# 3D Hadamard Transform Based Perceptual Video Hashing

Navaneeth S Rao
Vidyavardhaka College of Engineering
Mysuru- 570002

Shruthi S H
Vidyavardhaka College of Engineering
Mysuru- 570002

Achutha D
Vidyavardhaka College of Engineering
Mysuru- 570002

Dileep M K
UG student
Vidyavardhaka College of Engineering
Mysuru- 570002

Sandeep R
Associate professor
Vidyavardhaka College of Engineering
Mysuru- 570002

Girijamba D L
Assistant professor
Vidyavardhaka College of Engineering
Mysuru- 570002

## ABSTRACT
As there is a dynamic exchange of multimedia data over the internet, content identification and copyright protection has emerged as a serious issue. Perceptual video hashing helps to overcome this problem by providing user authenticity and security to the video data stored. The perceptual video hash function generates a compact code called the hash using the perceptual content of the video. This hash must be robust to any content preserving alterations and sensitive to content changing alterations. The paper proposes a robust video hashing algorithm using 3D Hadamard transformation. This algorithm is well suited for the hardware implementation as the basis functions of Hadamard transform involves only +1 and -1 values.

## General Terms
Hash, Algorithm, Simulation, Transform, Function and Co-efficients.

## Keywords
Perceptual video hashing, Hadamard transform, Near-identical videos, Indexing and video retrieval.

## 1. INTRODUCTION
The hashes computed from cryptographic hash functions are highly sensitive to every single bit of the input. These functions are extremely fragile and cannot be adopted for hashing multimedia data. In video hashing, the hashes computed must be sensitive to the content of the video rather than the actual binary representation. For instance, the original video, its brightness manipulated version, contrast manipulated version, histogram equalized version and noisy version should produce identical hashes as their perceptual content remain same although their binary representations differ. So a different method is required to compute the hash values that result in same output unless the perceptual content of the video is significantly changed. Content tracking and copyright protection is a great issue in recent days as there is a large dynamic flow of data in the internet. So it is necessary to differentiate the video uploaded by the genuine owner and the hacked version of the same uploaded by the invader [1]. The hash must be robust to the content preserving alterations and sensitive to the content changing alterations.

The remaining paper is structured as follows. In section 2 Literature survey is discussed in brief. In section 3, the various properties of the perceptual hash function is defined. In section 4, the mathematics related to the perceptual video hashing using 3D Hadamard transform is discussed. The simulation results and discussion are briefly explained in section 5. Finally in section 6, conclusion are drawn and future work is studied.

## 2. LITERATURE SURVEY
There are various video hash algorithms proposed in the literature.

Coskun and Sankur [2] have proposed a video hashing algorithm by projecting the luminance components of video onto a DCT basis. The input video is normalized and the projected low frequency 3D-DCT co-efficients are quantized to get the hash. Li and Monga [1] proposed a video hashing technique by modeling videos as tensors and using subspace projections of tensors, such as Low-Rank Tensor Approximations (LRTAs) via Parallel Factor Analysis (PARAFAC) for generation of perceptual hash from the videos. A rank one was used for tensor approximations. This method showed excellent robustness to large class of content preserving distortions. Sandeep et al. [3], extracted video hash using Tucker decomposition technique. It decomposes a tensor into a core tensor multiplied by the matrix along each mode. It flexibly decomposes n-way data arrays into a lower dimensionality space. The algorithm showed average performance against malicious video attacks. Inspired from the work in [4], Sandeep and Bora [5] extracted the hashes from the videos by projecting the normalized sub videos onto Achilioptas's random basis. The algorithm showed average performance against content preserving attacks and good performance against content changing attacks. This technique is secure, computationally efficient and retains essential features in reduced dimensionality sub-space with minimal distortion. This algorithm is robust to most of single and multiple image processing attacks.

Many hash algorithms extract hashes using certain key frames from the video. When the video loses the key frames during its transmission or reception, the hash value degrades. Thus to overcome this problem, Sandeep et al. [unpublished], proposed a 3D Radial Projection Technique (3D-RPT) that uses all the frames of the video to compute the hash rather than a particular set of key frames. In this technique, video is treated as an order 3 tensor. The video is broken down into

randomly overlapping sub-blocks and these sub-blocks cover the entire video to compute the hash.

# 3. PROPERTIES OF PERCEPTUAL HASH FUNCTION

The notations utilized for defining the properties are shown in Table 1.

**Table 1. Notations involved in defining the properties of perceptual hash functions**

| Symbol | Description |
|---|---|
| $V_S$ | Finite video space |
| $K_S$ | Finite secret key space |
| $V$ | Video input |
| $K$ | Secret key |
| $P$ | Perceptual hash function |
| $d_h$ | Hamming distance |
| $V_{sim}$ | Video perceptually similar to $V$ |
| $V_{diff}$ | Video perceptually different from $V$ |
| $\emptyset_1, \emptyset_2, \emptyset_3$ | Suitable thresholds |

The various properties of perceptual hash function are as follows:

## 3.1 One-way function

The mapping of the video to the hash must be a one-way function i.e., it should be practically infeasible to reconstruct the whole video from the hash generated.

$$V \rightarrow P(V,K) \qquad (1)$$

Where$V \in V_S \; and \; K \in K_S$

## 3.2 Compactness

The size of the hash must be very small compared to the size of the video.

$$size\big(P(V,K)\big) \ll size(V) \qquad (2)$$

## 3.3 Perceptual robustness

The Hamming distance $d_h$ calculated between the hashes of perceptually similar videos using the same secret key should be very small. The probability of these measurements being close should be very high i.e., near to 1. Mathematically, it is represented as

$$P_r\big\{d_h\big(P(V,K), P(V_{sim},K)\big) \leq \emptyset_1\big\} \approx 1 \qquad (3)$$

## 3.4 Diffusion

The Hamming distance $d_h$ calculated between the hashes of perceptually different videos using the same secret key should be very large. The probability of these measurements being large should be very high. Mathematically, it is represented as

$$P_r\left\{d_h\left(P(V,K), P\big(V_{diff},K\big)\right) > \emptyset_2\right\} \approx 1 \qquad (4)$$

## 3.5 Confusion

The Hamming distance measured between the hashes of perceptually different keys should be very large. The probability of these measured being large should be very high. Mathematically, it is represented as

$$P_r\big\{d_h\big(P(V,K_1), P(V,K_2)\big) > \emptyset_3\big\} \approx 1 \qquad (5)$$

# 4. PERCEPTUAL VIDEO HASHING USING 3D HADAMARDTRANSFORM

The proposed hashing method is an extended idea of 3D DCT based video hashing proposed by Coskun and Sankur in [2]. Here the 3D DCT is replaced by 3D Hadamard transform. The elements of the basis vectors of the Hadamard transform takes only binary values ±1 and are therefore well suited for hardware implementation of the algorithm.

## 4.1 Properties of Hadamard Transform [6]

a) The Hadamard transform is real, symmetric and orthogonal.

b) The Hadamard transform is a fast transform since it contains only $\pm\mathbf{1}$ values, no multiplications are required in transform calculations.

c) The Hadamard transform has good to very good energy compaction for highly correlated images.

## 4.2 Proposed Method

The basic block diagram of perceptual video hashing via 3D Hadamard transform is as shown in Figure 1. The steps involved are as follows: Pre-processing and Normalization, 3D Hadamard transform and Co-efficient selection and Hash selection.

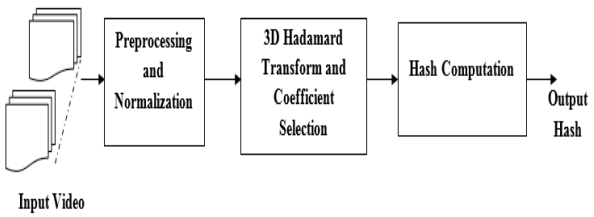


**Fig 1: Block diagram of proposed hashing method**

**a) Pre-processing and Normalization**

The videos to be hashed exist in various frame dimensions and frame rates. Hence, the video data has to be standardized in terms of frame dimensions and number of frames. In-order to standardize, the input video is subjected to temporal sub-sampling followed by spatial sub-sampling. In temporal sub-sampling, the input video is subjected to standard frame dropping to obtain 128 normalized frames. These frames are input to spatial sub-sampling where the normalized frames are re-sized to a standard size. We have standardized the frame dimensions to 64 × 64. It is as shown in Figure 2.

Let us use the notation $V_{input}\,(w,h,f)$ that denotes any input video where$w$ is the frame width, $h$ is the frame height and $f$ is the number of frames. In this step, the input video $V_{input}\,(w,h,f)$ is converted to standard size$V_{norm}$. This standard size was experimentallydetermined based upon the fact that smaller sizes risk losing semantic content.

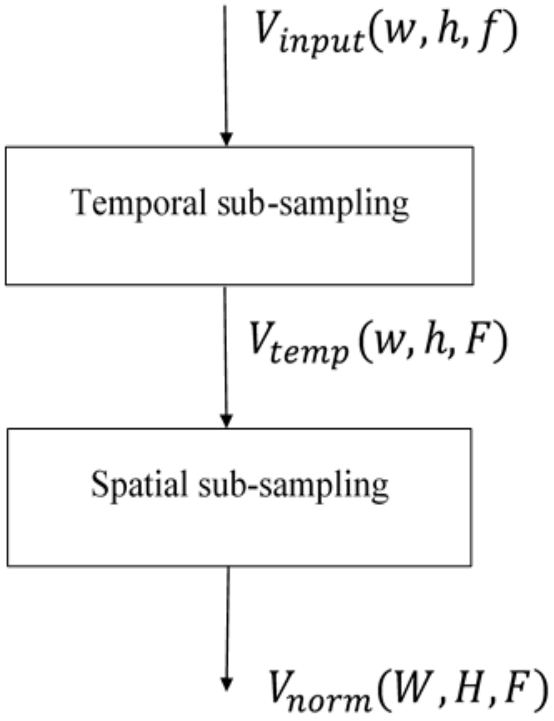


**Fig 2: Normalization before Hash extraction**

**b) 3D Hadamard transform and co-efficient selection**

The 3D Hadamard transform is applied on the normalized video. After applying the 3D Hadamard transform on the normalized video $Vnorm(W,H,F)$ , we obtain 3D array of Hadamard co-efficients. The low frequency Hadamard co-efficients contain most of the energy and are robust against various signal processing attacks. In our work, we have extracted hashes from a $8 \times 8 \times 2$ cube. It is as shown in Figure 3.

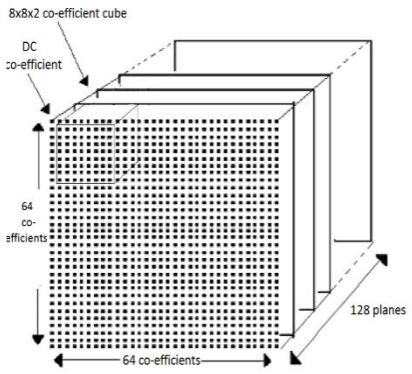


**Fig 3: The 3D-Hadamard array of normalized video with dimensions W = 64, H = 64 and F = 128.**

**c) Hash computation**

After the 3D transform is applied and co-efficients are selected, the hash calculation procedure remains the same irrespective of the transformation used. The selected 128 co-efficients are converted to binary string using the method of median based quantization. Let the co-efficients be denoted by $Cn$ where $n$ ranges from 1 to 128. The hash is binarized using the Equation 6.

$$h_n = \begin{cases} 1: C_n \geq median(C_n) \\ 0: C_n < median(C_n) \end{cases} \quad (6)$$

Where n = 0, 1, 2….128

## 5. SIMULATION RESULTS AND DISCUSSION

The details of our original database used for experimentation is tabulated in Table 2. A total of 1000 videos were selected for the experimentation purpose. Totally 19 attacks were performed on the database. They are: Brightness increase, brightness decrease, contrast increase, contrast decrease, adding logo (of size $\mathbf{64 \times 64}$) adding logo (of size $\mathbf{128 \times 128}$), histogram equalization, frame dropping, frame rate increase (to 60fps), frame rate decrease (to 15fps),spatial resolution increase, spatial resolution decrease, average blurring, gaussian blurring, addition of salt and pepper noise, addition of gaussian noise, bit rate changes (to 100 kbps),frame rotation and frame cropping.

**Table 2. Database details used for experimentations**

| **Parameter** | **Description** |
|---|---|
| Database | [trecvid.nist.gov][open-video.org] |
| Number of videos | 1000 |
| Minimum spatial resolution | $192 \times 144$ |
| Maximum spatial resolution | $640 \times 480$ |
| Minimum frame rate | 4 |
| Maximum frame rate | 60 |
| Minimum number of frames | 1 |
| Maximum number of frames | 11415 |
| Video format | MP4 |

A total of 20,000 video database was created (including the original videos). The videos were normalised to $\mathbf{64 \times 64 \times 128}$. A 128 bit hash is extracted from $\mathbf{8 \times 8 \times 2}$sub-cube for a single video. It is repeated for total number of videos in the database and is saved in an excel spreadsheet. For any query video selected, hash is computed and is checked for matching with the hashes in the spreadsheet. In-order to compare the hashes, hamming distance metric is used. The normalized hamming distance is computed between the hashes of the query video and the videos in the database. A threshold is setup for retrieval of videos by trial and error method. If the normalized hamming distance is less than the threshold, the videos are said to be near identical and are retrieved. If the normalized hamming distance is greater than the threshold, the videos are said to be different and are not retrieved. To compare the perceptual content between the query video and a randomly chosen video from the database, a graphical user interface (GUI) is created for clean and easy interface between the application and the user. It is as shown in Figure 4 and Figure 5 respectively.

## 6. CONCLUSION AND FUTURE WORK

The Proposed video hashing technique works fine for common signal processing attacks but under frame insertions or dropping of frames, there is a degradation of performance. The hardware implementation is easy as the basis vectors in Hadamard transform involves only ±1. The retrieval of near

identical videos is less complex and hardware implementation involve only addition and subtraction operations. The performance of the algorithm is yet to be studied using precision-recall curves and Receiver Operating Characteristic curves (ROC).

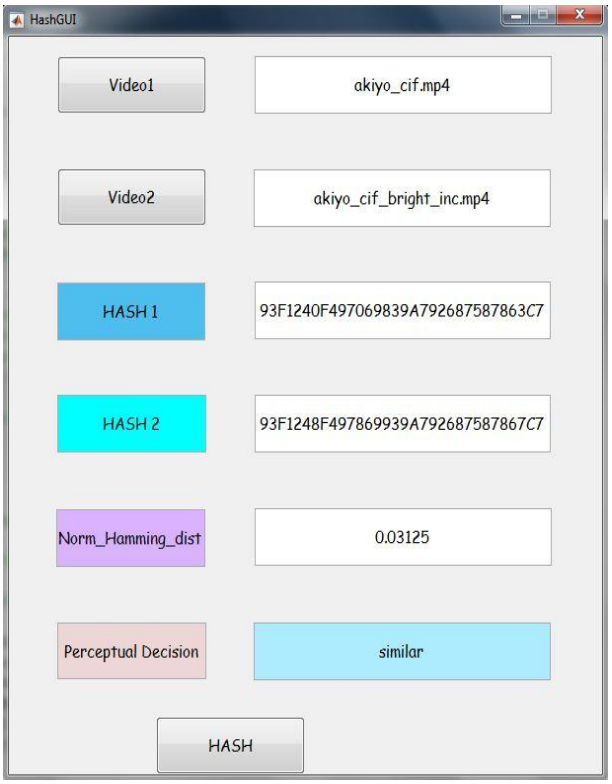


**Fig 4: GUI showing similar videos**


**Fig 5: GUI showing dissimilar videos**

## 7. REFERENCES

[1] M. Li and V. Monga, "Robust video hashing via multi linear subspace projections," *IEEE Transactions on Image processing*, vol. 21, no. 10, pp. 4397-4409, Oct 2012.

[2]B. Coskun, B. Sankur, and N. Memon, "Spatio-temporal transform based video hashing," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1190–1208, 12 2006.

[3] R. Sandeep, S. Sharma, T. Mayank, and P. K. Bora" Perceptual video hashing based on tucker decomposition with application to indexing and retrieval of near-identical videos," *Multimedia Tools and Applications*, vol. 75, no. 13, pp. 7779–7797, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11042-015-2695-1

[4]M. Li and V. Monga, "Desynchronization resilient video fingerprinting via randomized, low-rank tensor approximations," in *2011 IEEE 13th International workshop on Multimedia Signal processing*, Oct 2011, pp. 1–6.

[5] R. Sandeep and P. K. Bora, "Perceptual video hashing based on the achlioptas's random projections," in 2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Dec 2013, pp. 1–4.

[6] A. K. Jain, *Fundamentals of Digital Image Processing*, ser. Prentice-Hall Information and System Sciences Series. Prentice-Hall, 1989.