# PERCEPTUAL SIMILARITY BASED ROBUST LOW-COMPLEXITY VIDEO FINGERPRINTING

*Karthikeyan Shanmuga Vadivel*$^{1,2}$, *Felix Fernandes*$^{1}$, *Zhan Ma*$^{1}$, *PoLin Lai*$^{1}$, *and Ankur Saxena*$^{1}$

$^{1}$ Samsung Telecommunications America, 1301 E. Lookout Drive, Richardson, TX-75082, USA
$^{2}$ Department of ECE, University of California Santa Barbara, CA-93106, USA
$^{1,2}$karthikeyan@ece.ucsb.edu, $^{1}$\{ffernandes, zhan.ma, w.lai, asaxena\}@sta.samsung.com

## ABSTRACT

In this paper, we present a novel video fingerprinting algorithm which leverages the concept of perceptual similarity between different video sequences. Inspired by the popular structural similarity (SSIM) index, we quantify the perceptual similarity between different video sequences by proposing a perceptual distance metric (PDM) which is utilized in the matching stage of our proposed video fingerprinting algorithm. PDM requires very simple features, viz., block means and therefore has extremely low complexity in both the feature extraction part, as well as during the matching stage. We also show how to use an order statistic in the proposed distance measure to improve the system performance for localized block-based artifacts such as the logo artifact. Simulation results for the proposed fingerprinting algorithm show significant gains over other video fingerprinting techniques on different video datasets for numerous heavy video artifacts.

***Index Terms—*** Video fingerprinting; SSIM; perceptual distance, low-complexity video fingerprinting.

## 1. INTRODUCTION

In today's world, video takes the largest amount of user bandwidth, and is widely distributed via various transport streams. During the distribution, video may be altered intentionally or otherwise through various processes such as encoding artifacts, logo insertion, resizing, etc. At a playback device where the video sequence arrives for viewing, a mechanism for correct identification of the altered video is generally desirable. Several methods exist to allow video identification at a playback device. Textual tagging of video content is a simple method for video identification. Unfortunately, the tags are often destroyed during the distribution process or by unscrupulous pirates, and have to be manually placed most of the times. Steganography is another video-identification method in which the metadata is embedded within the video. But this method is thwarted by alterations, particularly by noise insertion.

Video fingerprinting is an identification method that survives noise attacks readily. This method consists of two stages: The first is the feature extraction stage where compact fingerprints/signatures are extracted from the query video. This is followed by the second stage of matching wherein the extracted signatures are matched against a database of copyright videos, and the status of the query videos is determined: whether they are the same or close to videos in the database. Next, we briefly review some of the well-known feature extraction and matching algorithms.

In video fingerprinting, various global features such as Scalable Color descriptor, Color Layout descriptor [1] and Edge histogram descriptor have been used [2]. But, in general local image features are more robust to artifacts (video tampering/modification) which are localized and hence preferred over the global features. Compact Fourier Mellin Transform (CFMT) descriptor [3] provides a concise and descriptive fingerprint for matching. However, transforming the image frames to a different domain incurs significant computational complexity and is expensive in hardware. Other local interest point based features such as SIFT [4] and its compact version, PCA-SIFT [5] have yielded promising results for the video fingerprinting problem. In SIFT, the matching algorithm involves comparison of large number of interest point pairs without ordering, and this also requires significant processing resources. Motivated by the need of low-complexity local feature based algorithms for video fingerprinting, Centroid of Gradient Orientations and Centroid of Gradient Magnitudes were proposed in [6] and [7], and are quite popular. But these gradient based features are very sensitive to noise, and therefore not robust to artifacts which affect the high frequency content of the video. The MPEG-7 video signature method [8] has a simple feature extraction process, but its performance is primarily dependent on the pre-processing steps. Their approach takes predetermined pairs of blocks, which may be specifically trained to the video database on which tests were performed. Thus, the approach may not work for other video databases.

The second stage of a fingerprinting algorithm after feature extraction is matching, where the distance between two fingerprints is computed. Euclidean distance is a popular distance measure, but fails when the artifact is heavy and localized. More sophisticated distance measures such as Hausdorff distance, partial Hausdorff distance [9] and its variant proposed in [10] outperform Euclidean distance when the query length is short. However, the Hausdorff based distance measures are computationally expensive because they are designed to work well in very difficult, often impractical cases such as when the video frames are permuted. Such a matching technique may be computationally too expensive for simple video fingerprinting applications. All these factors necessitate the need of a distance measure which can be efficiently computed in a video fingerprinting technique, and is robust to heavy artifacts.

In general, artifact videos are perceptually similar to the original videos and therefore using a perceptual similarity metric, which is robust to such artifacts, rather than the conventional Euclidean distance measure (equivalently PSNR) can be useful for video fingerprinting. With this motivation, we propose a low complexity video fingerprinting method in this paper, which uses a perceptual similarity metric, and is robust to a variety of artifacts. The primary contributions of our work are:

The work was performed when Karthikeyan S. V. was an intern with Samsung Telecommunications America.

- A novel low-complexity perceptual distance metric (PDM) based video fingerprinting algorithm which is inspired by the structural similarity (SSIM) index, and is robust to heavy artifacts. In addition the feature extraction part of our algorithm requires very simple features, viz., block means.
- A low-complexity matching algorithm using order statistics to compute the matching score across two video clips.

## 2. PROPOSED VIDEO FINGERPRINTING APPROACH

A high level block diagram of our video fingerprinting system is shown below in Fig. 1 and Fig. 3. We first perform fingerprint extraction on the video query, i.e., extract compact digests (fingerprints) of videos. Here, we assume that similar pre-computed fingerprints already exist for all the videos in the database. After obtaining the query video clip fingerprint, we compare the query fingerprint with the fingerprints from the video database. Finally, in the fingerprinting matching stage, we decide whether the query video clip is part of the database or not and identify the video clip in the database which is similar to the query clip. We next describe the proposed fingerprint extraction module and fingerprint matching module.

### 2.1. Fingerprint extraction module

Fig. 1 shows our proposed feature extraction module. Given a video clip, we extract only the luminance component (Y) in the YUV space. Next, we pre-process the frames by cropping the margins out. The motivation behind margin removal is that margins can be sometimes corrupted with padding artifacts, and therefore may provide very little useful information. We remove r% of the margin on all the sides of a frame in the Pre-processing step of Fig. 1 (here r=10). In fact, removing margin only has a minor change in the performance of non-padding artifacts.

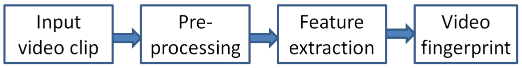

**Fig. 1**. Fingerprint extraction block diagram

Following margin removal, we divide the remaining image portion into $m \times n$ rectangular blocks and compute sample "mean" as features in all these blocks. To compute *finer* features such as second moment for a block, we divide every block into sub-blocks and then compute the sub-block "means" within a block. The motivation of such an approach would become clearer when we discuss the proposed matching algorithm. Fig. 2 shows an example where an image frame is divided into $4 \times 4 = 16$ blocks and $2 \times 2 = 4$ sub-blocks for every block. Typically the number of blocks can be m×n and the number of sub-blocks can be p×q.

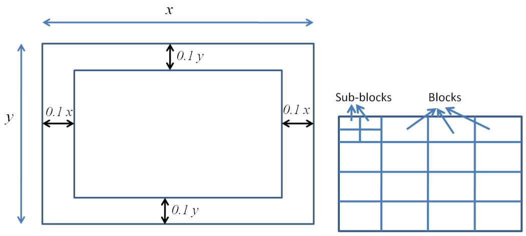

**Fig. 2**. Margin removal (left) and dividing an image frame into blocks and sub-blocks (right) during feature extraction

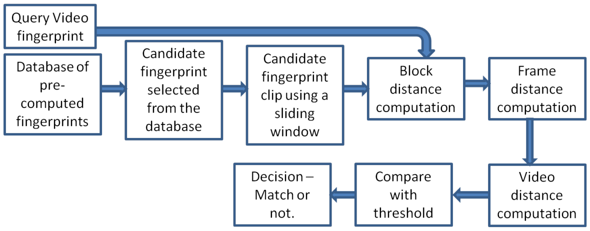
Query Video fingerprint
Database of pre-computed fingerprints
Candidate fingerprint selected from the database
Candidate fingerprint clip using a sliding window
Block distance computation
Frame distance computation
Video distance computation
Compare with threshold
Decision – Match or not.

**Fig. 3**. Fingerprint matching block diagram

### 2.2. Fingerprint matching module

Fig. 3 depicts the proposed fingerprint matching module. We select a candidate video *sequence* from the video database and have to identify whether the given query *clip* is a modified version of a video *clip* in the candidate video sequence. For this, we use a sliding window approach (Fig. 4) where we look at a window size exactly equal to the query video clip length in the candidate video, and verify if the query and the candidate video clip match. This match is determined by computing a distance measure between the query clip and the candidate clip using their video signatures.

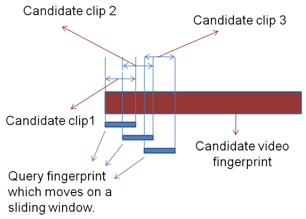
Candidate clip 2
Candidate clip 3
Candidate clip1
Candidate video fingerprint
Query fingerprint which moves on a sliding window.

**Fig. 4**. Sliding window technique for fingerprint matching

Next, we describe how the distance measure between the two video clips is computed via three stages: Block level distance, Frame level distance and Overall video distance as described below.

**Block level distance** Our block distance is based on the perceptual similarity. For this, we are motivated by the popular structural similarity (SSIM) index [11] in which the perceptual similarity between a candidate clip block $X$ and a query clip block $Y$ is given by:

$$SSIM(X,Y) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right)\left(\frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}\right) \qquad (1)$$

where the two terms correspond to a mean factor and a variance factor. In the above equation, $\mu_x$, $\sigma_x^2$ are the block mean and block variance of $X$ and similarly $\mu_y$, $\sigma_y^2$ are the block mean and block variance of $Y$. $\sigma_{xy}$ is the co-variance between $X$ and $Y$, and $C_1$ and $C_2$ are constants as specified in [11]. Also, note that 0 <SSIM(X,Y)< 1 and a larger value denotes higher similarity.

Unfortunately, SSIM cannot be used directly for video fingerprinting because for covariance computation between $X$ and $Y$, all the pixels in $X$ and $Y$ are required. However, after the feature extraction module, only the block features which are computed *independently*, and not all block pixels are available. An alternate would be to store all pixels for blocks, but such an approach would be computationally too expensive and would, in fact, defeat the purpose of video fingerprinting. To overcome this limitation of SSIM for a fingerprinting application, we derive a Perceptual Distance Metric (PDM) as explained next.

First, we approximate every block by a group of sub-blocks as shown in Fig. 2. Let the sub-block means of blocks $X$ and $Y$ be denoted by $\{{\mu'_x}^i\}_{i=1..N}$ and $\{{\mu'_y}^i\}_{i=1..N}$ respectively, where $N = p \times q$ denotes the number of sub-blocks in blocks for $X$ and $Y$. Next, we approximate the second moments from these sub-block means as follows:

$$(\sigma'_x)^2 = \frac{1}{N}\sum_{i=1}^{N}({\mu'_x}^i)^2 - \frac{1}{N^2}(\sum_{i=1}^{N}({\mu'_x}^i))^2 \tag{2}$$

$$(\sigma'_y)^2 = \frac{1}{N}\sum_{i=1}^{N}({\mu'_y}^i)^2 - \frac{1}{N^2}(\sum_{i=1}^{N}({\mu'_y}^i))^2$$

$$\sigma'_{xy} = \frac{1}{N}\sum_{i=1}^{N}({\mu'_x}^i)({\mu'_y}^i) - \frac{1}{N^2}\sum_{i=1}^{N}({\mu'_x}^i)\sum_{i=1}^{N}({\mu'_y}^i)$$

Note that even though the individual variances for block $X$ and $Y$ can be computed during the feature extraction part from the original pixels, they need to be consistent with the co-variance computation, which can only be approximated from the sub-block means and not from the original image pixels. Hence the variance also needs to be approximated as above. Note that the block-mean feature can trivially be obtained from sub-block means as: $\mu_x = (\sum_{i=1}^{N}({\mu'_x}^i))/N$ (since all the $N$ sub-blocks have same number of pixels), and similarly for $\mu_y$.

Next, we define our block perceptual distance metric (PDM) as:

$$PDM(X,Y) = 1 - \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}\right)\left(\frac{2\sigma'_{xy} + C_2}{\sigma'^2_x + \sigma'^2_y + C_2}\right) \tag{3}$$

so that it has a small value when $X$ and $Y$ are similar and a large value when $X$ and $Y$ are perceptually different. We should also mention here, that the choice of sub-block means as a very simple feature is very effective in approximating a perceptual similarity based block distance metric, which cannot be solely performed using the original SSIM in the context of video fingerprinting. As an analogy to the source coding literature, this can be viewed as a fine-coarse level quantization of an image into various sub-blocks and blocks.

**Frame level distance** We compute the frame level distance by using an order statistic ($k^{th}$ smallest value) of the block level distances $PDM(X_i, Y_i)_{i=1..M}$, where $M$ is the number of blocks in the image frame. This approach is computationally and storage wise efficient as the number of blocks in a frame are considerably small. The selection algorithm [12] efficiently computes the frame level distance from the block level distances. Order statistics are particularly robust to localized block level artifacts. For example, a median based metric is robust to logo artifacts such as *closed captions*. However, when more than 50% of the blocks are affected by severe artifacts, median might not be a good choice. In our work, we choose the rank of the order statistic such that heavily altered blocks will be ignored. For example, when 16 blocks are used, we set the rank as 7.

**Video level distance** We compute the video level distance as the mean of all the frame level distances. The number of frames in a video clip can be typically large and using a simple statistic such as the mean reduces the overall matching computational complexity. Also, most practical artifacts affect the frames spatially and not temporally. Hence, computing the mean of the frame level distances temporally is sufficient to compute distance between the video clips.

## 3. EXPERIMENTS AND RESULTS

We show the performance of our algorithm on two different datasets. The first dataset consists of five different types of artifacts introduced by us. The second dataset is the MPEG-7 dataset [13], and we tested our proposed algorithm on the challenging heavy artifacts subset of MPEG-7 dataset. A brief description of the datasets follows.

| Dataset 1 | Dataset 2 |
|---|---|
| Frame Cropping - 70%, 80%, 90% | Camera Capture (CC) |
| Resizing - ×2, ×1/2 | Resolution Reduction (RR) |
| Encoding - 512 kbps, 1 Mbps, 2 Mbps | Logo Insertion (LO) |
| Padding - 10% | Black strip (BS) |
| Scaling - 70%, 80%, 90% | Scaling (SC) |
| | Brightness Change (BC) |
| | Analog Video Conv. (AVC) |

**Table 1**. List of all the artifacts in both the datasets

**Dataset 1:** This dataset consists of 42 videos at resolutions varying from 320 × 240 to 1440 × 1040 totaling about 200 GB of data. We created 12 artifact clips for each original video shown to Table 1. In total, we have 42× 12= 504 artifact clips.

**Dataset 2:** The MPEG-7 dataset consists of 84 videos including high definition videos, totaling about 250 GB of data. MPEG-7 dataset also has numerous artifact clips [13], out of which we selected 672 artifact videos: 8 per video as listed in Table 1. The black strip (BS) artifact in Table 1 was further categorized into two artifacts : vertical BS and horizontal BS.

### 3.1. Setup

In our experiments, we divided each frame into 4×4 blocks and extracted the features from these blocks. Each block was in turn divided into 2×2 sub-blocks. Then, we performed independence and robustness tests similar to [6] and compare our proposed PDM algorithm to Centroid of Gradient Orientations (CGO) [6], Centroid of Gradient Magnitudes (CGM) [7] and Average block luminance (ABL) [6], all of which have similar complexity to PDM. Our implementation of CGO and CGM is exactly same as [6], but for consistency with PDM, we use 4×4 blocks for all the analysis. Note that, we also use order statistic based matching algorithm for all the feature extraction techniques (CGO, CGM, ABL, PDM) as described in Frame Level Distance section, since it gives better performance than mean based frame level distance metric, as shown later in the results.

**Independence test**

In this test, we select about $15 \times 10^6$ video clip distances obtained from different videos in the datasets. Then, we fix a threshold on the distance values and compute the corresponding false alarm rate, or the percentage of distances below the threshold.

**Robustness test**

The overall block diagram of this test is shown in Fig. 3. Here every artifact clip is slid along all the videos in database, and distance of artifact clip from the clips of video sequence in the database is determined. A match is confirmed if the distance between the artifact clip and the candidate original video falls below a threshold. The true positive rate and its complement, the false negative rate is determined using this test.

### 3.2. Observations and Results

**Comparison of different feature extraction techniques**

In the following section we show the False Positive (FP)-False Negative (FN) performance curves for the different algorithms for Dataset 1. The breakdown of the results on all the artifacts is shown in Fig. 5. We observe that our proposed algorithm consistently outperforms other techniques for all artifacts. Next for the MPEG-7 dataset, we show the performance of the different algorithms for a false positive rate of $3 \times 10^{-4}$ in Table 2.

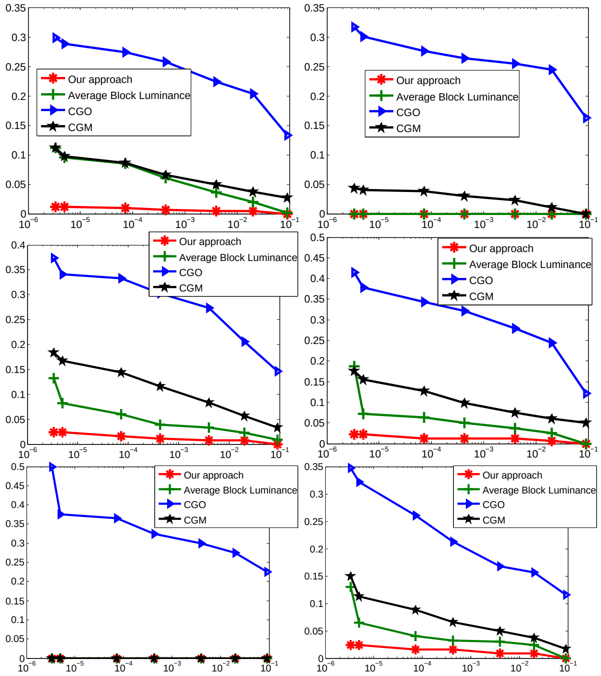

**Fig. 5**. Performance breakdown on different artifacts in Dataset 1, left-right, top-bottom - Overall, Encoding, Cropping, Padding, Resizing, Scaling. The x-axis shows false positive probability, while the y-axis shows false negative probability (Best viewed in color)

| | CGO | CGM | Avg. Blk Lum. | PDM |
|---|---|---|---|---|
| AVC | 42.85 | 72.72 | 85.71 | 85.71 |
| BS | 52.0 | 69.04 | 60.12 | 96.83 |
| BC | 100 | 95.23 | 6.06 | 100 |
| CC | 58.33 | 80 | 39.24 | 93.58 |
| RR | 15.38 | 95.0 | 100 | 100 |
| SC | 7.69 | 65.21 | 100 | 100 |
| LO | 83.33 | 81.05 | 100 | 97.46 |
| Overall | 50.98 | 78.34 | 68.73 | 96.34 |

**Table 2**. True positive rate breakdown for different artifacts in MPEG-7 dataset

Again, for all artifacts except logo insertion, our proposed PDM technique performs better than other algorithms. In the logo artifact, sometimes the variance term in the PDM metric gets affected more than the mean term by the logo artifacts, and the PDM metric therefore performs slightly worse as compared to the ABL, which simply comprises of the mean term. However, note that the ABL is not at all robust to brightness change (BC), since the mean term gets drastically affected by BC, while the proposed PDM metric is very robust to BC due to the variance term. When the artifacts affect the high frequency content of the image (resize, scaling), the gradient based features get affected severely and perform worse as compared to ABL and PDM. Camera capture (CC) is a combination of noise and brightness change, and therefore only PDM is robust to such artifacts. Also, in AVC and BS artifacts, PDM outperforms other algorithms.

To summarize, video artifacts are, in fact, perceptual distortions, and using a perceptual distortion metric such as proposed PDM which is robust to high frequency noise and brightness variations is desirable for a video fingerprinting system.

**Frame level distance analysis:** The commonly used matching algorithm in [6] computed the frame distance metric as the mean of block distance. In our approach we compute an order statistic (rank 7) of the block distances. We notice that on all artifacts except the logo artifact, the performance of mean and order statistic based frame distances are almost similar. However, for logo artifacts, we gain around 40% in accuracy using an order statistic over mean as shown in Table 3. Here, many blocks in the frame corrupted by the logo artifact act as outliers as they contain spurious distance values and corrupt the mean of the block, thereby affecting the frame level distance computation. An order statistic naturally ignores these outlier values and hence is robust to such localized block based artifacts.

| | AVC | BS | BC | CC | RR | SC | LO |
|---|---|---|---|---|---|---|---|
| Order sts. | 85.71 | 96.83 | 100 | 93.58 | 100 | 100 | **97.46** |
| Mean sts. | 86.42 | 95.36 | 100 | 93.45 | 100 | 100 | **56.15** |

**Table 3**. Performance of mean and order statistic based frame level distance measures in MPEG-7 dataset on individual artifacts

## 4. CONCLUSION

In this work, we proposed a new perceptual similarity based matching algorithm for a video fingerprinting system. The proposed matching algorithm requires very simple block mean features during the feature extraction part, and hence has very low complexity. We also use an order statistic in our query to original video distance measure, which improves the system performance, specifically for the case of localized block-based artifacts such as logo artifact. Extensive simulation results are provided, and the proposed video fingerprinting algorithm consistently outperforms other popular algorithms of similar complexity. Finally, a detailed analysis of the proposed PDM measure and comparison with other popular measures by probabilistically modeling different artifacts will be a part of the journal version of this paper.

## 5. REFERENCES

[1] E. Kasutani and A. Yamada, "The MPEG-7 color layout descriptor: a compact image feature description or high speed image/video segment retrieval," *IEEE ICIP*, 2001.

[2] M. Bertini et al., "Video clip matching using MPEG-7 descriptors and edit distance," *ACM CIVR*, 2006.

[3] S. Derrode and F. Ghorbel, "Robust and efficient Fourier-Mellin transform approximations for gray-level image reconstruction and complete invariant description," *Elsevier CVIU*, 2001.

[4] C. Chiu et al., "Efficient and effective video copy detection based on spatio-temporal analysis," *IEEE Int. Symp. on Multimedia*, 2007.

[5] X. Xu et al., "Practical Elimination of near-duplicates from web video search," *Proc. ACM Multimedia*, 2007.

[6] S. Lee and C. D. Yoo, "Robust video fingerprinting for content based video identification," *IEEE Transactions on CSVT*, 2008.

[7] A. Hampapur and R. Bolle, "Video copy detection using inverted file indices," *IBM Research Tech. Report*, 2001.

[8] M. Bober et al., "Study text of ISO/IEC 15938-3:2002/FPDAM4 video signature tools," in MPEG N11085, Kyoto, Japan, 2010.

[9] A. Hampapur and R. Bolle, *Comparison of distance measures for video copy detection*, *Proc. IEEE ICME*, 2001.

[10] A. Sarkar et al., "Efficient and robust detection of duplicate videos in a large database," *IEEE Transactions on CSVT*, 2010.

[11] Z. Wang et al., "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, 2004.

[12] M. Blum et al., "Time bounds for selection," in *Journal of Computer System Science*, 1973.

[13] "Updated call for proposals on video signature tools," in *ISO/IEC JC T1/SC29/WG11, MPEG N10155, Busan, Korea*, 2008.