Sensors and Materials, Vol. 33, No. 9 (2021) 3283–3293
MYU Tokyo

S & M 2688

Algorithm to Merge Images to Increase Browsing Speed of Smart Video Surveillance System

Xiaomei Zhang,[1] Jinbin Zheng,[1] Wenjuan Wang,[1]
Hsien-Wei Tseng,[2] and Cheng-Fu Yang[3,4*]

[1]College of Mathematics and Information Engineering, Longyan University, Fujian 364012, China
[2]College of Artificial Intelligence, Yango University, Mawei District, Fujian 350015, China
[3]Department of Chemical and Materials Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
[4]Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

(Received March 23, 2021; accepted June 29, 2021)

Keywords: fast image browsing technology, smart video surveillance system, discrete wavelet transform, fake motion

With the increasing use of digital video surveillance systems in the past few years, a greater number of images are being stored and analyzed. Thus, large amounts of time and labor are required to browse stored images when searching for specific people and objects. Based on the concept of video synopsis, we proposed an image processing flow to decrease the number of images requiring browsing, thus reducing the browsing time of surveillance images. We first used the discrete wavelet transform (DWT) to reduce the image resolution. We also used a low-pass filter to filter out the noise and fake motion generated from the external environment. Next, we removed the redundant information from the timeline, which allowed us to delete images in which no incidents occurred, then we used the proposed algorithm to extract the foreground objects. Finally, we calculated the space information of the recorded images, then a continuous event was integrated and merged in an image, thus reducing the time required for video playback.

1. Introduction

With the advances in cameras in recent years, digitization and high-resolution monitoring or surveillance systems have become increasingly popular. A video monitoring or surveillance system is composed of cameras, monitors (or display units), recorder units, and so on. Cameras may be digital or analog, and there are over four million surveillance camera systems used for the long-term monitoring of streets in cities in the United Kingdom. Thus, digitization monitoring or surveillance systems are playing an increasingly important role in security systems. However, it is difficult to manage the large amount of surveillance data, and people responsible for watching surveillance images may become less vigilant after long-term monitoring, reducing the effectiveness of systems. When using past video browsing technologies, for example, video skimming[1,2] and video summarization browsing,[3–5] it was found that most

*Corresponding author: e-mail: cfyang@nuk.edu.tw
https://doi.org/10.18494/SAM.2021.3391

ISSN 0914-4935 © MYU K.K.

processed videos are of no interest. Therefore, image summarization technologies are needed to perform quick browsing to search for specific events in long videos or a large number of images.

Video synopsis is a technology that simultaneously presents different events and enables the review of hours of images or videos in a few minutes.[6,7] This technology can track and analyze moving objects or events and can convert video streams into a database of activities and objects.[8] Nowadays, most multimedia surveillance systems use multiple or different types of sensors, which have different costs and have different sensing capabilities. How to select the most suitable and optimal number of sensors and how to determine their positions in the given monitored area are important tasks when constructing multimedia surveillance systems.[9] In this paper, an algorithm was proposed to delete images that do not show events of interest (incidents) and merge images to increase the browsing speed of a smart video surveillance system. Because a long video needed to be processed before it was browsed, the first step of the proposed algorithm was to delete the images without incidents or any changes. Next, the tracking or surveillance foreground objects were extracted and the spatial information of the images was calculated; then, the continuous events were integrated or merged into single images. Therefore, long-term continuous events were integrated into a small number of simplified images for storage. Our experimental results demonstrated that when a long video was processed as required, we needed to integrate and merge the incidents occurring in the same images as much as possible. The novelty of this study is that we proposed an algorithm that can process videos and images to enable quick browsing. As a result, we can merge images of a quickly moving object obtained by a surveillance system and decrease the space required for data storage.

2. Object Detection

2.1 Color space transformation

Color transformations of an image involve converting a three-layer optical wave (red, green, blue: RGB) into another three-layer optical wave. To decrease the volume of data required for image processing, grayscale image calculation was used to process long videos or a large number of images. The grayscale values were obtained using Y (luminescence) in Eq. (1), where R, G, and B are the values for red, green, and blue, respectively.[10]

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \tag{1}$$

2.2 Application of discrete wavelet transform (DWT)

The higher the resolution of digital images, the more computing time is needed to process them. Therefore, it is important to propose technologies for decreasing the processing time. The algorithm proposed in this study reduces the resolution of images through the adjustment of the grayscale, allowing us to reduce the number of subsequent operations. As compared with the use of an image mean filter, the low-frequency images obtained from the DWT are more complete, and the DWT is suitable for performing various treatments on the foreground of images.[11] The

important point is that the DWT can retain not only the low-frequency part required, but also some of the high-frequency part, enabling important features of images to be preserved. In 2009, the symmetric mask-based DWT was proposed,[12,13] which used four mask-based operations to obtain four frequency bands by a two-dimensional DWT, which were low-low frequency, low-high frequency, high-low frequency, and high-high frequency. In this study, we only used the low-low frequency information to extract objects; therefore, we only needed one type of mask operation to obtain the low-low frequency images. The two-dimensional DWT needed to be performed for the four-band transform to obtain the low-low frequency information. As compared with the use of the two-dimensional DWT, our proposed algorithm can reduce the computational complexity and required memory.

2.3 Image binarization

Because we obtained the foreground from the result of background subtraction, we first corrected the error of each photosensitive element. The Gaussian function is often used to express the probability density function of a normally distributed random variable. The Gaussian function has different formats and is widely used to describe normal distributions in statistics. Various Gaussian filters are used in signal processing, and two-dimensional Gaussian functions are usually used for the operation of Gaussian blurring in image processing. When the average value of the Gaussian distribution is 0, the probability density function of image binarization is expressed as

$$P(X) = \frac{1}{\sigma_c \sqrt{2\pi}} e^{-\frac{(X-u)^2}{2\sigma_c^2}}, \tag{2}$$

in which the average value u is 0, X is the difference between the pixels of two images at the corresponding positions, and σ_c^2 is the variance of the lens noise. When Eq. (2) was used to confirm the calculation result, the amount of noise left was about 4%, which matched the characteristic value of 2σ; thus, we learned that the prediction σ is approximately equivalent to σ_c, as shown in Fig. 1. Therefore, when the difference between the two images obtained by subtraction was smaller than the threshold value, we recognized this difference as the influence of noise and its value was treated as zero. When the threshold value was found, we performed image binarization. For convenience, the foreground of the images was expressed using the binarization image F_B. Thus, the pixels were obtained from the subtraction of images, and we used Eq. (3) to generate the binarization image used for the judgment of objects of interest.

$$F_B(x,y) = \begin{cases} 1, & (Y_{x,y} - T) > 0 \\ 0, & (Y_{x,y} - T) \le 0 \end{cases} \tag{3}$$

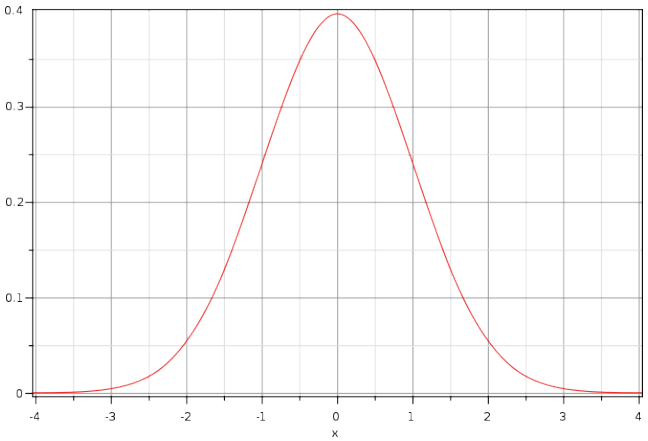

Fig. 1. (Color online) Schematic diagram of noise distribution of lens and threshold value.

2.4 Connection and component labeling of objects

Gaussian mixture models with different formats are one kind of probabilistic model, and they can present normally distributed subpopulations within an overall population. When we used a Gaussian mixture model to extract foreground objects, we imported the images to transform them into binarization images. To connect the motion of continuous objects into complete and consecutive events, it was necessary for the system to label every object automatically. This was because when objects were not labeled, it was difficult to track all the objects in the subsequent process. Different from past recursive labeling algorithms, we used the connected-component labeling (CCL) algorithm to label the detected objects, which included two steps of assignment and grouping.[14] The input binarization image is shown in Fig. 2(a). First, the CCL algorithm specifies a temporary label number for every foreground, as shown in Fig. 2(b). Through the temporarily assigned labels, it is established which foregrounds are connected together to create a group table, as shown in Table 1. In the second step, the values in the group having the same group label are merged using the group table, then each foreground is separated to complete the labeling process.

3. Image Extraction

3.1 Application of stroboscopic effect

The stroboscopic effect is a visual phenomenon caused by aliasing, which occurs when continuous rotational or other cyclic motion is represented by a series of instantaneous or short samples at a sampling rate close to the period of the motion. We used the stroboscopic effect to process the extracted continuous object tube, then the well-divided multisegmented object tubes were placed on the same image, as shown in Fig. 3.

After the above processing, we found that at different times, the same object appeared in the same image. Therefore, it was necessary to process the complete flows of objects and consider

255	0	255	0	255
255	0	255	0	255
255	0	255	0	255
255	0	0	0	255
255	0	0	0	255
255	255	255	255	255

(a)

254	0	234	0	214
254	0	234	0	214
254	0	234	0	214
254	0	0	0	214
254	0	0	0	214
254	254	254	254	254

(b)

Fig. 2. Representation of image object before and after labeling: (a) binarization image and (b) specified temporary label.

Table 1
Group table.

Group ID	Equivalent labels
1	254, 214
2	234

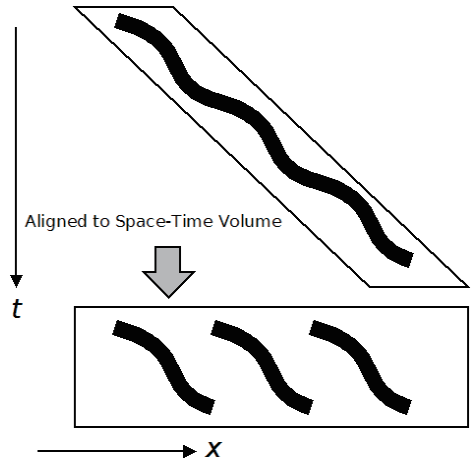

Fig. 3. Schematic diagram of stroboscopic effect.

whether objects had left the image. However, the objects remaining in the image still could not be analyzed; therefore, the proposed algorithm required a temporary storage space to store the object stream. Thus, the proposed method of this paper only processed objects in the current image. Because there was no tube, we were unable to know the number of times that the tube would be divided in advance. In this way, a brief image-processing algorithm was produced as follows. We start with an original image S that consisted of n continuous images s_0 to s_{n-1}, with the original image S expressed as follows:

$$S = \{s_0, s_1, s_2, ..., s_{n-1}\}. \tag{4}$$

Some continuous images in S contain object A, which form the set

$$S_{objA} = \{(s_{A1}, s_{A2}, s_{A3}, ..., s_{At}) \in S : A1 \geq 0,\ At \leq n-1\}. \tag{5}$$

By applying the operation $P_{objA}(s_{At})$, we obtain a location description of object A in image At:

$$P_{objA}(s_{At}) = (x_{min}, x_{max}, y_{min}, y_{max}) \in objA|_{At}. \tag{6}$$

Using Eqs. (5) and (6), we find the minimum At that satisfies the following condition:

$$P_{objA}(s_{At}) \text{ does not overlap } P_{objA}(s_{A1}). \tag{7}$$

Then the obtained rough length L of a sequence of simple shortcut images is

$$L = At - 1. \tag{8}$$

When the length L is set as a basic unit, the simple shortcut images for the contents of video s_{g0} to s_{gL-1} are expressed as follows.

$$\begin{aligned}
&\left\{objA|_{A1}, objA|_{A\lambda L}: \lambda \geq 1, \lambda L \leq At\right\} \in s_{g0} \\
&\left\{objA|_{A2}, objA|_{A\lambda L+1}: \lambda \geq 1, \lambda L + 1 \leq At\right\} \in s_{g1} \\
&\left\{objA|_{A3}, objA|_{A\lambda L+2}: \lambda \geq 1, \lambda L + 2 \leq At\right\} \in s_{g2} \\
&\qquad\qquad \vdots \\
&\left\{objA|_{AL}, objA|_{A\lambda L}: \lambda \geq 2, \lambda L \leq At\right\} \in s_{gL-1}
\end{aligned} \tag{9}$$

The simple shortcut image S_s is expressed as

$$S_s = \left\{s_{g0}, s_{g1}, s_{g2}, ..., s_{gL-1}\right\}. \tag{10}$$

The collection of all simple shortcut images can be expressed as shown in Fig. 4.

3.2 Production of simple shortcut images

The proposed theorem was based on the bounding box of an object; if every image had to be judged, the frequency of memory reading and the time required would be too high. To solve this problem, the flow proposed in this paper had a simplified process for judging overlap. From the judgment in the first round of image production, we obtained the rough length of a simple shortcut image; then, we predicted that in the next L rough images, there would be little change in the speed of an object moving in them. Thus, we only needed to extract objects and incorporate them in the belonging set until we had completed the analysis of L images; then, we performed the judgment of overlap. The object used for judgment here was $P_{objX}(s_{A1})$, where X represents all detected objects used to decide whether to maintain the length of the image L. If we judged that images overlap, then this result indicated that the moving speed of the object had decreased and L must be changed to L + 1; then, the object was incorporated into the added

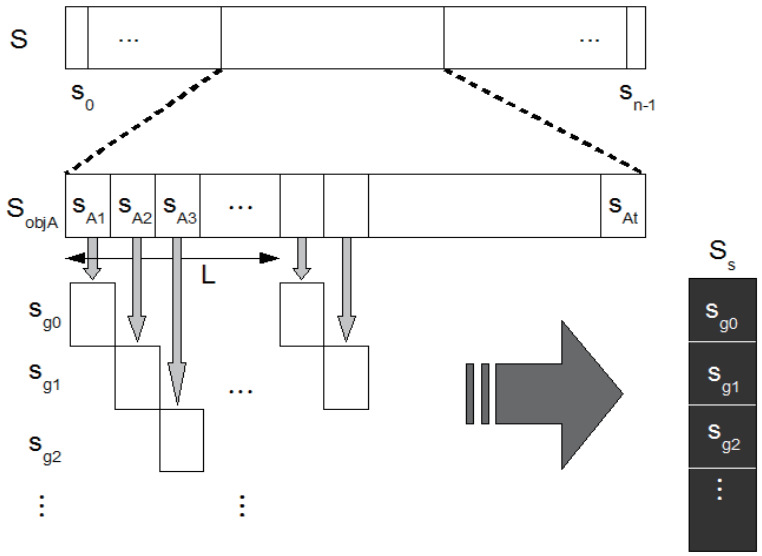

Fig. 4. Proposed classification of simple shortcut images.

image. Next, we analyzed the length of L + 2 images and made the judgment of overlap on $P_{objX}(s_{A1})$ until no overlap occurred. Then, we obtained a new image length L and recorded the current location $P_{objX}(s_{AL+2})$ for subsequent judgment and processing. However, there were two situations where overlap occurred. First, after the objects were interlaced, they followed the same route, and second, after the objects were staggered, they did not follow the route taken by another person. The second situation was relatively simple: we used the previous process to record the route. However, when we faced the first situation, there were problems with the above process. Thus, in this study, we added the following improvements when recording the location of an object.

The area through which the object passed was set as a dead zone. When the description of the location of any object overlapped with the dead zone, it was determined that the condition for increasing L was satisfied, but in this situation, the images were still not condensed. Thus, when L increased to twice its original length, we started another round of simple shortcut image production and added the s_{g0} image to solve the problem of object obscuration. The approach used to handle the dead zone is shown in Fig. 5. For consistency of expression, the dead zone also used the same description as that of Marco Block (MB, a processing unit in image and video compression formats based on linear block transforms), i.e., the maximum and minimum values on the x and y axes. The initial value $P_{objX}(s_{A1})$ was the position of the object when it first appeared. Every time the images were subjected to the judgment of overlap, if the current object location was described as (x_{Cmin}, x_{Cmax}, y_{Cmin}, y_{Cmax}) and the compared target object was described as (x_{Tmin}, x_{Tmax}, y_{Tmin}, y_{Tmax}) while we were recording the object location, we only needed to find min(x_{Cmin}, x_{Tmin}), min(y_{Cmin}, y_{Tmin}), max(x_{Cmax}, x_{Tmax}), and max(y_{Cmax}, y_{Tmax}); then, we updated the coordinates of the dead zone to complete the establishment of the dead zone.

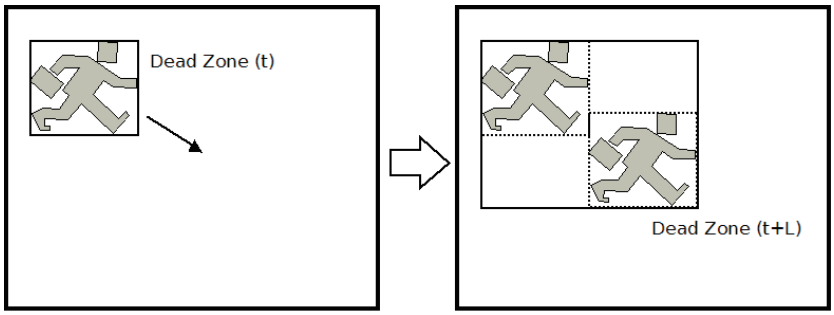

Fig. 5. Construction of dead zone. (a) Initial value of the dead zone of an object, where the arrow is the direction of movement. (b) Updating the location of the dead zone at the next time.

Next, we treated the closest object as the same object, and only objects belonging to the s_{g0} image were subjected to judgment. As the algorithm for distance calculation, we used the sum of the differences of the x and y coordinates of the two centroids as the basis for the decision, and we took the smaller sum and considered it as the same object. The $O(n)$ notation is used to classify an algorithm according to how its run time or space requirement grows as the input size grows, and the time or space requirement of the proposed algorithm used for image processing is $O(n^2)$ for an image size of $n \times n$. Thus, the complexities in both the difference of calculations and the number of comparisons are approximately $O(n^2)$, but only a very small proportion of images in the process need to be processed like this. When Eq. (11) is satisfied, where F_T is the total number of images and f is the number of images that need to be tracked, then the tracking complexity of the process flow is roughly equal to $O(n)$, which means that even if the number of objects increases, the complexity will only increase linearly.

$$\frac{F_T}{f} \geq n \tag{11}$$

For the algorithm proposed in this paper, the features of the identified objects were the centroid positions because only two coordinate values needed to be stored, which is less than the four coordinate values required to describe a location. Therefore, the recognition speed was enhanced and the space required for data storage was reduced.

4. Experimental Results

To demonstrate the effectiveness of the proposed algorithm, we evaluated it using two videos with video graphics array (VGA) and quarter video graphics array (QVGA) resolutions, and we used the proposed algorithm to process the video surveillance or monitoring images. As important parameters for the processing, we found from the measured values when we aimed the target camera that the critical value was equal to 14 and the number of executions for the DWT was level one (2D 1-Level DWT).

4.1 First test video

Figure 6(a) shows the original test image, which had QVGA resolution, was shot indoors, and the light source was natural light. Figure 6(b) shows the simple shortcut image, which was obtained by processing with our proposed algorithm. It can be seen from the result that the total number of simple shortcut images was only 8.73% of the number of original images and that the image processing speed was 29 frames per second (FPS), i.e., real-time image-processing speed was achieved. In addition, the original images were found to include three independent events after they were analyzed, and the simple shortcut images were also divided and saved as three image files. Table 2 shows a comparison of the original image and the simple shortcut images.

4.2 Second test video

Figure 7(a) shows the second test image, which had VGA resolution, was shot outdoors, the light source was natural light, and the weather was rainy. Figure 7(b) shows the simple shortcut image, which was obtained by processing with our proposed algorithm. It was seen from the result that the total number of simple shortcut images was only 12.96% of the number of original images and the image processing speed was 23 FPS, which was slower than the processing speed of the QVGA image in Fig. 6, although real-time image-processing speed was still achieved. The original images were also found to include three independent events after they were analyzed, and the simple shortcut images were divided and saved as three image files. A comparison of the original image and simple shortcut images is shown in Table 3.

(a) (b)

Fig. 6. (Color online) Result for the first test video: (a) original image and (b) test result.

Table 2
Comparison table for the first test video.

	Test image	Simple shortcut images
Processing speed	29.97 FPS	29 FPS
Total number of sheets	1053	92

(a) (b)

Fig. 7. (Color online) Result for the second test video: (a) original image and (b) test result.

Table 3
Comparison table for the second test video.

	Test image	Simple shortcut images
Processing speed	29.97 FPS	23 FPS
Total number of sheets	1420	184

5. Conclusions

In this study, we proposed a method to effectively decrease the number of images by over 85% in monitoring or surveillance systems by producing simple shortcut images. For example, for a QVGA system, the total number of simple shortcut images was only 8.73% of the number of original images, and for a VGA system, the total number of simple shortcut images was only 12.96% of the number of original images. Because the number of images was greatly reduced, the playback time was much shorter than that for the original images, enabling users browsing images to find events of interest in a short time. Even when the original images had high resolution, we still used the low-low frequency of the DWT to reduce the resolution of original images and the high-frequency noise. This algorithm can reduce the number of calculations when processing images and enhance the efficiency of browsing the images of surveillance systems.

Acknowledgments

This work was supported by the Young and Middle-aged Teacher Program of the Education Department of Fujian (JAT200604), project numbers MOST 108-2221-E-390-005 and MOST 109-2221-E-390-023.

References

1 N. Petrovic, N. Jojic, and T. Huang: Multimedia Tools Appl. **26** (2005) 327.
2 L. Zhang, Y. Cao, G. Ding, and Y. Wang: IEEE Int. Symp. Multimedia (December 2008) 667–672.
3 C. Pal and N. Jojic: IEEE Conf. Computer Vision and Pattern Recognition 2 (June 2005) 1192.

4 A. Pope, R. Kumar, H. Sawhney, and C. Wan: Proc. 32nd Asilomar Conf. Signals, Systems & Computers (1998) 915–919.
5 Y. Gong and X. Liu: IEEE Int. Conf. Image Process. **3** (2001) 362–365.
6 S. A. Ahmed, D. P. Dogra, S. Kar, and P. P. Roy: IEEE Trans. Circuits Syst. Video Technol. **29** (2019) 1985.
7 S. Ghatak, S. Rup, B. Majhi, and M. N. S. Swamy: IEEE Trans. Consum. Electron. **66** (2020) 144.
8 C. H. Hsia, S. C. Yen, and J. H. Jang: Sens. Mater. **31** (2019) 1803.
9 G. S. V. S. Siva Ram, K. R. Ramakrishnan, P. K. Atrey, V. K. Singh, and M. S. Kankanhalli: Video Surveillance & Sensor Networks, Santa Barbara, California, USA (VSSN06, 2006).
10 J. S. Chiang, C. H. Hsia, H. W. Peng, and C. H. Lien: J. Appl. Sci. Eng. **17** (2014) 341.
11 C. H. Hsia and C. F. Lai: Sens. Mater. **32** (2020) 3221.
12 C. H. Hsia, J. M. Guo, and J. S. Chiang: IEEE Trans. Circuits Syst. Video Technol. **19** (2009) 1202.
13 C. H. Hsia and J. S. Chiang: Int. J. Innovative Comput. Inf. Control **8** (2012) 1.
14 K. Wu, E. Otoo, and K. Suzuki: J. Pattern Anal. Appl. **2** (2009) 117.