Movie2Comics: Towards a Lively Video Content Presentation

Meng Wang, Member, IEEE, Richang Hong, Xiao-Tong Yuan, Shuicheng Yan, Senior Member, IEEE, and Tat-Seng Chua

Abstract—As a type of artwork, comics is prevalent and popular around the world. However, despite the availability of assistive software and tools, the creation of comics is still a labor-intensive and time-consuming process. This paper proposes a scheme that is able to automatically turn a movie clip to comics. Two principles are followed in the scheme: 1) optimizing the information preservation of the movie; and 2) generating outputs following the rules and the styles of comics. The scheme mainly contains three components: script-face mapping, descriptive picture extraction, and cartoonization. The script-face mapping utilizes face tracking and recognition techniques to accomplish the mapping between characters’ faces and their scripts. The descriptive picture extraction then generates a sequence of frames for presentation. Finally, the cartoonization is accomplished via three steps: panel scaling, stylization, and comics layout design. Experiments are conducted on a set of movie clips and the results have demonstrated the usefulness and the effectiveness of the scheme.

Index Terms—Cartoonization, comics, descriptive picture extraction.

I. INTRODUCTION

COMICS is a graphical medium which describes a story with a sequence of cartoon pictures. It started to gain popularity in the early 20th century with the newspaper comic strip. Early precursors of comics include Trajan’s Column and the work of William Hogarth [1]. Nowadays, comics is widely used in newspapers, magazines, and graphic novels. Although with varied forms, there are several common styles for comics. For example, word balloons and boxes are usually used to indicate conversation, and the flow of the story is conveyed by panels, layout, and zip ribbon.

Converting videos to comics has received lots of interests in the past years [9], [15], [29]. First, by turning a temporal frame sequence to a condensed set of 2-D pictures, it will make it easier for readers to browse and have a quick view of the content. Second, as a type of artwork, comics enables a lively presentation of video content. It offers readers a special enjoyment since such artwork can invoke strong emotional reactions, create identification, and effectively convey a story in an appealing manner [23], [31]. However, automatically turning videos to comics is not an easy task. Although several cartoonization and animation methods have been proposed for images and videos, it lacks a complete and fully automatic scheme. In fact, many existing approaches for turning movies to comics involve human efforts at different stages. For example, in [15], the frames for cartoonization are manually selected. In [29], the word balloon placement and comics layout are manually designed. Since the manual operations are usually expensive and time consuming, a scheme that can fully automate the conversion from videos to comics is highly desired.

In this paper, we propose a scheme named Movie2Comics for automatically turning a movie clip to comics. Our approach mainly contains the following steps. First, a script-face mapping is performed such that the conversation contents can be placed around the face of the speaking character in the generated comics. Second, we have a descriptive picture extraction step to summarize the visual contents in movies via a sequence of descriptive pictures. Finally, we perform a cartoonization on the pictures to generate the cartoonized panels in comics.

The main contributions of the work can be summarized as follows:

1) We propose an automatic scheme to convert movies to comics. While there exist several related research efforts, our scheme is the first fully automated and complete approach to generate comics from movies, including automatic picture extraction, word balloon placement, and layout.

2) We design an automatic script-face mapping algorithm to identify the speaking character when multiple characters are involved in a scene.

3) We propose a method to generate the descriptive and cartoonized pictures according to the characteristics of “narrative” in comics.

We would like to highlight that, although our scheme involves several existing techniques, such as subshot detection and image cartoonization, there are novel methods used in our approach. For example, the script-face mapping is a novel method for placing word balloons. The layout design is also not touched
before. But the most important contribution is the integration of a variety of techniques, such as subshot detection, keyframe extraction, face detection, face recognition, and cartoonization, to enable an automatic Movie2Comics scheme. Although the scheme is designed to turn movie clips to comics, other conversational videos can also be processed.1

The rest of the paper is organized as follows. Section II presents a brief review of related work. Section III provides an overview of the scheme. Section IV describes the script-face mapping component. Descriptive picture extraction is detailed in Section V. Section VI describes the cascaded steps towards the cartoonization including panel scaling, stylization, and comics layout design. Experiments are described in Section VII. We conclude the paper in Section VIII.

II. RELATED WORK

Extensive research efforts have been dedicated to video abstraction in the past decades. Most previous works aim to produce effective video abstraction with a friendly and compact form. Existing methods can be classified into two categories, i.e., dynamic representation and static representation. Dynamic representation generates a video sequence that is composed of a series of sub-clips extracted from one or multiple video sequences or generated from a collection of photos [10], [22], [30], whereas static representation generally generates one or multiple images from video key-frames to facilitate not only viewing but also transmission and storage [4], [7], [8], [17], [27], [35]. In comparison with the conventional video abstraction methods that mainly target at reducing the redundancy in video streams, our work focuses on the better transformation from movies to comics instead of redundancy reduction. It can also be regarded as a novel paradigm for presenting videos in a lively manner.

There are several recent efforts that are related to turning movies to comics. The first one is the cartoonization of video frames [15]. It manually selects frames with important features and transforms them into simplified illustrations. Stylized comic effects, including speed line, rotational trajectory, and background effects, are inserted into each illustration while word balloons are automatically placed. Further work in [9] seeks an automated approach of word balloon placement based on a more in-depth analysis of comic grammars. The second work [29] employs the screenplay of movie for turning movie into a comic strip since the screenplay information is able to offer important clues for segmenting the film into scenes and creating different types of word balloons. However, these two works still involve human efforts and can be categorized into the “computer aided design” paradigm, such as the manual selection of important frames in [15] and the word balloon placement and comic layout re-arrangement in [29]. Furthermore, an important issue is not touched in their methods, that is, how to identify the speaking character. Several other schemes are developed to summarize videos using comic-like layout according to the narrative grammars of comics and its universal and intuitive rules [5], [7], [35]. However, word balloon and stylization are not considered in these works. Shamir et al. create a sequence of comic-style images to summarize the main events in a virtual world environment and present them in a coherent, concise, and visually pleasing manner [31]. Wang et al. [37] and Agarwala et al. [2] transform the source video to a highly abstracted, spatially, and temporally coherent cartoon animation with a range of styles where user interaction is added to help determine the animation effects. The key issue is to automatically obtain the abstract segmented image and then animate it using the cartoon-like stylized images since cartoon animations are typically composed of large regions which are semantically meaningful and highly abstracted by artists [26]. Kim et al. [17] propose a semi-automatic method for expressive and non-realistic illustration of motion using video streams. Three features, i.e., temporal-flare, time-lapse, and particle-effects, are investigated for rendering source videos [17]. In comparison with the these works, our approach is fully automated by investigating a variety of computer vision and multimedia technologies, such as face detection and recognition, human detection, and lip motion analysis. Conversation contents have also been embedded in the generated comics.

III. SCHEME OVERVIEW

Typically, a sequential narrative is conveyed by a series of cartoon pictures and word balloons in comics. Intuitively, an effective Movie2Comics system that transforms movies to comics should consider the following two principles:

1) **Preserve as much informative content as possible.** In order to help readers better capture and understand the story of a movie, the scheme is expected to preserve as much informative content as possible in the generated comics. This requires an analysis to identify informative frames or compose descriptive pictures from video streams.

2) **Stylize the extracted key-scenes following the rules of comics creation.** Considering the characteristic of narrative in comics, the conversation contents need to be effectively presented around the faces of the characters. Therefore, we have to automatically map the characters with their scripts. In addition, comics has its specific style, in which shape and stylization of each panel, word balloons and comics layout have to be designed following several patterns.

Motivated by the above principles, we propose a Movie2Comics scheme that contains three main components, as illustrated in Fig. 1. The script-face mapping component is designed to present the conversation contents around the corresponding faces. In the descriptive picture extraction component, we extract informative frames or generate mosaic pictures from subshots. After that, the cartoonization process is carried out on the descriptive pictures. Panel scaling is used to resize the panel and we further perform several cropping operations. The panels are then stylized by combining bilateral filtering and difference-of Gaussian (DoG) edge detection operator. Word balloons are placed around the faces. Eight pre-defined templates are employed for organizing the processed panels.

---

1For videos that do not contain conversation content, it will be less meaningful to generate comics as the results will be merely a set of cartoonized pictures.
IV. SCRIPT-FACE MAPPING

This section describes the implementation of the script-face mapping component, which maps the speaking characters in a movie clip to their scripts to facilitate the placement of word balloons. Since the script file contains the identities of the speaking characters, the problem can be solved by labeling the faces appeared in frames. However, training data collection is a problem for this task. Here we apply a strategy as follows. We first observe the frames that only contain one face. For these frames, we then perform a lip motion analysis to verify whether the characters are speaking (even though only one face is contained in a frame, the voice may come from another character). For those verified cases, the faces can be labeled by the identities recorded in the script file and they form a training set for face recognition. In the following sub-sections, we describe the details of our approaches, including script-subtitle alignment, face detection and grouping, training data collection, and multi-task face recognition.

A. Script-Subtitle Alignment

We consider the case that there are two files associated with the given movie, i.e., a script file that contains speech contents and speaker identities and a subtitle file that contains the time information, as illustrated in Fig. 2. Then, the first step is to merge the speech contents, speaker identity, and time information from the two files. Here we utilize the dynamic time warping method [12] to align the subtitle and the script. Fig. 2 demonstrates an example of the alignment. If there is a script file that records all the information, this step can be skipped. Note that if such information is not available, we can generate the files by exploring speech recognition and speaker recognition techniques.

B. Face Detection and Grouping

For face detection, we adopt the algorithm in [36]. The detection is only performed on the frames that are within the conversation parts of the movie. As a video can contain thousands or even more detected faces, we group continuously detected faces of a particular character as a face “track” with a robust foreground correspondence tracker [43]. The tracker mainly works as follows. Given a pair of faces in adjacent frames, the size of the overlapped area between the two bounding boxes of faces is estimated. If it occupies a large proportion of the face region (we set the threshold to 80%), a matching is established. This tracking procedure is also able to deal with the cases that faces are not continuously detected due to pose variation or expression change. In this way, the number can be significantly reduced (typically we only need to deal with hundreds of such tracks for a movie). As a consequence, face track is regarded as the unit for labeling.

C. Training Data Collection

As previously mentioned, we perform a lip motion analysis to verify whether the character is speaking for those frames that contain only one face. The lip motion analysis is performed as follows. First, we detect a rectangular mouth region within each detected face region using a Haar feature-based cascade mouth detector. We then compute the mean squared difference of the
pixel values within the mouth region between each two continuous frames. To keep translation invariance, the difference is calculated over a search region around the mouth region in the current frame and we take the minimal difference for decision. A threshold is set to establish two statuses, namely “speaking” and “non-speaking”. For the verified “speaking” cases, we can label the face tracks with the speaker identity recorded in the script file. For example, if over half of the faces in a track are detected as speaking status and the script shows that merely “EDWARD” is speaking in this period, then we can label this track as “EDWARD”.

D. Face Recognition

For the cases that the frames contain multiple faces or the lip motion is not detected, the face track is regarded as unlabeled and we need to perform face recognition in order to find the speaker’s face. Each unlabeled face track is simply represented as a set of history image feature vectors. A simple approach for identification, as conducted in [12], is to directly calculate the feature distance between a testing face track and the training face tracks and then obtain classification results by the nearest neighbor principle. Another approach is to classify the history image independently via certain classification methods, such as sparse representation-based classification [42], and then assign the face track to the class that achieves the highest frequency.

In this work, by regarding the identity of each history image in a testing face track as a task, we formulate the face track identification challenge as a multi-task face recognition problem. This motivates us to apply the multi-task joint sparse representation model [28] to accomplish the task. The key advantage of multi-task learning is that it can efficiently make use of the complementary information embedded in different sub-tasks. We construct the representation of face appearance by a part-based descriptor extracted around local facial features [12]. Here we first use a generative model [3] to localize nine facial key-points in the detected face region, including the left and right corners of two eyes, the two nostrils and the trip of the nose, and the left and right corners of the mouth. We then extract the 128-dim SIFT descriptor from each key-point and concatenate them to form a 1152-dimensional face descriptor (SIFTD).

Our multi-task joint sparse representation model works as follows. Suppose that we have a set of exemplar face tracks with $M$ subjects. Denote $X = [X_1, \ldots, X_M]$ as the feature matrix in which the track $X_m \in \mathbb{R}^{d \times p_m}$ is associated with the $m$th subject consisting of $p_m$ samples. Here $d$ is the dimensionality of features and $\sum_{m=1}^{M} p_m = p$ is the total number of samples. Given a testing face as an ensemble of $L$ history images $y_l \in \mathbb{R}^d$, we consider a supervised $L$-task linear representation problem as follows:

$$y_l = \sum_{m=1}^{M} X_m w_m^l + e_l, \quad l = 1, \ldots, L, \tag{1}$$

where $w_m^l \in \mathbb{R}^{p_m}$ is a reconstruction coefficient vector associated with the $m$th subject, and $e_l$ is the residual term. Denote $w_l^* = [(w_1^l)^T, \ldots, (w_M^l)^T]^T$ as the representation coefficients for probe image feature $y_l$, and $w_m = [w_1^m, \ldots, w_l^m]$ as the representation coefficients from the $m$th subject across different case images. For simplicity, we denote $W$ as $[w_m^l]_{m \times l}$. Our proposed multi-task joint sparse representation model is then formulated as the solution to the following multi-task least square regressions with $\ell_1, 2$ mixed-norm regularization problem:

$$\min_{W} \sum_{l=1}^{L} \frac{1}{2} \sum_{m=1}^{M} \|y_l - \sum_{m=1}^{M} X_m w_m^l\|^2_2 + \lambda \sum_{m=1}^{M} \|w_m\|_{\ell_1, 2}, \tag{2}$$

We can see that the objective function contains two terms. The first term is a least square loss and the second term is a $\ell_1, 2$ regularizer on the weights $W$, which can enforce the weights for different tasks to be close in the multi-task learning [28]. Here we use the accelerated proximal gradient (APG) approach [34] to solve the optimization problem in (2).

When the optimal $W - [w_m^l]_{m \times l}$ is obtained, a testing image $y_l^*$ can be approximated as $y_l^* = X_m w_m^l$. For classification, the decision is ruled in favor of the class with the lowest total reconstruction error accumulated over all the $L$ tasks:

$$m^* = \arg \min_{m=1}^{M} \sum_{l=1}^{L} \|y_l^* - X_m w_m^l\|^2_2. \tag{3}$$

After labeling each face track with speaker identity, we can establish the speaking character even there is than one face in a frame. Hitherto we have accomplished the mapping between scripts and faces. It is worth mentioning that there also exist scripts that cannot be successfully mapped to faces, and in this work we directly put them on the left-top corner of a panel (off-screen voice is also processed in the same way).

V. DESCRIPTIVE PICTURE EXTRACTION

This section discusses the extraction of descriptive pictures, i.e., the essential parts of movie clips that convey the main story and will be cartoonized in the next step. Considering the intrinsic characteristic of narrative in comics, an intuitive way of descriptive picture extraction is to merely retain the frames marked by the time index in subtitle (we call them index frames as they correspond to the start moments of conversations). However, it is insufficient for representing the movie’s theme since there are many non-index frames and skipping them will inevitably degrade the user’s understanding of the story. Therefore, we need to investigate a more intelligent approach. In our method, we first segment the movie into subshots and then apply several strategies to extract one or more pictures from each subshot (as illustrated in Fig. 3).

A. Subshot Detection and Classification

We first segment the movie into a series of shots by performing the color-based method in [44]. Each shot is then decomposed into one or more subshots by a motion threshold-based approach [19], and each subshot is further classified into one of the six categories according to the camera motion, i.e., static, pan, tilt, rotation, zoom, and object motion. The algorithm in [20] is employed for estimating the following affine model parameters between two consecutive frames:

$$\begin{align*}
    v_x &= a_0 + a_1 x + a_2 y \\
    v_y &= a_3 + a_4 x + a_5 y
\end{align*} \tag{4}$$
Fig. 3. Example of descriptive picture extraction from a movie clip.

where $u_i, i = 0, \ldots, 5$ denote the motion parameters and $(u_x, v_y)$ is the flow vector at the pixel $(x, y)$. The motion parameters in (4) can be represented by a set of more meaningful parameters to illustrate the dominant motion in each subshot as follows [6]:

$$\begin{align*}
    b_{pan} &= a_0 \\
    b_{tilt} &= a_3 \\
    b_{zoom} &= \frac{(a_3 + a_5)}{2} \\
    b_{rot} &= \frac{(a_1 - a_2)}{2} \\
    b_{rev} &= \frac{(a_6 - a_8)}{2} \\
    b_{rev} &= \sum_{i=1}^{W} \sum_{j=1}^{H} \frac{p(i,j) - p_{\tilde{i}}(i,j)}{H \times W}
\end{align*}$$

(5)

where $p(i,j)$ and $p_{\tilde{i}}(i,j)$ denote the pixel values of the pixel $(i,j)$ in the original and the wrapped frame, respectively. The parameters $W$ and $H$ denote the width and the height of the frame, respectively. Based on the parameters in (5), we can categorize each subshot into one of the following six classes: pan, tilt, zoom, object motion, and static [19].

B. Descriptive Frame Extraction

After subshot detection and classification, we employ different strategies to extract one or more descriptive pictures from each subshot.

For zoom subshots, it is well known that they can be further categorized into zoom-in and zoom-out based on the tracking direction. In a zoom-in subshot, successive frames describe the gradual change of a scene from a distant view to a close-up view, as shown in Fig. 3. Therefore, the first frame captures more comprehensive information than other frames and it can be regarded as the descriptive picture. On the contrary, the last frame is regarded as the descriptive picture for a zoom-out subshot.

For pan and tilt subshots, camera is moved horizontally or vertically to capture a scene. In this case, one keyframe is usually insufficient to describe the whole scene, and image mosaic can be a choice to describe the wide field-of-view of the subshot in a compact form. Existing algorithms for mosaic typically involve two steps [16], [25]: motion estimation and image wrapping. The first step builds the correspondence between two frames by estimating the parameters in (4), while the second step uses the results of the first step to wrap the frames with respect to the global coordinates. Before generating the mosaic for each subshot, we first segment the subshot into units to ensure homogeneous motion and avoid excessive wide views [32]. As the wide-view mosaic is derived from a large number of successive frames may result in distortion, each subshot is segmented into units using the leaky bucket algorithm [18], [33]. As shown in Fig. 3, if the accumulation of $b_{pan}$ and $b_{tilt}$ exceeds a threshold (we empirically set the threshold to 200), a unit is segmented from the subshot. For each unit, we generate a mosaic image to represent it [16].

For other types of subshots, including rotation, object, and static, we simply select the middle frame in the subshot as the descriptive picture.

VI. CARTOONIZATION

As previously mentioned, comics are usually composed of cartoonized pictures and word balloons with a specific layout. Therefore, we describe the cascaded steps towards the cartoonization of the extracted descriptive pictures in this section. Our approach contains three steps, i.e., panel scaling, stylization, and panel organization.

A. Panel Scaling

Panel refers to an individual drawing that depicts a single moment in the comics sequence. A simple approach is regarding the extracted descriptive pictures as panels, but existing studies demonstrate that keeping the diversity of the scales of panels is important to prevent tedious reading [21]. Therefore, we design a panel scaling method to process the extracted descriptive pictures.

For zoom subshots, it is well known that they can be further categorized into zoom-in and zoom-out based on the tracking direction. In a zoom-in subshot, successive frames describe the gradual change of a scene from a distant view to a close-up view, as shown in Fig. 3. Therefore, the first frame captures more comprehensive information than other frames and it can be regarded as the descriptive picture. On the contrary, the last frame is regarded as the descriptive picture for a zoom-out subshot.

For pan and tilt subshots, camera is moved horizontally or vertically to capture a scene. In this case, one keyframe is usually insufficient to describe the whole scene, and image mosaic can be a choice to describe the wide field-of-view of the subshot in a compact form. Existing algorithms for mosaic typically involve two steps [16], [25]: motion estimation and image wrapping. The first step builds the correspondence between two frames by estimating the parameters in (4), while the second step uses the results of the first step to wrap the frames with respect to the global coordinates. Before generating the mosaic for each subshot, we first segment the subshot into units to ensure homogeneous motion and avoid excessive wide views [32]. As the wide-view mosaic is derived from a large number of successive frames may result in distortion, each subshot is segmented into units using the leaky bucket algorithm [18], [33]. As shown in Fig. 3, if the accumulation of $b_{pan}$ and $b_{tilt}$ exceeds a threshold (we empirically set the threshold to 200), a unit is segmented from the subshot. For each unit, we generate a mosaic image to represent it [16].

For other types of subshots, including rotation, object, and static, we simply select the middle frame in the subshot as the descriptive picture.

B. Descriptive Frame Extraction

After subshot detection and classification, we employ different strategies to extract one or more descriptive pictures from each subshot.

For zoom subshots, it is well known that they can be further categorized into zoom-in and zoom-out based on the tracking direction. In a zoom-in subshot, successive frames describe the gradual change of a scene from a distant view to a close-up view, as shown in Fig. 3. Therefore, the first frame captures more comprehensive information than other frames and it can be regarded as the descriptive picture. On the contrary, the last frame is regarded as the descriptive picture for a zoom-out subshot.

For pan and tilt subshots, camera is moved horizontally or vertically to capture a scene. In this case, one keyframe is usually insufficient to describe the whole scene, and image mosaic can be a choice to describe the wide field-of-view of the subshot in a compact form. Existing algorithms for mosaic typically involve two steps [16], [25]: motion estimation and image wrapping. The first step builds the correspondence between two frames by estimating the parameters in (4), while the second step uses the results of the first step to wrap the frames with respect to the global coordinates. Before generating the mosaic for each subshot, we first segment the subshot into units to ensure homogeneous motion and avoid excessive wide views [32]. As the wide-view mosaic is derived from a large number of successive frames may result in distortion, each subshot is segmented into units using the leaky bucket algorithm [18], [33]. As shown in Fig. 3, if the accumulation of $b_{pan}$ and $b_{tilt}$ exceeds a threshold (we empirically set the threshold to 200), a unit is segmented from the subshot. For each unit, we generate a mosaic image to represent it [16].

For other types of subshots, including rotation, object, and static, we simply select the middle frame in the subshot as the descriptive picture.

VI. CARTOONIZATION

As previously mentioned, comics are usually composed of cartoonized pictures and word balloons with a specific layout. Therefore, we describe the cascaded steps towards the cartoonization of the extracted descriptive pictures in this section. Our approach contains three steps, i.e., panel scaling, stylization, and panel organization.

A. Panel Scaling

Panel refers to an individual drawing that depicts a single moment in the comics sequence. A simple approach is regarding the extracted descriptive pictures as panels, but existing studies demonstrate that keeping the diversity of the scales of panels is important to prevent tedious reading [21]. Therefore, we design a panel scaling method to process the extracted descriptive pictures.

For zoom subshots, it is well known that they can be further categorized into zoom-in and zoom-out based on the tracking direction. In a zoom-in subshot, successive frames describe the gradual change of a scene from a distant view to a close-up view, as shown in Fig. 3. Therefore, the first frame captures more comprehensive information than other frames and it can be regarded as the descriptive picture. On the contrary, the last frame is regarded as the descriptive picture for a zoom-out subshot.

For pan and tilt subshots, camera is moved horizontally or vertically to capture a scene. In this case, one keyframe is usually insufficient to describe the whole scene, and image mosaic can be a choice to describe the wide field-of-view of the subshot in a compact form. Existing algorithms for mosaic typically involve two steps [16], [25]: motion estimation and image wrapping. The first step builds the correspondence between two frames by estimating the parameters in (4), while the second step uses the results of the first step to wrap the frames with respect to the global coordinates. Before generating the mosaic for each subshot, we first segment the subshot into units to ensure homogeneous motion and avoid excessive wide views [32]. As the wide-view mosaic is derived from a large number of successive frames may result in distortion, each subshot is segmented into units using the leaky bucket algorithm [18], [33]. As shown in Fig. 3, if the accumulation of $b_{pan}$ and $b_{tilt}$ exceeds a threshold (we empirically set the threshold to 200), a unit is segmented from the subshot. For each unit, we generate a mosaic image to represent it [16].

For other types of subshots, including rotation, object, and static, we simply select the middle frame in the subshot as the descriptive picture.
the number of recognized faces in each frame are available, we can design rules to perform cropping based on the information.

We consider the following four cropping operations:

- **Operation 1**: only keep the face region;
- **Operation 2**: vertically crop the speaking character out;
- **Operation 3**: horizontally crop the speaking character out;
- **Operation 4**: keep the whole picture.

In order to crop out the speaking character, we need to localize the person with a bounding box. In this work, we accomplish the human detection task by employing the part-based model in [13] learned with the annotated human images from the PASCAL Visual Object Classes (VOC) Challenge 2010 dataset [11].

Denote the height of recognized face and the number of faces in the frame as \( h_f \) and \( n_f \), respectively. We set the rules for deciding the cropping operation as follows:

\[
\begin{align*}
\text{Operation 1,} & \quad \text{if } n_f = 1 \text{ and } h_f \geq th_1; \\
\text{Operation 2,} & \quad \text{if } n_f > 1, h_f > th_1, \text{ and } \min_{i \neq 1} d(f, f_i) \geq th_2; \\
\text{Operation 3,} & \quad \text{if } n_f > 1, h_f < th_1, \text{ and } \min_{i \neq 1} d(f, f_i) < th_2; \\
\text{Operation 4,} & \quad \text{otherwise}
\end{align*}
\]

where \( th_1 \) and \( th_2 \) are two thresholds, \( f \) and \( f_i \) denote the speaker’s face and the \( i \)-th non-speaker’s face, respectively, and \( d(\cdot, \cdot) \) is the pixel-level distance between two faces and \( H \) is the height of the picture. From (6), we can see that the rule is in favor of zooming in the face to ensure that it is sufficiently large. If the size of the speaker’s face is small while its distances to other faces are large enough, the rule prefers vertically cropping the whole speaker out. Fig. 4 illustrates the cropping operations 1 and 2. In our implementation, we empirically set \( th_1, th_2 \) to 0.3 and \( W/3 \), respectively.

### B. Panel Stylization

We employ a stylization method that is analogous to [41] where an automatic, real-time video, and image abstraction framework is presented. It is able to abstract imagery by modifying the contrast of visually important features, i.e., luminance and color opponency. The basic workflow of our stylization scheme is shown in Fig. 5. First, we exaggerate the given contrast in an image using nonlinear diffusion. Second, we add highlighting edges to increase local contrast. Finally, we stylize and sharpen the resulting images.

Given a panel \( f \), we define the following filter:

\[
g(p, \sigma_d, \sigma_r) = \frac{\int_{\mathbb{R}^2} e^{-\|p - \hat{p}\|^2 / 2\sigma_d^2} w(p, \hat{p}, \sigma_r) f(p) \, dp}{\int_{\mathbb{R}^2} e^{-\|p - \hat{p}\|^2 / 2\sigma_d^2} w(p, \hat{p}, \sigma_r) \, dp}
\]

where \( \hat{p} \) is a pixel location, \( p \) indicates the neighboring pixels, \( \sigma_d \) is the blur radius parameter, \( w(\cdot, \cdot) \) is the range weighting function which is used to smooth or sharpen a given image. The function \( w(\cdot, \cdot) \) usually takes the following form:

\[
w(p, \hat{p}, \sigma_r) = e^{-\|p - \hat{p}\|^2 / 2\sigma_r^2}
\]

where \( \sigma_r \) determines how contrast will be preserved or blurred.

After performing the filtering, the DOG operator is utilized to detect edges. We define the blur function as

\[
S(\hat{p}, \sigma_e) = \frac{1}{2\pi\sigma_e^2} \int f(p) e^{-|p - \hat{p}|^2 / 2\sigma_e^2} \, dp
\]

where \( \sigma_e \) controls the spatial scale for edge detection. We then define

\[
D(\hat{p}, \sigma_e, \tau, \varphi_e) = \begin{cases} 1 & \text{if } (S(\hat{p}, \sigma_e) - \tau S(\hat{p}, 1.6\sigma_e)) > 0 \\
1 + \tanh(\varphi_e (S(\hat{p}, \sigma_e) - \tau S(\hat{p}, 1.6\sigma_e))) & \text{otherwise}
\end{cases}
\]

where the parameter \( \tau \) controls the amount of center-surround difference required for cell activation and \( \varphi_e \) controls the sharpness of the activation falloff. In (10), the factor of 1.6 relates the
typical receptive field of a cell to its surroundings [24]. Furthermore, we perform a color quantization step on the abstracted images, which results in cartoon or paint-like effects

$$Q(\hat{p}, q, \varphi_q) = q_{\text{nearest}} + \frac{\Delta q}{2} \tanh(\varphi_q \cdot \{f(\hat{q}) - q_{\text{nearest}}\}).$$

Here $Q(\cdot)$ is the pseudo-quantized image, $\Delta q$ is the bin width while $q_{\text{nearest}}$ is the bin boundary closest to $f(\hat{q})$, and $\varphi_q$ is a parameter controlling the sharpness of the transition from one bin to another. We summarize our stylization algorithm in Algorithm 1. In our implementation, the parameters $\sigma_\epsilon$, $q$, $\varphi_\epsilon$, and $\varphi_q$ are empirically set to 5, 10, 4.25, and 9.5, respectively.

Algorithm 1. The process of panel stylization

1. The bilateral filtering $\hat{H}(\hat{p}, \sigma_d, \sigma_\epsilon)$ is utilized to abstract the image details by preserving the edges;
2. DoG $D(\hat{p}, \sigma_\epsilon, \tau, \varphi_\epsilon)$ is used for detecting edges;
3. Color quantization in $Q(\hat{p}, q, \varphi_q)$ is applied to reduce colors;
4. Image with cartoon effect are generated by combining the edge image with the quantized image.

Word balloons are widely-used styles to present the speech contents in comics. As shown in Fig. 6, there are different types of word balloons. In this study, we simply choose the second type in the figure. Our strategy for placing word balloons is putting them to the right of character’s face, above the character’s head, or left of the character’s faces according to the position of the face.

C. Layout Design

The last step is to present all the cartoonized panels in a comic-like way by organizing the panels in a grid-based layout that contains several rows and columns. Initially, all the panels except the mosaic pictures are of the same size; these panels can thus be resized with keeping the width-height ratio according to the given layout template. The initial layout template is designed as illustrated in Fig. 8(a), which contains three rows and two columns. The width and height of the page as well as the intervals between each panel are fixed. Furthermore, the organization of panels must be consistent with the reading custom, i.e., from top to bottom and from left to right.

Considering the rules in comic layout, we pre-define several layout templates. We denote the standard width and height as $W_p$ and $H_p$, which are obtained by scaling the original frame size in movie while keeping the ratio invariant. In addition to the standard size $W_p \times H_p$, we design panels with three other sizes, namely, $W_p/2 \times H_p$, $W_p/2 \times 2H_p$, and $3W_p/2 \times H_p$, which are used for the pictures that are generated with face cropping, horizontal character cropping, and vertical character cropping, respectively (see Section VI-A). By organizing the panels of the four types with different combinations, we have manually defined eight templates, two of which are illustrated in Fig. 7(b) and (c), respectively.

In order to choose templates for a sequence of panels, we mark the panels with sizes $W_p \times H_p$, $W_p/2 \times H_p$, $W_p/2 \times 2H_p$, and $3W_p/2 \times H_p$, as type 4, 1, 2, and 3, respectively. We can see that the definition of the types is consistent with the types of cropping rule in Section VI-A. Each template thus can be represented as a numerical sequence of which the length varies from 6 (111111) to 9 (221122121). Based on the corresponding cropping operations, the generated panel list can also be represented as a numerical sequence and we let $\mathcal{S}$ denote this sequence. The method for choosing template is performed as follows. Each time, we read a sub-sequence of $\mathcal{S}$ and then compare it with the numerical sequences of the eight templates. The template that achieves the minimal hamming distance is selected for presenting the sub-sequence of panels. If the panels exactly match the template, we directly present them. Otherwise, we resort to the cropping operations in Section VI-A and we re-generate the panels such that they can match the template (that means, fitting the panels to the template has higher priority than the rules in Section VI-A for cropping operations). Therefore, we can see that actually the minimal hamming distance guarantees the least...
changes of the cropping operations. For example, if the ham-
ing distance is 3, then we will need to change the cropping
operations for 3 panels and re-generate them. If we assume that
a change of the cropping operations leads to a fixed loss, then
our strategy actually minimizes the overall cost for fitting panels
into the templates. This process continues until the end of $S$.

VII. EXPERIMENTS

In this section, we first introduce our experimental settings
and then detail the experimental results. We provide the results
of not only objective but also subjective studies. For objective
study, we mainly focus on the accuracy of script-face mapping
and subshot classification. For subjective study, we will inves-
tigate content comprehension and user impression.

A. Experimental Settings

We conduct the experiments with 15 movie clips that are
extracted from three movies: “Titanic”, “Sherlock Holmes”,
and “The Message”. The videos are in MPEG4 format and
associated with script and subtitle files (as introduced in
Section IV-A). The durations of the clips mainly vary from 2 to
7 min and the numbers of the detected face tracks mainly vary
from 10 to 50. Table I describes the details of the movie clips.
For script-face mapping, the parameter $\lambda$ in (2) is set to 0.1
throughout the experiment. Fig. 8 illustrates the two examples
of the comics generated by the proposed scheme.

B. On Script-Face Mapping

We now evaluate the performance of the script-face mapping
method. For comparison, two other face recognition algorithms
are employed: 1) the nearest-neighbor (NN) classifier; and 2)
the sparse representation (SR) classifier [43]. The script-face
mapping accuracies achieved by the three methods are illus-
trated in Table II. From the results, we can see that our method
achieves the best performance on 12 out of the 15 clips. For all
the clips, our method achieves accuracy above 0.7. The average
accuracy of the three methods are 0.817, 0.837, and 0.852, re-
spectively. The results demonstrate the effectiveness of the pro-
posed method.

For computational cost, the most expensive calculation lies in
the multi-task regression problem [see (2)]. As aforementioned,
we apply the APG method to optimize (2) which converges at

![Fig. 8. Examples of the generated comics using the proposed scheme. (a) A page that is generated from the movie clip of “Titanic”; and (b) a page that is generated from the movie clip of “The Message”.](image-url)
Table II

<table>
<thead>
<tr>
<th>Clip</th>
<th>NN</th>
<th>SR</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_1</td>
<td>73.33</td>
<td>72.26</td>
<td><strong>76.19</strong></td>
</tr>
<tr>
<td>C_2</td>
<td>86.90</td>
<td><strong>90.90</strong></td>
<td>90.24</td>
</tr>
<tr>
<td>C_3</td>
<td>80.89</td>
<td>87.47</td>
<td><strong>93.75</strong></td>
</tr>
<tr>
<td>C_4</td>
<td>83.74</td>
<td><strong>91.51</strong></td>
<td>88.89</td>
</tr>
<tr>
<td>C_5</td>
<td><strong>95.00</strong></td>
<td><strong>95.00</strong></td>
<td>87.50</td>
</tr>
<tr>
<td>C_6</td>
<td>90.43</td>
<td>90.25</td>
<td><strong>95.00</strong></td>
</tr>
<tr>
<td>C_7</td>
<td>87.81</td>
<td>91.91</td>
<td><strong>94.44</strong></td>
</tr>
<tr>
<td>C_8</td>
<td>65.06</td>
<td>69.70</td>
<td><strong>71.46</strong></td>
</tr>
<tr>
<td>C_9</td>
<td>77.97</td>
<td>81.47</td>
<td><strong>83.58</strong></td>
</tr>
<tr>
<td>C_10</td>
<td>70.79</td>
<td>73.17</td>
<td><strong>74.60</strong></td>
</tr>
<tr>
<td>C_11</td>
<td>86.34</td>
<td>91.44</td>
<td><strong>92.86</strong></td>
</tr>
<tr>
<td>C_12</td>
<td>87.50</td>
<td>82.08</td>
<td><strong>83.33</strong></td>
</tr>
<tr>
<td>C_13</td>
<td>70.09</td>
<td>66.30</td>
<td><strong>70.83</strong></td>
</tr>
<tr>
<td>C_14</td>
<td>80.37</td>
<td>82.59</td>
<td><strong>83.33</strong></td>
</tr>
<tr>
<td>C_15</td>
<td>89.17</td>
<td>90.00</td>
<td><strong>91.67</strong></td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th></th>
<th>zoom</th>
<th>pan/tilt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.87</td>
<td>0.96</td>
</tr>
<tr>
<td>Recall</td>
<td>0.93</td>
<td>0.91</td>
</tr>
</tbody>
</table>

The rate of $O(1/T^2)$, where $T$ is the number of iterations. In our experiment, the APG algorithm converges mainly after 10 to 20 rounds of iterations. The average running time is 0.76 s per testing face track.

C. On Subshot Categorization

Note that we have adopted three different descriptive picture extraction strategies for different kinds of subshots (one for zoom, one for pan and tilt, and one for other categories). Here we evaluate the accuracy of subshot categorization. A volunteer is asked to label the category of each subshot by observing its motion pattern. The performance for the classification of zoom and pan/tilt are listed in Table III. Note that we will not distinguish pan and tilt as the same strategy is used for these two categories. From the results, we can see that precision and recall are fairly high and the lowest value is 0.87.

D. User Evaluation

To validate the usefulness of the proposed approach, we conduct a user evaluation. The evaluation actually contains three parts. In the first part, we investigate content comprehension which measures how well users can understand the stories by observing the generated comics. In the second part, we study user experience in terms of enjoyment and naturalness. In the third part, we directly compare different paradigms based on a scoring method. For all the studies, there are 24 participants (19 males and 5 females). They are familiar with movies and comics, and their ages vary from 20 to 30.

1) Content Comprehension: Content comprehension measures the degree of story understanding. As we know, some questions, such as “how many characters are there in this movie clip?” have specific answers. Therefore, it is possible to judge how many questions can be correctly answered and use this value as the measurement of content comprehension.

In our study, we design 20 questions for each video clip to broadly cover the details of the story. For performance comparison, we define a metric of quality of perception (QoP) as the ratio of the correctly answered questions to the total number of the questions.

We compare the following three paradigms:

1. Movie, i.e., each user observes the original movie clips and then answer questions;
2. Comics, i.e., each user observes the comics generated by our approach and then answer questions;
3. Comics without Conversation, i.e., we do not present the conversation contents in the generated comics. That means we do not place the word balloons in the comics.

We randomly divide all the participants into 3 groups (each group has 8 participants) to avoid the repeated observation of a story which will cause knowledge accumulation. Therefore, each group merely evaluates one of the three paradigms.

Fig. 9 illustrates the comparison of the QoP scores. From the results, we can see that the QoP score of the “Comics without Conversation” paradigm is much lower than the other two paradigms. This indicates that presenting conversation contents plays an important role in conveying the stories of the movie clips. By adding conversation, the QoP score of the “Comics” paradigm can be greatly improved. The average QoP scores of “Comics” and “Movie” are 0.78 and 0.85, respectively.

2Here we do not compare the “Comics without Cartoonization” since it should have no difference with the Comics paradigm in terms of content comprehension.

3It is worth noting that we cannot let a user to directly compare different paradigms in this study, as the user will get accumulated knowledge if he/she watches a story for more than one time (for example, if a user first watches the movie and then watches the generated comics, he/she will easily answer the corresponding questions). Thus, we divide the participants into groups, and this setting is reasonable as we have sufficient participants. In Section VII-F, we will perform a statistical test (two-way ANOVA test) to compare different paradigms and analyze the impact of users in our study.

Fig. 9. QoP comparison of different paradigms.
This indicates that there still exists certain loss for content comprehension after turning movies to comics. The loss can be attributed to two facts. The first is that the comics are generated by exploring only a part of the video frames. It is clear that such loss is inevitable for almost all video abstraction and presentation schemes. The other is that the mistakes in script-face mapping will introduce several cases that put scripts around an incorrect face (from Table II, we can see that there are 14.8% of the cases that scripts are put around an incorrect face), and this will degrade the users’ understanding of the story. This problem can be solved by improving the performance of the script-face mapping.

2) User Impression: We now turn to the user experience about the generated comics. We ask each of the 24 participants to assign a score of 1 to 10 for each of these two criteria: enjoyment and naturalness. Here higher score indicates better experience.

Enjoyment: it measures the extent to which the viewers feel that the presentation style is enjoyable. The scores will be assigned based on whether the presentation is easy to perceive, read, and browse, and if it can invoke the emotional reactions of viewers and offer them a good experience.

Naturalness: it measures whether the viewers feel the presentation is natural. For example, unnatural cartoonization, inappropriate frame cropping, and word balloon insertion will lead to a low score.

We compare the following four paradigms:
1) Comics, i.e., the proposed presentation approach;
2) Comics without Conversation, i.e., we remove the word balloons;
3) Comics without layout design. That means we simply stylize the frames in a simple way (without cropping and presented with the simplest template that contains six panels);
4) Comics without stylization. That means we do not perform stylization, and the panel scaling and layout design are the same with the proposed approach.

Fig. 10 shows the average user scores of the four paradigms. From the results, we have the following observations:
• We first observe the results about enjoyment. We can see that the “Comics” paradigms gains the highest user score. The score of the “Comics without layout design” is much lower than the other three schemes. This demonstrates that the panel scaling and layout design are important in offering viewers a good enjoyment. Another problem is that, without speaking character cropping operations, for several frames that the characters are small, it will be difficult to recognize them from a single picture. In comparison with “Comics without layout design”, the gap between “Comics without stylization” and “Comics” is much smaller. Comparing “Comics” and “Comics without conversation”, we can see that the word balloons are critical in not only content comprehension but also viewers’ enjoyment as they can make our presentation closer to real comics.
• We now observe the results about naturalness. We can see that the other three paradigms slightly outperform “Comics”. This is due to the fact that the placement of word balloons, stylization, and layout design will slightly degrade the visual naturalness in comparison with the approach that simply presents video keyframes. But the gap is actually very small, and it indicates that our approach will not degrade user experience much in terms of naturalness.

3) Comprehensive Comparison of Different Paradigms: Finally, we compare the “Comics” paradigm with the “Comics without Conversation”, “Comics without layout design”, and “Comics without stylization” with considering both the content comprehension and the user impression. The users are asked to freely browse the comics generated by different paradigms. Then, they are asked to make a comparison. When comparing two paradigms, each user can make a “better”, “much better”, and “comparable” choice. To quantify the results, we convert the results into ratings. We assign score 1 to the worse scheme, and the other scheme is assigned a score 2, 3, and 1 if it is better, much better, and comparable than this one, respectively. Tables IV–VI illustrate the comparison of the average scores, from which we can clearly see that the users prefer the “Comics” paradigm. Especially for the comparison of “Comics” and “Comics without Conversation”, we can see that the score gap is very large. This indicates that the users cannot accept the presentation that does not contain word balloons, as the lack of conversation presentation will seriously hurt the content comprehension. We also perform an ANOVA test to statistically analyze the comparison. The p-values show that the difference of the paradigm comparison is significant and the difference of users is insignificant.

To summarize, we have the following conclusions from our user evaluation. First, our comics generation method can well preserve the information of the original video clips. It will not significantly degrade users’ story understanding. Second, presenting conversation content in the generated comics is critical for users’ story understanding. Third, the layout design, word balloons, and stylization can effectively improve users’ enjoyment while only slightly degrade naturalness. Our studies demonstrate that users prefer using all these three components.

4Note that each time there are only two paradigms compared. The users were asked “what is your preference if you can choose only one from the two comics styles?”.
TABLE IV

<table>
<thead>
<tr>
<th>Comics vs. Comics without Conversation</th>
<th>The factor of paradigms</th>
<th>The factor of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comics</td>
<td>Comics without Conversation</td>
<td>F-statistic</td>
</tr>
<tr>
<td>2.62±0.49</td>
<td>1.0±0.0</td>
<td>222.31</td>
</tr>
</tbody>
</table>

TABLE V

<table>
<thead>
<tr>
<th>Comics vs. Comics without layout design</th>
<th>The factor of paradigms</th>
<th>The factor of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comics</td>
<td>Comics without layout design</td>
<td>F-statistic</td>
</tr>
<tr>
<td>2.19±0.60</td>
<td>1.10±0.30</td>
<td>36.48</td>
</tr>
</tbody>
</table>

TABLE VI

<table>
<thead>
<tr>
<th>Comics vs. Comics without stylization</th>
<th>The factor of paradigms</th>
<th>The factor of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comics</td>
<td>Comics without stylization</td>
<td>F-statistic</td>
</tr>
<tr>
<td>2.05±0.50</td>
<td>1.29±0.46</td>
<td>31.22</td>
</tr>
</tbody>
</table>

E. Discussion

In our experiment, it takes less than 3 min to process a video clip on average on a PC with Pentium 4 3.0G CPU and 2G memory (all the steps have been taken into account for the time cost). The computational costs can be further reduced such as by speeding up the solution process of (2) and the bilateral filtering process.

We also observed several situations that our approach cannot well handle. Here we list several of them.

1) Scripts are put around incorrect faces due to the mistakes in script-face mapping. As discussed in Section VII-B, the accuracy of script-face mapping is about 85.2%, and thus there are several failure cases.
2) The extracted descriptive pictures (see Section V-B) may not well represent the whole keyshots.
3) The frame cropping introduced in Section VI-A may lead to the loss of context information since for several frames, we only keep the speaking characters.
4) Word balloons cannot be well inserted. For example, in several cases, the inserted word balloons cover important elements in a scene.
5) Several visual details are lost in the stylization step.

For several problems, we can further improve our approach to reduce the failure cases. For example, we can improve our face recognition method to reduce the mistakes in script-face mapping. For word balloon placement, we can integrate image saliency analysis to put the balloons to more appropriate places. We can investigate different stylization methods or combine them, such as those in [29] and [41], to achieve better stylization. Scene and object understanding [38]–[40] may also be integrated to help to achieve better descriptive picture extraction and frame cropping. But several problems are not easy to tackle. For example, the information loss in descriptive picture extraction is inevitable. This problem exists in nearly all video summarization methods. One possible solution to address all these problems is to involve human efforts. We can regard the results generated by each step of our approach as a kind of recommendation, and users can manually adjust the results. In this way, we will be able to obtain much better results, while the labor cost is much less than pure manual operation as our approach can provide reasonable recommendations.

Finally, we would like to mention that, although the current scheme targets at videos along with scripts, it can be extended to process conversational videos without scripts. For example, we can employ speech recognition to convert speech contents and then explore speaker recognition to accomplish the script-face mapping.

VIII. Conclusion

This paper describes a scheme that automatically turns movies to comics. The scheme mainly consists of three components, namely, script-face mapping, descriptive picture extraction, and cartoonization. The script-face mapping accomplishes the mapping between characters’ faces and their scripts, whereby conversation contents can be presented appropriately in the generated comics. The descriptive picture extraction generates a sequence of informative frames, and they are processed to generate the comics by the cartoonization component. We have conducted experiments on multiple movie clips, and both subjective and objective evaluations demonstrate the effectiveness of the scheme.
REFERENCES

[23] S. McCloud, “Reinventing comics: How imagination and technology than 50 publications in these areas.

Meng Wang (M’09) received the B.E. degree and Ph.D. degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, respectively.

He is a Professor in the Hefei University of Technology. He previously worked as an Associate Researcher at Microsoft Research Asia, and then a core member in a startup in Silicon Valley. After that, he worked in the National University of Singapore as a Senior Research Fellow. His current research interests include multimedia content analysis, search, mining, recommendation, and large-scale computing. He has authored more than 100 book chapters, journal, and conference papers in these areas.

Prof. Wang received the best paper awards continuously in the 17th and 18th ACM International Conference on Multimedia and the best paper award in the 16th International Multimedia Modeling Conference. He is a member of ACM.

Richang Hong received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2008.

He worked as a Research Fellow in the School of Computing, National University of Singapore, as a Research fellow from September 2008 to December 2010. He is now a Professor in Hefei University of Technology. His research interests include multimedia question answering, video content analysis, and pattern recognition. He has coauthored more than 50 publications in these areas.

Dr. Hong is a member of the Association for Computing Machinery. He won the Best Paper Award in ACM Multimedia 2010.
Xiao-Tong Yuan received the M.Sc. degree in electrical engineering from the Shanghai Jiaotong University, Shanghai, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent system from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2009. During 2009–2011, he worked as a postdoctoral research fellow in the Department of Electrical and Computer Engineering, National University of Singapore. He is currently a postdoctoral research fellow in the Department of Statistics, Rutgers University, New Brunswick, NJ. His main research interests include machine learning, data mining, and computer vision.

Shuicheng Yan (SM’11) is currently an Assistant Professor in the Department of Electrical and Computer Engineering at the National University of Singapore, and the founding lead of the Learning and Vision Research Group (http://www.lv-nus.org). His research areas include computer vision, multimedia, and machine learning, and he has authored or co-authored over 240 technical papers over a wide range of research topics. Dr. Yan is an associate editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and has been serving as the guest editor of special issues for TMM and CVIU. He received the Best Paper Awards from PCM’11, ACM MM’10, ICME’10, and ICIMCS’09; the winner prizes of the classification task in both PASCAL VOC’10 and PASCAL VOC’11; the honorable mention prize of the detection task in PASCAL VOC’10; 2010 TCSVT Best Associate Editor (BAE) Award; 2010 Young Faculty Research Award; 2011 Singapore Young Scientist Award; and the co-author of the best student paper awards of PREMIA’09 and PREMIA’11.

Tat-Seng Chua received the B.S. and Ph.D. degrees from the University of Leeds, Leeds, U.K., in 1979 and 1983, respectively. He then joined the National University of Singapore, where he is now a full Professor at the School of Computing. He was the Acting and Founding Dean of the School of Computing during 1998–2000. He spent three years as a research staff member at the Institute of Systems Science in the late 1980s. His main research interest is in multimedia information processing, in particular on the extraction, retrieval, and question-answering (QA) of text and video information. He focuses on the use of relations between entities and external information and knowledge sources to enhance information processing. He is currently working on several multi-million-dollar projects: interactive media search, local contextual search, and event detection in movies. His group participates regularly in TREC-QA and TRECVID news video retrieval evaluations.

Dr. Chua is active in the international research community. He has organized and served as program committee member of numerous international conferences in the areas of computer graphics, multimedia, and text processing. He is the conference co-chair of ACM Multimedia 2005, CIVR (Conference on Image and Video Retrieval) 2005, and ACM SIGIR 2008. He serves in the editorial boards of: ACM Transactions of Information Systems (ACM), The Visual Computer (Springer), and Multimedia Tools and Applications (Springer). He is a member of the steering committee of CIVR, Computer Graphics International, and Multimedia Modeling conference series; and is a member of International Review Panels of two large-scale research projects in Europe.