Implementing Content Based Video Retrieval Using Speeded-Up Robust Features

Jos Timanta Tarigan
Faculty of Computer Science and Information Technology
Universitas Sumatera Utara
Medan, Indonesia
jostarigan@usu.ac.id

Poltak Sihombing
Faculty of Computer Science and Information Technology
Universitas Sumatera Utara
Medan, Indonesia
poltak@usu.ac.id

Evi P. Marpaung
Faculty of Computer Science and Information Technology
Universitas Sumatera Utara
Medan, Indonesia
evimarpaung131401107@gmail.com

***Abstract* - In this paper, we propose an implementation of Content-Based Video Retrieval (CBVR) using Speeded-Up Robust Features (SURF). Given an image as a query, the application looks through a set of videos and pick the ones contain frame similar to the image query. Our objective is to measure the performance of the algorithm. The performance is measured using three variables: recall, precision, and running time. We used two sets of samples to perform the test: in-frame and not-in-frame. Furthermore, we limit the samples only to contain these 5 categories: body parts, kitchen and eating utensils, fruits, and pets. The test shows the program gives a 57.75% average recall value and 37.5% precision value for not-in-frame test, while the in-frame-test gives 51% and 59% for recall and precision value respectively. Moreover, the running time data shows there is no relation between in-frame/not-in-frame and speed. Running time performance highly depends on the image query and the length of the video.**

Keywords - Content Based Retrieval, Video Retrieval, Feature Extraction

I. INTRODUCTION

One of the most popular subject in information retrieval is video retrieval; a subject that discuss a method to collect valuable or related information from a video. With the massive amounts of videos available on the internet, video retrieval is important to automatically search through the contents. Automated copyright infringement search engine and video-based face detection are two of the many examples of video retrieval implementation. Both of these examples is practically impossible to be performed manually by humans due to the massive amount of content available. To tackle this problem, some researches have developed a method to automate this process.

The early phased of video retrieval is based on keywords attached the the video. The idea of this method is simple; given a set of videos, the algorithm will look through the video's keywords and find the one similar to the query given by the user. While the implementation is easy and can be accurate, this concept highly depends on the availability of the keywords. Hence, the method requires human intervention to provide the suitable keywords and cannot be fully automated by computers.

To make it fully automated, the system must have the capability to gather and process video content such as color, shape, and texture, and conclude the content based on the gathered information. Speeded-Up Robust Features (SURF) algorithm is capable to perform this task by detecting local features of an image. Published by Herbert Bay in 2006 [1], this algorithm is fast enough to detect the features in an image, hence it is suitable to be performed in a video retrieval which contains massive amount of images. Closely related to this research is a work Asha et al. [6] that developed a content based video retrieval based on SURF. The result gives a 78% accuracy in detecting objects in videos.

In this paper, we would like to present a performance test of a SURF-based CBVR system. Similar to the work by Asha et al., our objective is to investigate the performance of SURF-based CBVR using a different set of samples. To perform the test, we build a CBVR application based on SURF detector. Later, we collect categorized video and image samples to be used on the test. During the test, we collect the necessary data for performance observation. We use two measurements to value the performance: accuracy, and run-time. Moreover, we measure accuracy by collecting precision and recall values from the test.

II. RELATED WORKS

A. Content Based Video Retrieval

Video content retrieval or Content Based Video Retrieval (CBVR) is a method of retrieving content-based video files based on the visual features of the video. In this context, the

content includes colors, textures, object shapes, or other information that can be obtained to represent the frame of images in the video. There are many methods of CBVR have been proposed in previous researches. Amir et al. developed TRECVID, a content based retrieval for video that uses global descriptor by computing color histogram and color moments for video retrieval [9]. Shen et al. developed UQLIPS, a Near-duplicate video clip (NDVC) detection system that summarized each video to a single vector [10].

The system proposed on this paper however uses local features. The overall process of the whole system is shown in figure 1 below.

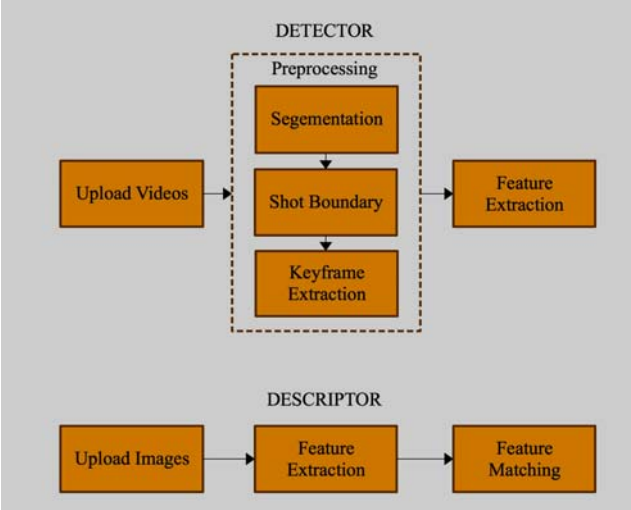

Fig. 1. Common CBVR Workflow

The first step of CBVR is video segmentation. The purpose of this process is to split the videos into parts with meaningful sections called segments. While segmenting video can be done easily by user, computer-automated unsupervised segmentation process is complicated.

The segmentation process can be summarized as follow: Given a video V with n number of frames, the segments of $V = f(I_i, t)$ where $i = 1, 2, 3, \ldots, n$ are represented as $I_1, I_2, I_3, \ldots I_n$. These segments are then processed by implementing shot boundary detection. The objective of this process is to define a group of consecutive frames that are temporally and visually close to each other using an auto dual threshold approach. A single shot S can be defined as $S = g(I_k, t), k = i, i+1 \ldots j$ where $1 \leq i < j \leq n$. The next step is to pick a key frame that can represent a scene from a segment. Usually, key frames selection can be performed by looking for a frame with the least difference from the other frame. In our system, we use SURF algorithm to perform feature extraction.

B. Speeded-Up Robust Features

Speeded-Up Robust Features (SURF) is an algorithm to detect and local feature of an image. It is capable to detect a specific feature that represent the image. This feature usually called local descriptor. Introduced by Bay et al [1], SURF performance is capable to outperform other implementation of interest point detector proposed by Harris et al. [2] and Lindeberg et al. [3]. Both of these research is capable to detect interest point based on corner while Lindeberg's research is capable to perform a scale invariant detection.

Many feature extraction researches use SURF as its descriptor. Huang et al. developed a wood image retrieval using SURF descriptor [7] to extract the feature of wood image, which can be used to automate plant species detection. Kim et al. developed an automated face detection using SURF as its descriptors and Support Vector Machines as its classifiers [8].

SURF uses Hessian matrix for its detector since it has a considerably good performance in both time and accuracy. Given a point $X = (x, y)$ in an image I, the Hessian matrix $\mathcal{H}(X, \sigma)$ in X at scale σ is defined as follows:

$$\mathcal{H}(X, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad (1)$$

where $L_{xx}(x, \sigma)$ is the convolution of the Gaussian second order derivative $\frac{\partial^2}{\partial x^2} g(\sigma)$ with the image I in point x, and similarly for $L_{xy}(x, \sigma)$ and $L_{yy}(x, \sigma)$.

For descriptor, SURF is heavily based on previous work proposed by Lowe called Scale-Invariant Features (SIFT) [4]. It is considered as a descriptor with the best performance [5]. It begins with constructing a square region centered around the interest point. The region is then split up regularly into smaller 4x4 square sub-regions. We then compute a simple feature at 5x5 regularly spaced sample points and calculate the Haar wavelet response in horizontal direction (d_x) and in vertical direction (d_v). Based on these values, we define a four-dimensional descriptor vector for each sub-region is defined as follow:

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \quad (2)$$

Hence, each region with 4x4 sub-regions will have a descriptor of length 64.

III. IMPLEMENTATION

Our implementation consists of two process: development of the software and the test and observation. We decided to build our own system so we have a thorough understanding and control over the internal process of the system which is important to optimize the process.

In this test, we use a quad core i5 @ 2.5 GHz laptop with 8 Gb of DDR3 Memory. The program runs on Windows 10 Home. To optimize the environment, we stop any unnecessary programs and services that may affect the performance. All video files are rendered with MPEG-4 Part 10 Advanced Video Coding (commonly known as H.264).

There is no restriction on video resolution; videos are rendered between 480p and 1080p.

A. System Development

To test our application, we built a system based on the proposed algorithm. The application consists of two parts: detector and descriptor. Detector is responsible in segmenting videos, selecting key frames, and extracting the features from the key frames. Figure 1 is the interface for adding videos to the database.

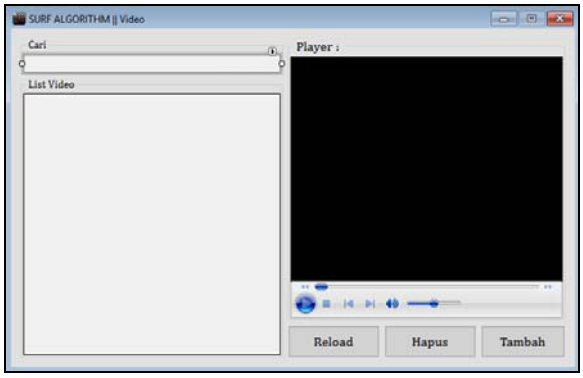

Fig. 2. User Interface of detector

The second system is descriptor system that performs matching based on image query. User pick an image from a file and by pressing the search button, the system will search related content on the video.

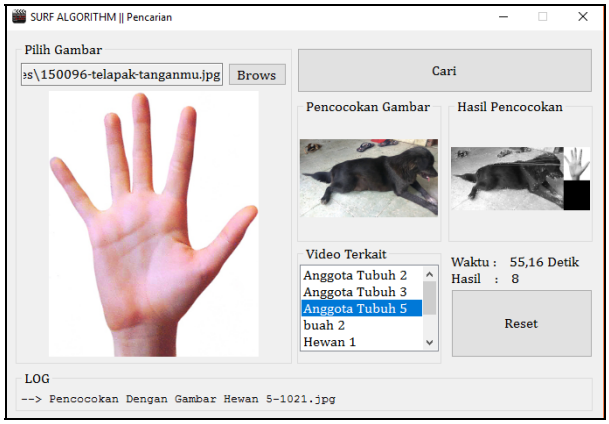

Fig. 3. Unser interface for Descriptor.

B. Tests and Results

For the testing purpose, we have collected 20 videos divided into 4 categories; body parts, eating utensils, fruits, and pets. We chose these categories since we have not found any research using these objects as their samples. We also build a set of images that will be used as query image. The images are divided into two group: in-frame samples and not-in-frame samples. In-frame samples are images taken from the video sample, which is similar to the key frame but not the key frame itself. On contrary, not-in-frame sample contain images similar to the in-frame samples but not taken from the video.

The first step of the system is to collects key frames from the sample videos. Table 1 below shows the detailed data of the video.

TABLE I. LIST OF VIDEOS TO BE EXTRACTED

Category	Name	Duration (in seconds)	Total *Frame*	Total Key frame	Description
Body Part	bp1	13	421	15	Hand
	bp2	39	657	42	Hand
	bp3	38	641	41	Hand
	bp4	30	931	32	Ear
	bp5	33	1021	35	Feet
Eating Utensil	eu1	35	1081	37	Glass
	eu2	40	1231	42	Spoon
	eu3	40	1231	42	Plate
	eu4	45	1381	47	Glass, Spoon, Plate
	eu5	37	1141	39	Glass
Fruit	fr1	14	241	16	Apple, Oranges, Lemon, Kiwi
	fr2	36	457	39	Apple
	fr3	18	305	20	Lemon
	fr4	21	617	23	Orange
	fr5	15	177	17	Kiwi
Pet Animal	pa1	36	1111	38	Cat
	pa2	49	1174	52	Cat
	pa3	33	1021	35	Cat
	pa4	18	571	20	Dog
	pa5	39	1201	41	Dog

As for the query, we split the set into two categories: in-frame query and not-in-frame query. Figure 4 shows examples of in-frame and not-in-frame query.

Fig. 4. Not-in-frame (top) and in-frame (bottom) image query

In the first test, we enter not-in-frame images and run the program. As shown in table 2 below, our system is capable to find 2 from 3 videos containing hand sequence which gives a 66% recall value. However, the system also picked 6

others videos that does not contain hand which gives a 25% precision value. Other categories also had the same result that the system is capable to have a acceptable recall value but failed to perform at precision value. However, in the last category (animal), the system was able to find one video related to the query (hence 100% precision) but failed to recognize the other two related videos (25% recall). On average, not-in-frame test gives an average value 62.25% and 37.5% for precision and recall value respectively. Run-time observation shows that query_3 (apple) gives the best performance amongst all query due to the system match apple in almost all videos (18). The average run-time performance is 73.57 seconds.

TABLE II. RESULTS FROM NOT-IN-FRAME IMAGE QUERY TEST

Name	Description	Videos Found	Related	Running Time (Seconds)
query_1	hand	8	2	55.16
query_2	glass	7	1	77.94
query_3	apple	18	2	34.14
query_4	cat	1	3	127.04

In the second test, we entered in-frame query and collected the precision and run-time performance data. The behavior is similar with apple images resulted the most match hence the lowest precision value (0.18%). It is interesting to see that query image fr1 that contains all fruits, as shown in figure 3, could be matched to all fruit based videos. It is interesting to notice that in-frame test shows a significant increase in run-time performance with average value 115.9 compared to 73.57 for not-in-frame test. However, precision and recall value for in-frame test are 51% and 59% respectively.

Table 3 shows a sample of data collected from in-frame test.

TABLE III. RESULTS FORM IN-FRAME IMAGE QUERY TEST

Name	Description	Videos Found	Related	Running Time (Seconds)
bp2	hand	2	2	103.84
bp4	feet	1	1	141.44
eu2	spoon	7	2	71.29
eu4	glass, spoon, plate	1	5	159.73
fr1	Apple, Oranges, Lemon, Kiwi	5	5	102.78
fr2	Apple	11	2	74.55
pa2	cat	1	3	124.64
pa4	dog	1	2	148.94

Fig. 5. Image query fr3 shows all fruits from other video

IV. CONCLUSION AND FUTURE WORKS

In this research, we have managed to build a content based video retrieval based on image using Speeded-Up Robust Features (SURF). We use 20 videos as the source videos. We test the system with in-frame and not-in-frame images to test the precision performance. The not-in-frame test gives a 25% average precision value and 66% average recall value while the in-frame tests gave a 59% precision value and 51% recall value. Moreover, running time performance data gives a 73.56 seconds for not-in-frame images and 121.67 seconds for in-frame images.

In our current test, we use random categories for both images and videos. While we are confident that the behavior will be similar regarding the data, it would be interesting to see the relation of performance based on the criteria of videos and images used in the test. Our next research will focus on understanding the performance behavior of SURF based on a specific category of videos. Understanding this behavior will give a slight idea how to optimize the algorithm based on a particular category.

ACKNOWLEDGMENT

This research is possible only by the support of the authors' institution, Universitas Sumatera Utara. The author would like to thank Prof. DR Runtung Sitepu, S.H., M.Hum as the rector of Universitas Sumatera Utara, Prof. Dr. Erman Munir, MSc as the head and staff of "Lembaga Penelitian USU (Research Center of Universitas Sumatera Utara)". The author would also like to thank the Dean of Faculty of Computer Science and Information Technology University of Sumatera Utara, Prof. Dr. Opim Salim Sitompul, M.Sc for the support during the research.

REFERENCES

[1] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," Computer Vision and Image Understanding, vol. 110 (3), p. 346-359, 2008.

[2] C. Harris and M. Stephens, "A combined corner and edge detector," Proceedings of the Alvey Vision Conference, p. 147-151, August 1988.

[3] T. Lindeberg, "Feature detection with automatic scale selection," International Journal of Computer Vision, vol. 30(2), p. 79-116, 1988.

[4] D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, p. 1150-1157, 1999.

[5] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 27(10), p. 1615-1630, 2005.

[6] S. Asha and M. Sreeraj, "Content Based Video Retrieval using SURF Descriptor," Proceedings of The Third International Conference on Advances in Computing and Communications, p. 212-215, August 2013

[7] S.L. Huang, C. Cai, F.F. Zhao, D.J. H, and Y. Zhang, "An efficient wood image retrieval using SURF descriptor," Proceedings of 2009 International Conference on Test and Measurement, p. 59-62, December 2009.

[8] D. Kim and R. Dahyot, "Face Components Detection Using SURF Descriptors and SVMs," Proceedings of International Machine Vision and Image Processing Conference, p. 51-56, September 2008.

[9] A. Amir, W. Hsu, G., Iyengar, C.Y. Lin, M. Naphade, A. Natsev, C. Neti, H.J. Nock, J.R. Smith, B.L. Tseng, Y. Wu, and D. Zhang, "IBM research TRECVID-2003 video retrieval system," in Proceeding of TREC Video Retrieval Evaluation, 2003.

[10] H. T. Shen, X. Zhou, Z. Huang, J. Shao, X. Zhou, "UQLIPS: A Real-time Near-duplicate Video Clip Detection System," in Proceeding of International Conference on Very Large Data Bases, p. 1374-1377, 2007