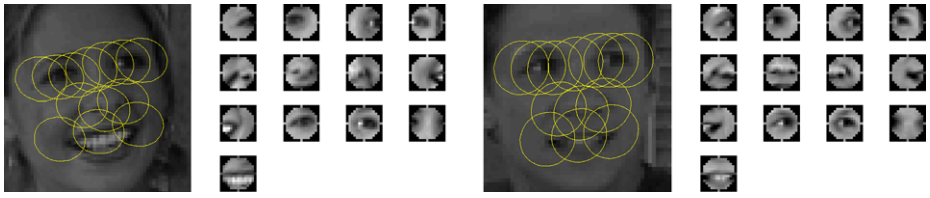


Fig. 4. Face appearance descriptors. For the two faces shown, ellipses show the affine-transformed regions around the localized facial features from which the descriptor is computed. Patches on the right show the extracted image regions.

Fig. 5. Matching characters across shots using clothing appearance. In the two examples shown the face is difficult to match because of the variation in pose, facial expression and motion blur. The strongly coloured clothing allows correct matches to be established in these cases.

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

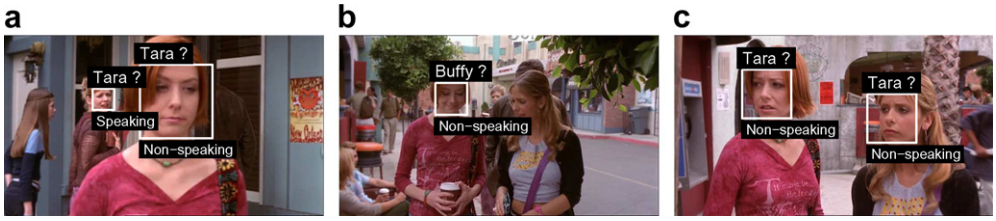


Fig. 6. Examples of speaker ambiguity. In all the cases shown the aligned script proposes a single name, shown above the face detections. (a) Two faces are detected but only one person is speaking. (b) A single face is detected but the speaker is actually missed by the frontal face detector. (c) A "reaction shot" – the speaker is not visible in the frame. The (correct) output of the speaker detection algorithm is shown below each face detection.

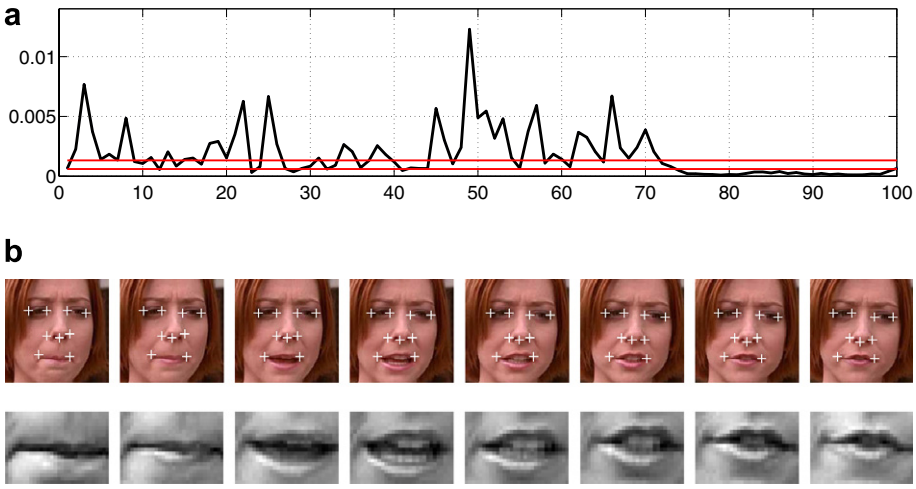


Fig. 7. Speaker identification by detecting lip movement. (a) Inter-frame differences for a face track of 101 face detections. The character is speaking between frames 1–70 and remains silent for the rest of the track. The two horizontal lines indicate the "speaking" (top) and "non-speaking" (bottom) thresholds, respectively. (b) Top row: Extracted face detections with facial feature points overlaid for frames 47–54. Bottom row: Corresponding extracted mouth regions.

Visual speaker detection [35] is achieved here by the intuitive approach of finding face detections with significant lip motion. A rectangular mouth region within each face detection is identified using the located mouth corners (Section 3.2). Examples of the extracted mouth region are shown in Fig. 7b. The sum of squared difference of the pixel values within the region is computed between the current and previous frame as a measure of the amount of motion in the mouth region. To achieve moderate translation invariance, giving some robustness to pose variation of the head, the inter-frame difference is computed over a search region around the mouth region in the current frame and the minimum taken. Fig. 7a shows a plot of the inter-frame difference for a face track where the character speaks then remains silent.

Two thresholds on the inter-frame difference are set to classify face detections into "speaking" (difference above a high threshold), "non-speaking" (difference below a low threshold) and "refuse to predict" (difference between the thresholds). Thresholds were set by eye and kept fixed for all the experiments reported here – it should be noted that generating ground truth for speaking/non-speaking so that these thresholds could be set systematically is in general quite difficult because of natural pauses in the speech and the production of sound with little movement of the lips. This simple lip motion detection algorithm works well in practice as illustrated in Fig. 7. Fig. 8 shows further examples where the method correctly assigns a class "non-speaking" despite significant changes in head pose and mouth shape (smiling). Note that in choosing the method and thresholds it is somewhat more important to achieve a low false positive (detector predicts speaking when character is silent) rate than false negative rate. As discussed in Section 4.2, false positive speaker detections cause incorrectly labelled faces to enter the set of exemplars used for naming, which may propagate incorrect names to other face detections.

The speaker detector produces a classification for each frame of a face track. Names proposed by the script for the corresponding face detections classified as speaking are accumulated into a single set of names for the entire face track. In many cases this set contains just a single name, but there are also cases with multiple names, due to merging of script lines into a single subtitle (Section 2.3) and imprecise timing of the subtitles relative to the video.

4. Naming by classification

The combination of subtitle/script alignment and speaker detection gives a number of "exemplar" face tracks for which, with high probability, the single proposed name is correct. Fig. 9 shows examples of exemplar face tracks extracted for two characters. Note that each face track consists of multiple face detections, so the number of exemplar faces is much greater than the number of tracks, as shown in the figure.

The overall naming problem is effectively transformed into a standard supervised classification problem: for some tracks, the corresponding name (class) is extracted from the text and speaker detection, with high probability of being correct (Section 5.1); from these tracks a model or classifier may be built for each character in the video; this classifier is then applied to assign names to tracks which have no, or an uncertain, proposed name.

We consider here two classification methods. First, a "nearest neighbour" method presented in an earlier version of this work [11]; second, use of a support vector machine (SVM) classifier

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

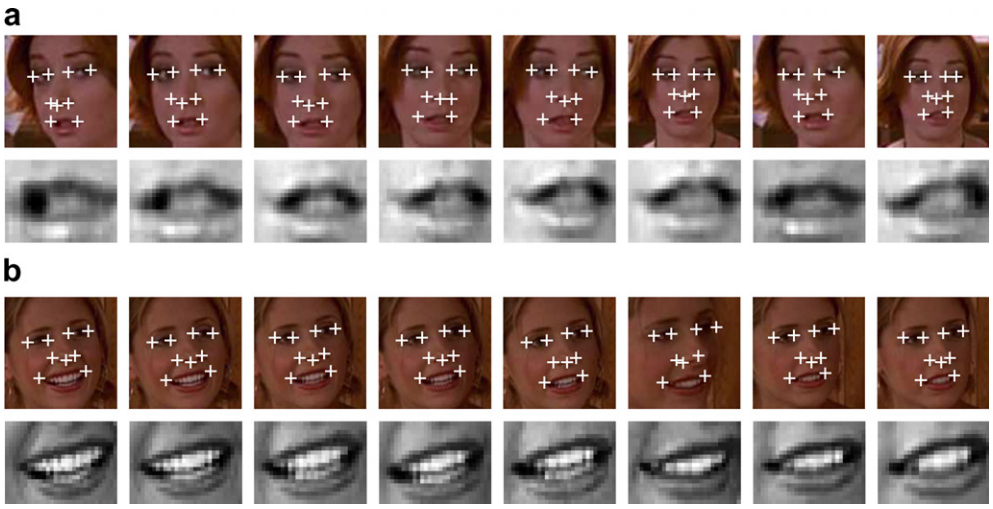


Fig. 8. Correct classification of tracks as "non-speaking". Examples of two face tracks are shown. (a) Frames 1,6,11,...,36 from a 44 frames long face track. All frames in this face track are correctly classified as "non-speaking" despite significant head pose variation. (b) Frames 1,11,21,...,71 from a 75 frames long face track. The track is correctly identified as "non-speaking" despite the shape and appearance variations in the mouth due to expression change (smiling). 73 frames are classified as "non-speaking" and 2 as "refuse to predict". In both (a) and (b) the top row shows the extracted face detections with facial features overlaid and the bottom row shows the corresponding extracted mouth regions.

(a) Buffy (2,300 faces) (b) Willow (1,222 faces)

Fig. 9. Examples of exemplars for two of the main characters. Each track may consist of tens of faces – a single example is shown for each track. The total number of exemplar faces for each character is shown in parentheses.

which can, to some extent, cope with errors in the names obtained from speaker detection. Central to both methods is that the model for a character has multiple modes (in the sense of density), consisting of a (weighted) set of exemplars in appearance space. This allows the model to capture distinct "phases" of a person's appearance, for example mouth open vs. mouth closed. An alternative view is that the multiple modes of the model represent sparse samples on an underlying person-specific appearance manifold. Note that this choice of multi-modal model is possible because the subtitle/script processing and speaker detection gives multiple examples of a character's appearance without the need for further manual intervention. This is in distinct contrast to classical face recognition where the number of examples of an individual's appearance is typically very small (often one) but only a limited range of pose, expression, and lighting is considered.

4.1. Similarity measure

Common to the two classification methods considered here is the definition of a similarity measure between a pair of face tracks. Recall that a face track consists of a bag of face and clothing descriptors, one per frame of the track (Section 3.1), and that measures of the distance between a pair of face descriptors (Section 3.3) and clothing descriptors (Section 3.4) have been defined.

Given a pair of "person" detections (faces and associated clothing) p_i and p_j, and the definitions for the distance between face descriptors d_f and clothing descriptors d_c, we define the similarity $s(p_i, p_j)$ between the two persons as:

$$s(p_i, p_j) = \exp\left\{-\frac{d_f(p_i, p_j)}{2\sigma_f^2}\right\} \exp\left\{-\frac{d_c(p_i, p_j)}{2\sigma_c^2}\right\} \quad (2)$$

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

The scale factors σ_f and σ_c control two aspects: (i) the relative influence of the face and clothing descriptors, and (ii) the overall "peakiness" of the similarity measure, that is how quickly the similarity decays about a pair of faces. The relevance of the latter will become clear in Section 4.2.

The similarity $S(F_i, F_j)$ between a pair of face *tracks* F_i and F_j is defined based on the person similarity as:

$$S(F_i, F_j) = \max_{p_i \in F_i, p_j \in F_j} s(p_i, p_j) \quad (3)$$

This defines the similarity between a pair of face tracks as the maximum similarity over any pair of person descriptors taken across the tracks, and has also been referred to as the "min–min" distance [4]. Note, we are assuming here that a good match requires a similarity of both face and clothing. Other possibilities could also be considered, for example that a track corresponds to the same character if the faces have a high similarity even if the clothing does not (to allow for unobserved changes of clothing).

Equipped with these definitions and suitable choice of constants, the similarity between all pairs of face tracks can be computed.

4.2. "Nearest neighbour" classifier

The first classification method we investigate, first reported in [11], uses a "nearest neighbour" approach. Let us define the name proposed for a track F_j by the text processing and speaker detection as n_j. A tuple of face track and corresponding name will be referred to as an exemplar. We then define the "quasi-likelihood" that an unlabelled track F_u arose from the person with name λ_i as:

$$p(F_u|\lambda_i) = \max_{F_j : n_j = \lambda_i} S(F_u, F_j) \quad (4)$$

This definition is "nearest neighbour" in that only the similarity to the most similar exemplar with a given name is used to assign the likelihood. Assuming that the person associated with each name λ_j may appear in the video with equal prior probability, and applying Bayes' rule, we can derive an approximation of the posterior probability that the track should be assigned the name λ_i:

$$P(\lambda_i|F_u) = \frac{p(F_u|\lambda_i)}{\sum_j p(F_u|\lambda_j)} \quad (5)$$

A predicted name is then assigned to the track as the name λ_i for which the posterior probability $P(\lambda_i|F_u)$ is maximal. Note that this is equivalent to the name for which the likelihood (Eq. (4)) is maximum. However, the utility in defining an approximation of the posterior probability (Eq. (5)) is that it gives an indication of the certainty of the predicted name – if a given face track is similar to exemplars for several characters, the posterior probability for each name falls, indicating the uncertainty in the prediction. It is in defining the posterior that the overall scale of the face and clothing distances (Eq. (2)) becomes relevant, controlling the scale at which the difference between two similar exemplars is considered "uncertain".

By *thresholding* the posterior, a "refusal to predict" mechanism is implemented – faces for which the certainty of naming does not reach some threshold will be left unlabelled; this decreases the recall of the method but improves the accuracy of the labelled tracks. In Section 5 the resulting precision/recall tradeoff is reported.

The "nearest neighbour" classifier described here has appeal in its simplicity, and captures the multi-modal distribution of appearance for a single character which we advocate; it also captures the notion that some tracks may be implicitly difficult to label reliably, and might best be left unlabelled. However, there are two potential weaknesses with the method: (i) it is assumed that the names assigned to exemplar tracks by the text processing and speaker detection are *correct*; (ii) it is assumed that all exemplar appearances are equally valid, e.g., regardless of whether they are blurred, show particularly extreme facial expressions, are partially occluded, etc. Both these assumptions may cause errors since the prediction made for an unlabelled track is made on the basis of the *single* nearest exemplar, and cannot be corrected.

4.3. SVM classifier

A possible solution to the assumptions made in the nearest neighbour classifier we have investigated is the use of a SVM classifier (see [36]). In this approach, the same definition of similarity between face tracks is retained, but is now used as a kernel for the SVM. One SVM is trained per name using a 1-vs-all scheme. All the exemplar tracks for that name are used as positive data, and the exemplars for all other names provide the negative training data. The SVM defines the confidence $Q(\lambda_i|F_u)$ that the name λ_i should be assigned to an unlabelled track F_u as:

$$Q(\lambda_i|F_u) = \sum_j \mathbb{W}_{ij} S(F_u, F_j) + k_i \quad (6)$$

where $\mathbb{W}_{ij}$ is the weight assigned to exemplar j for the name λ_i, and k_i is a (bias) constant. Note that the form of the confidence measure is similar to that of the likelihood defined in the nearest neighbour model (Eq. (4)). The max function is replaced with a sum, analogous to the choice of nearest neighbour density estimator versus a Parzen estimate (see [37]). Additionally, weights are introduced for *all* exemplars, so that the confidence depends on both the positive and negative data (not only on the closest positive example as in Eq. (4)).

The potential strength in the SVM method comes then not from the form of discriminant, but the criterion used to choose the weights $\mathbb{W}$. The SVM training minimizes a weighted sum of two terms: the margin of the classifier on the training set and a penalty on the norm of the weight vector $\mathbb{W}_i$. This latter term regularizes the solution, penalizing "non-smooth" discriminants. The effect is that elements of $\mathbb{W}$ may become small or zero, effectively discarding "outlier" exemplars which may have either incorrect names assigned by speaker detection, or have extreme or non-discriminative appearance which does not aid classification in general. The SVM can thus potentially correct errors made in the names proposed by the text processing and speaker detection, increasing the accuracy in the name assignment both in the labelled exemplar tracks and unlabelled tracks.

To implement the SVM method we used the publicly available LIBSVM software [38], with a custom kernel defined by the track similarity measure of Eq. (3). The same values for the parameters (σ_f, etc) are used as in the nearest neighbour classifier. The "refusal to predict" mechanism was implemented by thresholding the maximum of the confidence $Q(\lambda_i|F_u)$ over names λ_i.

5. Experimental results

The proposed method was applied to three episodes of "Buffy the Vampire Slayer" – in total around two hours of video. Episode 05-02 contains 62,157 frames in which 25,277 faces were detected, forming 516 face tracks; episode 05-05 contains 64,083 frames, 24,170 faces, and 477 face tracks; episode 05-13 contains 64,075 frames, 26,826 faces, and 533 face tracks.

Ground truth names for every face detection were produced by hand. While the task of assigning ground truth to every one of around 75,000 face detections might appear daunting, the use of the face tracking algorithm (Section 3.1) makes this a relatively cheap procedure in terms of time. A two stage approach was used:

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

first, all face tracks are visually checked to ensure that they contain only a single character. As noted in Section 3.1 the tracking algorithm proves extremely reliable, and in practice no false merges of tracks are found, but an interface was provided to manually split tracks in the case that errors occurred. Second, a single ground truth name is assigned to every face detection making up that track. This approach reduces the task of ground truth labelling from that of labelling 75,000 faces to around 1500 tracks.

The ground truth cast list has twelve named characters: Anya, Buffy, Dawn, Giles, Glory, Harmony, Joyce, Riley, Spike, Tara, Willow, Xander. In addition, a single name "Other" is applied to faces of other people appearing in the video – this includes un-named incidental characters and extras. False positive face detections are assigned the name "FalsePositive". To be considered a correct name, the algorithm must distinguish between the main characters, unnamed characters and false positive face detections. It should be noted that, while the set of people to be distinguished is smaller than might be used in classical face recognition research where a "gallery" of 100 people might be typical, the imaging conditions (pose, expression, lighting, etc.) are far more varied in the domain considered here, making this a challenging task.

Note that ground truth is only established for the face detections produced by the frontal face detector used [15] (whether true or false positive). The results reported here, as in previous work [4], are therefore relative to the proportion of appearances of a character detected by a state-of-the-art frontal face detector. Section 7 discusses the question of how many of the actual appearances of a character in any pose, for example in profile views or facing away from the camera, are represented by this proportion.

The parameters of the speaker detection, weighting terms in the quasi-likelihood (Eq. (4)), and weight parameter in SVM learning were coarsely tuned on episode 05-02 and all parameters were left unchanged for the other episodes. *No* manual annotation of any data was performed other than to evaluate the method (ground truth label for each face track).

5.1. Speaker detection

We first report the accuracy of the speaker detection algorithm. The performance of this part of the method is important since, for the nearest neighbour classifier (Section 4.2), errors in speaker detection cannot be corrected. The speaker detection method (Section 3.5) allows for three outputs: "speaking", "non-speaking" and "refuse to predict". Across the three episodes, the method labels around 25% of face tracks as speaking, and of those the corresponding label from the script has around 90% accuracy.

Fig. 10 shows two examples where the speaker detection fails. In Fig. 10a, the character shouts and is correctly identified as "speaking" but the timing information on the subtitles is inaccurate such that the face is attributed to a character who appears at the beginning of the next shot. Ambiguities such as this occur because the timing information on the subtitles does not precisely indicate the time at which a spoken line starts and finishes, for example when a long line is spoken quickly the subtitle display time may have been extended to facilitate reading. In Fig. 10b, the face is incorrectly classified as "speaking". In this case the shot is a "reaction shot" in which the visible character (silently) gasps in shock at what is being said by another character off-screen. Such cases of speech-like motion are difficult to detect based on visual information alone. Other errors in the speaker detection are due to complex appearance changes of the mouth region such as partial occlusion by another person, severe head pose changes, and complex lighting effects (e.g., a moving shadow cast by another person). Such changes cause large apparent motion of the mouth which is incorrectly classified as speech. Greater accuracy in such cases might be obtained by using a more complete model of the mouth region, and is left for future work.

5.2. Naming accuracy

We turn now to the performance of the entire method on the naming task. In this section we concentrate on the performance of the nearest neighbour method (Section 4.2) previously proposed [11], and comparison to baseline methods based on the subtitle/script alone. In the next section the performance of the SVM method (Section 4.3) and the influence of errors in speaker detection are considered.

Fig. 11 shows precision/recall curves for the proposed nearest neighbour method. Quantitative results at several levels of recall are shown in Table 1. The term "recall" is used here to mean the proportion of tracks which are assigned a name after applying the "refusal to predict" mechanism (Section 4). The term "precision" refers to the proportion of correctly labelled tracks. Note that reporting performance in terms of face tracks, rather than individual face detections, gives a more meaningful assessment since the faces in a track can be associated in a rather straightforward manner by tracking (Section 3.1). Reporting performance by individual face detections would allow the presence of some long tracks with little or unchallenging motion to bias the apparent results.

These results illustrate the benefit of learning from the exemplars to label other tracks. The recall and precision of the exemplars alone (i.e., only those tracks for which speaker detection assigns a name from the text, without any visual labelling of other tracks) is 31.0% recall, 90.6% precision for episode 05-02; 27.9% recall, 91.7% precision for episode 05-05; 34.5% recall, 82.1% precision for episode 05-13.

Two baseline methods were compared to the proposed method:

(i) "Prior" – label all tracks with the name which occurs most often in the script (e.g., Buffy). It is expected that the main characters will appear in the video rather more frequently

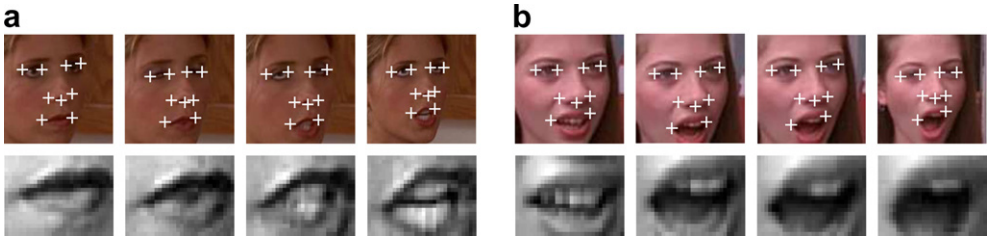
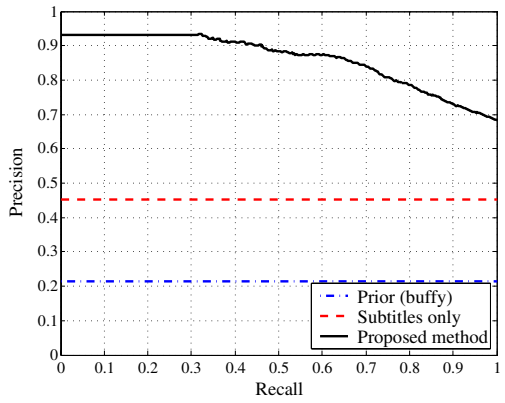
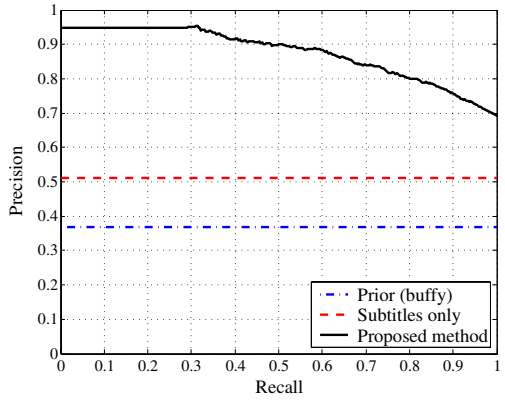


Fig. 10. Examples of errors in speaker identification. (a) Four frames from a 19 frames long face track where the actor shouts and is detected as speaking. Despite valid visual detection, due to inaccurate subtitle timing information this shout is attributed to a person speaking in the next shot. (b) Four frames from a 23 frames long face track where the actor silently opens her mouth and is wrongly classified as speaking. In both (a) and (b) the top row shows extracted face detections with facial features overlaid and the bottom row shows the corresponding extracted mouth regions.

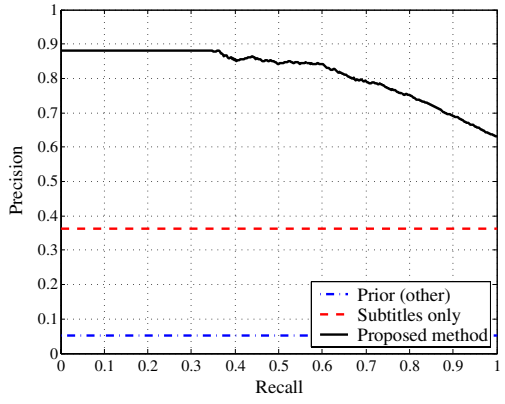
Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018



(a) Episode 05-02



(b) Episode 05-05



(c) Episode 05-13

Fig. 11. Precision/recall curves for three episodes. Recall is the proportion of face tracks which are assigned labels by the proposed method at a given confidence level, and precision the proportion of correctly labelled tracks. The graphs show the performance of the proposed method and two baseline methods using the subtitles to propose names for each face track (see text for details).

than secondary characters so it is important to establish the extent to which this is true so that the true accuracy of the method can be distinguished from "chance".

(ii) "Subtitles only" – label any tracks with proposed names from the script (not using speaker identification) as one of the proposed names, breaking ties by the prior probability of the name occurring in the script; label tracks with no proposed names as the most-frequently occurring name (e.g., Buffy). This baseline allows us to assess to what extent the visual processing improves accuracy over the use of text alone. It is interesting to note that in previous work [8] which combined transcripts of news footage with Eigenface-based face recognition, only small improvements in accuracy were obtained by incorporating visual face recognition.

As expected, the distribution over the people appearing in the video is far from uniform – labelling all face tracks "Buffy" gives correct results 21.9% of the time in episode 05-02 and 36.9% of the time in episode 05-05. In episode 05-13 minor characters dominate, and the prior labels only 5.1% of tracks correctly. The cues from the text alone (subtitles and script) increase this accuracy to around 35–50% in each episode. While an improvement over chance, this reveals the relative weakness of the text as a cue to identity.

Using the proposed nearest neighbour method, if we are forced to assign a name to *all* face tracks, the accuracy obtained is around 63–69% across episodes. Requiring only 80% of tracks to be labelled increases the accuracy to around 75–80%. We consider these results extremely promising given the challenging nature of this data.

Fig. 12 shows some examples of correctly detected and named faces. Note that correct naming is achieved over a very wide range of scale, pose, facial expression and lighting. The ability of the proposed method to give good results in such conditions is attributable to (i) the automatic extraction of exemplars throughout the video such that the changes in appearance are, to some extent, spanned by the exemplar set; (ii) the use of a multi-modal model of a person's appearance which enables a representation of the distinctly different appearances to be maintained.

5.3. SVM method and errors in speaker detection

As noted in Section 4.2, errors in the speaker detection and the presence of "outlier" faces in the exemplar set may contribute to errors on the naming task. A possible solution is the use of a SVM classifier (Section 4.2), which is theoretically robust to such errors in the training data. In this section, we examine the influence of errors in the speaker detection on the nearest neighbour method, and report the performance of the SVM classifier.

Fig. 13 shows precision/recall curves for the original nearest neighbour method ("NN-Auto") using automatic speaker detection, and reported in the previous section. The results of two additional experiments are reported: (i) "NN-Manual" is the nearest neighbour method using *manually* labelled exemplars. This corrects

Table 1
Quantitative precision results at different levels of recall

Recall	Episode 05-02				Episode 05-05				Episode 05-13			
	60%	80%	90%	100%	60%	80%	90%	100%	60%	80%	90%	100%
Proposed method	87.5	78.6	72.9	68.2	88.5	80.1	75.6	69.2	84.1	75.2	69.2	63.0
Subtitles only				45.2				51.1				36.2
Prior				21.3				36.9				5.1

The baseline methods do not provide a means for ranking, so only the overall accuracy is reported.

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

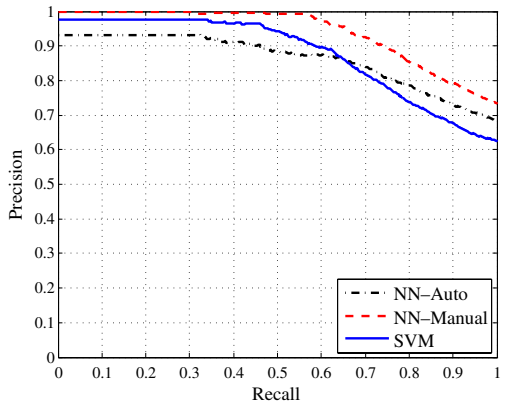
Fig. 12. Examples of correct detection and naming throughout episode 05-02.

any exemplars which have been assigned an incorrect name by the automatic speaker detection method. Note that this should be considered for discussion alone, since the manual labelling of exemplars requires more user intervention than we desire; (ii) "SVM" is the SVM classifier proposed in Section 4.3, trained using automatic speaker detection. In this case, the hope is that the SVM training criterion can remove errors in the names assigned by speaker detection, and remove "outlier" exemplars which are not helpful to discrimination. We also tried training the SVM using manually labelled exemplars; the results were indistinguishable from those obtained using automatically labelled exemplars, and are omitted here for the sake of clarity. Quantitative results for each experiment are reported in Table 2.

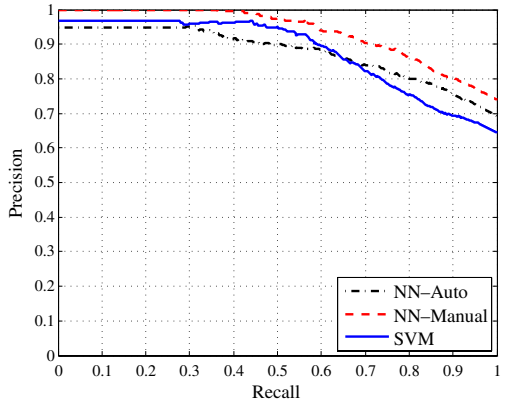
The first result of note is that the errors in the exemplar labels caused by errors in speaker detection do indeed impact the overall naming accuracy of the nearest neighbour classifier. The precision using manually labelled exemplars is consistently greater, at 40% recall increasing from 91.3% to 99.6% (+8.3%) for episode 05-02, from 91.7% to 99.5% (+7.8%) for episode 05-05, and from 86.4% to 99.6% (+13.2%) for episode 05-13. The increase diminishes slightly at higher recall, with precision at 100% recall of 73.3% versus 68.2% (+5.1%) on episode 05-02, 74.0% versus 69.2% (+4.8%) on episode 05-05, and 75.4% versus 63.0% (+12.4%) on episode 05-13, but the improvement obtained by using manually labelled exemplars is consistent. The notable improvement in results on episode 05-13 can be attributed to the low accuracy of labels from speaker detection (82.1%) obtained for this episode due to factors including imprecise alignment of the video and subtitle. The decrease in accuracy at high recall is likely indicative of the failure of the face track similarity measure at "long range" – when there are examples in the video for which the similarity to any exemplar is low, those examples cannot be labelled reliably.

As shown, use of the SVM classifier does, to some extent, overcome the errors in the exemplar labels from the speaker detection. On episode 05-02 at a recall level of 40%, the SVM method gives 96.7% precision versus 91.3% (+5.4%) using the nearest neighbour method, 96.7% versus 91.7% (+7.8%) on episode 05-05, and 91.2% versus 86.4% (+4.8%) on episode 05-13. These improvements are considerable, however, at higher levels of recall the accuracy of the SVM method decreases such that above around 65% recall it gives worse results than the nearest neighbour method: at 100% recall the precision decreases from 68.2% to 62.4% (−5.8%) on episode 05-02, from 69.2% to 64.6% (−4.6%) on episode 05-05, and from 63.0% to 62.3% (−0.7%) on episode 05-13. The decrease in the precision of the SVM classifier at high recall levels might be explained by the outlier rejection effected by the SVM training. If

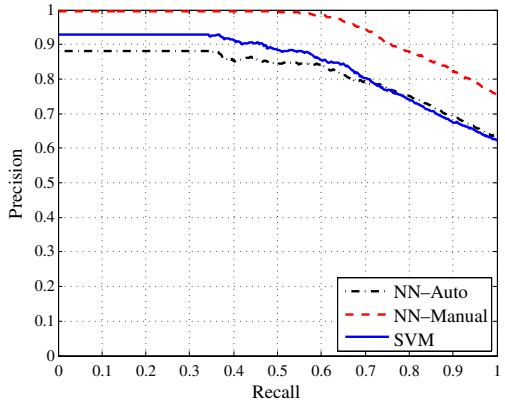
Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018



(a) Episode 05-02



(b) Episode 05-05



(c) Episode 05-13

Fig. 13. Effect of errors in the exemplar labels and the SVM method. "NN-Auto" is the originally proposed nearest neighbour method with automatically labelled exemplars; "NN-Manual" uses the same method with manually labelled exemplars; "SVM" is the SVM method trained with automatically labelled exemplars.

there is an exemplar which lies far from the other exemplars, but is nevertheless correctly labelled, it may be pruned as an outlier; at testing time, the loss of this exemplar can cause tracks to be incorrectly classified which lie far from any of the reduced set of exemplars. However, the initial improvement in results obtained by the SVM classifier show promise, and should motivate more application-oriented detection of errors in the labels or visual outliers.

6. Discussion

In the original version of this work [11], the proposed (nearest neighbour) classification method had no explicit mechanism for error correction. The SVM classifier proposed here shows some potential for dealing with errors in the speaker detection and "outlier" appearances, but as noted does not represent a full solution to the problem. Rather than requiring the classifier training algorithm to cope with errors in the annotation, a more global approach which considers the resultant labelling of the entire video may be more successful. A promising approach is to cast the labelling problem as one of solving a conditional random field (CRF) over the graph of connections generated by face and clothing similarities. In this setting, rather than viewing the annotation extracted from speaker detection as ground truth, yielding a fully supervised learning problem, the annotation is viewed in a "softer" manner as a prior on the labels.

The success of the CRF method would require more "long-range" interactions between the tracks to be generated in order to build a richer, more connected graph structure. This requires that the descriptors computed for the tracks have greater generalization (e.g. over pose or expression) than the current pixel-based descriptor adopted here. For example, replacing the pixel-based descriptor with a SIFT [28] descriptor or using Eigen facial-features would give some robustness to image deformation. Similarly the 2D face description could be replaced by a 3D description by fitting a parameterized 3D model to the detected face [39,40]. This can be thought of as "engineering in" some level of invariance or generalization. In the current exemplar framework slightly worse results on the naming task were obtained by using SIFT (compared to the simple pixel-based descriptor), but this might reasonably be attributed to the SIFT descriptor incorporating *too much* invariance to slight appearance changes relevant for discriminating faces. In a CRF framework this lack of discrimination may not be such a problem as other information may be available to correct such errors.

7. Conclusions

We have proposed methods for incorporating textual and visual information to automatically name characters in TV or movies and demonstrated promising results obtained without any supervision beyond the readily available annotation.

We consider of particular interest the use of visual speaker detection to improve the specificity of the ambiguous textual annotation. The idea of using lower-level vision methods to improve the annotation does not appear to be widespread, and could

Table 2
Quantitative results showing the effect of errors in the exemplar labels and the SVM method

Recall	Episode 05-02					Episode 05-05					Episode 05-13				
	40%	60%	80%	90%	100%	40%	60%	80%	90%	100%	40%	60%	80%	90%	100%
NN-Auto	91.3	87.5	78.6	72.9	68.2	91.7	88.5	80.1	75.6	69.2	86.4	84.1	75.2	69.2	63.0
NN-Manual	99.6	97.2	85.3	79.1	73.3	99.5	94.1	86.2	80.2	74.0	99.6	98.5	87.9	82.3	75.4
SVM	96.7	89.7	73.8	67.5	62.4	96.7	89.6	75.5	69.4	64.6	91.2	85.6	74.0	67.6	62.3

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

be applied in domains beyond that addressed here. An example is the area of learning object recognition from images annotated with keywords [41], e.g., learning to recognize cars from images annotated with the word "car" but with no segmentation of the image specified. For images annotated with some additional appearance properties, e.g., "red car", lower-level vision methods, i.e., colour classification, could be used to "target" the object referred to by the annotation in a manner similar to that used here in the form of speaker detection.

It is also worth noting that while there is previous work on recognizing people in video using text, the *video* properties have not been exploited, treating a segment of video as an unrelated collection of still images. The use of face tracking and speaker detection here shows the benefits of exploiting the specific properties of video. The general framework proposed here has also recently been applied successfully to face recognition from a wearable camera [42], using the same principle of face tracking to collect exemplars, and the same feature localization and representation methods proposed here.

In contrast, one aspect of TV and movie footage which has been neglected here is the *audio*. While the availability of script and subtitles makes the audio track seemingly redundant, since the script specifies *who* is speaking, and the subtitles specify *when*, there might be more information to be extracted from the audio. One area where the audio might usefully be applied is resolving the ambiguity in the subtitle/script timing mentioned in Section 2.3. Another interesting possibility is to attempt to *localize* the speaker in the frame based on the audio, augmenting the visual speaker detection. Related work in this direction [43] has used the correlation between video and audio to discover which pixels are "responsible" for a sound, and a similar approach might be used for identifying which person in the image is speaking.

The detection method and appearance models used here could be improved, for example by bootstrapping person-specific detectors [2] from the automatically obtained exemplars in order to deal with significantly non-frontal poses, and including other weak cues such as hair or eye colour. Further use of tracking, for example using a specific body tracker rather than a generic point tracker, could propagate detections to frames in which detection based on the face is difficult. As noted in Section 5, the results reported here are for frontal faces only. In other work [40], ground truth was prepared for all occurrences of characters in a TV show ("Fawlty Towers"), whether facing toward the camera or not. It was estimated that frontal faces account for only around one third of the occurrences of a character's face in the video, with the remainder being approximately one third profile, and one third facing away from the camera. This clearly leaves substantial space for improving the coverage of the proposed method.

In general, it seems promising to pursue further contextual cues such as co-occurrence of particular people or recognition of location. In the particular domain of TV and movies, there is also "grammar" of editing in cinematography, for example alternating close-up shots during a dialogue, which could be exploited.

Acknowledgements

This work was supported by EC project CLASS and an EPSRC Platform grant. This publication only reflects the authors' views.

References

[1] O. Arandjelovic, A. Zisserman, Automatic face recognition for film character retrieval in feature-length films, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, 2005, pp. 860–867.
[2] M. Everingham, A. Zisserman, Identifying individuals in video by combining 'generative' and discriminative head models, in: Proceedings of the 10th International Conference on Computer Vision, Beijing, China, 2005, pp. 1103–1110.
[3] A.W. Fitzgibbon, A. Zisserman, On affine invariant clustering and automatic cast listing in movies, Proceedings of the 7th European Conference on Computer Vision, vol. 3, Copenhagen, Denmark, 2002, pp. 304–320.
[4] J. Sivic, M. Everingham, A. Zisserman, Person spotting: video shot retrieval for face sets, in: Proceedings of the International Conference on Image and Video Retrieval, Singapore, 2005, pp. 226–236.
[5] P. Felzenszwalb, D. Huttenlocher, Pictorial structures for object recognition, International Journal of Computer Vision 61 (1) (2005) 55–79.
[6] O. Arandjelovic, R. Cipolla, Automatic cast listing in feature-length films with anisotropic manifold space, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, New York, 2006, pp. 1513–1520.
[7] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y.W. Teh, E. Learned-Miller, D. Forsyth, Names and faces in the news, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, 2004, pp. 848–854.
[8] J. Yang, A. Hauptmann, M.-Y. Chen, Finding person X: correlating names with visual appearances, in: Proceedings of the International Conference on Image and Video Retrieval, Dublin, Ireland, 2004, pp. 270–278.
[9] D. Ozkan, P. Duygulu, Finding people frequently appearing in news, in: Proceedings of the International Conference on Image and Video Retrieval, Tempe, AZ, 2006, pp. 173–182.
[10] J. Yang, Y. Rong, A. Hauptmann, Multiple-instance learning for labeling faces in broadcasting news video, in: Proceedings of the ACM International Conference on Multimedia, Singapore, 2005, pp. 31–40.
[11] M. Everingham, J. Sivic, A. Zisserman, "Hello! My name is... Buffy" – automatic naming of characters in TV video, in: Proceedings of the 17th British Machine Vision Conference, Edinburgh, UK, 2006, pp. 889–908.
[12] SubRip – DVD subtitles ripper, http://zuggy.wz.cz/.
[13] SlayerMagic, http://uk.geocities.com/slayermagic/.
[14] C.S. Myers, L.R. Rabiner, A comparative study of several dynamic time-warping algorithms for connected word recognition, The Bell System Technical Journal 60 (7) (1981) 1389–1409.
[15] K. Mikolajczyk, C. Schmid, A. Zisserman, Human detection based on a probabilistic assembly of robust part detectors, Proceedings of the 8th European Conference on Computer Vision, vol. 1, Prague, Czech Republic, 2004, pp. 69–82.
[16] S.Z. Li, Z.Q. Zhang, Floatboost learning and statistical face detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1112–1123.
[17] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, 2005, pp. 886–893.
[18] B. Leibe, E. Seemann, B. Schiele, Pedestrian detection in crowded scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, 2005, pp. 878–885.
[19] J. Shi, C. Tomasi, Good features to track, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, 1994, pp. 593–600.
[20] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, 2001, pp. 511–518.
[21] M. Meila, M.I. Jordan, Learning with mixtures of trees, Journal of Machine Learning Research 1 (2000) 1–48.
[22] D. Cristinacce, T.F. Cootes, Feature detection and tracking with constrained local models, in: Proceedings of the 17th British Machine Vision Conference, Edinburgh, UK, 2006, pp. 929–938.
[23] M. Everingham, A. Zisserman, Regression and classification approaches to eye localization in face images, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Southampton, UK, 2006, pp. 441–446.
[24] B. Heisele, P. Ho, J. Wu, T. Poggio, Face recognition: component–based versus global approaches, Computer Vision and Image Understanding 91 (1–2) (2003) 6–21.
[25] G. Shakhnarovich, B. Moghaddam, Face recognition in subspaces, in: S. Li, A. Jain (Eds.), Handbook of Face Recognition, Springer, 2004.
[26] M. Turk, A.P. Pentland, Face recognition using eigenfaces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1991, pp. 586–591.
[27] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.
[28] D. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150–1157.
[29] P.J. Phillips, H. Moon, P.J. Rauss, S. Rizvi, The feret evaluation methodology for face recognition algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (10) (2000) 1090–1104.
[30] L. Zhang, L. Chen, M. Li, H. Zhang, Automated annotation of human faces in family albums, in: Proceedings of the ACM International Conference on Multimedia, Berkeley, 2003, pp. 355–358.
[31] G. Jaffre, P. Joly, Costume: a new feature for automatic video content indexing, in: Proceedings of RIAO, Avignon, France, 2004, pp. 314–325.

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018

[32] J. Sivic, C.L. Zitnick, R. Szeliski, Finding people in repeated shots of the same scene, in: Proceedings of the 17th British Machine Vision Conference, Edinburgh, UK, 2006, pp. 909–918.
[33] Y. Song, T. Leung, Context-aided human recognition – clustering, Proceedings of the 9th European Conference on Computer Vision, vol. 3, Graz, Austria, 2006, pp. 382–395.
[34] W. Press, B. Flannery, S. Teukolsky, W. Vetterling, Numerical Recipes in C, Cambridge University Press, 1988.
[35] K. Saenko, K. Livescu, M. Siracusa, K. Wilson, J. Glass, T. Darrell, Visual speech recognition with loosely synchronized feature streams, in: Proceedings of the 10th International Conference on Computer Vision, Beijing, China, 2005, pp. II: 1424–1431.
[36] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.
[37] C. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, 1995.
[38] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, software available at http://www.csie.ntu.edu.tw/cjlin/libsvm (2001).
[39] V. Blanz, S. Romdhani, T. Vetter, Face identification across different poses and illumination with a 3D morphable model, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, 2002, pp. 192–197.
[40] M. Everingham, A. Zisserman, Automated detection and identification of persons in video using a coarse 3-D head model and multiple texture maps, IEE Proceedings on Vision Image, and Signal Processing 152 (6) (2005) 902–910.
[41] P. Duygulu, K. Barnard, J.F.G. de Freitas, D.A. Forsyth, Object recognition as machine translation: learning a lexicon for a fixed image vocabulary, Proceedings of the 7th European Conference on Computer Vision, vol. 4, Copenhagen, Denmark, 2002, pp. 97–112.
[42] N.E. Apostoloff, A. Zisserman, Who are you? real-time person identification, in: Proceedings of the 18th British Machine Vision Conference, Warwick, UK, 2007, pp. 509–518.
[43] E. Kidron, Y. Schechner, M. Elad, Pixels that sound, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, 2005, pp. 88–96.

Please cite this article in press as: M. Everingham et al., Taking the bite out of automated naming of characters in TV video, Image Vis. Comput. (2008), doi:10.1016/j.imavis.2008.04.018