



3D Average Common Submatrix Measure

Federica Franco, Alessia Amelio^(✉), and Sergio Greco

University of Calabria, DIMES, Via Pietro Bucci 44, Rende, CS, Italy
{a.amelio,s.greco}@dimes.unical.it

Abstract. This paper introduces a new measure for computing the similarity among 3D objects as the average volume of the largest sub-cubes matching in the objects. The match is approximate and only verified within a neighbourhood from the position of the sub-cubes. Preliminary tests performed on random and synthetic datasets prove the efficacy of the similarity measure in capturing the visual similarity among the 3D objects and a reduction in the execution time when the neighbourhood is considered.

Keywords: 3D objects · Pattern matching · Image similarity

1 Introduction

In multimedia retrieval, a similarity measure is used on a digital library of 2D images, 3D objects of medical imaging, and other data for finding the k -most similar items to a query item. The problem of automatically computing the similarity in the context of the digital libraries is still challenging, due to the gap between the human perceived appearance of the data and their features which are captured by the machine (semantic gap) [10].

In the last years, different approaches based on the concept of *common sub-matrix* have flourished for similarity computation in image retrieval tasks, whose advantages versus competing methods have been deeply investigated in the literature [1, 2, 4]. Specifically, [3] introduced the *Average Common Submatrix* (ACSM) measure for computing the similarity of images represented as matrices. ACSM computes the similarity between two images as the average area of the largest sub-matrices matching in the two images. A more efficient version of ACSM was proposed in [4] using a tree data structure for indexing the sub-matrices. To avoid the time and space requirements of constructing the tree index, [1] introduced an approximate version of ACSM, where the match between two sub-matrices is only verified for a portion of pixels at regular intervals, with any additional data structure. Finally, [2] framed ACSM within a new framework laying the theoretical foundations of the *common sub-matrix* concept and extending it. It noticeably reduced the temporal cost of computing the similarity without using additional data structures or omitting the match of a portion of pixels.

The *common sub-matrix*-based methods have been introduced for 2D objects so far, thus making not possible their usage in multidimensional contexts, including video retrieval [6] where the semantic gap still remains an open problem, medical image registration [11], and extensions in the context of multidimensional network analysis [7–9]. Especially in medical image registration, the correct selection of a similarity measure is a key aspect for reliably monitoring the time evolution of a patient’s state for critical pathologies (e.g. stroke lesions) [5].

In this paper, we focus on extending the concept of *common sub-matrix* revisited in [2] for 3D objects, which is a brand new idea in the state-of-the-art. In particular, we provide the following contributions: (i) the ACSM measure is extended for objects in three dimensions, (ii) the match is verified in a neighbourhood of the objects, (iii) the match is approximate and computed in terms of the Hamming similarity extended in three dimensions. Accordingly, we aim to provide in progress results of our research.

2 The Proposed Method

In this section, we present 3D Average Common Submatrix (3D-ACSM), a new similarity measure for 3D objects extending the revisited version of ACSM introduced in [2]. Specifically, we use this new measure for computing the similarity between data in three dimensions, which can be represented as parallelepipeds.

2.1 3D Average Common Submatrix

Intuitively, let A , B and C be three parallelepipeds of volume $v = n \times p \times m$ defined on the same alphabet Σ . Then, A can be considered as more similar to B than to C if the average volume of sub-cubes of A matching approximately in B is larger than the same average volume in C [5].

More specifically, for each position (i, j, z) of A , 3D-ACSM finds the largest sub-cube in A starting at that position and with edge length greater than or equal to α approximately matching with a sub-cube in B within an ε -neighbourhood from (i, j, z) . Restricting the area of search makes the similarity evaluation more efficient while it preserves the accuracy. The parameter α represents the minimum edge length of the sub-cubes to consider for the match. The parameter ε is related to the size of the neighbourhood with centre in (i, j, z) where a sub-cube can be found. Two sub-cubes approximately match to each other if the number of identical elements at the same positions exceeds or is equal to a given similarity threshold τ . From all aforementioned, 3D-ACSM between A and B is computed as follows:

$$S_\alpha(A, B) = \frac{1}{v} \sum_{i=\alpha}^n \sum_{j=\alpha}^p \sum_{z=\alpha}^m W(i, j, z), \quad \text{s.t. } W(i, j, z) \geq \alpha \quad (1)$$

where $W(i, j, z)$ is the volume of the largest sub-cube starting at position (i, j, z) in A of edge length greater than or equal to α approximately matching a sub-cube of B within an ε -neighbourhood from (i, j, z) in B .

Algorithm 1. 3D-ACSM algorithm

```

1: function compute3D-ACSM( $A, B, \alpha, \tau, \varepsilon$ )
2:    $W_\alpha(A, B) = 0; W_\alpha(A, A) = 0; d = 0; k = 0;$ 
3:   for  $i = \alpha \dots n$  do
4:     for  $j = \alpha \dots p$  do
5:       for  $z = \alpha \dots m$  do
6:          $k = \min\{i, j, z\};$ 
7:          $W_\alpha(A, A) = W_\alpha(A, A) + k^3;$ 
8:          $found = false;$ 
9:         while  $(k \geq \alpha \wedge \neg found)$  do
10:          if  $matchApproximate(A_{i,j,z}^k, B, \varepsilon) \geq \tau$  then
11:             $W_\alpha(A, B) = W_\alpha(A, B) + k^3;$ 
12:             $found = true;$ 
13:          end if
14:           $k = k - 1;$ 
15:        end while
16:      end for
17:    end for
18:  end for
19:  return  $SN_\alpha(A, B) = W_\alpha(A, B)/W_\alpha(A, A);$ 
20: end function

```

A value $SN_\alpha(A, B)$ between 0 (minimum similarity) and 1 (maximum similarity) can be obtained as the ratio of $S_\alpha(A, B)$ with $S_\alpha(A, A)$, which is the similarity of A with itself.

2.2 Algorithm

The Algorithm 1 shows the steps for computing the 3D-ACSM similarity between two parallelepipeds A and B , given the minimum edge length of the sub-cube α , the similarity threshold τ and the ε size of the neighborhood. We denote with $A_{i,j,z}^k$ the sub-cube at position (i, j, z) in A whose edge length is k .

Firstly, for each position (i, j, z) (lines 3–5), the algorithm computes the ideal matching at that position. This value, denoted by $W_\alpha(A, A)$, is given by the volume of the maximal sub-cube matching itself, whose value is k^3 (line 7).

The global match of parallelepipeds A and B is denoted by $W_\alpha(A, B)$ and is computed by accumulating at each step the values of the matching between sub-cubes of A and B , respectively. In particular, at each step, the algorithm looks for the maximal sub-cube $A_{i,j,z}^k$ in A , with size k and start position (i, j, z) , having a match with a sub-cube of B , with size k and start position in a neighbourhood of position (i, j, z) of size ε (lines 8–15). To determine the maximal size of the sub-cube k , it starts with a maximal size $k = \min\{i, j, z\}$ and, at each step, if the matching value (computed by function *matchApproximate*) is not greater than or equal to a given threshold τ , decrements the size of the sub-cube k . If a feasible match is not found, the position (i, j, z) gives no contribution to $W_\alpha(A, B)$.

Finally, the ratio of $W_\alpha(A, B)$ with $W_\alpha(A, A)$ is returned (line 19).

The core of Algorithm 1 is given by the method *matchApproximate*, which finds the match of a sub-cube $A_{i,j,z}^k$ in B . The match is verified in a neighbourhood with centre at (i, j, z) in B and extent of ε hops along the three dimensions (where possible). Also, the match is approximate and resembles to the notion

Table 1. CPU time of Algorithm 1 on the first dataset with $m = \{15, 20\}$. For $\varepsilon = \{1, 3\}$, Δ_α is the average absolute time difference between $\alpha = 3$ and 1, and $\alpha = 5$ and 3. For $\alpha = \{1, 3, 5\}$, Δ_ε is the average absolute time difference between $\varepsilon = 3$ and 1

m	15						20					
ε	1			3			1			3		
$(n = p)/\alpha$	1	3	5	1	3	5	1	3	5	1	3	5
4	0.02	0.01	—	0.06	0.03	—	0.02	0.01	—	0.08	0.06	—
16	0.44	0.24	0.19	1.06	1.02	0.92	0.34	0.37	0.34	1.67	1.62	1.47
64	4.28	3.41	4.07	8.22	8.13	7.92	6.91	6.19	6.23	16.24	17.02	15.32
128	15.40	16.44	16.35	24.30	23.98	22.86	41.32	31.54	31.57	50.88	49.09	48.60
256	68.20	65.67	64.39	78.49	77.12	74.93	146.58	140.04	129.75	161.07	158.35	151.89
512	273.83	261.29	255.56	278.01	269.51	262.20	565.34	549.59	538.24	570.80	554.49	551.86
Δ_α	—	2.86	1.56	—	1.73	2.19	—	5.47	4.35	—	3.61	2.29
Δ_ε	4.66	5.46	5.65	—	—	—	6.70	8.82	12.60	—	—	—

of Hamming similarity [2] and its direct extension to 3D objects. Specifically, a match between two sub-cubes exists if their Hamming similarity (number of identical elements at the same positions) is greater than or equal to a threshold parameter τ , $0 \leq \tau \leq 1$.

3 Experiments

Preliminary experiments have been conducted for testing the performances of 3D-ACSM on two different datasets. The experiments have been performed in Matlab R2017a on a computer laptop with CPU Quad-Core 2.30 GHz, 8 GB RAM and Windows 7 (64 bit) operating system.

The first dataset includes a set of randomly generated numerical parallelepipeds of size $n = p$ varying from 4 to 512 and depth m varying from 5 to 20 with steps of 5. The size of the alphabet $|\Sigma|$ is set to 2. This dataset is used for testing the CPU time (in seconds) of Algorithm 1 at different values of the input parameters $\varepsilon = \{1, 3\}$, $\alpha = \{1, 3, 5\}$, and $\tau = 0.5$.

The second dataset includes a set of synthetic parallelepipeds each composed of three greyscale illusory and colour miscellaneous images of size 128×128 ($n = p = 128$ and $m = 3$). The used images are extracted from the online database of the Computer Vision Group, University of Granada¹. They represent synthetic objects with frequent patterns and shapes useful for assessing the effectiveness of the similarity measure. The 3D-ACSM similarity is computed for each pair of parallelepipeds at different values of the input parameters $\varepsilon = \{1, 2, 3\}$, $\alpha = \{1, 4, 8\}$, and $\tau = \{0.5, 0.6, 0.7\}$. In the end, the obtained 3D-ACSM similarity is compared with the visual similarity, which is the similarity as perceived by the human eye.

¹ <http://decsai.ugr.es/cvg/dbimagenes/>.

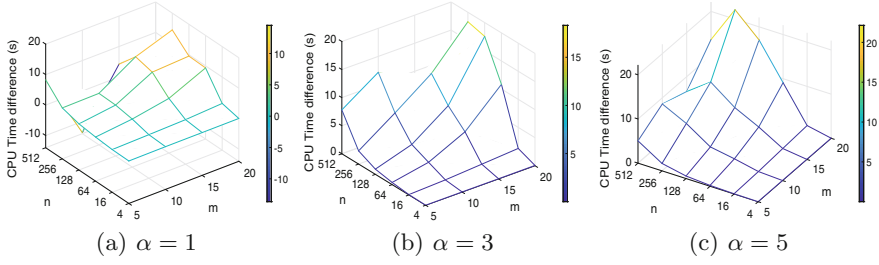


Fig. 1. CPU time difference between $\varepsilon = 3$ and 1

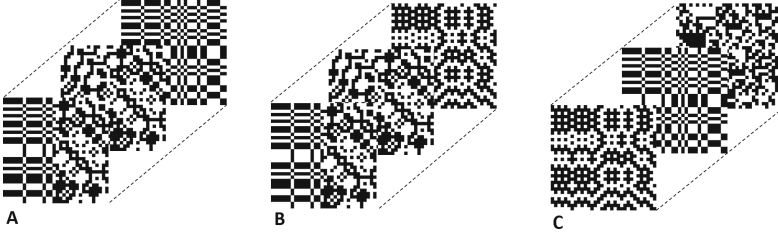


Fig. 2. Sample parallelepipeds A , B and C from the second dataset. The size n, p is 128 with depth m equal to 3

Table 2. 3D-ACSM similarity computed among the sample parallelepipeds in Fig. 2

τ	ε/α	1		4		8	
		$SN_1(A, B)$	$SN_1(A, C)$	$SN_4(A, B)$	$SN_4(A, C)$	$SN_8(A, B)$	$SN_8(A, C)$
0.5	1	0.999	0.963	1.000	0.964	1.000	0.946
	2	1.000	0.994	1.000	0.994	1.000	0.988
	3	1.000	0.999	1.000	0.999	1.000	0.999
0.6	1	0.999	0.853	0.999	0.842	1.000	0.864
	2	0.999	0.931	0.999	0.924	1.000	0.930
	3	0.999	0.972	0.999	0.969	1.000	0.973
0.7	1	0.978	0.700	0.977	0.680	0.966	0.615
	2	0.991	0.819	0.990	0.807	0.985	0.739
	3	0.995	0.870	0.995	0.863	0.991	0.792

4 Results and Discussion

Table 1 reports the CPU time of Algorithm 1 on the first dataset of randomly generated numerical parallelepipeds with $m = \{15, 20\}$. It is worth noting that changing ε determines a higher time variation than changing α (the average absolute time difference Δ_ε is higher than Δ_α), which confirms the need of restricting the match within a neighbourhood.

Also, Fig. 1 shows the CPU time difference between $\varepsilon = 3$ and 1 for $\alpha = \{1, 3, 5\}$. It is worth noting as the time difference increases when the 3D objects become larger (higher size n, p and depth m) regardless of the α value. It confirms that restricting the neighbourhood can be time-saving especially for large 3D objects for which the CPU time has the tendency to normally increase (see Table 1).

Finally, Table 2 reports the 3D-ACSM similarity values for three sample parallelepipeds from the second dataset (see Fig. 2). From a visual inspection, it can be observed that A is more similar to B than to C , since the first and second slices of A and B are exactly the same. From Table 2, it is worth noting that 3D-ACSM similarity is compliant with the human perception of similarity, since $SN_\alpha(A, B)$ is higher than $SN_\alpha(A, C)$ for all parameters' combinations. It confirms that reducing the neighbourhood size still obtains reliable results with the advantage of reducing the execution time.

5 Conclusions

This paper introduced the 3D-ACSM measure as an extension of the ACSM similarity measure for 3D objects. Preliminary experiments were presented as a part of an ongoing work employing 3D-ACSM in the medical context of CT images. In such a context, an efficient 3D similarity measure is required in the registration process for monitoring the correct evolution of a patient's state [5]. Also, a comparison of 3D-ACSM similarity is being performed with other state-of-the-art measures. Finally, the Matlab code of the project is going to be available in open source for the scientific community.

Acknowledgments. This work was performed during the student internship of the first author at DIMES, University of Calabria.

References

1. Amelio, A.: Approximate matching in ACSM dissimilarity measure. *Proc. Comput. Sci.* **96**, 1479–1488 (2016). Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 20th International Conference KES-2016
2. Amelio, A.: A new axiomatic methodology for the image similarity. *Appl. Soft Comput.* **81**, 105474 (2019)
3. Amelio, A., Pizzuti, C.: Average common submatrix: a new image distance measure. In: Petrosino, A. (ed.) *ICIAP 2013*. LNCS, vol. 8156, pp. 170–180. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41181-6_18
4. Amelio, A., Pizzuti, C.: A patch-based measure for image dissimilarity. *Neurocomputing* **171**, 362–378 (2016)
5. Amelio, L., Amelio, A.: CT image registration in acute stroke monitoring. In: 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1527–1532, May 2018
6. Gurrin, C.: Content-based video retrieval. In: Liu, L., Özsu, M.T. (eds.) *Encyclopedia of Database Systems*, pp. 466–473. Springer, Boston (2009). <https://doi.org/10.1007/978-0-387-39940-9>

7. Hu, W., Xie, N., Li, L., Zeng, X., Maybank, S.: A survey on visual content-based video indexing and retrieval. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **41**(6), 797–819 (2011)
8. Kankanhalli, M.S., Rui, Y.: Application potential of multimedia information retrieval. *Proc. IEEE* **96**(4), 712–720 (2008)
9. Li, Y., Gu, C., Dullien, T., Vinyals, O., Kohli, P.: Graph matching networks for learning the similarity of graph structured objects. *CoRR* abs/1904.12787 (2019). <http://arxiv.org/abs/1904.12787>
10. Patel, T., Gandhi, S.: A survey on context based similarity techniques for image retrieval. In: 2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), pp. 219–223, February 2017
11. Roche, A., Malandain, G., Ayache, N., Prima, S.: Towards a better comprehension of similarity measures used in medical image registration. In: Taylor, C., Colchester, A. (eds.) *MICCAI 1999*. LNCS, vol. 1679, pp. 555–566. Springer, Heidelberg (1999). https://doi.org/10.1007/10704282_60