

# CINEMA COMICS : CARTOON GENERATION FROM VIDEO STREAM

Won-Il Hwang, Pyung-Jun Lee, Bong-Kyung Chun, Dong-Sung Ryu, Hwan-Gue Cho  
Dept. of Computer Science and Engineering, Pusan National University  
San-30, Jangjeon-dong, Keumjeong-gu, PUSAN, 609-735, KOREA

Keywords: non-photorealism, comics cartooning, stylized rendering, video summary.

Abstract: This paper presents CORVIS(COmic Rendering system on VIdeo Stream) which helps to create comic strips from video streams in semi-automated manner. For this, first we manually select a set of important featuring scenes in a cinema and transform them into simplified illustrations by Mean-Shift segmentation. Then we insert the stylized comic effects to each illustration by considering the before/after video images. We newly proposed some techniques for this cartoon rendering effects. These stylized effects include the speed line, rotational trajectory and the background effect. Finally CORVIS automatically places the word balloons to represent the dialogues of actors. And echoic words e.g., "BANG", will be inserted in the comic cut to imitate the sound effects of the original film. We tested CORVIS with the well-known cinemas, "Spider Man II" and "I ROBOT". The final results show that our technique is quite effective and efficient to create a comic booklet from video streams.

## 1 MOTIVATION

Comics has been studied and practiced for so long time as the oldest genre in non-photorealistic rendering. One famous book, "Understanding Comics" and "Reinventing Comics" have successfully explained the basic structure of traditional comics and the power of comics as a communication tool(McCloud, 1999; McCloud, 2000).

Comics is very easy to give emotion to each character with exaggerated motions, special background effects and the manipulation of dialogue text in a word balloon. And a single comic cut can represent much more things than 3D scenery. This *smart depiction* feature of comics can show the before/after-scene of a still scene, which is very effective in explaining the dynamics of a moving object(Nienhaus and Dollner, 2005)(See Figure 1).

One of difficulties in generating the traditional comics is that it depends on totally manual work, so it is hard to be automated and it takes long time to be completed. Another difficulty is that the stylized comics rendering techniques highly depends on human artists. So there are no general rules and principles in evaluating the quality, expressiveness and cognitive effects of comics. Also since the quality of

Figure 1: Typical examples for Western(left) and Asian(right) comics.

comics is very subjective, it is hard to define or evaluate the quality of comics.

In this paper we give a novel system, CORVIS (COmic Rendering system on VIdeo Stream), which helps to transform a video(a cinema film) into a comics book with a semi-automated manner. This comic book can be regarded as a summarization of a video. The main difficulty in transforming a video into a comic book is how to put a sequence of video stream cuts in a single comic cut.

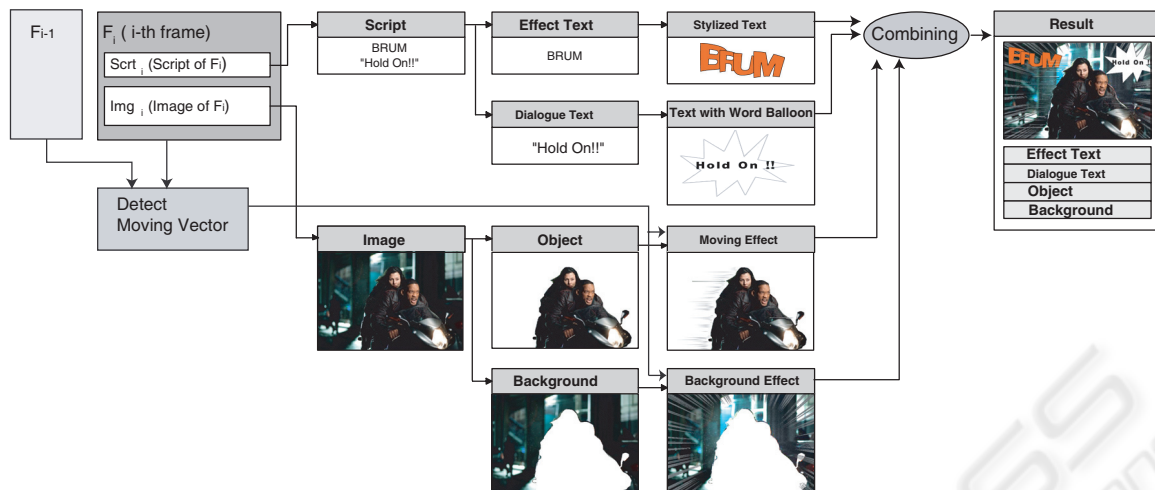


Figure 2: The architecture of CORVIS(COMics Rendering system with Video Stream) system.

## 2 PREVIOUS WORK

There are lots of work for video tooning(Wang et al., 2004; Agarwala et al., 2004), which converts an input video into a highly abstracted, spatial-temporally coherent cartoon animation with a range of style. The prevailing technique in cartooning is Mean-Shift methodology, which has been studied for long time. The main issue of this work is how to get the abstract segmented image with automated manner. Then these cartoon-like stylized images are used for animation, since cartoon animations are typically composed of large regions which are semantically meaningful and highly abstracted by artists(McCloud, 1999; McCloud, 2000).

We are not the first to present methods for transforming video stream to a set of still images. (Kim and Essa, 2005) have proposed one semi-automated method for expressive and non-realistic illustration of motion using video streams. They studied three features (temporal-flare, time-lapse and particle-effects) for cartoon-like rendering.

For video stream processing, there have been work for video summarization and visualization by an abstracted version. The purpose of video MANGA was similar to ours in that it transforms a video film into a sequence of 2D still images(Uchihashi et al., 1999). The main objective of these approaches is to develop the method to choose relevant key frames, emphasize meaning events and find a quantitative measure that how well a summary captures the salient events. However, they do not propose any comics-like features such as the speedline, word balloon relocation(Kurlander et al., 1996) *et al.*

Stylized cartoon rendering techniques have been studied so long time. For example, (Hanl et al., 2004)

studied cartoon-line rendering method for motion representation in computer game. He introduced how to deform objects using their squash-and-stretch technique to add more functional realism to games.

One interesting and cute application of comics rendering was introduced for internet chatting (Kurlander et al., 1996). This is the the first work on how to place the word balloon in a comic cut automatically.

We know only one commercial tool for this purpose, "Comics Creator", which allows only word balloon placing on a still image (Planetwide Games, 2005). Comparing to our CORVIS, Comics Creator does not provide any features for automated word balloon, background texture generation, speedline and rotational trajectory effects.

## 3 STYLIZED EFFECTS FOR COMICS

### 3.1 System Overview of CORVIS

Let  $F_i$  be an image frame of a video stream. CORVIS transforms a sequence of consecutive frames, e.g.,  $F_i - F_j$  into one comic cut  $C_i$ (Figure 2). Each image frame  $F_i$  consists of two components, the sound/dialogue script and pixel frame. Cinema sound will be represented by the stylized characters or the size of word balloon. And the dialogue text will be placed in the word balloon.

First the main object  $O_m$  should be identified and segmented. There are lot of algorithms and systems for this intelligent object segmentation. Since this intelligent segmentation and scissoring is not a main issue of our work, we do not explain further in detail.

We have manually located the main object in each cinema frame in CORVIS.

In order to make  $C_i$ , two basic procedures are applied. One is the script processing and the other is image processing to reflect moving effects. If  $O_m$  is in the state of moving in the video, we need to compute the direction of moving  $O_m$ . This can be done by comparing the previous frame,  $F_{i-1}$ ,  $F_{i-2}$  to the current frame  $F_i$ . But in this paper, we do not apply the automated approach as (Brostow and Essa, 2001), but we only estimated the speed and direction of main object by watching.

If there is a loud sound around  $F_i$ , we need to express it in a still comic cut (volume, duration and pitch) by manipulating the shape and size of word balloon shape. Some background texture effects could also be added if the concentration or focusing effect is required. CORVIS keeps each processed outputs (stylized text lettering, word balloon, speedline for moving direction and background texture mapping for focusing effect) in a separated layer. Then finally all these layers are put together to produce one comic cut.

### 3.2 Speedline for Linear Movement

Speedline can give an important stylistic effect in a comics, though the physics of speedline is very subjective and hard to define. Speedline easily indicates former position and future position by drawing repetition of its contour. Generally speedline effect consists of segmented speedline, fading the ratio of contour and partial contour repetition. This speedline technique has been used widely in all kinds of comics and classical animation film.

Since the main goal of our approach is to transform a sequence of consecutive video shots into a single cut, one automated method should be devised. In this chapter we give an automated algorithm for speed line generation for a moving object in a still comic cut.

First we need to know the velocity and direction of the moving object. However, it is nearly impossible to measure the exact and physical velocity(e.g., as  $34.7km/hour$ ) by watching a cinema. We only classify the speed of objects as five categories: halt, quivering, walking speed, running and fast-running.

Here we give one automated procedure for speedline rendering from an estimated speed we guess. The control parameter for speedline is the moving vector representing the direction and velocity.

Let  $O$  be the main object for speedline effect, and  $\vec{V}_o$  be the vector for the velocity and its direction.  $x$  is the center position of  $O$  in the previous video frame,  $F_{i-1}$ . Let  $E$  be all boundary edges of  $O$ , which is also visible from point  $x$ . Each speedline  $s_i$  is placed to  $E$  in the direction of  $\vec{V}_o$ . See Figure 3.

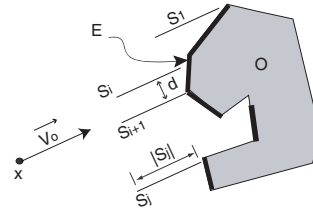


Figure 3: The procedure for speedline rendering.

Let the gap between  $s_i$  and  $s_{i-1}$  be  $d$ . We forces that  $d$  gets smaller if the velocity of  $O$  is higher. And the length of the speedline  $s_i$  itself gets longer according to the velocity. That is defined as follows.

$$d = k_a \cdot \frac{1}{|\vec{V}_o|} + random[0, w]$$

$$|s_i| = k_b \cdot |\vec{V}_o| + random[0, w]$$

, where  $k_a, k_b$  are adjusting constants.

Figure 4 shows six different kinds of speedline effects. (a) staying, and (b) shows a warming-up state. (c), (d), (e) and (f) show the 4 different speedlines according to the velocity. In Figure 4 (f), we added shadowing background to show the very fast moving. Since the physical speed of object in a video can not be measured exactly, these speedline effects depend on human manipulator.

If the speed of object is negligible, for example less than  $0.5km/hour$ , then we add a special effect(quivering) to show that it is ready to move, but not in a stationary state. We call this “quivering” effect. The generation procedure of quivering effect is of two steps. First we take a few boundary corners of the main object with a small circle  $L$ . Next we compute the boundary shape of the main object contained in  $L$ . Let this small fragment of boundary shape be  $f_i$ . This  $f_i$  will be transformed into a smooth curve(in this paper, we modified  $f_i$  as a single Bezier curve, which is  $B(f_i)$ ). Finally we place two or three copies of  $B(f_i)$  outward from the original boundary of  $O$  which is covered by  $L$ . See Figure 5 for this procedure.

If the trajectory of moving object is not a straight line, then the circular trajectory curve should be added by artist in comics. Figure 6 shows two different motions of a soccer player. (a) shows that the player rushes forward and the ball is passed on the ground from an opposite direction. Figure 6 (b) shows that the ball is falling down and the player is trying to kick it immediately.

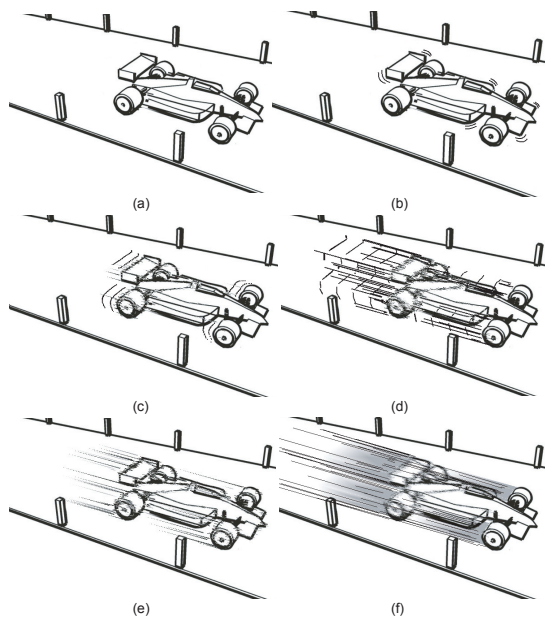


Figure 4: The six different of speedlines according to the velocity of moving object. (a) staying. velocity=0. (b) warming up for starting, quivering. (c) Just moving with velocity=10. (d) velocity=20. (e) velocity=30. (f) velocity=60, very fast moving.

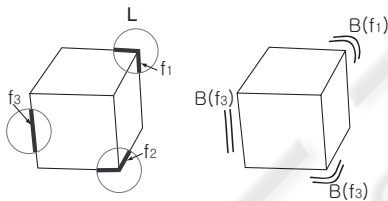


Figure 5: The quivering effect for slowly moving object.

### 3.3 Speedline for Rotation

Sometimes we need to render the speedline for the rotational objects such as airplane propeller in comics. In this chapter we propose an automated method to render the rotational speedline by proposing a virtual *rpm*(rotation per minutes) unit.

In order to render this effect, first we should locate the center of rotating object, which is invariant in rotation,  $x$  in Figure 7 (a). The characteristic features of rotational speedline are the number of partial contour repetition and the density of circular speedline. In Figure 7 (a), let  $A_i$  be a partial contour boundary. The angle between two adjacent  $A_i$ s is denoted by  $\theta_i$ . We determine the number of  $\{A_i\}$  and  $\theta_i$  with the *rpm* unit given. Suppose that we are asked to render a comic cut for rotating object  $O$  with *rpm*  $K$ . In practice, the observed scene of rotational is depended on the interference between the rotation speed and light-

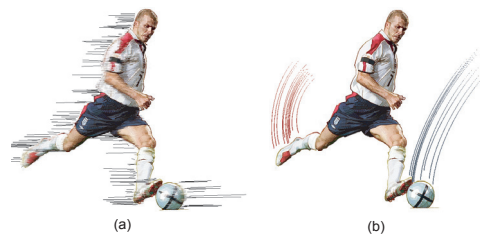


Figure 6: Two different comic cuts by assigning different speedlines. (a) The result of straight speedline effect. (b) The result of curved speedline effect.

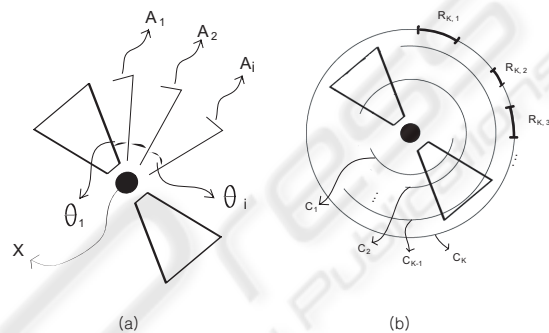


Figure 7: The speedline for the rotational object.

ing frequency. If the *rpm*  $= K$  is less than a threshold value of  $R_{mid}$ , the number of  $A_i$  is proportional to  $K$  in somewhat extent. Also  $\theta_i$  is decreasing according to  $K$ . Let  $n_K$  be the number of  $A_i$  to be rendered as a partial contour of original object.

$$n_K = k_a \cdot K, \text{ if } K \leq R_{mid}$$

$$\theta_i = k_b \cdot K, \text{ if } K \leq R_{mid}$$

If *rpm*  $K$  is higher than  $R_{mid}$ , then the situation is reversed as follows.

$$n_K = k_a \cdot \frac{1}{K}, \text{ if } K > R_{mid}$$

$$\theta_i = k_b \cdot \frac{1}{K}, \text{ if } K > R_{mid}$$

, where  $k_a, k_b$  are adjusting constants.

For the higher *rpm* rotation, more rotational arc speedline will be added as in Figure7 (b). Let  $C_i$  be a rotational circle and  $d(K)$  be the distance gap between two adjacent  $C_i$ s of the *rpm* rotating object.  $C_i$  will be used as the speedline effect for  $O$ . But only a set of partial arc of  $C_i$  should be rendered to make a more comics-like rendering. Let  $R_{i,p}$  be the  $p$ -th visible arc in  $C_i$  in clockwise. See Figure 7(b). The most important procedure to show an object with *rpm*  $= K$  is to determine  $d(K)$  and the number of  $C_i$ 's  $= m(K)$ , and how to break the  $C_i$  and how to

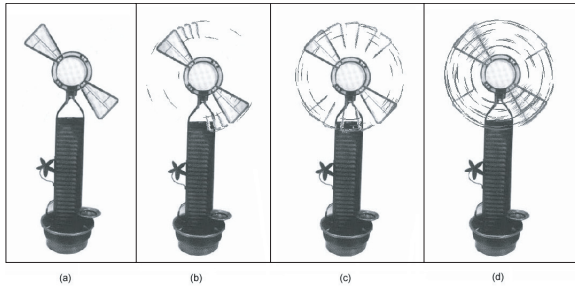


Figure 8: The results of rotational effect according the different rpms. (a) rpm = 0. (b) rpm = 20. (c) rpm = 50. (d) rpm = 100.

choose  $R_{i,p}$ . All these parameters are decided experimentally in our CORVIS. We propose the following principle to control the rotational arc line for an rotating object with  $rpm = K$ .

$$\begin{aligned} d(K) &= k_c \cdot K \\ m(K) &= k_d \cdot K \end{aligned}$$

For  $R_{i,k}$ , we randomly select an interval of  $C_i$  (we consider the boundary of  $C_i$  as a bounded straight line). Let  $\varphi(K)$  be the average length of  $R_{i,j}$  and  $\eta(K)$  be the number of  $R_{i,k}$  arc appeared as a visible part on  $C_i$  in the rotating object with rpm  $K$ . We select  $R_{i,k}$  to meet the following principle.

$$\begin{aligned} \varphi(K) &= k_e \cdot K \\ \eta(K) &= k_f \cdot K \end{aligned}$$

, where  $k_e, k_f$  are adjusting constant and  $K$  is the virtual  $rpm$  given. We applied our method to a real electric fan. Figure 8 (a) is a photo taken from a real fan. (b), (c) and (d) show the final results with the different  $rpm$  speed.

### 3.4 Background Texture

Background texture in comics is crucial to imply the atmosphere of the scenery of cinema film. For long time, there are typical textures in comics to represent anger, joy, madness, anxiety, cold, quietness and warm. The idea that a picture can evoke an emotional or sensual response in the view is vital to art of comics (McCloud, 1999; McCloud, 2000). Though there are unaccountably many background textures for comics, CORVIS allows three basic background textures with the control parameters.

First one is the background texture for concentration lines around the main object with blurring the tedious background objects (Figure 9 (a)). Second and third are the textures for representing some obscure feeling of the actor or scene (Figure 9 (b), (c)). Our background textures can be controlled with a

couple of parameters to emphasize the emotional atmosphere. Especially there are two kinds of common control variables,  $D$  and  $T$ .  $D$  denotes the average density of component of texture, e.g., lines in concentrated texture, and  $T$  denotes the average thickness of component. In some cases, certain texture has the unique parameters such as the type of component. For example, lines can be replaced as a sequence of small circles to represent a romantic atmosphere (Figure 10 (b)), which is well-known and stereotype implication in Asian comics.

Figure 9 clearly shows the effect of background texture. Figure 9 is the video frame obtained from "I ROBOT". In Figure 9 (a), we tried to draw a man shouting by adding concentration background texture and burst word balloon. (b) and (c) was rendered to show obscure feeling of the actor by adding complex background pattern.

Figure 10 shows snap shots of one television drama, "Winter sonata", which was very famous in Asia, especially in Korea, Japan and China. The bubble background texture in Figure 10 (b) presents more romantic atmosphere comparing (a) the original scene with focusing texture.

## 4 CONCLUSION AND FUTURE WORK

Comics is one artistic genre, and this has been studied for so long time and so widely in the world. Since the 2D comics works in a very limited environment (only on the paper), we need some specialized and stylized rendering techniques to depict the real-time video frames. The motivation of this paper is how to make a comic book from a well-known and famous cinema film with the minimal human interaction.

In this paper we propose CORVIS, which helps to generate comic strips from video streams. For this, the general method for speedline for moving objects in video was proposed and three kinds of different background textures, which can be controlled with some parameters, were proposed.

Till now CORVIS needs lots of manual work for selecting featuring scenes, and we should guess the direction of main object and its speed and moving direction. But it should be noted that this manual work can be systematically combined in CORVIS and some semi-automation can relieve burdensome human work to create comic strips using CORVIS. Currently we completed the automatic placement procedure for word balloon, which was not explained due to the limit of pages. We will announce more automated version of CORVIS in near future.

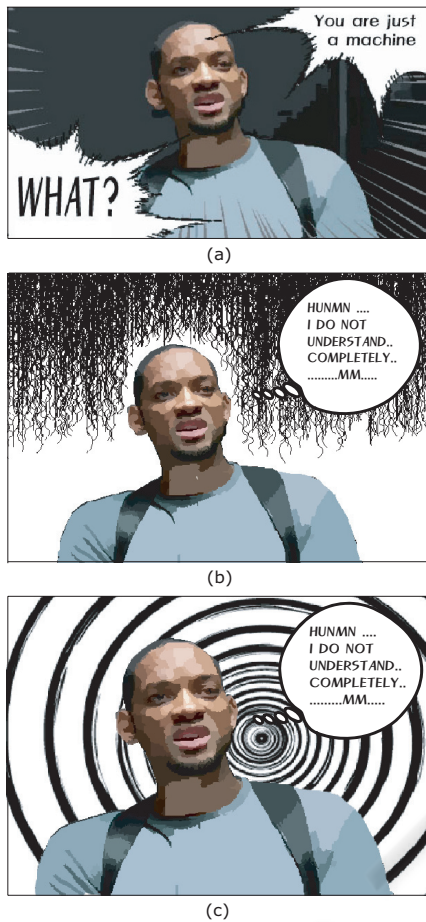


Figure 9: Three different background textures. (a) Concentration background texture. (b) Background texture for gloomy feeling. (c) Spiral background texture to imply confusion.

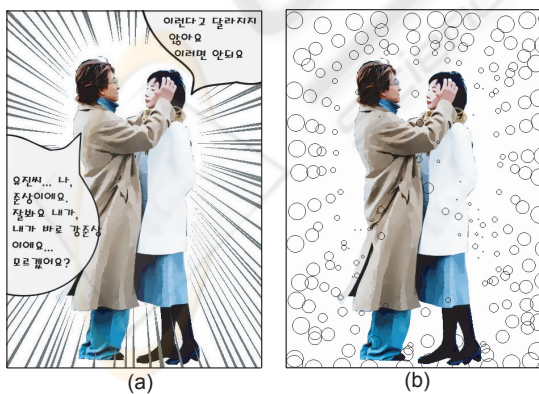


Figure 10: Two different background effects. (a) Concentration background texture with straight lines. (b) Concentration background with bubbles to show romantic atmosphere.

## ACKNOWLEDGMENTS

This work was supported by a grant from Non-Photorealistic Animation Technology project of ETRI(2005). We gratefully credit the thoughtful reviewers, who provided substantial constructive criticism on an earlier version of this paper.

## REFERENCES

Agarwala, A., Hertzmann, A., Salesin, D. H., and Seitz, S. M. (2004). Keyframe-based tracking for rotoscoping and animation. In *Proceedings of SIGGRAPH 2004*, pages 584–591. ACM Press.

Brostow, G. J. and Essa, I. (2001). Image-based motion blur for stop motion animation. In *Proceedings of SIGGRAPH 2001*, pages 561–566. ACM Press.

Hanl, C., Haller, M., and Diephuis, J. (2004). Non-photorealistic rendering techniques for motion in computer games. *Computers in Entertainment*, 2(4):11.

Kim, B. and Essa, I. (2005). Video-based nonphotorealistic and expressive illustration of motion. In *Proceedings of CGI*, pages 32–35.

Kurlander, D., Skelly, T., and Salesin, D. (1996). Comic chat. In *Proceedings of SIGGRAPH 1996*, pages 225–236. ACM Press.

McCloud, S. (1999). *Understanding Comics: The Invisible Art*. Kitchen Sink Press.

McCloud, S. (2000). *Reinventing Comics: How Imagination and Technology Are Revolutionizing an Art Form*. Perennial.

Nienhaus, M. and Dollner, J. (2005). Depicting dynamics using principles of visual art and narrations. *IEEE Computer Graphics and Applications*, 25(3):40–51.

Planetwide Games, I. (2005). Comic book creator.

Uchihashi, S., Foote, J., Girgensohn, A., and Boreczky, J. (1999). Video manga: Generating semantically meaningful video summaries. In *Proceedings of the 7th ACM International Conference on Multimedia*, pages 382–392. ACM Press.

Wang, J., Xu, Y., Shum, H.-Y., and Cohen, M. F. (2004). Video tooning. *ACM Transactions on Graphics*, 23(3):574–583.