Multi-modal Interactive Video Retrieval with Temporal Queries

Keyword: Video Browser Showdown, Interactive video retrieval, Content-based retrieval

S.Heller et al.

Ngày 31 tháng 10 năm 2022

Purpose/Output

- Trình bày phiên bản Vitrivr tham gia Video Browser Showdown (VBS) 2022. Vitrivr đã hỗ trợ một loạt các phương thức truy vấn, chẳng hạn như phác thảo màu và ngữ nghĩa, OCR, ASR và text embedding.
- Trong bài báo này, giới thiệu ngắn gọn hệ thống, sau đó mô tả cách tiếp cận mới đối với các truy vấn chỉ định bối cảnh tạm thời, ý tưởng cho các bản phác thảo dựa trên màu sắc trong môi trường truy xuất có tính cạnh tranh và cách tiếp cận mới lạ đối với các pose-based queries.

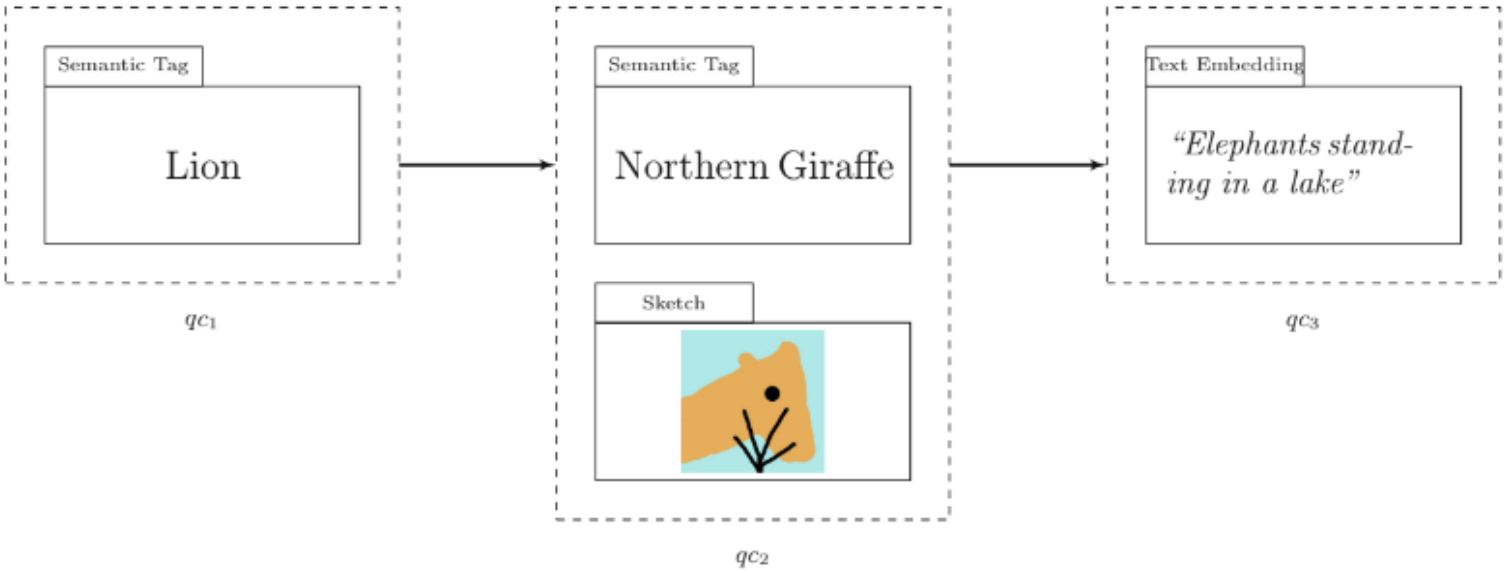
Data source/ input

- Bộ dataset V3C được sử dụng cả 2 phần với khoảng 2300 giờ video.

Partition	V3C1	V3C2	V3C3	Total
File Size (videos)	1.3TB	1.6TB	1.8TB	4.8TB
File Size (total)	2.4TB	3.0TB	3.3TB	8.7TB
Number of Videos	7'475	9'760	11'215	28'450
Combined Video Duration	1'000 hours, 23 minutes, 50 seconds	1'300 hours, 52 minutes, 48 seconds	1'500 hours, 8 minutes, 57 seconds	3801 hours, 25 minutes, 35 seconds
Mean Video Duration	8 minutes, 2 seconds	7 minutes, 59 seconds	8 minutes, 1 seconds	8 minutes, 1 seconds
Number of Segments	1'082'659	1'425'454	1'635'580	4'143'693

Hình 1. Tổng quan về các phân vùng của V3C

Method



Hình 2. Truy vấn tạm thời đa phương thức bao gồm ba container truy vấn (Dashed, QC1, QC2 và QC3)

- Đây là một truy vấn được xây dựng cho các bức ảnh của một con Lion đầu tiên (thẻ ngữ nghĩa), tiếp theo là một con hươu cao cổ (thẻ ngữ nghĩa) + phần (phác thảo), được kết luận "Những con voi đứng trong hồ" (text embedding).
- Thứ tự tạm thời được đưa ra thông qua thứ tự của các thùng chứa truy vấn.

1. Temporal Queries

- Một truy vấn tạm thời (**Temporal Query**) (Hình 2) bao gồm nhiều truy vấn con giống nhau, được sắp xếp theo thứ tự của nhiều phương thức (có thể khác nhau) và khoảng cách thời gian giữa các truy vấn con.
- Kết quả của các truy vấn con sau đó được tổng hợp và cho điểm theo thứ tự của các vùng chứa truy vấn và khoảng cách thời gian được chỉ định.
- Đối với mục đích ký hiệu, xác định một video V được phân đoạn thành danh sách các phân đoạn $S = (s_1, s_2, ...s_m)$.
- Đưa ra truy vấn tạm thời TQ và danh sách các vùng chứa truy vấn $TQ = (qc_1, ..., qc_n)$, với mỗi vùng chứa truy vấn có thể là 1 truy vấn đa phương thức, mỗi vùng chứa truy vấn được thực thi riêng biệt.
- Sau đó, kết quả là một tập hợp RS, $RS_i = ((s_f, s_g, ..., s_l), qc_i)$, với một phân đoạn có thể xuất hiện trong nhiều phần tử của tập kết quả nếu nó là một kết quả phù hợp cho nhiều vùng chứa.

1. Temporal Queries

- Quá trình tổng hợp tạo ra các chuỗi phân đoạn, trong đó các chuỗi có nhiều phân đoạn kết quả được chấm điểm ngày càng cao từ nhiều vùng chứa truy vấn được ưu tiên cho từng đối tượng.
- Chuỗi tạm thời TS được định nghĩa là danh sách có thứ tự gồm các bộ chứa phân đoạn $TS = ((s_i, qc_a), (s_j, qc_b), ..., (s_z, qc_n))$.
- Mô tả thuật toán tính điểm được **Vitrivr** sử dụng tại đây:
  - Đối với mỗi tuple (s_i, qc_a) trong list result được làm phẳng, các ứng cử viên trình tự thời gian được xây dựng. Các phân đoạn được nối vào một trình tự thời gian nếu chúng tuân theo thứ tự do người dùng chỉ định, tức là đối với TS = (s_1, qc_i), chỉ các bộ giá trị (s_x, qc_j) với j>i mới được xem xét.
  - Điểm được tổng hợp trong một chuỗi TS (với phần tử đầu tiên là (s_i, qc_a)) với một hàm decay. Đối với sự chênh lệch thời gian t giữa phần tử, tất cả các đoạn ngoại trừ đoạn đầu tiên (TS (s_i, qc_a)) có điểm được nhân với adj(t) = $e^{-|l*(t-m)|}$, với l là hình phạt của không ở khoảng cách do người dùng chỉ định (sử dụng l = 0,1) và m $\leq$ 0 là thời gian đến phân đoạn tiếp theo như được xác định bởi truy vấn tạm thời.
  - Sau khi chuẩn hóa điểm theo số lượng truy vấn phụ (trong ví dụ là 3), các chuỗi có điểm cao nhất sẽ được chọn.

2. Color Sketches

Phương pháp này được sử dụng trong lĩnh vực ảnh Y sinh và ảnh vệ tinh viễn thám (image classification, image denoising,...).

- $I \in R^{MxM}$ biểu diễn một ảnh có kích thước M $\times$ M.
- Kết quả của Low-rank và sparse decomposition của **I** là 2 ma trận $L \in R^{MxM}$ và $S \in R^{MxM}$. Low-rank component: phản ánh cấu trúc nguyên tắc của I và sparse component: đại diện cho các thành phần nổi bật của I, được biểu diễn:

Partition	V3C1	V3C2	V3C3	Total
File Size (videos)	1.3TB	1.6TB	1.8TB	4.8TB
File Size (total)	2.4TB	3.0TB	3.3TB	8.7TB
Number of Videos	7'475	9'760	11'215	28'450
Combined Video Duration	1'000 hours, 23 minutes, 50 seconds	1'300 hours, 52 minutes, 48 seconds	1'500 hours, 8 minutes, 57 seconds	3801 hours, 25 minutes, 35 seconds
Mean Video Duration	8 minutes, 2 seconds	7 minutes, 59 seconds	8 minutes, 1 seconds	8 minutes, 1 seconds
Number of Segments	1'082'659	1'425'454	1'635'580	4'143'693

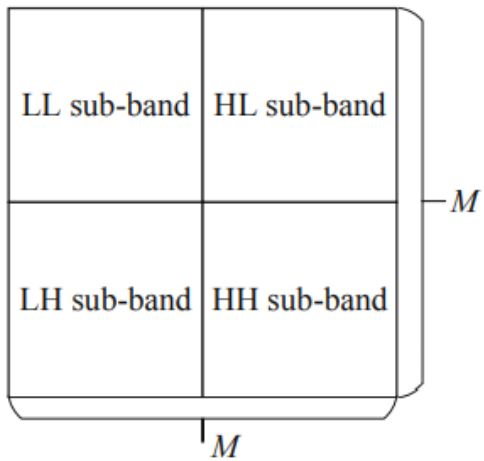
Trong đó: $||\cdot||_0$ biểu thị l_0 norm của ma trận và λ. Bài toán tối thiểu hóa lỗi bị ràng buộc có thể được giải quyết bằng một phương pháp là hệ số nhân Lagrange tăng cường không chính xác (IALM - inexact augmented Lagrange multipliers).

(a)low-rank component

(b)sparse component

3. DWT

- DWT là một kỹ thuật xử lý ảnh và một single-level 2-D DWT có thể phân tách hình ảnh đầu vào thành bốn sub-band như hình dưới LL sub-band, LH sub-band, HL sub-band và HH sub-band.
- **Trong đó:** LL sub-band có tần số thấp và 3 sub-band khác có tần số cao.



- Chọn LL sub-band để biểu diễn để lấy mã ngắn dựa trên hai chú ý:
  - Một số hoạt động bảo toàn nội dung, như lọc low-pass và ô nhiễm tiếng ồn, có ảnh hưởng nhẹ đến hệ số dải tần phụ tần số thấp.
  - Hệ số DWT trong LL sub-band là hệ số xấp xỉ của ma trận đặc trưng và chỉ là hệ số phần tư dành riêng.

$\Rightarrow$ *giúp hàm băm dẫn xuất thành một đoạn mã ngắn với các tính năng dựa trên nội dung.*

4. Minh họa thuật toán

- **Step 1:** Một video đầu vào được chuyển đổi thành normalized video V_{norm}^{MxMxM} với 2 lần resampling, sau đó được tạo thêm V_{rank}^{MxMxM} bằng cách sử dụng chuỗi vị trí tương đối tham chiếu đến một chuỗi hỗn loạn **g**.
- **Step 2:** Áp dụng low-rank và sparse decomposition cho từng frame của V_{rank}^{MxMxM} và trích xuất ma trận cấp thấp **L** dưới dạng content-based feature matrices cho từng frame.
- **Step 3:** Áp dụng 2D-DWT cho từng frame của video, và thu thập giá trị trung bình của tất cả hệ số LL sub-band để tạo thành 1 short string **s** (có tổng số M phần tử). Tiếp theo, các phần tử của s được lượng tử hóa theo quy tắc dưới đây:

$$h(i) = \begin{cases} 1 & \text{If s(i)} < \text{s(i+1)} \\ 0 & \text{Otherwise,} \end{cases} \qquad i = 1, 2, ..., M,$$

Trong đó: s(i) là phần tử thứ i của **s** $\rightarrow$ hàm băm được tạo như sau:

$$\text{h} = [\text{h(1)},\text{h(2)},...,\text{h(M)}].$$

5. Hash Similarity

- Vì hàm băm của chúng ta là một chuỗi bit, nên Hamming distance được lấy để đánh giá mức độ giống nhau giữa hai hàm băm.
- Giả sử V_1 và V_2 đại diện cho hai video khác nhau và h_1 và h_2 lần lượt đại diện cho hai hàm băm tương ứng của chúng. Khoảng cách giữa hai hàm băm có thể được tính theo phương trình sau:

$$d_H(h_1, h_2) = \sum_{i=1}^{M} |h_1(i) - h_2(i)|$$

Trong đó: phần tử thứ i của h_1 và h_2 lần lượt là h_1(i) và h_2(i), và M đại diện cho độ dài của hàm băm. Hamming distance càng nhỏ thì 2 video càng giống nhau.

  - Ngưỡng T xác định trước có thể được sử dụng để đánh giá mức độ giống nhau của hai video. Nói cách khác, hai video có thể được đánh giá là video giống nhau về mặt hình ảnh khi $d_H \leq T$. Nếu không, hai video có thể được coi là video khác nhau về hình ảnh.

Results

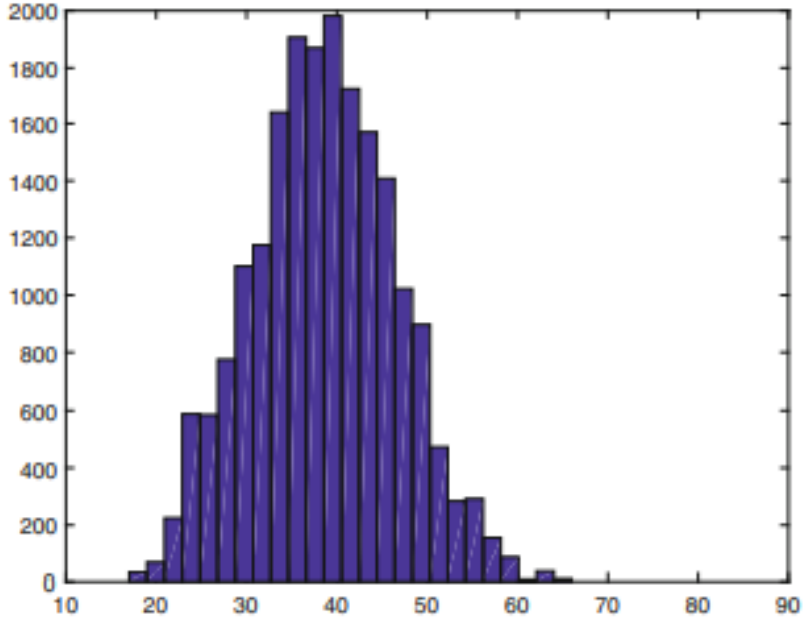
1. Robustness

- Video chuẩn hóa ngẫu nhiên có kích thước $128 \times 128 \times 128$
- Một bộ lọc Gaussian low-pass với kernel size là 1×20. Do đó, độ dài hàm băm là 128 bits.
- Chi tiết cách cài đặt hoạt động như sau:
  - Điều chỉnh độ sáng với thang đo của Photoshop là 20, 15, ..., 20
  - Bộ lọc Gaussian low-pass 3×3 với Độ lệch chuẩn là 0.1, 0.2, ..., 1
  - Salt và pepper noise của mật độ là 0.001, 0.002, ..., 0.01
  - AWGN của tỷ số nhiễu tín hiệu là 1, 2, ..., 6
  - Tốc độ bit cho MPEG-2 là 100, 200, ..., 1000
  - Số lượng khung hình drop là 5, 10, ..., 20
  - Tỷ lệ mở rộng khung hình là 0.8, 0.85, ..., 1.2 và góc quay là 2, 1, ..., 2
- Sau khi thử nghiệm, rút ra được hầu hết các giá trị trung bình đều nhỏ hơn 6 ngoại trừ hoạt động của Rotation và giá trị trung bình của Rotation không lớn hơn 15. Điều này ngụ ý rằng ngưỡng được chọn là $T = 15$

Results

2. Discrimination

- Trích xuất các hàm băm cho mỗi video và tính toán khoảng cách Hamming giữa hàm băm ban đầu của nó và các hàm băm của 199 video khác. Do đó, cặp khoảng cách $200 \times 199/2 = 19900$ Hamming thu được.
- Hình dưới cho thấy khoảng cách Hamming tối thiểu giữa hai video khác nhau là 16 và tối đa là 89.
- Giá trị trung bình và độ lệch chuẩn lần lượt là 38.52 và 8.06
- Đặt ngưỡng là 35, thì có 0,003% các cặp video khác nhau bị coi là các video giống nhau về mặt hình ảnh.

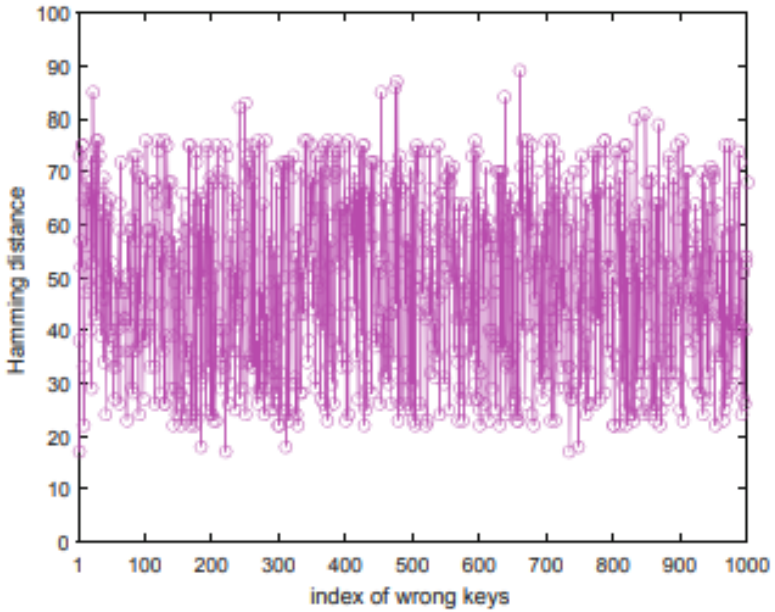


Results

3. Key Dependence

Để xác thực bảo mật của hàm băm, lấy video của phần Robustness và discrimination test video làm video kiểm tra.

- Đặt các nhóm key bí mật khác nhau k_1 và k_2 của logistic map để tạo các hàm băm của mọi video thử nghiệm và tính toán khoảng cách Hamming giữa các hàm băm của video thử nghiệm.
- Sau khi thử nghiệm hơn 1000 nhóm key khác nhau (k_1 và k_2) thì thấy khoảng cách Hamming của các khóa sai phân bố từ 17 đến 89, lớn hơn ngưỡng T = 15.



$\Rightarrow$ **phép băm phụ thuộc vào key-dependence.**