International Journal of Basic and Applied Sciences, 14 (8) (2025) 505-511

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/fe3h7455
Research paper

Performance Assessment of Enhanced Chio-Like Method in Video Retrieval a Non-Square Determinant Approach

Besnik Duriqi [1] *, Halil Snopçe [1], Armend Salihu [2], Artan Luma [1]

[1] Computer Sciences, South East European University, Tetovo, North Macedonia
[2] Computer Science, UNI Universum International College, Lipjan, Kosovo
**Corresponding author E-mail: bd30521@seeu.edu.mk*

Received: November 20, 2025, Accepted: December 20, 2025, Published: December 23, 2025

Abstract

The rapid growth of digital video data makes efficient Content-Based Video Retrieval (CBVR) increasingly important, yet traditional similarity measures often fail to capture high-er-order dependencies between video features. This paper introduces a CBVR pipeline that uses a novel non-square determinant kernel as the direct similarity score with a faster Chio-like algorithm that reduces matrix order by four in each step. Experiments show an average execution-time decrease of about 25 % compared to the standard Chio-like method and 3.1 % compared to its modified version. Integrating this kernel into the CBVR demonstrates that the Chio-enhanced determinant kernel outper-forms similarity measures across benchmark vid-eo datasets. By demonstrating superior retrieval efficiency and accuracy, the proposed meth-od is well-suited for efficient and accurate similarity evaluation in large-scale or real-time CBVR applications.

Keywords: *Content-Based Video Retrieval; Determinant Kernels; Non-Square Determinants; Enhanced Chio Method; Similarity Score.*

1. Introduction

The very fast growing of video data on the internet, surveillance systems, and digital libraries has created an urgent demand for efficient video retrieval systems. Unlike metadata-based retrieval, which depends on textual annotations, Content-Based Video Retrieval (CBVR) focuses on analyzing the intrinsic content of videos, such as colors, textures, motion, and spatial-temporal patterns. The challenge of similarity measurement, which is at the core of CBVR, is how to identify the videos in a database that share higher similarity to a query video. Traditional metrics like trace kernels, Euclidean distance, and cosine similarity have been applied extensively. These methods are computationally simple but they mostly use vector-based comparisons, which might miss more profound structural relationships in feature matrices. Recently, determinant kernels have emerged as promising similarity functions. By considering the determinant of feature matrix products, these kernels inherently capture multi-dimensional correlations. However, their adoption has been limited by the computational overhead of determinant calculation, particularly for non-square matrices that arise in CBVR when video shots of different lengths are compared. In our previous work [1] introduced an enhanced Chio-like method to compute non-square determinants efficiently. This paper takes the next step by providing an experimental validation of determinant kernels within a full CBVR pipeline. This article has two complementary goals: (i) to review and synthesize determinant-based similarity functions and non-square determinant computation strategies in CBVR, and (ii) to experimentally assess an enhanced Chio-like algorithm that reduces determinant order by four at each step, enabling practical determinant-kernel similarity evaluation for variable-length video shots.

2. Related Work

The Content Based Video Retrieval (CBVR) is considered one of the most practical methods for attaining high-quality video retrieval [2]. The integration of video content offers great opportunities for enhancing actual search engines, indicating that the area of CBVR holds promise for creating more efficient video search engines in the future. Most online video retrieval systems function by indexing and re-trieving videos using text linked to them [3]. The core of video retrieval lies in extracting features. Different types of features exist, such as color features, texture features, and shape features [4], which can be obtained from images and video data. The most powerful aspect of video retrieval is based on color. In particular, the color histogram is an easy but efficient method [5]. Kernel-based multimedia retrieval methods have proven effective in several applications such as shape recognition, image retrieval [6], and event detection [7]. Many methods start by creating a kernel function using supervised data [8], on which a classifier, such as support vector machines (SVM), is developed. Numerous recent efforts [9] have created kernel methods for semi-supervised metric learning too. In [10], two algorithms for kernel-based metric learning utilizing pairwise similarity constraints (Kernel-A and Kernel-B) are presented. Authors in [11] created an extension of the Kernel-b method presented in [10]. Moreover, there exists a range of query types that consist of example-based queries sketch-based queries [12] object-based queries [13], keyword-based queries, natural language-based queries [14], and combination-based queries [15].

This paper explores query by example. When a query comes in, the most similar shots can be retrieved using a similarity method. To obtain a recorded video, measures of similarity are essential. There are many similarity techniques nowadays, including: combination-based matching [16], ontology-based matching [17], text matching [18], and feature matching [19]. This paper introduces a novel similarity calculation method using a determinant kernel approach based on the features extracted. Deep metric learning trains an embedding such that similar videos are close under cosine and Euclidean distance. These approaches can provide strong semantic retrieval but usually require labeled pairs/triplets and significant training resources. In contrast, determinant-kernel similarity is training-free once features are extracted and provides a matrix-level score that captures higher-order dependencies across key frames. The approaches are complementary: the determinant kernel can also be computed on deep embeddings, combining learned representations with matrix-level similarity [20-21]. Kernel functions determine the inner product of the shot-feature matrices and produce a real number. Various kernel functions are available, such as KPCA, KLDA, RBF, and Fisher kernel [22]. In essence, kernel functions function on vectors and square matrices; however, the representation matrices of real data, such as shot-feature matrices, are not square. Almost all current techniques reduce input non-square matrices to make them compatible with kernel functions, leading to a loss of certain data. In this study, we have broadened the idea of the determinant concerning non-square feature matrices processing. We will proceed with the kernel presented in [22] for non-square matrices based on the non-square determinant definition [23] and utilize Chio-like techniques modified by the authors [24] and [25], evaluating the efficacy of the resulting kernels by retrieving movie video clips where shots are manually segmented. Kernels and determinants represent two essential concepts in linear algebra, especially regarding matrices. A matrix is a set of numbers arranged in rows and columns so as to form a rectangular array. The kernel of a matrix consists of the set of vectors that yield a zero vector when multiplied by the matrix. The determinant of a matrix is a scalar quantity derived from its elements that signifies specific characteristics of the matrix. Kernels and determinants find applications across various fields, such as physics, engineering, economics, and computer science. In computer science, image processing uses kernels as well as determinants [26].

Kernels are mathematical functions that accept two inputs and calculate a similarity metric between them. Determinant kernels are a specific type of kernel that employs the determinant of matrices (commonly obtained from feature or covariance matrices) to measure the similarity between two data points or sets of features. In CBVR, determinant kernels are utilized by incorporating them into different phases of the retrieval process: extracting features, calculating similarity through determinant kernels, and ordering results according to the similarity scores. CBVR has evolved from manual descriptors toward deep representations, and from simple vector distances toward learned similarity functions. However, variable-length shots naturally form non-square feature matrices, and forcing these into square representations can discard structural information. Determinant kernels provide matrix-level similarity that captures higher-order dependencies, but their use in CBVR is constrained by the computational cost of non-square determinant evaluation. This gap motivates the enhanced Chio-like method evaluated in this work.

3. Methodology

The methodology in this research is comprised of an analytical approach as well as simulation with Matlab and a video dataset. In our simulation, we selected about 500 video shots which were manually annotated and splitted and where during sampling process of selection of key-frames, we always selected larger number of key-frames for database videos. By simulation we have compare the processing efficiency of video retrieval of newly developed Chio-like algorithm against earlier one [24] and [25]. Although the algorithms' pseudocode are generalized to be executed in any environment, for this simulation we have used in MATLAB 2021b version environment, in Asus ROG Flow Z13, with 12th Gen Intel(R) Core(TM) i7-12700H, 16.0 GB of RAM, NVIDIA GeForce RTX 3050 GPU. For this simulation we have used a video dataset segmented into precise shots. During the video transformation process for video retrieval purposes, initially the features of the query video as well as the features of the videos in the database are transformed and represented with the relevant matrices. Features in our case are colors. Although we use color histograms as the baseline feature representation, the proposed similarity computation is agnostic to feature type. Any per-frame/shot descriptor (e.g., LBP texture, motion descriptors, optical-flow statistics, or deep CNN/video embeddings) can be arranged into a shot-feature matrix, after which the same Gram-matrix and non-square determinant kernel computation applies. Further, the features of the videos, in order to compare their similarity, are transformed into the matrix known as the Gram matrix, which represents the scalar product of dots (further dot product) between two matrices. After creating the Gram matrix, the kernel-determinant is calculated for video retrieval. The input to the kernel determinant function is a scalar product from the feature matrices. The Kernel function takes the input points (features) and returns a real number that represents the level of their similarity, while the kernel itself in this case is calculated by means of the determinant. Since videos generate non-square feature matrices due to their dynamicity, their kernel is very complex to calculate depending on the size of the feature matrix. Two feature matrices were compared using the determinant kernel for non-square matrices where the output is a real number; the determinant-kernel produces a real-valued similarity score whose absolute magnitude can be very small; therefore, in retrieval we interpret it comparatively, where higher scores indicate higher similarity between the two shots. Determinant-kernel scores may appear very small in absolute magnitude because the determinant combines correlation information across multiple key frames into a single scalar value, and its scale depends on the size of the matrices involved. This behavior is normal for determinant-based similarity measures in high-dimensional settings. In our CBVR pipeline, the determinant score is used as a similarity indicator: for a fixed query, database shots that are more similar produce higher scores than less similar shots. To ensure consistent interpretation of similarity scores, we apply the same feature normalization procedure to all key-frame descriptors before forming the Gram matrix. In addition, we report determinant scores consistently (in scientific notation) and interpret retrieval outcomes using score ordering for each query. This makes the comparison transparent and ensures that the ranking reflects similarity rather than differences in raw feature scaling. If A is feature-matrix of a shot with m key frames and B is feature-matrix of a shot with n key frames, then for $C_{mxn}(m \leq n)$ we have:

$$C_{mxn} = A * B^T \tag{1}$$

Following the works of authors in [24] and [25], we have enhanced further the mathematical and algorithmic model which is published in our previous [1] by which we have performed the simulations and the results are presented in the result section and discussion. In the following we will present initially the two initial theorems and algorithms of above mentioned authors which have provided the foundation in our new theorem 3 and corresponding algorithm det_Chio3:

Theorem 1: *[24] (Chio's-like method for rectangular determinants): For rectangular determinants of the order* $m \times n$*, in cases for* $2 \times 3, 2 \times 4$ *and* 3×4*, the following formula holds:*

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{vmatrix}_{m\times n} = \frac{|A_c|}{a_{11}^{m-2}} + (-1)^m \begin{vmatrix} a_{12} & a_{13} & \cdots & a_{1n} \\ a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m2} & a_{m3} & \cdots & a_{mn} \end{vmatrix}_{m\times(n-1)} \tag{2}$$

Where:

$$|A_c| = \begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{1n} \\ a_{21} & a_{2n} \end{vmatrix} \\ \vdots & \ddots & \vdots \\ \begin{vmatrix} a_{11} & a_{12} \\ a_{m1} & a_{m2} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{1n} \\ a_{m1} & a_{mn} \end{vmatrix} \end{vmatrix}_{(m-1)\times(n-1)} \tag{3}$$

And $a_{11} \neq 0$.
Proof of Theorem 1: See Theorem 2.2 in [24].

Theorem 2: *[25] Suppose that A is rectangular matrix of order $m \times n, m > 3$ and $m \leq n-1$, its determinant can be calculated using formula below:*

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{vmatrix}_{m\times n} = \tag{4}$$

$$= \frac{|A_{c1}|}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}^{m-3}} + (-1)^m \begin{vmatrix} a_{12}-a_{11} & a_{13} & \cdots & a_{1n} \\ a_{22}-a_{21} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m2}-a_{m2} & a_{m3} & \cdots & a_{mn} \end{vmatrix}_{m\times(n-1)}$$

Where:

$$|A_{c1}| = \begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ a_{31} & a_{32} & a_{3n} \end{vmatrix} \\ \vdots & \ddots & \vdots \\ \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{m1} & a_{m2} & a_{m3} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{12} & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ a_{m1} & a_{m2} & a_{mn} \end{vmatrix} \end{vmatrix}_{(m-2)\times(n-2)} \tag{5}$$

And

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \neq 0.$$

Proof of Theorem 2: See Theorem 4 in [25]
Following the approach of authors in [24] and [25] of using Chio-like methods for non-square determinants we have further improved, since our approach reduces determinant for four orders.

Theorem 3: *Suppose that A is rectangular matrix of order $m \times n, m > 5$ and $m \leq n-1$, its determinant can be calculated using formula below:*

$$\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{vmatrix}_{m\times n} =$$

$$= \frac{|A_{c2}|}{\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix}^{m-5}} + \begin{vmatrix} a_{11} & a_{12} & a_{14}-a_{13} & a_{15} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{24}-a_{23} & a_{25} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m4}-a_{m3} & a_{m5} & \cdots & a_{mn} \end{vmatrix}_{m\times(n-1)} \tag{6}$$

$$+ \begin{vmatrix} a_{12}-a_{11} & a_{14}-a_{13} & \cdots & a_{1n} \\ a_{22}-a_{21} & a_{24}-a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m2}-a_{m1} & a_{m4}-a_{m3} & \cdots & a_{mn} \end{vmatrix}_{m\times(n-2)} + (-1)^m \begin{vmatrix} a_{12}-a_{11} & a_{13} & \cdots & a_{1n} \\ a_{22}-a_{21} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m2}-a_{m1} & a_{m3} & \cdots & a_{mn} \end{vmatrix}_{m\times(n-1)}$$

$$|A_{c2}| = \begin{vmatrix} \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{3n} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{4n} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{5n} \end{vmatrix} \\ \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{3n} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{4n} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{6n} \end{vmatrix} \\ \vdots & \ddots & \vdots \\ \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & a_{m5} \end{vmatrix} & \cdots & \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{3n} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{4n} \\ a_{m1} & a_{m2} & a_{m3} & a_{m4} & a_{mn} \end{vmatrix} \end{vmatrix}_{(m-4)\times(n-4)} \quad (7)$$

Proof of Theorem 3: See Theorem 3 in [1].
In the following is the corresponding algorithm of Theorem 3 as well as essential steps of the algorithms of Theorem 1 and 2 essential to our simulation:
Pseudocode of det_Chio3 algorithm for Theorem 3 [1]: (Enhanced Chio's-like) method to calculate rectangular determinants of order $m \times n, m > 5$ and $m \leq n - 1$

```
Step 1: Check edge cases
        if m > n
                d = 0
                return d
end
if m == n or m < 5
                d = Radic_Definition(A) // use Radic or standard determinant
                return d
end

Step 2: Calculate Pivot Block

Pivot = det(A(1:4, 1:4)) // 4×4 determinant
if Pivot == 0
                Interchange rows to make Pivot ≠ 0
                Pivot = det(A(1:4, 1:4)) // recompute
end

Step 3: Build Condensed Matrix B
Initialize B = zeros(m−4, n−4)
        for i = 5 to m
        for j = 5 to n
        B(i−4, j−4) = det(A([1,2,3,4,i], [1,2,3,4,j])) // 5×5 determinant
        end
 end

Step 4: Compute Subtracted Columns

Sub1 = A(:,2) − A(:,1) // m×1

Sub2 = A(:,4) − A(:,3) // m×1

Step 5: Recursive Chio4 Calls
T1 = det_Chio4(B)
T2 = det_Chio4([A(:,1:2) Sub2 A(:,5:n)]) // m×(n−1)
T3 = det_Chio4([Sub1 Sub2 A(:,5:n)]) // m×(n−1)
T4 = det_Chio4([Sub1 A(:,3:n)]) // m×(n−2)

Step 6: Combine Determinants

d = T1 / Pivot^(m−5) + T2 + T3 + (−1)^m * T4

return d
```

Essential step of the algorithm of Theorem 1 [24]:
Step 5: Calculate the final result of non-square determinant

```
d = 1/A(1,1)^(m - 2) * det_Chio(B) + (-1)^m * det_Chio(A(1:m, 2:n));
```

Step 5: Display the result of the determinant
Essential step of the algorithm of Theorem 2 [25]:
Step 4: Calculate the result of non-square determinant

Sub1 = A(:,2) − A(:,1) //m×1

d=det_Chio2(B)/Pivot^(m- 3)+(1)^m*det_Chio2([Sub1 A(:,3:n)]);

In the following is shown the pseudocode if the simulation:

```
load "query"
Query ← QueryFeatureMatrix

for each file in filenames:

 load featureMatrix from file
 A ← DataSetFeatureMatrix

 B ← Query × transpose(A)

 start timer
 det1 ← det_C1(B)
 t1 ← elapsed time

 start timer
 det2 ← det_C2(B)
 t2 ← elapsed time

 start timer
 det3 ← det_C4(B)
 t3 ← elapsed time

 Δ12 ← t1 − t2
 Δ13 ← t1 − t3
 Δ23 ← t2 − t3

 P12 ← 100 × (t1 / t2 − 1)
 P13 ← 100 × (t1 / t3 − 1)
 P23 ← 100 × (t2 / t3 − 1)

end for
```

4. Results and Discussion

This section evaluates the performance of the enhanced Chio-like method within the determinant-kernel CBVR pipeline. Computation time and similarity score behavior were examined across about 500 video shots comparisons using three determinant computation methods: the original Chio-like method (Theorem 1), the modified version (Theorem 2), and the proposed enhanced method. In the following we have tested the video retrieval by generating similarity scores as well as execution time of the algorithm presented above and compared with the execution time of algorithms based on Theorem 1 (see: Algorithm 2.3 in [24]), and Theorem 2 (see: P2 in [25]), and the results are presented in the following Table 1:

Table1: Similarity Scores and Processing Efficiency

Shot Nb	Similarity Score	Theorem1	Theorem2	Theorem3	1-3	2-3	1-3%	2-3%
14	3.44E-40	3.87	2.96	2.84	1.03	0.12	27%	4.2%
7	1.41E-39	4.38	3.32	3.24	1.14	0.09	26%	2.6%
8	1.56E-39	5.91	4.50	4.36	1.55	0.14	26%	3.0%
17	3.35E-34	3.82	2.92	2.84	0.99	0.08	26%	2.9%
18	7.62E-34	3.84	2.96	2.86	0.98	0.10	25%	3.3%
36	4.28E-33	0.00	0.00	0.00	0.00	0.00	27%	2.2%
40	9.40E-33	5.76	4.42	4.58	1.18	0.16	20%	3.7%
3	9.57E-33	3.81	2.92	2.87	0.94	0.06	25%	1.9%
18	1.93E-32	0.10	0.07	0.07	0.04	0.00	35%	2.8%
6	4.52E-32	0.00	0.00	0.00	0.00	0.00	30%	3.8%
18	8.15E-32	3.77	3.03	2.90	0.87	0.13	23%	4.4%
4	2.39E-31	0.94	0.69	0.71	0.23	0.01	25%	2.1%
3	4.14E-31	3.72	2.83	2.89	0.83	0.06	22%	2.1%
33	7.91E-31	4.19	3.07	2.98	1.21	0.09	29%	2.9%
7	1.16E-30	3.76	2.89	2.84	0.92	0.05	24%	1.8%
41	1.22E-29	4.26	3.09	2.99	1.26	0.09	30%	3.0%
29	6.11E-29	3.72	3.03	2.94	0.79	0.09	21%	2.9%
16	8.65E-29	0.01	0.01	0.01	0.00	0.00	27%	1.9%
24	2.06E-28	0.09	0.06	0.07	0.02	0.00	25%	2.8%
38	3.36E-28	3.78	2.90	3.02	0.76	0.12	20%	4.3%

36	4.53E-28	3.83	2.97	3.11	0.72	0.14	19%	4.8%
42	5.25E-28	6.02	4.57	4.43	1.59	0.14	26%	3.0%
40	8.37E-28	0.56	0.39	0.41	0.15	0.01	27%	2.7%
13	1.07E-27	0.02	0.02	0.02	0.01	0.00	24%	4.7%
25	2.52E-27	3.74	2.84	2.96	0.78	0.11	21%	4.0%
11	4.44E-27	0.00	0.00	0.00	0.00	0.00	26%	4.9%
63	6.29E-27	3.93	2.98	2.92	1.01	0.06	26%	1.9%
4	7.30E-26	0.00	0.00	0.00	0.00	0.00	18%	2.0%
5	8.02E-26	3.80	2.89	2.82	0.98	0.07	26%	2.6%
37	3.82E-25	0.18	0.13	0.13	0.04	0.00	25%	3.7%
			Average improvement				25.0%	3.09%

Beyond computational efficiency, an important conceptual advantage of determinant-kernel similarity is its ability to reflect higher-order (multilinear) dependencies across the entire key-frame feature matrix. Vector-based similarities such as cosine and Euclidean similarity typically require pooling frame information into a single descriptor, which can suppress cross-frame dependency structure. In contrast, determinant-based similarity operates at the matrix level and is sensitive to joint correlation structure across multiple frames; it naturally penalizes redundancy among key-frame feature vectors (driving the determinant magnitude toward zero) and yields higher scores when correlations across frames are jointly consistent. This provides a mechanism to exploit multi-frame structure that may be under-represented in purely vectorized similarity measures.

4.1. Computational efficiency and similarity behaviour of determinant kernels

Table 1 shows that the proposed method achieves the fastest computation time for every individual shot in the dataset. Across about 500 executions, it attains an average execution time reduction of 25% compared to Theorem 1 and 3.1% compared to Theorem 2. The improvement is consistent across all trials, indicating that reducing determinant order by four in each step—produces a measurable and stable computational benefit. The enhanced algorithm maintains both structural equivalence and identical determinant outputs while reducing intermediate operations, which explains the consistent speed increase. All three methods output identical similarity scores for every shot, confirming that the computational optimization does not alter the determinant value. The determinant magnitudes are in the range of approximately 10^{-31} to 10^{-60}, reflecting the high-dimensional Gram-matrix products commonly produced in video-feature correlations: database video shots visually more similar to the query video shot consistently yield higher determinant values, while unrelated shots produce smaller values. This behavior aligns with the determinant-volume interpretation of the similarity kernel used in the experiment, where higher cross-correlation between key frame distributions leads to larger determinant values. Figure 1 illustrates this behavior for high-similarity and low-similarity examples, where score differences correspond to observable differences in visual content. In the following is presented the graphical view Figure1 of the comparison presented on Table1:

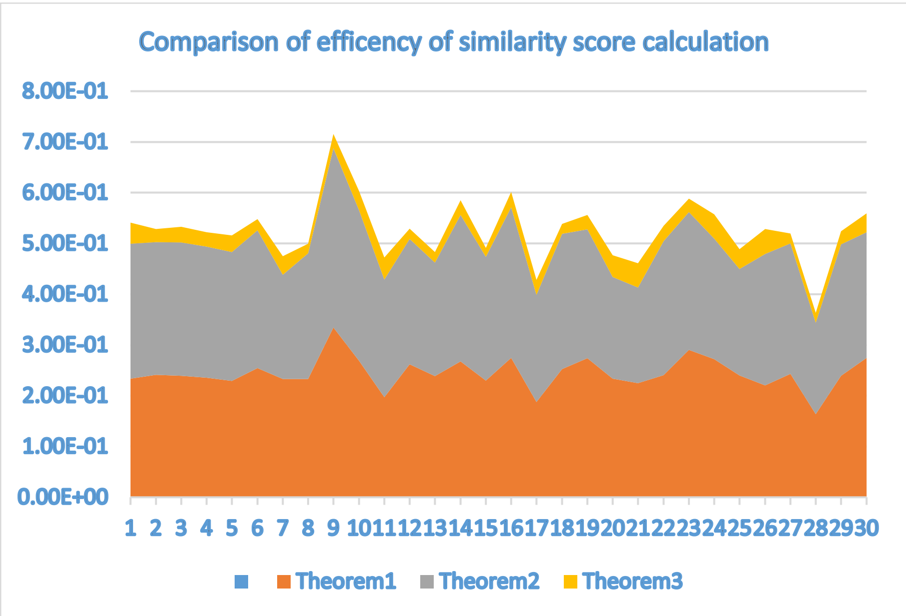

Fig. 1: Runtime Comparison of Determinant Computation Methods (Theorem 1: Original Chio-Like; Theorem 2: Modified; Theorem 3: Enhanced) for the Representative Query–Shot Comparisons Listed in Table 1.

Highest similarity score is achieved with itself that is 3.82E-25. The experimental findings demonstrate that determinant-based similarity functions are practical and effective for Content-Based Video Retrieval when combined with an efficient determinant computation algorithm. The enhanced Chio-like method provides a clear computational advantage without modifying the mathematical output of the determinant kernel. The ability of determinant kernels to capture higher-order dependencies is clearly reflected in the similarity-score distribution: visually similar shots consistently produce higher determinant values. This behavior supports the view that determinants encode multi-perspective correlation structures across all key frames, unlike vector-based similarity measures that collapse frame information into low-dimensional descriptors. Another key insight is that the optimized algorithm scales well with dataset size. The consistent 25% speed-up over Theorem 1 and the further 3.1% gain over Theorem 2 show that the enhanced method reduces computation while preserving the structural properties of determinant kernels. This efficiency makes determinant kernels suitable for large-scale or near-real-time CBVR systems, where traditional determinant computation would be very costly.

5. Conclusion and Future Work

This work demonstrates that determinant kernels can be applied efficiently and effectively in Content-Based Video Retrieval when supported by an optimized Chio-like determinant algorithm. The enhanced method reduces the determinant order by four per iteration, resulting in a consistent 25% faster performance over the standard Chio-like method and 3.1% faster performance compared to its modified variant, while preserving identical similarity outputs. The results confirm that determinant kernels can be computed efficiently without sacrificing retrieval performance. Future work will explore optimized implementations in lower-level languages, the integration of learning-based feature representations, and the application of the determinant kernel to additional domains such as video classification and cross-modal retrieval. On the other hand, scalability on the for large-scale CBVR can be addressed by (i) offline pre-computation of shot-level feature matrices, (ii) candidate-set pruning using inexpensive approximate similarity measures or approximate nearest-neighbour indexing, and (iii) batched determinant computations on GPU/parallel hardware. Together with a lower-level implementation of det_Chio3, these strategies enable near-real-time deployment.

References

[1] Duriqi, B. S. (2025). Enhanced algorithm based on Chio-like Method for Non-Square Determinant Calculations for application in CBVR. Journal of Applied Science and Technology Trends, 149-160. https://doi.org/10.38094/jastt62253.

[2] Patel, B. &. (2023). Content-based Video Retrieval Systems: A Review. International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 441-449). IEEE. https://doi.org/10.1109/ICIMIA60377.2023.10425939.

[3] Dong, J. W. (2022). Reading-strategy inspired visual representation learning for text-to-video retrieval. IEEE transactions on circuits and systems for video technology (pp. 5680-5694). IEEE. https://doi.org/10.1109/TCSVT.2022.3150959.

[4] Ge, Y. X. (2022). Contributions of shape, texture, and color in visual recognition. European Conference on Computer Vision (pp. 369-386). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-19775-8_22.

[5] Shamoi, P. S. (2022). Comparative overview of color models for content-based image retrieval. International Conference on Smart Information Systems and Technologies (SIST) (pp. 1-6). IEEE. https://doi.org/10.1109/SIST54437.2022.9945709.

[6] Kumar, S. S. (2023). Efficient deep feature based semantic image retrieval. Neural Processing Letters (pp. 2225-2248). https://doi.org/10.1007/s11063-022-11079-y.

[7] Xiao, S. Z. (2023). Multi-dimensional frequency dynamic convolution with confident mean teacher for sound event detection. International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1-5). https://doi.org/10.1109/ICASSP49357.2023.10096306.

[8] Yang, R. H. (2024). Data Imputation by Pursuing Better Classification: A Supervised Kernel-Based Method. arXiv.

[9] Wu, H. L. (2022). Semi-supervised segmentation of echocardiography videos via noise-resilient spatiotemporal semantic calibration and fusion. Medical Image Analysis. Elseiver. https://doi.org/10.1016/j.media.2022.102397.

[10] Hoi, S. C. (2007). Learning nonparametric kernel matrices from pairwise constraints. 4th international conference on Machine learning (pp. 361-368). ACM. https://doi.org/10.1145/1273496.1273542.

[11] Yeung, D. Y. (2007). A kernel approach for semisupervised metric learning. Transactions on Neural Networks (pp. 141-149). IEEE. https://doi.org/10.1109/TNN.2006.883723.

[12] Woo, S. J. (2024). Sketch-based video object localization. Winter Conference on Applications of Computer Vision (pp. 8480-8489). https://doi.org/10.1109/WACV57701.2024.00829.

[13] Jagtap, S. &. (2024). Object-based image retrieval and detection for surveillance video. International Journal of Electrical and Computer Engineering (IJECE) (pp. 4343-4351). IJECE. https://doi.org/10.11591/ijece.v14i4.pp4343-4351.

[14] Kim, D. L. (2024). MOVES: Motion-Oriented VidEo Sampling for Natural Language-Based Vehicle Retrieval. International Conference on Advanced Video and Signal Based Surveillance (AVSS) (pp. 1-7). https://doi.org/10.1109/AVSS61716.2024.10672583.

[15] Mishra, P. K. (2024). Skeletal video anomaly detection using deep learning: Survey, challenges, and future directions. Transactions on Emerging Topics in Computational Intelligence. IEEE. https://doi.org/10.1109/TETCI.2024.3358103.

[16] Adly, A. S. (2022). Development of an Effective Bootleg Videos Retrieval System as a Part of Content-Based Video Search Engine. International Journal of Computing Journal, 214-227. https://doi.org/10.47839/ijc.21.2.2590.

[17] Sathiyaprasad, B. (2023). Ontology-based video retrieval using modified classification technique by learning in smart surveillance applications. International Journal of Cognitive Computing in Engineering, 55-64. https://doi.org/10.1016/j.ijcce.2023.02.003.

[18] Ali, A. S. (2022). Video and text matching with conditioned embeddings. CVF winter conference on applications of computer vision (pp. 1565-1574). IEEE. https://doi.org/10.1109/WACV51458.2022.00055.

[19] Gao, M. H. (2023). Video object segmentation using point-based memory network. Pattern Recognition. Elsevier. https://doi.org/10.1016/j.patcog.2022.109073.

[20] Kulis, B. (2013). Metric learning: A survey. Foundations and Trends® in Machine Learning, 5(4), 287-364. https://doi.org/10.1561/2200000019.

[21] Hoffer, E., & Ailon, N. (2015, October). Deep metric learning using triplet network. In International workshop on similarity-based pattern recognition (pp. 84-92). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-24261-3_7.

[22] Zhang, G. Z. (2022). Probability Loop Closure Detection with Fisher Kernel Framework for Visual SLAM. International Conference of Pioneering Computer Scientists, Engineers and Educators (pp. 219-239). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-5194-7_17.

[23] Perronnin, F. &. (2007). Fisher kernels on visual vocabularies for image categorization. Conference on computer vision and pattern recognition (pp. 1-8). IEEE. https://doi.org/10.1109/CVPR.2007.383266.

[24] Zhou, S. K. (2004). Trace and determinant kernels between matrices. NIPS.

[25] Salihu, A. &. (2021). Chio's-like method for calculating the rectangular (non-square) determinants: Computer algorithm interpretation and comparison. European Journal of Pure and Applied Mathematics, European Journal of Pure and Applied Mathematics. https://doi.org/10.29020/nybg.ejpam.v14i2.3920.

[26] Salihu, A. S. (2023). Modified Chios-Like Method for Rectangular Determinant Calculations. Advanced Mathematical Models & Applications, Advanced Mathematical Models & Applications.

[27] Lafferty, R. K. (2002). Diffusion kernels on graphs and other discrete input spaces. ICML. ICML.