www.cad-journal.net
Computer-Aided Design AND APPLICATIONS™

Fast shape retrieval based on differential chain code descriptor and fuzzy contour matching

Wei Pan[1] , Ning Li[2] , Wenming Tang[3] Lei Lu[4]

[1]OPT Machine Vision Tech Co.,Ltd., vpan@foxmail.com
[2]Henan University of Technology, ln_1013@163.com
[3]Shenzhen Institute of Information Technology, tangwenming@sziit.edu.cn
[4]Henan University of Technology, lulei@haut.edu.cn

Corresponding author: Lei Lu, lulei@haut.edu.cn

Abstract. To address the issues of slow processing, poor robustness, and performance degradation in complex scenarios often seen in traditional image matching methods, a fast contour matching approach based on differential chain codes is proposed. This method leverages the structural and topological information inherent in object contours by encoding contour curves into directional differential chain code sequences, effectively transforming contour matching into a highly efficient sequence comparison problem. During the matching process, a segmented contour matching strategy is adopted, dividing the overall contour into smaller segments for individual matching. A robust verification mechanism is introduced to reduce the risk of mismatches caused by noise interference. Extensive experiments on both synthetic and real-world datasets demonstrate that the proposed method significantly improves processing speed while maintaining high matching accuracy. Moreover, it shows strong stability and robustness under challenging conditions, such as noise, blur, and occlusion. This algorithm has great potential for real-time applications in areas such as robotic perception, industrial visual inspection, and mobile vision systems.

Keywords: Mesh Processing, Surface Denoising, Dimensional Accuracy, Process Optimization
DOI: https://doi.org/10.14733/cadaps.YYYY.aaa-bbb

1 INTRODUCTION

Image matching is a fundamental task in computer vision and plays a crucial role in a wide range of applications such as object recognition, image stitching, augmented reality, 3D reconstruction, and visual localization. The objective of image matching is to identify corresponding regions or features between different images, allowing higher-level vision tasks to infer spatial relationships, track movements, or reconstruct scenes.

Historically, image matching methods have evolved significantly and can be broadly categorized into three main classes: feature-based, template-based, and deep learning-based approaches.

Feature-based methods extract local descriptors from key points in images. Classical algorithms such as SIFT [14], SURF [2], and ORB [18] have been widely adopted for their robustness to scale, rotation, and moderate changes in illumination. These methods detect salient regions in images and compute distinctive descriptors that can be efficiently matched across images. However, they tend to perform poorly under conditions of heavy noise, blur, or significant viewpoint variation. Furthermore, due to the large number of feature points and complex matching procedures, these algorithms are often computationally intensive, posing challenges for real-time or resource-constrained applications.

Template-based methods approach the matching problem by directly comparing image patches using similarity measures such as normalized cross-correlation or sum of squared differences. Notable examples include fast match [9] and person search [10], which accelerate template matching through hierarchical or coarse-to-fine strategies. These methods are particularly effective for detecting known objects or patterns within cluttered scenes. However, they often suffer from high computational costs, especially when applied to large search spaces or high-resolution images, and are sensitive to scale, rotation, and illumination changes.

Deep learning-based methods represent the current state of the art, using powerful neural network architectures to learn robust feature representations. Techniques such as Siamese networks [8] and matching networks [20] have shown remarkable performance in tasks requiring one-shot learning and similarity-based matching. These approaches automatically learn a similarity function from the data, enabling them to match images effectively even with challenging variations. In addition, methods such as low-shot learning [11] and Collaborative Distribution Alignment [7] have been proposed to improve the robustness and adaptability of deep learning-based image matching systems.

To address the increasing need for fast and reliable matching, various shape-based matching and retrieval techniques have been developed. The use of Fourier descriptors [21], shape contexts [3], and inner-distance shape classification [12] have enabled better performance in shape-based image retrieval tasks. Further advances have been made with techniques that focus on articulation-invariant shape matching [4] and Zernike moments [15] for robust shape feature descriptors.

Furthermore, recent work has also focused on sketch-based shape retrieval, such as SketchCleanNet [16], which leverages deep learning to improve and correct query sketches in 3D CAD model retrieval systems. This is part of a larger trend towards integrating sketch-based recognition systems with deep neural networks to improve accuracy and retrieval times, as seen in recent work on CAD cut-piece retrieval [6].

The increasing availability of comprehensive datasets, such as Tari-1000 [1], ETH80 [13], and MPEG-7 [19], is also driving the development of new matching algorithms and evaluation benchmarks, further enhancing the accuracy and efficiency of image matching techniques.

To address the above issues, this paper proposes a fast contour matching method based on differential chain codes. By ensuring the ordering and uniformity of the contour point sets, contours with similar shapes produce highly similar differential chain codes, effectively transforming the contour matching problem into a differential chain code matching problem. However, due to the ordered nature of differential chain codes, they are sensitive to noise. To overcome this, the proposed method adopts a segmented contour matching strategy, dividing the contour into multiple segments for matching, and introduces a series of steps to suppress noise and eliminate false matches. This significantly reduces the impact of noise on the matching results and improves the robustness of the algorithm. Experimental results demonstrate that the proposed method performs excellently in both accuracy and speed, maintaining high matching precision even when dealing with noisy data.

2 Contour Extraction and Coding

For a binary image (black background with white regions), perform run-length encoding by first identifying the starting and ending columns of white regions (pixel value 255) in each row and storing this information sequentially to obtain the run-length encoding sequence for the entire image. Then, conduct a connectivity analysis on the run-length encoding sequence using four-connectivity or eight-connectivity to convert it into a graph data structure. Perform depth-first search (DFS) on the constructed graph data structure to obtain connected regions and their run vertices sequences. If the graph contains cycles, it indicates the presence of inner contours; otherwise, only outer contours are present. The resulting contours are preliminary, and since they are derived from the vertices of the runs, it is necessary to perform equidistant sampling in both horizontal and vertical directions to obtain the complete contour.

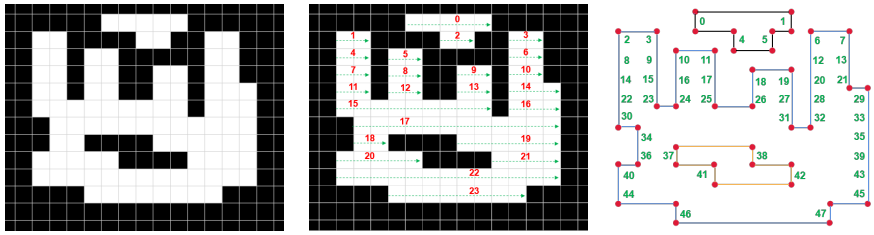


Figure 1: Correspondence between differential chain codes and angles

2.1 Contour Smoothing

In the processing of contour point sets, noise significantly impacts the uniformity of the subsequent sampled point set, which in turn affects the matching results. Therefore, smoothing the contour point set before sampling is necessary. This paper uses the mean of neighboring points for smoothing, defined as follows: Point set $P = \{P_0, P_1, \ldots, P_{n-1}\}$, where $P_i = (x_i, y_i)$ is the i-th point, and n is the total length of the point set. - The filter kernel size is k, where k is an odd number. The filter radius r is defined as $r = \frac{k-1}{2}$.

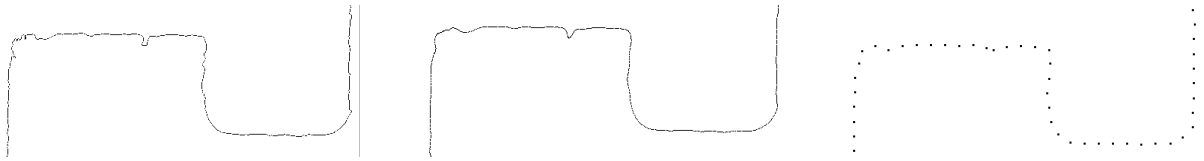
The smoothed result point set is defined as $P' = \{P'_0 P'_1, \ldots, P'_{n-}\}$, where $P'_i = (x'_i, y'_i)$ represents the smoothed point coordinates.Below are the specific formulas Equation 1 is the smoothing formula for closed contours, and Equation 2 is the smoothing formula for open contours.

$$\begin{cases} P'_0 = \frac{1}{2r+1}\left(\sum_{j=n-r}^{j=n-1} P_j + \sum_{j=0}^{j=r} P_j\right), & i = 0 \\ P'_i = \frac{1}{2r+1}\left(P_{(i+r) \mod n} - P_{(i-r-1) \mod n} + P'_{i-1}\right), & 1 \le i < n \end{cases} \tag{1}$$

$$\begin{cases} P'_i = P_i, & 0 \le i < r, n-r \le i < n \\ P'_i = \frac{1}{r+1}\left(\sum_{j=0}^{j=r} P_j\right) + \frac{1}{2r+1}\left(\sum_{j=i-r}^{i+r} P_j\right), & r \le i < n-r \end{cases} \tag{2}$$

2.2 Uniform sampling

The contour points obtained from the run-length encoding have a distance of approximately 1 pixel between them. To reduce computational load while preserving the features of the original data, we use uniform sampling. This method samples the point set at a fixed distance, effectively reducing the number of points while retaining the primary shape and characteristics of the contour.



(a) Original contour (b) Smoothed contour (c) Sampled contour

Figure 2: Illustration of uniform sampling on a contour: (a) the original contour with densely spaced points (approx. 1-pixel interval) from run-length encoding, (b) the smoothed contour, and (c) the sampled contour with points selected at a fixed distance.

2.3 Contour coding

Chain code is one of the commonly used methods to describe geometric information of shapes. The description of geometric information will have geometric invariance when considering the difference and normalization of the chain code. The 8-chain chain code [5] is effectively used in describing geometric information of a shape. The chain code of a digitized curve represents sequences of direction changes between adjacent points on the curve [17]. Therefore, any geometric shape can be represented by a chain code sequence. The 2D free-form shapes participating in the layout can be represented by a series of chain codes without losing accuracy.

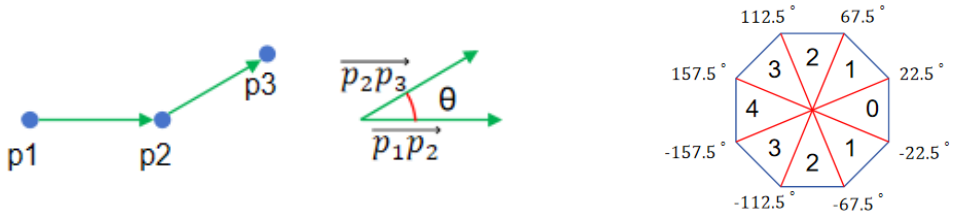
Starting from a starting point of the boundary (curve or line), observe the direction of each line segment clockwise, and represent it with the corresponding pointer of 8-chain codes. The result is a digital sequence representing the boundary (curve or line), which is called the original chain code. The original chain code has translation invariance (the index is not changed during translation), but when the starting point S is changed, a different chain code representation will be obtained, that is, it is not unique.

Differential coding is also known as incremental coding. Differential coding refers to the code of a digital data stream in which every element, except the first element, is represented as the difference between that element and its predecessor. Differential chain codes and original chain codes have the same properties, that is, translation invariance and scale invariance. The computation of the difference code is shown in Eq. 1.

$$\begin{cases} M_N = \sum_{i=1}^{N} a_i \\ a'_1 = (a_1 - a_i) \mod N, i = 2, 3, \ldots, N \\ a'_i = (a_i - a_{i-1}) \mod N \end{cases} \tag{3}$$

where the value of N is the number of difference codes, and a_i is the value of each sequence in the difference chain code. It can start from any code value in the difference code and recurse step by step in one direction.

The method of computing difference chain codes described in this paper differs slightly from traditional difference chain code calculations. Instead of first computing the original chain code, this method directly uses three consecutive points to form two vectors. The difference chain code for the middle point is then determined based on the rotation angle between these two vectors. This approach streamlines the process by bypassing the initial chain code computation, allowing for a more efficient determination of geometric information The statistical rules and the relationship between the difference chain codes and angles are illustrated in the figure below.



(a) Angle quantization rule (b) Eight-directional code

Figure 3: Differential Chain Code Computation: (a) Angle quantization rule using three sequential points to compute differential chain codes. (b) Eight-directional code used for representing angle changes.

3 Segmented Fuzzy Matching

This section employs a segmented fuzzy matching approach to enhance robustness. By dividing the contour into segments and applying fuzzy logic to match corresponding segments, the method can tolerate noise and minor distortions, leading to more reliable retrieval results. The overall process is illustrated in the figure below.

The ordered nature of differential chain codes can lead to noise sensitivity issues. As shown in the figure, when the noise point p_3 in point set P matches with q_3 in point set Q, the subsequent matching point pairs will be misaligned. This indicates that the errors caused by noise are transmissive; the longer the differential chain code, the more misaligned points there are, and the greater the error. To mitigate error transmission, the method segments the short differential chain codes sequentially. By reducing the length of the chain code array, the impact of transmission errors is minimized.

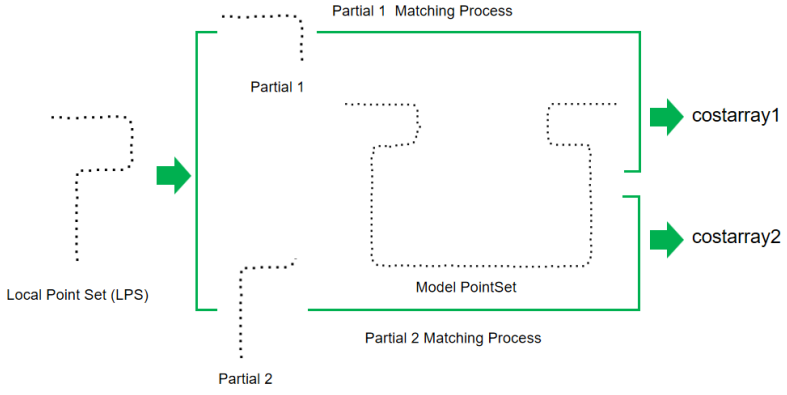


Figure 4: Segmented fuzzy matching. This figure provides an overview of the fuzzy matching process, which consists of segmenting the model and local pointsets, matching the partial pointsets and evaluating the cost array.

Based on the sub-sequence difference chain code arrays obtained in the previous step, they are matched with the longer difference chain code. During the matching process, the shorter difference chain code is used as a sliding window to traverse the longer difference chain code array one by one, and the cost value is calculated

according to the following formula:

$$C_i = \sum_{0}^{n} |b_i - a_i| \tag{4}$$

The index of the cost value c_i is the starting position of the match a_i. The smaller the matching cost value, the higher the probability of a successful match. When the matching position is close to the actual corresponding position, its cost value will be significantly different from the cost values of other positions. After matching all sub-sequence difference chain codes, multiple cost arrays of the same length as the long difference chain code will be obtained. The specific implementation is shown in the figure below.

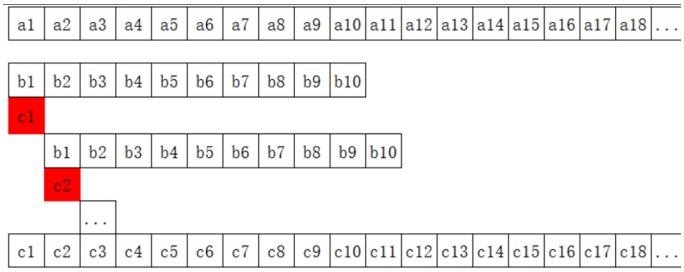


Figure 5: Matching cost calculation. This figure visualizes the process of obtaining cost array for each segmented chain code sequences.

According to equation [1], the template point set P is encoded as $A = \{a_1, a_2, a_3, \ldots, a_m\}$, and the point set to be matched Q is encoded as $B = \{b_1, b_2, \ldots, b_n\}$ (where $m \geq n$). To reduce errors, the point set Q is divided into k segments, each with a length of l. Thus, the sequence array to be matched is:$B^l = \{\{b_1, b_2, \ldots, b_l\}, \ldots, \{b_{n-l+1}, b_{n-l+2}, \ldots, b_n\}\} = \{B_1^l, B_2^l ... B_k^l\}$After segmentation, there are k subsequences, where $k = \lceil \frac{n}{l} \rceil$.At this point, according to the formula, the cost array for each subsequence B_i with A is calculated, resulting in C= $\{C_1, C_2, \ldots, C_k\}$, where C_i represents the cost array of the i-th subsequence,C_i= $\{c_1, c_2, \ldots, c_m\}$.

Although segmentation can effectively suppress the propagation of differential code matching errors, it still causes misalignment of the best matching position. Therefore, we subsequently use a sliding window to traverse each cost array and fill the minimum cost value within the window into the center position of the window to reduce errors. The specific calculation is shown in the figure below.

After calculating the cost array for each subsequence, the matching cost of the local contour point set is computed. The cost at each position is calculated using the following formula, where C_i is the cost array of the i-th sub-contour, and l_i is the length of the i-th sub-contour. The specific calculation can be referred to in the figure below.

Due to the use of a sliding window, the resulting total cost value is often distributed in segments, forming a pattern with increasing and decreasing values. Then, the positions with smaller cost values are retained, and the center of these segments is taken as the possible starting position for matching. The final best matching position is selected from these positions corresponding to the smaller cost values.

Due to the fuzzy matching strategy employed, the results obtained may include positions that are similar to the local contour shape as well as positions with erroneous matches. To eliminate erroneous matches, the method forms pairs of corresponding points between the local point set and each possible matching position. The transformation matrix is calculated using the Singular Value Decomposition (SVD) method. The quality of the match is evaluated by calculating the average distance between the corresponding point pairs after transformation. Finally, the positions with smaller average distances are retained.

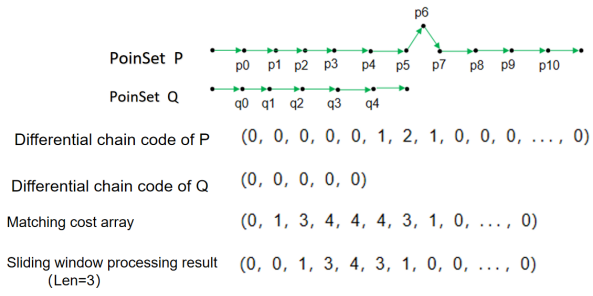


Figure 6: Sub-contour cost calculation process. This figure illustrates how a sliding window of size 3 is used to traverse the cost array and how minimum cost values are assigned to reduce error

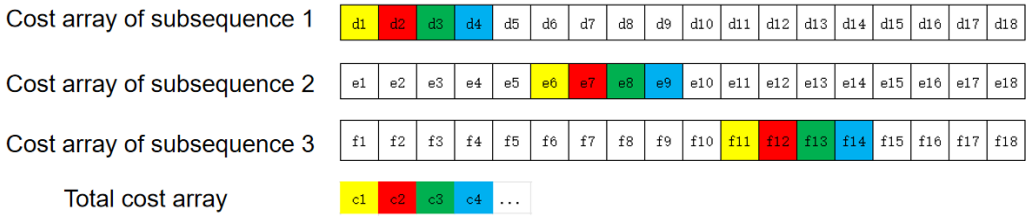


Figure 7: Statistical cost value. This figure presents the process of calculating total cost using the local cost array.

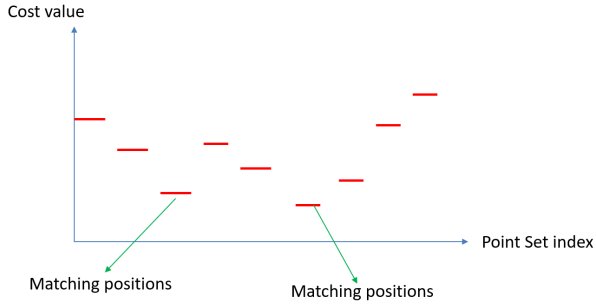


Figure 8: Cost distribution map. This figure illustrates how the local minimums of cost array are used to select the most suitable matching positions.

4 Experimental Results:

In this experiment, we used CAD drawings captured by the SmartFlash-2020 one-click measurement sensor as experimental materials. This sensor is suitable for inspecting large flat products, with a measurement range of up to 300×200 mm, and is widely used for batch measurement of component dimensions, offering high efficiency and consistent measurement results.

To test the effectiveness of the matching algorithm, we rotated each original image clockwise by 30, 60, 90, 120, and 150, generating a series of rotated images. We then compared the original images with these rotated images, and the experimental results are shown in Figure 11. Additionally, we conducted matching experiments under multi-target interference, and the results are shown in Figure 10. Through these experiments, we evaluated the performance and accuracy of the matching algorithm under different rotation angles and interference conditions. This process not only validated the robustness of the algorithm but also helped us further optimize and improve image matching technology.

Figure 9: SmartFlash-2020 Measuring Instrument

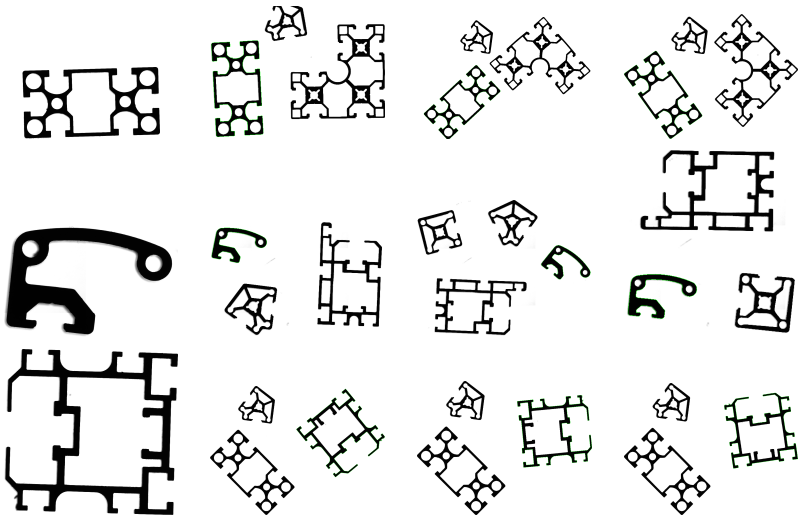


Figure 10: Multi-Target Matching Experimental Results

To further evaluate the effectiveness of the proposed method, we conducted quantitative experiments on

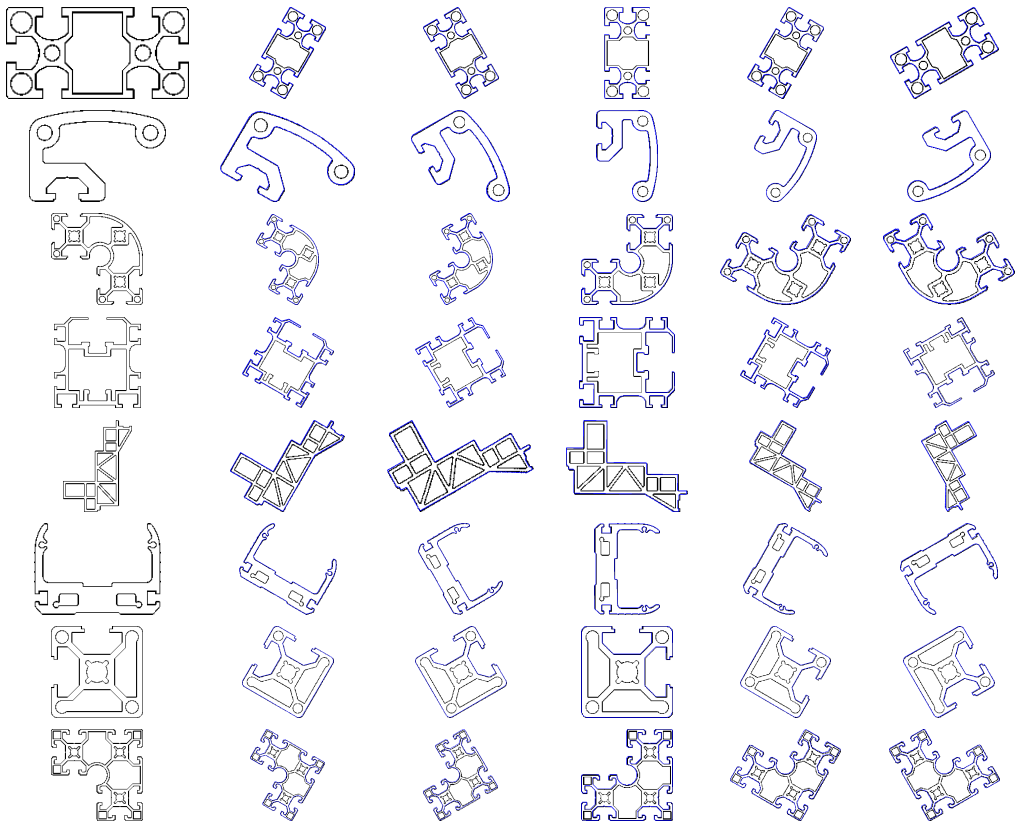


Figure 11: Matching experimental results

multiple groups of representative test images. Table 1 presents a comparison between our method and several existing approaches, including the widely used Halcon system and a baseline method referred to as Smart. For each test image, we report two error metricsMean Distance and Standard Deviation, which measure the discrepancy between the matched contour and the ground truth.

As shown by the experimental results, our method consistently achieves lower error values compared to Halcon and Smart, indicating higher matching accuracy. Furthermore, when combined with a fine alignment step using iterative closest point (ICP) -denoted Our+ICP our method demonstrates further improvements in accuracy across all test cases. Notably, the incorporation of ICP significantly reduces errors in test images 2, 3, 5, and 8, highlighting the effectiveness of this two-stage matching strategy.

Overall, the results confirm the robustness and high accuracy of the proposed method in contour matching tasks. It maintains strong performance even in scenarios involving shape deformation or moderate noise interference. Additionally, the method offers high computational efficiency while ensuring accuracy, making it well-suited for real-world applications that demand both precision and real-time performance.

Table 1: Performance Quantitative Evaluation

Test Image	Matching Method	Mean distance	Standard deviation
1	Halcon	0.964	0.452
	Smart	0.642	0.233
	Our	0.521	0.222
	Our+ICP	**0.447**	**0.189**
2	Halcon	1.027	0.430
	Smart	0.715	0.319
	Our	0.426	0.185
	Our+ICP	**0.387**	**0.155**
3	Halcon	0.754	0.353
	Smart	0.671	0.265
	Our	0.434	0.198
	Our+ICP	**0.372**	**0.163**
4	Halcon	0.768	0.351
	Smart	0.662	0.248
	Our	0.578	0.253
	Our+ICP	**0.419**	**0.177**
5	Halcon	0.856	0.412
	Smart	0.958	0.421
	Our	0.581	0.251
	Our+ICP	**0.391**	**0.160**
6	Halcon	0.924	0.390
	Smart	0.747	0.322
	Our	**0.376**	0.179
	Our+ICP	**0.376**	**0.168**
7	Halcon	0.694	0.269
	Smart	0.824	0.461
	Our	0.464	0.210
	Our+ICP	**0.415**	**0.187**
8	Halcon	0.960	0.483
	Smart	0.715	0.292
	Our	0.601	0.279
	Our+ICP	**0.372**	**0.160**

5 Conclusion

The proposed CAD matching method, based on the modified Freeman chain code and fuzzy contour matching, demonstrates significant improvements in both matching efficiency and robustness. Its ability to handle noise and complex shapes makes it a valuable tool for real-world CAD applications. As future work, further

optimizations and extensions of the method will be explored, such as integrating more advanced smoothing techniques, enhancing the fuzzy matching algorithm, and investigating the incorporation of deep learning to improve performance and scalability.

ACKNOWLEDGEMENTS

Wei Pan, https://orcid.org/0000-0002-0933-2453
Ning Li, https://orcid.org/0009-0006-9149-8087
Wenming Tang, https://orcid.org/0000-0002-1427-3216
Lu Lei, https://orcid.org/0000-0002-3050-6542

REFERENCES

[1] Baseski, E.; Erdem, A.; Tari, S.: Dissimilarity between two skeletal trees in a context. Pattern Recognition, 42(3), 370–385, 2009. http://doi.org/10.1016/j.patcog.2008.05.022.

[2] Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L.: Speeded-up robust features (surf). Computer vision and image understanding, 110(3), 346–359, 2008. http://doi.org/10.1016/j.cviu.2007.09.014.

[3] Belongie, S.; Malik, J.; Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE transactions on pattern analysis and machine intelligence, 24(4), 509–522, 2002. http://doi.org/10.1109/34.993558.

[4] Biswas, S.; Aggarwal, G.; Chellappa, R.: Efficient indexing for articulation invariant shape matching and retrieval. In 2007 IEEE Conference on Computer Vision and Pattern Recognition, 1–8. IEEE, 2007. http://doi.org/10.1109/CVPR.2007.383227.

[5] Freeman, H.: On the encoding of arbitrary geometric configurations. IRE Transactions on Electronic Computers, EC-10(2), 260–268, 1961. http://doi.org/10.1109/TEC.1961.5219197.

[6] Geng, L.; Yang, Y.; Xiao, Z.; Yin, C.: Cad cut-piece retrieval method based on representation learning. In Proceedings of the 2021 7th International Conference on Computing and Artificial Intelligence, 452–458, 2021. http://doi.org/10.1145/3467707.3467775.

[7] Hu, N.; Zhou, H.; Liu, A.A.; Huang, X.; Zhang, S.; Jin, G.; Guo, J.; Li, X.: Collaborative distribution alignment for 2d image-based 3d shape retrieval. Journal of Visual Communication and Image Representation, 83, 103426, 2022. http://doi.org/https://doi.org/10.1016/j.jvcir.2021.103426.

[8] Koch, G.; Zemel, R.; Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop, vol. 2, 1–30. Lille, 2015. http://doi.org/10.1109/ICACITE57410.2023.10182466.

[9] Korman, S.; Reichman, D.; Tsur, G.; Avidan, S.: Fast-match: Fast affine template matching. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2331–2338. IEEE, 2013. http://doi.org/10.1109/CVPR.2013.302.

[10] Lan, X.; Zhu, X.; Gong, S.: Person search by multi-scale matching. In Proceedings of the European Conference on Computer Vision (ECCV), 536–552, 2018. http://doi.org/10.1007/978-3-030-01246-5_33.

[11] Leong, O.F.; Sakla, W.: Low shot learning with untrained neural networks for imaging inverse problems. In LatinX in AI at Neural Information Processing Systems Conference (NeurIPS), 2019. http://doi.org/10.52591/lxai2019120832.

[12] Ling, H.; Jacobs, D.W.: Shape classification using the inner-distance. IEEE transactions on pattern analysis and machine intelligence, 29(2), 286–299, 2007. http://doi.org/10.1109/TPAMI.2007.41.

[13] Ling, H.; Yang, X.; Latecki, L.J.: Balancing deformability and discriminability for shape matching. In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III 11, 411–424. Springer, 2010. http://doi.org/10.1007/978-3-642-15558-1_30.

[14] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91–110, 2004. http://doi.org/10.1023/B:VISI.0000029664.99615.94.

[15] Ma, Z.M.; Zhang, G.; Yan, L.: Shape feature descriptor using modified zernike moments. Pattern Analysis and Applications, 14, 9–22, 2011. http://doi.org/10.1007/s10044-009-0171-0.

[16] Manda, B.; Kendre, P.P.; Dey, S.; Muthuganapathy, R.: Sketchcleanneta deep learning approach to the enhancement and correction of query sketches for a 3d cad model retrieval system. Computers & Graphics, 107, 73–83, 2022. http://doi.org/10.1016/j.cag.2022.07.006.

[17] Nixon, M.S.; Aguado, A.S.: Chapter 7 - object description. In M.S. Nixon; A.S. Aguado, eds., Feature Extraction Image Processing for Computer Vision (Third Edition), 343–397. Academic Press, Oxford, third edition ed., 2012. ISBN 978-0-12-396549-3. http://doi.org/https://doi.org/10.1016/B978-0-12-396549-3.00007-0.

[18] Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G.: Orb: An efficient alternative to sift or surf. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2564–2571. IEEE, 2011. http://doi.org/10.1109/ICCV.2011.6126544.

[19] Shen, W.; Jiang, Y.; Gao, W.; Zeng, D.; Wang, X.: Shape recognition by bag of skeleton-associated contour parts. Pattern Recognition Letters, 83, 321–329, 2016. http://doi.org/https://doi.org/10.1016/j.patrec.2016.02.002.

[20] Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.: Matching networks for one shot learning. In Advances in Neural Information Processing Systems, vol. 29, 3630–3638. NeurIPS, 2016. http://doi.org/10.48550/arXiv.1606.04080.

[21] Zhang, D.; Lu, G.: Generic fourier descriptor for shape-based image retrieval. In Proceedings. IEEE International Conference on Multimedia and Expo, vol. 1, 425–428. IEEE, 2002. http://doi.org/10.1109/ICME.2002.1035809.