2012 10th International Conference on Frontiers of Information Technology

Content based Video Retrieval using Particle Swarm Optimization

Ayesha Salahuddin, Alina Naqvi, Kainat Mujtaba
Department of Computer Science
Kinnaird College for Women
Lahore, Pakistan
Email: ayeshasalah18@gmail.com

Junaid Akhtar
Department of Computer Science
School of Science and Engineering, LUMS
Lahore, Pakistan
Email: jakhtar@lums.edu.pk

***Abstract*—Traditional video search engines retrieve the results on the basis of correspondence between users textual query and tags associated with the videos. Only that content that matches the tags is returned as a result to the user. Given the ever-increasing immensity of videos on the internet, especially those with zero or irrelevant tags, such traditional methodology has eventually led to rise in ratio of missing important context. Content based searching within a video library is definitely an alternative solution but it requires time consuming computations and comparisons which renders exhaustive search unpractical. The purpose of this paper is to provide an efficient methodology that will lead to incremental improvement in the video search results against a users query image. Our method employs Particle Swarm Optimization (PSO), an evolutionary population based search algorithm, to look for frames within the video library. The fitness of each swarm particle is the degree of similarity with respect to the content present in both the input image provided by the user and the video frame(s) fetched through PSO. This exempts us from the exhaustive and linear search of every frame of every video in the library. The relative best match in each generation of PSO is shown to the user for his engagement. For calculating the fitness of each swarm particle we have tested three similarity measures, 1) correlation based template matching, 2) score from scale-invariant feature transform (SIFT) algorithm and, 3) convolution. Preliminary results on real video library are promising.**

***Index Terms*—Content based Video Retrieval, Particle Swarm Optimization, Template Matching, Scale Invariant Feature Transform,Convolution, Correlation coefficient**

I. INTRODUCTION

Video search engines, such as Youtube [1], blinkx [2], clipblast [3], and Yahoo! [4] have been massively successful in sharing videos to a large number of users over the internet. 65,000 videos per day (45 terabytes of data) were being added to Youtube in the year 2006, while it received 100 million views per day [5], [6]. In 2012, that number has gone up to four billion views, with a staggering amount of videos added each day [6], [7]. This shows the exponential growth in the amount of video files present on the internet in a matter of just a few years. It is expected to grow at an increasing rate in the future as well.

All classical search engines, including the ones just mentioned, have one common search mechanism: they require a textual keyword based query as an input from the user, this query is then matched with the tags associated with each video, if the tags match, the associated video is provided to the user as a result. This mechanism for searching is common for all content on the internet, be it a search for a specific web page or a multimedia. An alternative method employs retrieval of videos on the basis of *content* present in the input image provided by the user, which is matched with the set of videos kept in a library. This technique has the potential to avoid the shortcomings of tag-based searching and retrieving.

Some related work regarding content based matching use linear exhaustive approach for searching, which makes the search system inefficient and impracticle. For instance, in [8], a linear combination of color and texture based features is used as a feature vector. For similarity measure euclidean distance is computed between feature vector of query image and database set of images, but to retrieve images from a certain class of images, exhaustive search is employed. Newsam et al. require an input image as well as a defined region of interest from the user. This region of interest is then matched linearly using color histogram and texture features. Euclidean distance between corresponding feature sets is then used for comparing the similarity with the tiles of images kept in the database [9].

Barrios et al. uses color histogram and Gabor wavelet descriptors for matching, but use exhaustive search to build a database of similarities [10]. This is working in reasonable time for now since they are reportedly just working with 115,000 images at the moment. The system does not seem to have the characteristics of scalability. Others have also used databases for searching, for instance, Grauman uses an indexed database where a set of images is kept in different bins with a unique reference index based on similarity. When an image is given as an input query to this system, it goes to the reference index and then searches all images present in that bin linearly to find the best match with the query image [11].

Instead of a linear exhaustive search within the video/image database, the community has also worked on stochastic search models. To retrieve images from within video clips, Hauptmann et al. use probablistic similarity measures based on the assumption that the query image had an underlying stochastic model through which it was generated in the first place. This underlying model must be similar with the those images in the database that are similar to the query image [12]. We think that although this is a step away from the linear search, but their

978-0-7695-4927-9/12 $26.00 © 2012 IEEE
DOI 10.1109/FIT.2012.23

IEEE computer society

assumption might not hold under all conditions. We propose a stochastic population based evolutionary search mechanism, known as Particle Swarm Optimization (PSO), which does not involve such assumptions.

Interestingly, most of the related research that attempts at solving content based video/image searching involving PSO, have used it for optimizing similarity parameters, not for searching within the video library space. For instance, Hasan et al. propose a face recognition system that first transforms the images into fourier space using Randon transform. Linear discriminant analysis is then used for computing similarity between the input image and the database images. PSO is employed to calculate the optimal direction in which the training set of images are to be transformed by the Randon transform. Every particle carries a different *theta* to begin with, and after a few generations the best angle for the transform is found [13].

In [14] and [15], same approach is used towards PSO. It takes image as an input query from the user and matches it with all the images predefined in the database by finding local features using either; 1) affine transformation invariant modeling or graph based method to map regions [14], or 2) using color, texture or shape feature extraction methods [15]. The feature vectors detected are then optimized by representing them as a particle in PSOs swarm.

Our survey gives us a warrant to believe that our work is different from others in a way that; instead of searching exhaustively for related video frames, or using simplistic assumptions about the underlying processes, or using PSO to optimize the matching parameters, we use PSO to generate the (video number, frame number) pairs. Using this stochastic search mechanism, we are able to retrieve results efficiently because it avoids sequential search and only those frames are matched with the input image that PSO generates, instead of time-consuming exhaustive search.

The details of this proposed approach are discussed in section 2. PSO requires calculation of fitness of each particle, which in our context is given by the similarity measure. Section 3 discusses the similarity measures that are paired with PSO. In section 4 we discuss an experimental setup that tests our hypothesis. The results and their analysis are discussed in section 5. Finally conclusions and future work are discussed in section 6.

II. Particle Swarm Optimization

Particle Swarm Optimization is a population based optimization technique proposed by Kennedy and Eberhart in 1995 [16]. The algorithm took its inspiration from the social behavior of the living organisms like a flock of birds or swarm of bees, where each individual is in search for food separately as well as in coordination with the swarm's leader. Each individual's best finding and swarm's global best finding help in setting the velocity of each individual, and that in turn helps decide the next points of exploration for *food*.

The *food* in our context is the frame number of the video that best matches with the input image from the user. To accomplish this, PSO initializes a swarm with uniform random distribution, where each particle of the swarm is a 2 dimensional representation- a video number and a frame number. This is also known as particle's position. Once the swarm's positions are initialized, fitness scores are calculated for each particle present in the swarm. Fitness in the current context means how similar is each particle's video's frame with the input query image. For calculating of fitness three similarity measure have been tried in this paper (discussed in section 3):

- Convolution,
- Correlation coefficient and,
- Score from SIFT algorithm

Once we have the fitness scores against each particle, this information is used for potentially updating each particle's best position and swarm's global best particle's position; where the *global best particle* represents the best content based match found so far in the current population and the *particle best* represents the current particles best fitness score achieved so far. Hence, taking each particle's fitness score the following condition is checked:

```
if $current_{\text{particle}} > particle_{\text{best}}$ then
    $particle_{\text{best}} \leftarrow current_{\text{particle}}$
end if
```

Once we have this information against all the particles, the initial generation has done it's job by giving us the starting batch of search points within the video library with which to compare the input image with. The last job for this generation is to produce the next searching points using each particle's current position. In order to move them to a new position for the next generation, a velocity for each swarm particle is computed. Velocity represents how much the particle can move in its search space therefore, it is calculated according to three factors; 1) particles previous velocity (initially zero), 2) particles previous fitness score that represents the local influence over the swarm and, 3) global best position which represents the social influence of the swarm over each particle. Therefore, a velocity of each particle is computed through eq. (1a). We have used parameters from a variant PSO that incorporates inertia or weight while updating the velocity. Eq. (1b) shows how it is done, where $weight_{min}$ is set as 0.4 and $weight_{max}$ is set as 0.9 which are the starting and ending inertia weights [17].

$$\begin{aligned} Velocity[] = (&weight * velocity[] + \\ &(f1 * (p_{\text{best}}[] - current_{\text{particle}}[])) + \\ &(f2 * (g_{\text{best}}[] - current_{\text{particle}}[]))) \end{aligned} \tag{1a}$$

Where,

$$f1 = cc1 * random_{\text{number}}(1,1);$$

$$f2 = cc2 * random_{\text{number}}(1,1);$$

cc1 and cc2 are the cognitive and social parameters respectivey, and are usually set to 2.0

$$Weight = ((weight_{\max} - weight_{\min})* \\ ((i_{\max} - i_{\text{current}})/i_{\max})/ \\ weight_{\min}) \tag{1b}$$

Where i represents the swarm size of the swarm.

Our optimization problem is in 2D space that is each particle is referred by a positive video number and their respective frame number, thus two velocities are computed in order to make the particle move about. Once this is done, velocity is added to particles current position using eq. (2).

$$current_{\text{particle}}[] = (current_{\text{particle}}[] + velocity[]) \tag{2}$$

There is a chance that this new position computed exceeds the search space (either in terms of the total number of videos or frames available). If any particle's position does then a uniformly random value is assigned to that specific particle ranging from [lower$_{\text{boundary}}$, upper$_{\text{boundary}}$] of the search space. This helps introduce variation within swarm and also preempts the situation where most of the swarm's particle become useless for any further exploration, by being out of search bounds. Once this is checked against all the particles present in the swarm, eventually these particles are now the next positions (video number, frame number) to be matched with the input image.

III. Similarity measures paired with PSO

SIFT, correlation and convolution are the similarity measures used in our method as an objective function in PSO that calculates fitness of every particle. Each of these measures are structurally distinctive but similar behaviorally.

A. Convolution

Convolution is used to find a common area that overlaps between pair of images, it is done by moving a mask from pixel to pixel in an image computing a predefined quantity, which is the fitness score of that particular pixel [18]. Therefore, we take a *template* image which consists an object or a region, and on the other hand a *frame*, which is fetched from a video through PSO, to find out whether the frame consists of the object or region we are trying to search from the template image. Using eq. (e) [18], template image is flipped about the region, shifting it with respect to the frame image by changing the values of (x,y) and then computing the sum of the products over all values of m and n for each displacement (x,y), this is done till both images stop overlapping each other. Convolution value generated ranges from [-1,1].

$$f(x,y) * t(x,y) = 1/MN \sum_{M-1}^{m=0} \sum_{N-1}^{n=0} f(m,n)* \\ t(x-m, y-n)) \tag{3}$$

where f is the frame image, and t is the template image. MxN is the size of both images and (X, Y) are incrementing positive factors.

A pixel with the maximum value of convolution in the frame image, contributes towards the fitness of the video frame, represented as a particle in PSO's swarm.

B. Correlation

Correlation is used to find a match between two images. In our approach these images are again, a frame that is fetched from a video through PSO and the other is the template image selected by the user, that also contains the object or region which the user aims to search for. Hence, to compute a matching value eq. (4) [18] is used whose value ranges from [-1, 1]. A value near 1, represents a good matching result, and as the value approaches 0, its matching extent between them decreases.

$$f(x,y) \circ t(x,y) = 1/MN \sum_{M-1}^{m=0} \sum_{N-1}^{n=0} f(m,n)* \\ t(x+m, y+n)) \tag{4}$$

where f is the frame image, and t is the template image. MxN is the size of both images and (X, Y) are incrementing positive factors.

Correlation and convolution are similar to each other, but there are two major differences which make correlation different; 1) the template image is not mirrored about the region and, 2) it is more naturally understood as a similarity measure for time domain signals.

C. Scale Invariant Feature Transform (SIFT)

Scale Invariant Feature Transform (SIFT), is a methodology that computes local features from an image. These features are invariant to changes in scale, 2D translation and rotation transformations [19]. This method provides a fitness score to PSO in a way that it first takes out frame *descriptors* of two images; 1) input image provided by user and, 2) a frame picked up from the PSO's particle. Using those descriptors it then calculates the euclidean distance using eq. (5) [18], which returns the result in terms of common matches with their scores found in both the images.

$$D_e(p,q) = (((x-s)^2 + (y-t)^2)^{1/2}\Big) \tag{5}$$

where p,q are the descriptors of both the images, (x,y) and (s,t) are coordinates of these points respectively.

The degree of similarity between the two images in SIFT is that; if the matching points of both the images is the exactly the same, then the euclidean distance will sum up to 0, which is considered to be the fittest score. But if there are absolutely no matching points between two images then no score is computed. If however, there are varying degree of matching points, the score computed is a positive number linearly proportional to the degree of mathing descriptors of the two images.

To conclude, these similarity measures discussed above provide PSO with a fitness value for each particle present in the swarm. This helps PSO to decide the global best and particle best which consequently contributes in producing the next generation of searching positions for the particles of the swarm.

IV. Experimental Setup

A video library consisting of 20 film trailers taken arbitrarily from the internet was used to test the performance of PSO. The videos range from 15 to 50 MB in size, having on average 3500 frames. For the experiment, specific frames from 5 different videos were taken as the input template image. To be specific, frames from the 1st, 5th, 10th and 20th video in the library were used for different experiments.

To evaluate the performance of our method, we monitored *first hitting time*, to track the number of generations it takes PSO after which the fitness score of any particle in the swarm achieved an acceptable fitness threshold (0.85 in the case of convolution and correlation, 0 for SIFT). The experiments with frames taken as template images from different videos were only meant to further test the consistency of PSO's efficiency to retrieve the results. An exhaustive search algorithm's retrieval time would have surely increased linearly, the further the actual location of template image moves into the video library. Other than the parameters already discussed in the above sections, we tested our approach by keeping a swarm size of just 100 and maximum generations up to 200, the results generated are shown in Figure 1(a). As we can see , in Figure 1(a) swarm particles were dispersed but as generations gradually grew the results started to get visible (Figure 1(b),1(c)). Figure 1(d) shows the final result of the test we performed, where the swarm points are now clustered verticlally around that video trailer number and respective frame number(s) that turned out to be the best match with the input image.

V. Results and Discussion

Since random and stochastic elements are involved in the initialization and working of the PSO, we conducted 25 runs against each setting of the experiment, and calculated standard statistics listed in table I, II and, III. The first row of the table defines the position number of the video in the library from which searching template was used. Against each of the input image and similarity measure, following statistics were monitored for PSO's *first hitting time* in the 25 experimental runs:

1) Best case scenario; minimum number of generations it took the PSO to find the best match,
2) Worst case scenario; maximum number of generations it took to find the best match,
3) Average case; stastitical mean of the number of generations it took the PSO to find the best match,
4) Standard deviation; the dispersion from the mean number of generations.

To give a fair perspective to the results let us state the fact that the 2-D search space for PSO consists of around 70,000 unique search locations (20 videos and 3500 frames on average per video). PSO in one generation has the capacity to test the similarity of input image with only 100 of these 70,000 potential points, as it maintains only 100 particles in the swarm.

Using the results obtained, it can be deduced that using our method, the PSO consistently found the best matching frames efficiently from a set of videos regardless of the fact that we took the input frames from different videos from within the library. Among the 25 experimental runs it can be said that at least once the best case scenario using any of the similarity measure, threshold set for frame was achieved right after one evolutionary updation in the the randomly initialized generation. This means that PSO (at best) found out the first match after looking at just 200 search points. This stastistic is helped by the fact of temporal redundancy in video data files. However, in stochastic solutions, usually the most significant statistic is the *average* behavior. The average first hitting time is roughly around 7 generations in the case of correlation and convolution, considering all 4 scenarios. This means that it took our PSO around just 700 evaluations out of the possible 70,000 to hit the first acceptable match, on average. This makes just 1% of the total search space, and hence is indicative of a boastful efficiency for a content-based search solution, as opposed to a linear exhaustive search technique that is in use at large. Also observable is the fact that SIFT works more efficiently on average as compared to other two methods. The reasons for this behavior needs proper investigation in the future.

TABLE I
Results obtained using Convolution

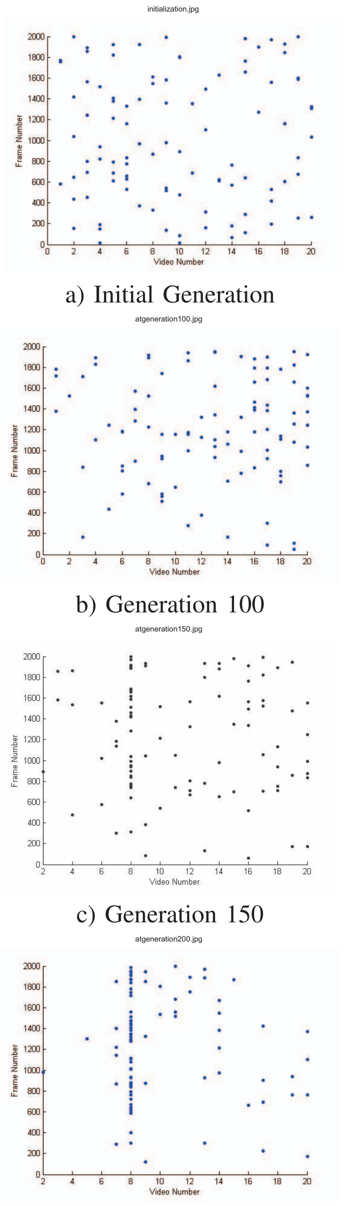
	Position number of video from library			
Cases	1st	5th	10th	20th
Best case	2	2	2	2
Worst case	22	25	21	22
Average	8.72	8.80	7.08	7.16
Standard Deviation	7.87	7.82	6.80	4.87

TABLE II
Results obtained using Correlation

	Position number of video from library			
Cases	1st	5th	10th	20th
Best case	2	2	2	2
Worst case	20	24	26	22
Average	4.76	8.44	9.04	7.76
Standard Deviation	5.00	6.15	6.91	5.91

TABLE III
Results obtained using SIFT

	Position number of video from library			
Cases	1st	5th	10th	20th
Best case	2	2	2	2
Worst case	6	9	6	5
Average	2.56	3.4	2.96	3.04
Standard Deviation	1	2.23	1.13	1.09



a) Initial Generation

b) Generation 100

c) Generation 150

d) Generation 200

Fig. 1. Convergence of PSO: From initialization to after 200 generations

VI. Conclusion

Knowing that with a traditional video search engine, results are retrieved on the basis of correspondence between users textual query and tags associated with the videos; we proposed a method that retireves the results on the basis of matching content present in the image query with set of videos that is kept in predefined library. We have tried three similarity measure in this regard: convolution, correlation, and SIFT. For the experiments, the input template image is taken from within the videos. An interesting future possibility is to test our method with an input image not taken from within the video library. Our original contribution however, is in the use of PSO to exploit the search within the video library, as shown in Figure 1. The current work has used a small sized video library. The idea needs to be tested on large scaled library in the future. Nevertheless, the initial results indicate that PSO makes the spatial location of the content within the order of video library irrelevant, as on average it hits the first match very efficiently. We believe that with this spatial independence, a big problem within the content based video/image searching gets addressed.

References

[1] www.youtube.com

[2] www.blinkx.com

[3] www.clipblast.com

[4] http://video.search.yahoo.com

[5] www.focus.com/fyi/10-largest-databases-in-the-world

[6] http://en.wikipedia.org/wiki/Youtube

[7] http://mashable.com/2012/01/23/youtube-4-billion

[8] Yanyan Gao, Honggang Zhang and Jun Guo. *Multiple Features-Based Image Retrieval*.Pattern Recognition and Intelligent System Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China. Proceedings of IEEE IC-BNMT2011.

[9] Shawn Newsam, Daniel Leung, Oscar Caballero, Justin Floreza, and Jesus Pulido. *CBGIR: Content-Based Geographic Image Retrieval* . Electrical Engineering and Computer Science, University of California at Merced. ACM GIS 10 , November 2-5, 2010. San Jose, CA, USA.

[10] Juan Manuel Barrios, Diego Diaz-Espinoza, and Benjamin Bustos. *Text-Based and Content-Based Image Retrieval on Flick*. University of Chile, Department of Computer Science, Santiago, Chile. 2009 Second International Workshop on Similarity Search and Applications.

[11] Kristen Grauman, *Efficiently Searching for Similar Images*. Department of Computer Sciences, University of Texas at Austin,TX, USA, 2009.

[12] Alexander G. Hauptmann, Rong Jin, and Tobun D. Ng, School of Computer Science, Carnegie Mellon University Pittsburgh, PA. *Video Retrieval using Speech and Image Information*, Electronic Imaging Conference (EI'03), Storage Retrieval for Multimedia Databases, Santa Clara, January 20-24, 2003.

[13] Mr. Hamid M. Hasan,Prof. Dr. Waleed A. AL.Jouhar and Dr. Majid A. Alwan. *Face Recognition Using Improved FFT Based Radon by PSO and PCA Techniques*. International Journal of Image Processing (IJIP),volume 6, issue 1, year 2012

[14] Nozomi Oka and Keisuke Kameyama. *Relevance Tuning in Content-Based Retrieval of Structurally-Modeled Images using Particle Swarm Optimization*,2009.

[15] Bae-Muu Chang,Hung-Hsu Tsai2 and Wen-Lin Chou2.*Content-Based Image Retrieval Based on Image Features and Particle Swarm Optimization*. Business and Information 2012

[16] J. Kennedy and R. Eberhart. Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2001.

[17] Prabha Umapathy, C. Venkataseshaiah,and M. Senthil Arumugam. *Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Power Flow Solution*. Hindawi Publishing Corporation Discrete Dynamics in Nature and Society, Volume 2010.

[18] Rafael C. Gonzalez and Richard E.Woods, *Digital Image Processing*

[19] Lowe, D.G.,Dept. of Comput. Sci., British Columbia Univ., Vancouver, BC ,*Object recognition from local scale-invariant features* ,Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference, Volume 2.

[20] Rafael C. Gonzalez and Richard E.Woods, *Digital Image Processing*,

[21] Bae-Muu Chang,Hung-Hsu Tsai2 and Wen-Lin Chou2.*Content-Based Image Retrieval Based on Image Features and Particle Swarm Optimization*. Business and Information 2012

[22] http://www.flickr.com

[23] Jame Blondin, *Particle Swarm Optimizaton*, 2009