computing@computingonline.net
www.computingonline.net

Print ISSN 1727-6209
On-line ISSN 2312-5381
International Journal of Computing

# VIDEO COPY DETECTION UTILIZING THE LOG-POLAR TRANSFORMATION

**Daniel Reynolds[1)], Richard A. Messner[2)]**

[1)] Portsmouth Naval Shipyard, Kittery, ME, 03904, dan_r123@yahoo.com
[2)] University of New Hampshire, ECE Department, Durham, NH, 03824, rich.messner@unh.edu, www.svpal.unh.edu

**Abstract:** Video copy detection is the process of comparing and analyzing videos to extract a measure of their similarity in order to determine if they are copies, modified versions, or completely different videos. With video frame sizes increasing rapidly, it is important to allow for a data reduction process to take place in order to achieve fast video comparisons. Further, detecting video streaming and storage of legal and illegal video data necessitates the fast and efficient implementation of video copy detection algorithms. In this paper some commonly used algorithms for video copy detection are implemented with the Log-Polar transformation being used as a pre-processing step to reduce the frame size prior to signature calculation. Two global based algorithms were chosen to validate the use of Log-Polar as an acceptable data reduction stage. The results of this research demonstrate that the addition of this pre-processing step significantly reduces the computation time of the overall video copy detection process while not significantly affecting the detection accuracy of the algorithm used for the detection process. 

**Keywords:** Video Copy Detection, Log-Polar Transform.

## 1. INTRODUCTION

In order to create robust video copy detection algorithms, content based video copy detection methods can be applied. The video content is utilized to create unique signatures that define that video. Once signatures are constructed for multiple videos each can be mathematically compared to determine if a candidate video is a copy of another. Once identified as copies the candidate video can be labeled as such. This simple concept of detecting copies becomes more complicated and time consuming when videos become altered in a copy or transformation process. Thus video copy detection algorithms must be able to withstand both spatial and temporal changes. Video alterations can include such modifications as: addition of logos, removal of content, change in brightness/contrast, frame freezing, mixing of video content, etc. To date most video copy detection algorithms rely on making signatures based on performing mathematical operations on entire frames.

Video copy detection becomes more problematic with larger frame sizes. This is particularly apparent with the introduction of High Definition video and soon Ultra High Definition video where frame size is approaching 8K. Increases in frame size can complicate the signature forming process as well as cause a significant time delay in the construction of signatures which make the detection of copied video streaming media difficult.

With such a large increase in frame size it is apparent that the current algorithms will become untenable unless consideration is given to algorithms which perform data reduction on the video itself prior to constructing of any signature which describes the video clip.

A simple solution to increase the speed of the overall process is to utilize a pre-process step to perform data reduction on a frame by frame basis before extracting signatures. The question that arises is what type of pre-process step should be chosen and how well does the video copy detection algorithm perform on the reduced data set. To be successful, the choice in the data reduction algorithm must be able to be completed quickly and not change the video in such a manner that would result in a reduction in the overall copy detection of the subject video. In this paper we consider the Log-Polar transform to accomplish the data reduction step.

The Log-Polar mapping process transforms a frame from Cartesian coordinates into Log-Polar coordinates where a natural data reduction takes place [1]. The Log-Polar process includes an adjustable compression variable and produces an

output which mimics the non-uniform sampling of a human vision system [2]. Based on the benefits that Log-Polar offers, this paper explores the benefits and drawbacks associated with utilizing Log-Polar as a pre-processing step.

## 2. BACKGROUND

Research into video copy detection began in the early 2000s and has since become an area of active investigation. Many different techniques have been created to solve the problem of detecting copied videos. Almost all of the techniques focus on methods to summarize the video into short, compact, and robust signatures, or methods to find and compare signatures in a database. Little research has been devoted to determining what pre-processing steps can be utilized to aid the signature creation.

In 2007 a paper titled "Video Copy Detection: a Comparative Study"[3], by Law-To et al., was created to summarize the research completed up to 2007. The paper describes multiple methods to calculate signatures and includes some discussions on signature comparisons and the results they achieved. There are two types of signature categories that are discussed; Global and Local. Global methods rely on calculating signatures based on global features such as contrast, brightness or other full frame comparisons. Local methods rely on finding specific points in a frame and computing a signature based on those points.

There are three different methods discussed for the Global category. The first is a Temporal method which relies on time based information alone to define the signature. The second is an Ordinal method which relies on calculating a signature based solely on a frame by frame basis. The third is a Temporal Ordinal method which utilizes both time and individual frame information to calculate the signature [3].

There are also three methods discussed for the Local category. The first is called AJ (for Alexis Joly) and relies on choosing key frames and calculating a signature based on specific points in each frame. The second is called Video Copy Tracking (ViCopT) which computes specific points for every frame and then tracks their trajectories. The third is called Space Time Interest Points (STIP) which computes a signature based on points in the videos that have a significant variation in both space and time [3].

Since 2007 there have been many different techniques discussed. Some of these techniques are ones that were previously known but have become common in new algorithms.

The term Key Frames has become very popular. Key Frames are ones that are taken out of a video and provide an overall description of the video. Key Frames often are chosen based on grouping similar frames and then choosing a representative frame from that group [4,5,6,7].

Shot Boundary Detection has been known and is now becoming more popular [8,9]. Shot Boundaries are calculated in different ways, but all describe the boundary between different scenes of a video. One method described both by Law-To et al. and Ping-Hao et al. is by subtracting two frames from time $t$ and $t$+1 [3,10]. When the frame difference exceeds a specified threshold the frame is considered at a scene change and thus labeled an anchor frame. Ping Hao et al. use the distance between anchors as the signature for their work [10].

Utilizing audio for video copy detection has now become a more popular method. Saracoglu et al. utilize two signatures, one for the audio signal and one for the video. The two signatures are compared separately and the final results compared [11].

Discrete Cosine Transformation (DCT) coefficients are used in multiple algorithms. Zhihua et al. use DCT coefficients because many videos and images are compressed using DCT coefficients. The paper determines Key Frames for the video and then accesses DCT coefficients directly; bypassing the need to decompress the data. This paper selects low to middle frequency DCT coefficients as the final signature [6].

Other methods that have become popular are SIFT and SURF [12,13]. SIFT, or Scale-Invariant Feature Transform, is a local method that utilizes scale-invariant key points and was originally designed for image matching [14]. Lowe describes identifying interest points that are invariant to scale and orientation and determined by calculating the gradients around the points [14]. SURF, or Speeded-Up Robust Features, is also a scale and rotation-invariant descriptor similar to SIFT. SURF relies on wavelet responses around the interest points rather than the gradient. SURF also relies on specific frames and minimizes the output signature [15].

Some recent algorithms have focused on combining different approaches. Corvaglia et al. use the term multi-feature to describe an approach that uses dominant colors, color layout and an ordinal measure that is based on average luminance [16]. Yonghong et al. use the term multimodal to describe an approach that uses SIFT, SURF, DCT and audio [17].

Finally there is a small segment that utilizes some pre-processing steps. Huamin et al. use a pre-processing step to smooth frames with Gaussian filters, then choose three representative frames from each shot boundary and then resize the frames to 128x128 pixels [18]. Esmaeili et al. use a pre-processing step that smooths the video frames by

applying a Gaussian filter spatially and temporally and then downsamples the video frames in both dimensions. The resulting videos are downsampled to 4 frames/s and 144x176 pixels [19]. Dutta et al. pre-process videos by converting to greyscale and applying a mean filter and histogram equalization [20].

The papers above that apply pre-processing steps are oriented toward improving the robustness of the algorithms vice improving speed. Huamin et al. and Esmaeili et al. do allude to the fact that the computational cost of the algorithms will be reduced, but do not provide quantitative analysis to the benefits [18,19]. This paper attempts to provide such a quantitative analysis and offer a new pre-processing step that can be implemented in most, if not all, of the current signature algorithms.

## 3. LOG-POLAR

Log-Polar is a transformation mapping that transforms Cartesian coordinates into Log-Polar coordinates. Log-Polar performs a data reduction during the mapping process. The transform mimics the mapping that the eye performs by having radial and data reduction properties that are centered around a focal point. Data reduction is performed by first choosing a focal point and then creating concentric circles whose distance from the focal point increases logarithmically. Each of these circles is divided at certain radius degrees and each segment created by these intersections defines what will become a single point in the final output [1,2]. Fig. 1 shows this process.

This paper utilizes Log-Polar for its data reduction capability and because videos are inherently a visual process for humans. The expectation is that the data reduction can be accomplished without affecting the algorithm as it is anticipated that the important information to describe a video is located in the center. Log-Polar accomplishes this by making the center of the original image more important in the destination image (larger in size) and content on the periphery of the original image to be less important (smaller in size).

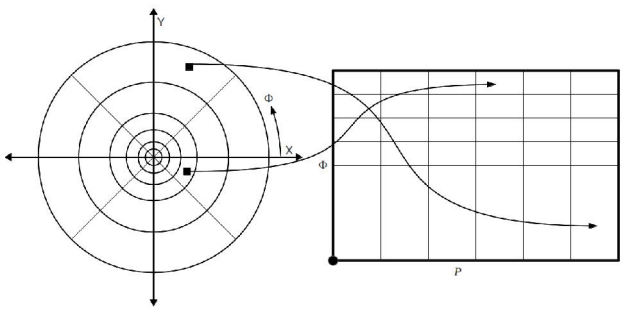



**Fig. 1 – Log-Polar.**

The Log-Polar transformation is accomplished by remapping the original image. To make the remapping process focus more on the center of the image and less on the periphery, the log function is utilized. The log function is applied to the length of the vector ($\sqrt{\boldsymbol{X^2 + Y^2}}$). To allow the magnitude ($\boldsymbol{P}$) of the destination image to change, a magnitude factor ($\boldsymbol{M}$) is utilized. The resulting magnitude is seen in (1).

$$\boldsymbol{P = M * \log\left(\sqrt{X^2 + Y^2}\right)} \tag{1}$$

To calculate the phase of the destination image the inverse tangent is utilized in the same manner as converting Cartesian coordinates to Polar coordinates. Equation (2) shows the phase calculation.

$$\boldsymbol{\emptyset = \arctan\left(\frac{Y}{X}\right)} \tag{2}$$

## 4. SIGNATURE METHODS

### 4.1. GLOBAL TEMPORAL METHOD

This method is based on calculating a signature from the temporal information of frames. The method utilized to calculate the signature is described by Law-To, et al. [3] and the signature comparison method was chosen separately for the purposes of this research.

This method calculates a vector that is based on the difference between subsequent frames and choosing the largest differences, as seen in (3). This vector exhibits large values where frames differ the most, which occurs at scene transitions and fast motion. The vector is then Fourier transformed and the phase information is saved as the final signature.

Once the signature is created for two videos, the signatures are compared by utilizing the Cosine Formula [21]. Depending on the similarity value, a decision is made if the videos are considered copies.

$$\boldsymbol{a(t) = \sum_{i=1}^{N} K(i)\big(I(i,t) - I(i,t-1)\big)^2} \tag{3}$$

where, $\boldsymbol{a(t)}$ = Temporal Activity; $\boldsymbol{K(i)}$ = Weight Function (emphasis on central pixels); $\boldsymbol{I(i,t)}$ = Intensity of each pixel (current time); $\boldsymbol{I(i,t-1)}$ = Intensity of each pixel (past time); $\boldsymbol{N}$ = Number of pixels in frame.

The Cosine Formula is given by (4), (5) and (6):

$$\boldsymbol{cosine(\emptyset) = \frac{v * w}{\|v\|\|w\|}} \tag{4}$$

$$\boldsymbol{v} \cdot \boldsymbol{w} = \sum_{i=1}^{N} \boldsymbol{v_i} * \boldsymbol{w_i} \quad (5)$$

$$\|\boldsymbol{v}\|\|\boldsymbol{w}\| = \sqrt{\sum_{i=1}^{N} \boldsymbol{v_i}^2} * \sqrt{\sum_{i=1}^{N} \boldsymbol{w_i}^2} \quad (6)$$

## 4.2. GLOBAL TEMPORAL ORDINAL METHOD

This method is based upon calculating a signature from the temporal and ordinal information of frames and is summarized by Law-To, et al. in "Video Copy Detection: A Comparative Study" [3] and originally created by Chen et al. in "Video Sequence Matching Based on Temporal Ordinal Measurement" [22].

This method splits each video frame into multiple segments and outputs a vector for each segment for the entire video that contains values that range from 1 to the frame count.

Once the signature is created for two videos, the signatures are compared by utilizing a custom comparison method defined by Chen et al. [22] which performs a correlation type comparison; as seen in (7), (8) and (9).

$$\boldsymbol{D}\left(\boldsymbol{V_q}, \boldsymbol{V_r^p}\right) = \frac{\boldsymbol{1}}{\boldsymbol{K}} \sum_{k=1}^{K} \boldsymbol{d^p}\left(\boldsymbol{\lambda_q^k}, \boldsymbol{\lambda_r^k}\right) \quad (7)$$

$$\boldsymbol{d^p}\left(\boldsymbol{\lambda_q^k}, \boldsymbol{\lambda_r^k}\right) = \frac{\boldsymbol{1}}{\boldsymbol{C_M}} \sum_{i=1}^{M} \left|\boldsymbol{\lambda_q^k}(\boldsymbol{i}) - \boldsymbol{\lambda_r^k}(\boldsymbol{p} + \boldsymbol{i} - \boldsymbol{1})\right| \quad (8)$$

$$\boldsymbol{C_M} = \sum_{i=1}^{M} |\boldsymbol{M} + \boldsymbol{1} - \boldsymbol{2} * \boldsymbol{i}| \quad (9)$$

where, $\boldsymbol{D}$ = Overall distance for all segments; $\boldsymbol{V_q}$ = Query video; $\boldsymbol{V_r}$ = Reference video; $\boldsymbol{p}$ = Comparison point between the two vectors; $\boldsymbol{K}$ = Number of segments; $\boldsymbol{d_p}$ = Distance between the two vectors at time $\boldsymbol{p}$; $\boldsymbol{C_M}$ = Normalizing factor; $\boldsymbol{\lambda^k}$ = Signature of segment $\boldsymbol{k}$; $\boldsymbol{M}$ = Length of query video.

## 5. SUMMARY OF RESULTS

The following results show the comparisons that were made for the Temporal and Temporal Ordinal algorithms described by Law-To et al. with modifications to the comparison algorithms. The results focus on the difference between the original algorithm and the algorithm with Log-Polar applied. It was not necessary to optimize the original algorithm since the difference with and without Log-Polar is the necessary information. Without optimization and with a different comparison method, the results do not correlate directly to the results achieved by Law-To et al.

All results were achieved with a C++ implementation utilizing OpenCV, Eclipse, GTKMM, FFTW and GNUPlot in a Linux operating system.

There were four total comparisons performed. Three compare actual videos with pre-defined Log-Polar parameters. The fourth varies the Log-Polar magnitude parameter to determine the affects that compression has on the videos. The different comparisons encompass:

1) Log-Polar pre-process effect with different transformations;
2) Log-Polar pre-process effect when comparing all different videos;
3) Log-Polar pre-process effect when comparing actual copied videos;
4) Log-Polar pre-process effect when changing the magnitude parameter (79% to 97% compression).

Tables I through IV describe the increase (↑) or decrease (↓) of either time or recall when comparing the algorithm with Log-Polar and without Log-Polar. For example, Table I - Temporal; Time decreased by 27% when Log-Polar was added to the algorithm process. For these tables, recall is the percentage change when comparing the number of videos correctly detected with Log-Polar versus without Log-Polar.

Overall, the results show that the time to compute the algorithm with the added step of computing the Log-Polar, results in a reduction in overall computation time. The recall is noted to be reduced most with the Temporal algorithm and least with the Temporal Ordinal algorithm. The Log-Polar magnitude results show that an increase in compression leads to an improved reduction in computation time but not linearly. Certain compression amounts result in significantly improved times. Overall, recall varies non-linearly as well but does not exhibit degrading values even with 97% compression of the original video.

**Table 1. Comparison 1 – Transformations**

| | Time | Recall |
|---|---|---|
| Temporal | 27% ↓ | 3% ↓ |
| Temporal Ordinal | 54% ↓ | 1% ↑ |

**Table 2. Comparison 2 – Different Videos**

| | Time | Recall |
|---|---|---|
| Temporal | 22% ↓ | 8% ↓ |
| Temporal Ordinal | 41% ↓ | 1% ↑ |

**Table 3. Comparison 3 – Copied Videos**

| | Time | Recall |
|---|---|---|
| Temporal | 15% ↓ | 17% ↓ |
| Temporal Ordinal | 38% ↓ | 17% ↑ |

**Table 4. Comparison 4 – Log-Polar Magnitude**

| | | Time | Recall |
|---|---|---|---|
| Temporal | 97% | 25% ↓ | 1% ↑ |
| | 95% | 15% ↓ | 1% ↑ |
| | 90% | 22% ↓ | 8% ↓ |
| | 84% | 6% ↓ | 1% ↑ |
| | 79% | 1% ↓ | 8% ↓ |
| Temporal Ordinal | 97% | 41% ↓ | 0% |
| | 95% | 33% ↓ | 0% |
| | 90% | 41% ↓ | 1% ↑ |
| | 84% | 21% ↓ | 0% |
| | 79% | 29% ↓ | 1% ↑ |

## 6. CONCLUSION

In order to perform any copy detection of video it is clear that fast and efficient algorithms must be developed if real time performance is to be achieved. Utilizing a pre-processing step to compress video frames is a necessary step toward the development of such efficient algorithms and is an active area of current research. This is especially evident as refresh rates and frame sizes continue to increase at the pace they are. It is demonstrated here that the Log-Polar transformation provides a fast and computationally efficient compression method that can keep pace and perform well by mimicking the human visual system sampling process. Pre-processing via Log-Polar provides a method that can withstand transformations that can significantly reduce frame size with little effect on the overall decision process being subsequently performed. By performing this technique in the afferent portion of the process to reduce the video data to a more manageable format it is shown that back-end processes used in this paper are not significantly altered in their performance. The examples shown in this paper suggest that further research work in this area may prove fruitful in attaining real time video comparison on streaming media. In addition, more complex methods might be applied for the copy detection process due to the significantly reduced data set created by the Log-Polar compression.

## 7. REFERENCES

[1] M. A. Abbott and R. A. Messner, "Use of coordinate mapping as amethod for image data reduction," in *Proceedings of the ConferenceBoston-DL tentative*, (1991), pp. 272-282.

[2] R. A. Messner and H. H. Szu, An image processing architecture for real time generation of scaleand rotation invariant patterns, *Computer Vision, Graphics, and Image Processing*, (31) 1 (1985), pp. 50-66.

[3] J. Law-to, O. Buisson, L. Chen, M. H. Ipswich, V.Gouet-brunet, A. Joly,N. Boujemaa, I. Laptev, F. Stentiford, and M. H. Ipswich, Video copydetection: a comparative study, in *Proceedings of the ACM International Conference on Image and Video Retrieval CIVR'07*, (2007), pp. 371-378.

[4] N. Guil, J. M. González-Linares, J. R. Cózar, and E. L. Zapata, Aclustering technique for video copy detection, in *Proceedings of the 3rdIberian conference on Pattern Recognition and Image Analysis, Part I,ser. IbPRIA'07*, (2007), pp. 451-458.

[5] H.-S. Kim, J. Lee, H. Liu, and D. Lee, Video linkage: group basedcopied video detection, in *Proceedings of the International Conference on Content-based Image and Video Retrieval, ser. CIVR'08*, (2008), pp. 397-406.

[6] T. S. Kok, C. Manders, and L. Chaisorn, Evaluation and analysis of anordinal-based approach to video signature, in*Proceedings of the IEEE Region 10 Conference TENCON'2009*, (2009), pp. 1-5.

[7] C. Wu, J. Zhu, and J. Zhang, A content-based video copy detectionmethod with randomly projected binary features, in *Proceedings of the IEEE Computer Society Conference on Computer Visionand Pattern Recognition Workshops CVPRW'2012*, (2012), pp. 21-26.

[8] X. Liu, J. Sun, and J. Liu, Shot-based temporally respective framegeneration algorithm for video hashing, in *Proceedings of the IEEE International Workshop on Information Forensics andSecurity WIFS'13*, (November 2013), pp.109-114.

[9] J. Baber, N. Afzulpurkar, M. Dailey, and M. Bakhtyar, Shot boundarydetection from videos using entropy and local descriptor, in *Proceedings of the 17th International Conference on DigitalSignal Processing DSP'11*, (July2011), pp. 1-6.

[10] P.-H. Wu, T. Thaipanich, and C.-C.Kuo, A suffix array approach to video copy detection in video sharing social networks, in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP'09*, (2009), pp. 3465-3468.

[11] J. Kim and J. Nam, Content-based video copy detection using spatiotemporal compact feature, in *Proceedings of the 11th International Conference on Advanced Communication*

*Technology ICACT'09*, vol. 03, (2009), pp. 1667-1671.

[12] W.-L. Zhao and C.-W. Ngo, Flip-invariant sift for copy and objectdetection, *IEEE Transactions on Image Processing*, (22) 3 (2013), pp. 980-991.

[13] S. Asha and M. Sreeraj, F-surf feature descriptor for video copy detection, in *Proceedings of the Fourth International Conference on Advances in Computing and Communications ICACC'12*, (August 2014), pp. 93-96.

[14] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, (60) 2 (2004), pp. 91-110.

[15] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, Speeded-up robust features (surf), *Comput. Vis. Image Underst.*, (110) 3 (2008), pp. 346-359.

[16] M. Corvaglia, F. Guerrini, R. Leonardi, P. Migliorati, and E. Rossi, Toward a multi-feature approach to content-based copy detection, in *Proceedings of the 17th IEEE International Conference on Image Processing ICIP'10*, (2010), pp. 2345-2348.

[17] Y. Tian, M. Jiang, L. Mou, X. Fang, and T. Huang, A multimodal video copy detection approach with sequential pyramid matching, in *Proceedings of the 18th IEEE International Conference on Image Processing ICIP'11*, 2011, pp. 3629-3632.

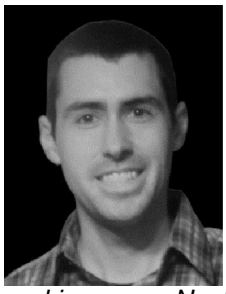
[18] H. Ren, S. Lin, D. Zhang, S. Tang, and K. Gao, Visual words basedspatiotemporal sequence matching in video copy detection, in *Proceedings of the IEEE International Conference on Multimedia and Expo ICME'09*, 2009, pp. 1382-1385.

[19] M. Esmaeili, M. Fatourechi, and R. Ward, A robust and fast videocopy detection system using content-based fingerprinting, *IEEE Transactions on Information and Security,* (6) 1 (20011), pp. 213-226

[20] D. Dutta, S. Saha, and B. Chanda, Photometric attack invariant video sequence matching, in *Proceedings of the 3rd International Conference on Electronics Computer Technology ICECT'11*, vol. 1, (2011), pp. 340-344.

[21] G. Strang, *Introduction to Linear Algebra*, 3rd ed. Wellesley-Cambridge Press, 2003.

[22] L. Chen, F. W. M. Stentiford, L. C. A, and F. W. M. S. B, Videosequence matching based on temporal ordinal measurement, *Tech. Rep.*, 2006.



***Daniel S. Reynolds****received his Bachelor and M.S. degrees in Electrical Engineering from the University of New Hampshire in 2007 and 2014, respectively. The main areas of his research interests are with image processing and home automation. Presently, he is working as a Nuclear Engineer at the Portsmouth Naval Shipyard.*

***Dr. Richard A. Messner*** *received the B.S. and M.S. degrees from Clarkson College of Technology, Potsdam, NY in 1979 and 1981 respectively. From 1981 to 1982 Dr. Messner was a member of the technical staff at the MITRE Corporation in Bedford, MA. He recieved an ONR fellowship award and conducted his Ph.D. research in the Applied Optics Branch of the Optical Sciences Division at the Naval Research Laboratory in Washington, D.C. under the direction of Dr. Harold H. Szu. He was awarded the Ph.D. degree in 1985 from Clarkson University. The same year he joined the Deptartment of Electrical and Computer Engineering as an Assistant Professor and was promoted to Associate Professor in 1989. His research interests include hybrid optical/digital signal processing, real time digital image processing, and biologically inspired signal processing. Dr. Messner is a senior member of IEEE, SPIE, Eta Kappa Nu, and Life member of Sigma Xi.*