Colorization of High-Frame-Rate Monochrome Videos Using Synchronized Low-Frame-Rate Color Data

Ching-Fan Chiang, Yang-Ming Yeh, Chi-Yun Yang, and Yi-Chang Lu(✉)

Graduate Institute of Electronics Engineering,
National Taiwan University, Taipei, Taiwan
{r01943001,d05943006,r03943126,yiclu}@ntu.edu.tw

Abstract. The frame rate of a color camera is usually limited by its maximum data bandwidth. To obtain high-frame-rate color videos, we propose to colorize high-frame-rate monochrome videos using data from a system composed of one high-frame-rate monochrome camera and two low-frame-rate color cameras. The cameras are synchronized by external triggering signals. With stereo matching and motion estimation algorithms, colorization of high-frame-rate monochrome videos can be realized. The system is very cost-effective, and the processing steps can be fully automated as demonstrated in the paper.

Keywords: Colorization · Multiple-camera systems · High-frame-rate video

1 Introduction

High-speed cameras are able to acquire videos at very high frame rates. These high-speed videos are very important in various fields such as scientific research and television production. However, there are some known disadvantages and limitations in high-speed cameras. For instance, a high-speed camera is often much more expensive than a low-frame-rate camera. Also, with a very high frame rate, the exposure time of each frame can be extremely short. Therefore, monochrome high-speed cameras are frequently used because of their stability in low light conditions. In addition, monochrome cameras provide the benefits of higher resolution since no demosaicing of color patterns is needed. Unfortunately, in most applications users favor color videos than monochrome videos. There are many studies to address general colorization issues. For example, the authors in [1] proposed to combine the methods of color transfer in [2] and feature mapping in [3] to transfer color characteristics from one color image to a greyscale image.

This work is partially supported by Ministry of Science and Technology, Taiwan, under Grant numbers MOST 106-2221-E-002-055, 107-2221-E-002-123-MY2 and 107-2622-8-002-009-TA.

S. Tominaga et al. (Eds.): CCIW 2019, LNCS 11418, pp. 276–285, 2019.
https://doi.org/10.1007/978-3-030-13940-7_21

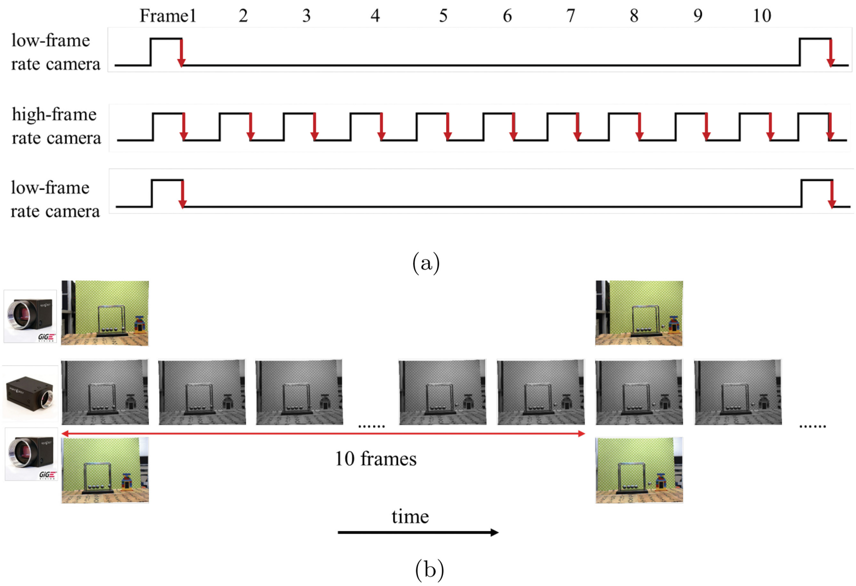

Fig. 1. (a) Triggering signals for cameras. (b) The captured image sequences.

On the other hand, some researchers [4,5] suggested using iterative methods with initial color annotation of a small portion within the image. Recently, deep learning networks are also applied to the field of colorization [6].

In this paper, to put emphasis on high-frame-rate applications, we set up a high-frame-rate monochrome video colorization system. We apply stereo matching and motion estimation techniques to colorize high-frame-rate monochrome videos using color information from nearby low-frame-rate color videos. The proposed system is capable of generating high-frame-rate colors video at a more affordable price while preserving the advantages of monochrome cameras.

2 Monochrome Video Colorization System

2.1 System Overview

The proposed system setup is illustrated in Fig. 1. Two Point Gray Flea3 low-frame-rate color cameras are used to provide color references. One Point Gray Gazelle high-frame-rate monochrome camera gives the complete image sequence.

To precisely control the triggering time of three cameras, we use FPGA to provide the triggers to three cameras. We set the triggering rate at 10 Hz for color cameras and 100 Hz for the monochrome camera. It means the FPS (frame per second) of the monochrome camera is ten times faster than that of color cameras. After rectification, two color cameras will provide color information to the high-speed camera. In the cross-camera colorization stage, we colorize every

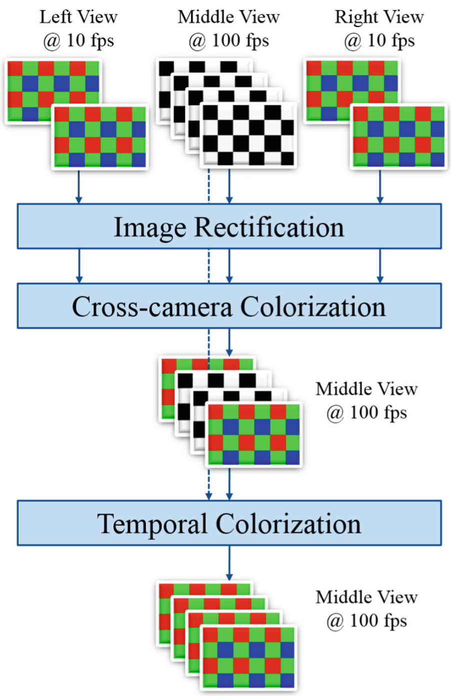

Fig. 2. The flow chart of our colorization system.

tenth monochrome image using the color information which captured at the same moment. In the temporal colorization stage, we used the colorized images in the monochrome sequence as the guide to generate the final high-speed color sequence. The flow chart is shown in Fig. 2.

2.2 Image Rectification Stage

Since our cameras are a set of stereo vision cameras, we should first calibrate three cameras and rectify each captured image. Secondly, we change the color domain of each color image from RGB to YCbCr since we prefer not modifying the luminance of monochrome images after colorization. The occlusion problem happens when using a two-camera stereo system. It means that, for certain parts of some objects in one view, we may not be able to find the corresponding pixels in the other view. To prevent this problem, we use three cameras to avoid this problem. For example, the insufficient information of occlusion regions between the left and middle views can be provided by the right view.

2.3 Cross-Camera Colorization

Since sensors of cameras are not totally identical, the luminance values captured are not the same even the same camera setting is used. If the luminance information differs between images, the result of stereo matching is not convincing. Thus,

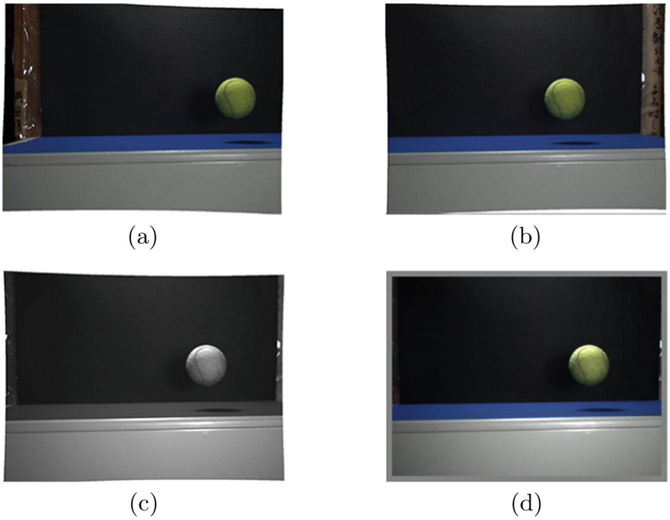

Fig. 3. (a) Left view, (b) right view, (c) middle view, and (d) the colorized image after applying cross-camera colorization.

before implementing stereo matching, we need to adjust the luminance of two low-frame-rate color images, to make their luminance similar to the one in the high-frame-rate monochrome image, using the transformation proposed by [3]:

$$C'(p) = \frac{\sigma_G}{\sigma_C}(C(p) - \mu_C) + \mu_G. \tag{1}$$

$C(p)$ and $C'(p)$ stand for the original and calibrated luminance in the low-frame-rate color image, respectively. μ_C and σ_C are the mean and standard deviation in the color image, while μ_G and σ_G are the mean and standard deviation in the monochrome image. This equation brings the mean and standard deviation of the color image to the same as those in the monochrome image.

After adjusting the luminance for two color images, we utilize a stereo matching algorithm to find the matching point for each pixel in the monochrome image. By creating a block of 17×17 pixels for each pixel in the monochrome image, we generate blocks of the same size along the corresponding 1-D search line in the two reference color images. Then, we perform the Sum of Absolute Difference (SAD) similarity check between these blocks with every possible horizontal shift which depends on the maximum disparity of the objects in the scene. Finally, we choose the best matching point from two color images and use its Cb, Cr as the color information for the current monochrome pixel. After matching all color information for each pixel in the monochrome image, one-tenth of the monochrome images are colorized. Then, we move to the next step to colorize other high-frame-rate monochrome images using colorized monochrome images obtained.

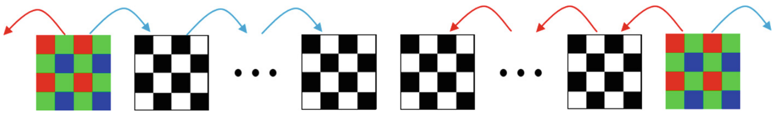

Fig. 4. The scheme of temporal colorization.

2.4 Temporal Colorization

In this step, we want to colorize the monochrome images which do not have corresponding color information from the low-frame-rate images. We use the colorized high-frame-rate monochrome images to colorize other monochrome images by finding motion vectors using motion estimation algorithm.

We treat the colorized monochrome images as the reference frames and the current monochrome image as the target frame to be processed. We first divide the current frame into macro blocks. Each macro block is 6×6 pixels, and its corresponding search region on the reference frame is 15×15 pixels. Then we compute the similarity between the current block and each candidate block in the search region. When the best match has been found, we record this motion vector as the spatial offset between the two blocks, which also means that the color information is copied from the best matched block on the reference frame. In our proposed system, we use full search method for motion estimation algorithm. Although larger block size is preferred for SAD, the colorization results would be discontinuous at the edges of blocks. Thus, we propose to use two different types of blocks. The macro block mentioned above is Matching block, and the other type is called Painting block. We should keep the size of Painting block small (2×2 pixels) enough to minimize the blocking effect of colorization results. On the other hand, we use a larger Matching block when computing the similarity to ensure the accuracy. The reason that we do not use a single pixel as Painting block is a result of the trade-off between image quality and computing time, since the 2-D search region in this stage contains far more candidates than the ones in the previous stage. Figure 4 shows the procedure to propagate the color information in the temporal colorization step. Each colorized middle frame obtained from the cross-camera colorization step provides color information to its previous four monochrome frames and the latter five monochrome frames as indicated by the red and blue arrows shown in Fig. 4, respectively.

3 Experimental Results

Since the images are directly obtained from the prototype system we built, there is no ground truth of color high-frame-rate images available to calculate PSNR and SSIM metrics. Nevertheless, we show the qualitative results of three sequences to demonstrate the feasibility of the system. The first and eleventh images are the colorized results from cross-camera colorization.

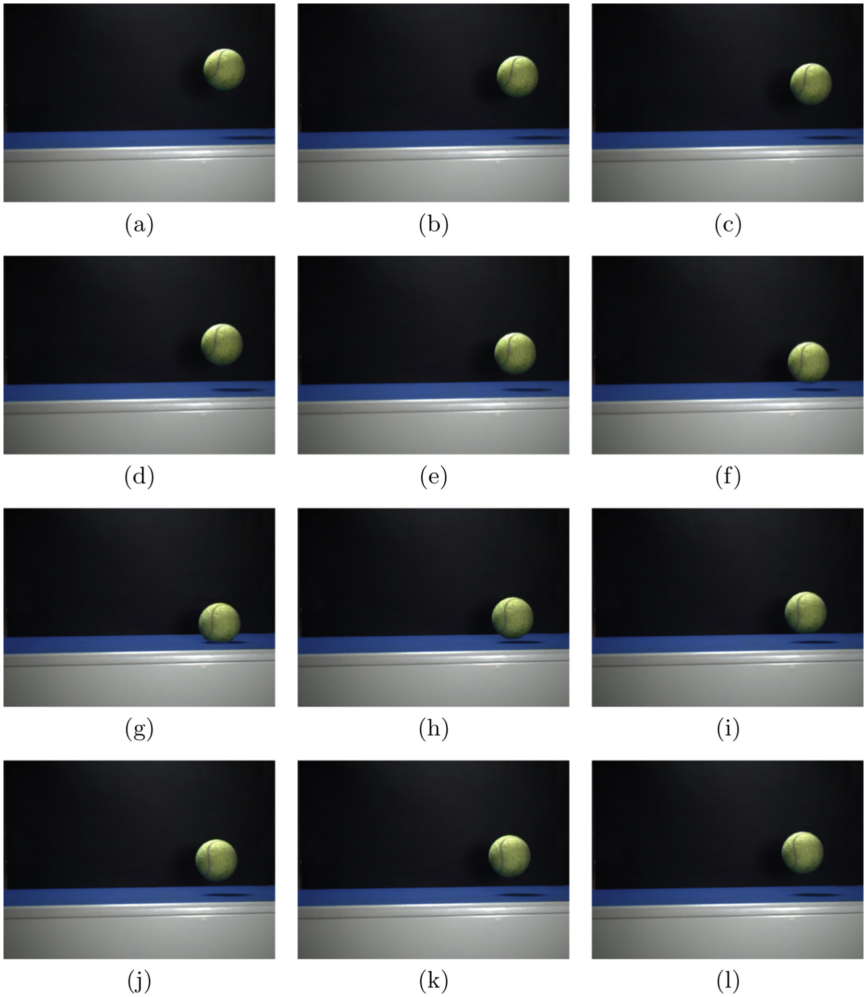

Fig. 5. Scene with an object in rigid motion: (a) and (k) are the colorized images after applying cross-camera colorization. The rest images are the colorized results after temporal colorization.

The rest images are the colorized results after temporal colorization. Since the frames are obtained directly from the monochrome camera, the information of the luminance channel is always correct. Combining with the fact that chrominance information is less sensitive to human eyes, the flaws of the colorized video are very difficult to be spotted.

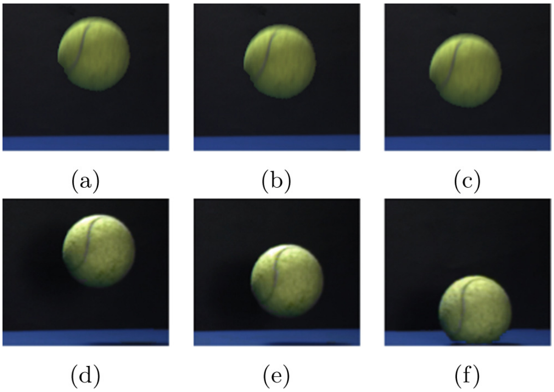

Fig. 6. (a)–(c) are from a sequence generated by interpolation of two adjacent frames from a low-speed-color camera. (d)–(f) are colorized frames using the proposed pipeline.

3.1 Scene with Objects in Rigid Motion

In the first experiment, we record the bouncing behavior of a tennis ball. The input images for cross-camera colorization are shown earlier in Fig. 3. The edges and patterns in Fig. 3(d) are clearer than those in the images from the left and right color cameras because the intensity information that comes from the middle camera has a very short exposure time. Since the trajectory of the ball is not linear, if an ordinary interpolation method is used, the interpolated frames could not look natural. In this work, with the high-frame-rate data, we are able to retain the real temporal information. Figure 5 shows the sequence after our temporal colorization flow. The non-linear motion, especially at the moment of impact, is well-preserved and colorized. Figure 6 compares the sequences obtained from the interpolation method and the proposed approach. The results in the first row of Fig. 6 are interpolated from the two adjacent frames captured by a low-speed color camera. Therefore, the results look blurred. When compared with our method, we can see that interpolation method could not generate the correct trace of the tennis ball.

The second experiment is to make the background more complicated. Besides, each ball in the Newton's cradle is not completely still. If we zoom in to a small region of Fig. 7, we can find some defects in the colorization results. Since we hold the intensity of high-frame-rate video constant, the impact of colorization errors are difficult to be discovered unless the video is played at a very low speed on a large screen.

3.2 Scene with an Object in Non-rigid Motion

Here we conduct another experiment, a scene with an object in non-rigid motion, which is considered harder to estimate the motion vectors due to the deformation of the object between frames. However, we can still obtain the colorized video frames as shown in Fig. 8. We can see that the colorized water balloon looks very realistic. We also try to increase the deformation level by dropping the water balloon from a higher place. In that case, the water balloon would form larger deformation within a shorter time period. Figure 9 shows the zoomed-in images of our colorized results of the case with larger deformation. Although there are some errors due to large deformation as highlighted in Fig. 9(c), the flaws in video are not obvious to human eyes.

Fig. 7. Scene with objects in rigid motion: (a) and (k) are the colorized images after applying cross-camera colorization. The rest images are the colorized results after temporal colorization.

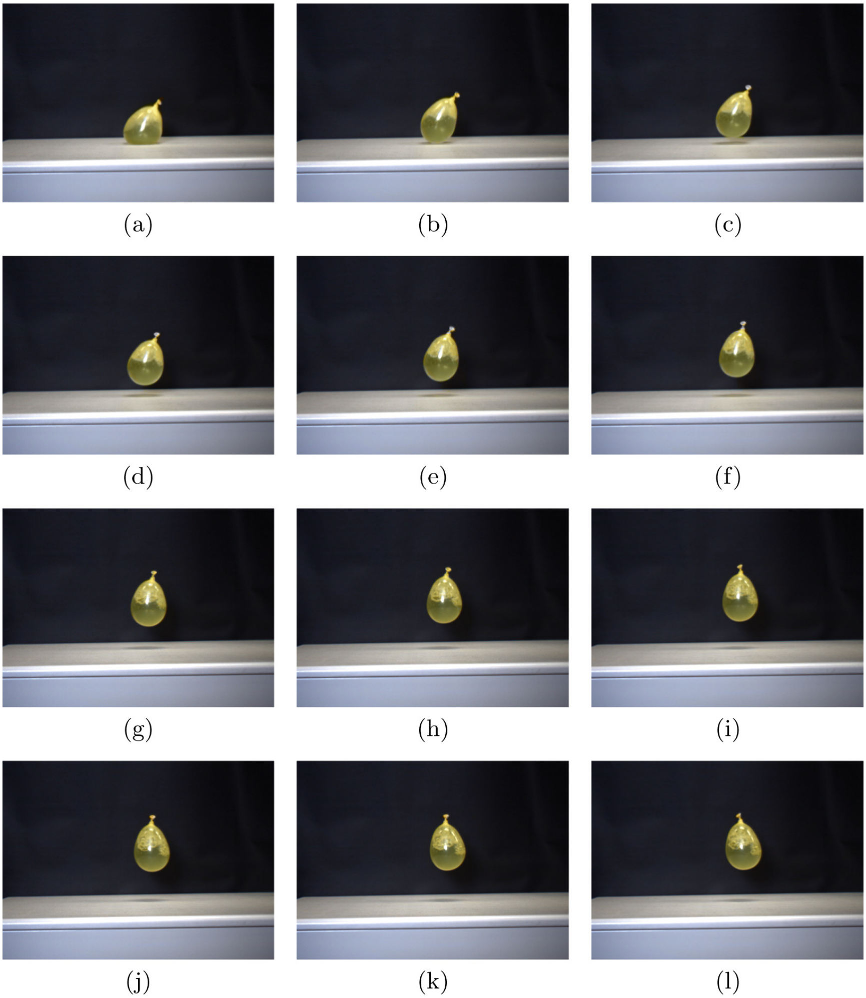

Fig. 8. Scene with an object in non-rigid motion: (a) and (k) are the colorized images after applying cross-camera colorization. The rest images are the colorized results after temporal colorization.

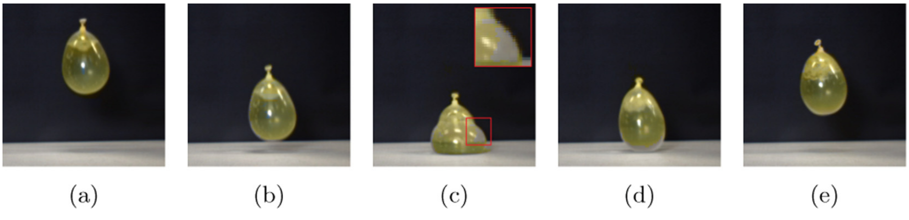

Fig. 9. The zoomed-in results, (a)–(e), are from the five consecutive frames of the case with larger deformation.

4 Conclusion

In this paper, we propose a high-frame-rate monochrome video colorization system which is composed of one high-frame-rate monochrome camera and two low-frame-rate color cameras. The three cameras are synchronized by external triggering signals. With stereo matching and motion estimation algorithms, colorization of high-frame-rate monochrome video can be realized using the proposed system. As demonstrated in the paper, the image quality is good and the process can be fully automated.

References

1. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. ACM Trans. Graph. **21**(3), 277–280 (2002)
2. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Comput. Graph. Appl. **21**(5), 34–41 (2001)
3. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 327–340 (2001)
4. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. ACM Trans. Graph. **23**(3), 689–694 (2004)
5. Nie, D., Ma, Q., Ma, L., Xiao, S.: Optimization based grayscale image colorization. Pattern Recogn. Lett. **28**(12), 1445–1451 (2007)
6. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pp. 415–423 (2015)