

LETTER

Fast and Simple 2D Shape Retrieval Using Discrete Shock Graph

Solima KHANAM[†], Seok-Woo JANG^{††}, Nonmembers, and Woojin PAIK^{†a)}, Member

SUMMARY In this letter, we propose an effective method to retrieve images from a 2D shape image database using discrete shock graphs combined with an adaptive selection algorithm. Experimental results show that our method is more accurate and fast than conventional approaches and reduces computational complexity.

key words: medial axes, shock graph, deform cost, visual transformation, computational complexity

1. Introduction

Object retrieval using shape information is one of the most important topics in CBIR (content-based image retrieval) areas. The visual instability and computational complexity are two significant issues for shape-based retrieval. Researchers have tried extensively to decrease the time taken for shape-based object retrieval without a loss of visual stability in many real-life applications like crime prevention and medical diagnoses. The problem of visual instability is solved in the work of Sebastian et al. [1] by using shock-graph-based skeletal approach, which allowed them to achieve 100% accuracy for 216 images of the Kimia database. Because of joint curve matching, the total time complexity of [1] is $O(x^3y^3)$, where x and y are the number of nodes from query and database image. To reduce time complexity, a skeleton-based approach is proposed by Latecki's group [2]. According to them, instability occurs due to the junction points (or branch points) of the skeleton, and therefore, matching is only done with respect to end points. In contrast, Zaboli et al. [3] and Goh [4] reduced time complexity using branch points only. However, considering only branch or only end points is not always accurate, because some images, like those of a circle, may not have any end points. On the other hand, different images may have the same branch points. Therefore, we developed an adaptive weighing algorithm to choose branch or end points based on the situation [5]. This reduces the sample points and decreases running time. However, it does not yet help to reduce the time complexity of the retrieval.

In this letter, we propose an idea of a discrete shock graph-based shape retrieval which reduces time complexity $O(x^3y^3)$ to $O(xy)$. Also, combining our new approach with adaptive selection [5], we decrease the value of x (number of nodes), which further helps to reduce the running time in

the case of $O(xy)$.

2. Shock-Based Representation of a Shape

For the purpose of similarity measure, shape can be represented relying on two main approaches: one is *shape contour* and the other is *shape interiors*. The interior-based approaches show better performance than contour-based approaches in handling instability existing in an image database. *Skeleton-based* shape interior which consists of different nodes and links represent a shape intrinsically [2]. The skeleton-based approaches show superiority to the contour-based approaches by providing topological and geometrical information as well as showing robustness against visual transformations. *Medial axis* is one of the most important skeleton-based features that can be defined as the locus of the centers (called singularities or shock points) of maximal circles which touches the boundary at least at two points, as in Fig. 1. The points which touches the boundary are referred to as characteristic points (a and b in Fig. 1) and endowed with geometric and dynamic information [1].

Shock graph is an idea which arises from the concept of *medial axis* augmented with some additional dynamic properties. In a shape, shock graph dynamically interprets the path of a moving particle with associated direction and speed of flow [1]. For these additional properties, shock graph can capture a shape that is complete and unique. According to the type of tangency and the number of touching points on the boundary, a shock point can be of first, second, third or fourth order. The loci of all the shock points in Fig. 1 give the Blum's medial axis and also the idea of the

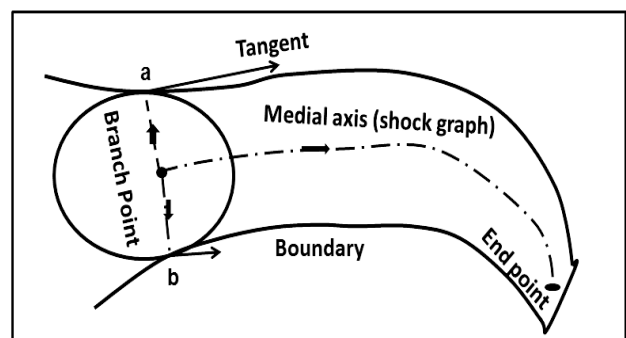


Fig. 1 Segmented shape showing bitangent circle, medial axis, branch point, end point and characteristic points (a, b).

Manuscript received June 14, 2011.

[†]The authors are with Konkuk University, Chungju, Korea.

^{††}The author is with Anyang University, Anyang, Korea.

a) E-mail: wjpaik@kku.ac.kr (Corresponding author)

DOI: 10.1587/transinf.E94.D.2059

whole shock graph [6].

The second and fourth order shocks are the generic cases of shock orders which are involved in occurring instability in shape retrieval [2], [3]. The second order shocks are the sources of flow while the fourth order shocks are termination points of flow [1], which represent branch and end points, respectively. These end and branch points will be used as shape descriptors in our proposed approach to represent and retrieve an image from a database. In our retrieval application, the shock points which constitute a shock graph of a query image are matched to the corresponding shock points of a database image. However, a shock graph descriptor makes any retrieval task complex. Our discrete shock graph-based approach solved the complexity in a simple way without a loss of accuracy. Adaptive weighing of the shock points [5] used by our approach selects the end points or branch points from the images, which reduces the complexity. The detailed description of the proposed shape retrieval is discussed in Sect. 3.

3. Shock Point-Based Shape Retrieval Scheme

At the starting of retrieval task, we adaptively select the nodes (shock points) of the images corresponding to the given query [5]. The time complexity of the selection algorithm is $O(x)$, where x is the number of nodes. For matching, we use shock point matching and edit operations in a discrete way. Instead of a joint curve (corresponding boundaries of the shock graph) idea [1], which is computationally complex, we will match the shock points of the query with those of a database image. In this part, matching cost between shock graphs is measured by discretizing the graph into branch or end nodes. Let two segments of shock graphs be discretized at sample nodes. These nodes can be considered as elements of a matrix (m by n). Let $C(i, j)$ and $d([k, j], [l, j])$ be the matching cost of graphs and segments, respectively. Therefore, the cost of shock point matching $C(i, j)$ will be

$$C(i, j) = \min_{k,l} [C(i-k, j-l) + d([i-k, i], [j-l, j])] \quad (1)$$

Here, the number of sub-problems solved by a dynamic programming algorithm is $x \times y$ where x and y are the number of shock points (nodes) between graphs from the query and data image. Therefore, computing the $C(i, j)$ needs complexity of order $O(xy)$.

To deal with a visual transformation, we will consider deform cost resulting from the edit operation like contract and splice. Contract is related to removing the branch points and splice cost is related to removing the end points in Fig. 2 (i) and Fig. 2 (ii), respectively [1]. Thus, contract cost is the difference between the total number of branch nodes from the query and data images. Similarly, splice cost is the difference between the total number of end points from query and data images. Let BP_q and EP_q be the branch point difference and end point difference, respectively. Therefore, the deform cost will be either contract cost or splice cost.

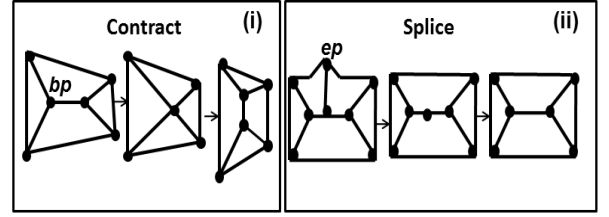


Fig. 2 The figures show edit operations: (i) contract removes branch point, bp and (ii) splice removes end point, ep . Finally, the interior point is merged.

Calculating splice or contract cost gives rise to the complexity of $\max\{O(x), O(y)\}$. Finally, the total cost will be the sum of the shock point matching cost and deform cost. The total matching cost, M_q can be written as

$$M_q = C(i, j) + D_q \quad (2)$$

where $C(i, j)$, D_q and q represent matching cost, deform cost and the number of images, respectively. By applying the algorithm 1, we can calculate the matching cost step by step.

Algorithm 1

A. Adaptive Shock Point(SP) Selection

1. Input 1: a query image I_i
2. Input 2: database images ($I_1, I_2, I_3, \dots, I_q$)
3. Find end point difference between query and database, EP_q
4. Find branch point difference between query and database, BP_q
5. For ($r=1, 2, 3, \dots, q$) do
6. IF ($EP_r > BP_r$) then
7. Select end point, ep as shock points
8. Else ($EP_r < BP_r$) then
9. Select branch point, bp as shock points
10. End if
11. End for r
12. End selection: $SP = bp$ or $SP = ep$

B. Distance Measure for Retrieval

1. Input 1: shock points, SP (for query I_i)
2. Input 2: shock points, SP_q (for database images $I_1, I_2, I_3, \dots, I_q$)
3. Find the minimum N between SP and SP_q
[N is an m by n matrix. Shock points are considered as elements of a matrix of m rows and n columns]
4. For ($i=1, 2, 3, \dots, m$) do
5. For ($j=1, 2, 3, \dots, n$) do
6. If ($SP(i, j) = SP_q(i, j)$) then
7. Matching cost, $C(i, j) = 0$
8. Else calculate a cost or distance,
 $C(i, j) = \min_{k,l} [C(i-k, j-l) + d([i-k, i], [j-l, j])]$
9. End if
10. End for j
11. End for i
12. Calculate deform cost, D_q (splice cost or contract cost)
13. Find the total matching cost, $M_q = C(i, j) + D_q$
14. Retrieve images according to ascending order of, M_q

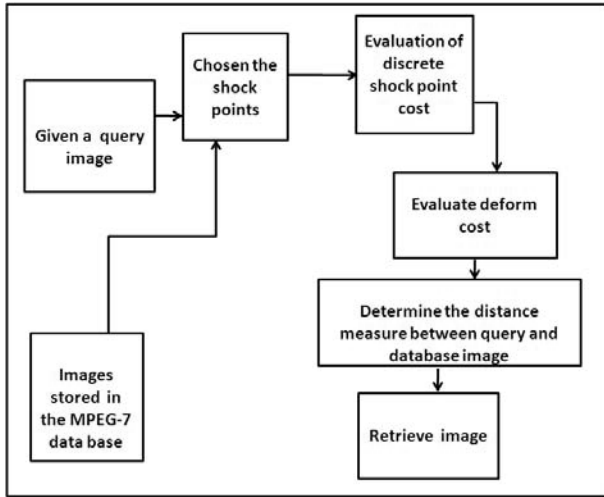


Fig. 3 Architecture of the proposed scheme.

Thus, the proposed retrieval measure has two steps: selection of shock points and cost measurement for retrieval. Figure 3 depicts the architecture of the proposed method.

4. Result and Discussion

To verify the performance of the suggested shape retrieving algorithm, we tested it on the MPEG-7 dataset [7]. We used this dataset without any kind of change to get the performance results in the presence of instability in the dataset. Retrieval performance of our approach is excellent (in Fig. 4) up to the 3rd retrieval both for transformed and non-transformed images. Moreover, retrieval rate is 100% where instability does not occur in the database (e.g. bone images in Fig. 4). Comparative study has not been provided due to the unavailability of skeleton-based shape retrieval approaches which use MPEG-7 database. We have also tested our approach with a Kimia dataset of 216 images [1]. It is clear from Fig. 5 that our method performed the same or better than the recent approaches [2], [4]. We see that, up to the 3rd retrieval, the proposed approach shows the same performance as the recent approaches [1], [2], [4] and better performance for the 5th to the 10th retrieval (except for the 4th retrieval). The skeletal graph method [2] shows better performance only for the 4th retrieval and has a sharp fall after the 5th retrieval. The gradient vector approach [4] shows lower performance than our approach after the 4th retrieval. Therefore, in terms of retrieval performance, the approach [2] and [4] are not as good as our approach. Though shock graph approach [1] has almost the same performance as our approach, however, the method [1] is avoided due to its computational complexity [2]. Therefore, our method does not affect the accuracy at all. Moreover, our main goal is to reduce the complexity without a loss of accuracy.

In terms of computational complexity, our approach shows best performance compared to the other skeletal-based methods (Table 1). The total complexity will be $O(xy)$, [total complexity= (selection algorithm = $O(x)$) +

	Image without transformation			Image with transformation		
Query						
Match 1	M=0	M=0	M=0	M=0	M=0	M=0
Match 2	M=4	M=4	M=6	M=66	M=268	M=72
Match 3	M=4	M=8	M=6	M=68	M=283	M=79
Match 4	M=4	M=8	M=6	M=70	M=290	M=79
Match 5	M=6	M=10	M=10	M=74	M=292	M=83

Fig. 4 Distance measure (matching cost) for shape retrieval using 3 groups of 15 images (bone, camel, and cattle) from the MPEG-7 database.

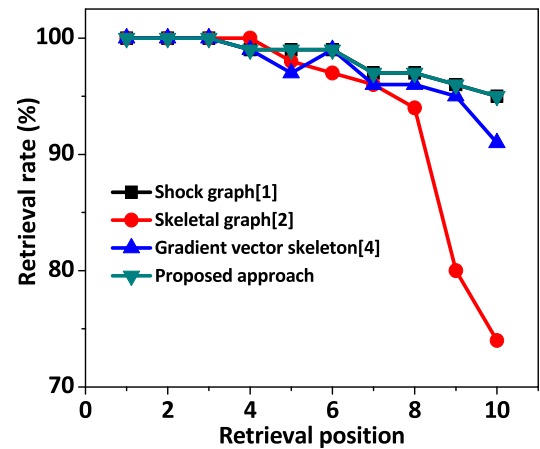


Fig. 5 Comparing shape retrieval performance (in percentage, up to 10th retrieval) of our approach to some other skeletal-based approaches using the Kimia 216 database [1], [2].

(shock points matching = $O(xy)$) + (deform cost = $O(x)$) = $\{O(x) + O(xy) + O(x)\} = \max\{O(xy), O(x)\} = O(xy)$. However, complexity tells little about running time; the efficiency of an algorithm depends upon data size [8]. A problem with complexity $O(xy)$ is still NP hard; however, we can decrease the running time by reducing the sample points. The important thing is that our approach reduces running time by selecting the sample points before matching. The average running time for matching two images is 0.09 seconds on a regular Pentium-IV 2.5 GHz PC, where the previous running time for a shock-based approach was 0.17 for two segments [1]. Moreover, the methods [2] and [4] are specialized to end points and branch points, respectively. However, our approach shows robustness in handling shapes with different complex interiors.

Table 1 Performance of the proposed approach compared to the other skeletal-based approaches with respect to complexity analysis.

Methods	Preferred Nodes	Time complexity	Comments
Shock graph [1]	All nodes	$O(x^3y^3)$	Computationally complex problem
Skeletal graph [2]	End nodes	$O(x^2y^2)$	Not applicable for images with no end points
Gradient vector skeleton [4]	Branch nodes	$O(x^3)$	Not applicable for different images with same branch points
Discrete shock graph	Nodes are selected adaptively	$O(xy)$	Time complexity and running time is less than [1] and applicable for all types of images

5. Conclusion

To the best of our knowledge, a skeleton-based approach

that adaptively considers all shock points by reducing the time complexity for shape retrieval has not yet been examined for a large database of MPEG-7. In this letter, we proposed and implemented an idea of discrete calculation of shock-based edit cost which is less complex than traditional shock-based shape retrieval. Moreover, the selection of nodes before matching helps to reduce running time without a loss of accuracy. A detailed comparative study of our approach applied in a real-life image database is under investigation.

References

- [1] T.B. Sebastian, P.N. Klein, and B.B. Kimia, "Recognition of shapes by editing their shock graphs," IEEE Trans. Pattern Anal. Mach. Intell., vol.26, no. 5, pp.550–571, 2004.
- [2] X. Bai and L.J. Latecki, "Path similarity skeleton graph matching," IEEE Trans. Pattern Anal. Mach. Intell., vol.30, no.7, pp.1282–1292, 2008.
- [3] H. Zaboli and M. Rahmati, "An improved shock graph approach for shape recognition and retrieval," Proc. First Asia Int. Conf. on Modeling and Simulation, pp.438–443, Thailand, March 2007.
- [4] W.-B. Goh, "Strategies for shape matching using skeletons," Comput. Vis. Image Underst., vol.110, no.3, pp.326–345, 2008.
- [5] S. Khanam, S.W. Jang, and W. Paik, "An improved shock graph-based edit distance approach using an adaptive weighting scheme," Proc. Int. Conf. on Future Generation Communication and Networking, Korea, vol.56, pp.501–508, Dec. 2009.
- [6] H. Blum, A Transformation for Extracting New Descriptors of Shape, ed. W. Whaten-Dunn, MIT Press, 1967.
- [7] http://www.imageprocessingplace.com/root_files_V3/image_databases.htm. Loudon
- [8] K. Loudon, Mastering Algorithms with C, O'Reilly Media, 1999.