Title:
**Fast Shape Retrieval Based on Differential Chain Code Descriptor and Fuzzy Contour Matching**

Authors:
Wei Pan, vpan@foxmail.com, OPT Machine Vision Tech Co.,Ltd
Ning Li, ln_1013@163.com, Henan University of Technology
Wenming Tang, tangwenming@sziit.edu.cn, Shenzhen Institute of Information Technology
Lei Lu, lulei@haut.edu.cn, Henan University of Technology

Keywords:
shape retrieval, contour based matching, pointset registration, chain code.

DOI: 10.14733/cadconfP.2025.300-305

Introduction:
Image matching is a core task in computer vision, widely applied in object recognition, image stitching, and visual localization. Early image matching methods can be divided into three main categories: feature-based methods (such as SIFT [1], SURF[2] and ORB[3]), which extract feature points from images and match their descriptors to effectively handle rotation, scale, and illumination changes. However, these methods perform poorly in high-noise or blurred images and are computationally expensive; template-based methods (such as Fast-Match[4], MSPD[5]), which compare templates to target images pixel by pixel using sliding windows, are suitable for object detection but incur high computational costs in complex scenarios; and deep learning-based methods (such as Siamese Networks[6], Matching Networks[7], which utilize convolutional neural networks and other techniques to extract image features, offering stronger robustness and the ability to handle issues such as lighting, scale variation, and occlusion. These methods are suitable for complex scenes but require large amounts of labeled data and computational resources, with long training times and slower inference speeds.

To address these issues, this paper proposes a fast contour matching method based on differential chain codes. By ensuring the ordering and uniformity of contour point sets, differential chain codes of similarly shaped contours exhibit high similarity, transforming the contour matching problem into a differential chain code matching problem. Due to the ordered nature of differential chain codes, they are sensitive to noise. To mitigate this, the method adopts a segmented contour matching strategy, dividing the contour into multiple segments for matching and introducing a series of steps to suppress noise and eliminate incorrect matches. This significantly reduces the impact of noise on the matching results and enhances the robustness of the algorithm. Experimental results show that the proposed method performs excellently in both accuracy and speed, maintaining high matching accuracy even when dealing with noisy data.

Contour Extraction and Coding:
In this section, a binary image undergoes run-length encoding by identifying the starting and ending columns of white regions in each row, and then storing this sequentially for the entire image. Connectivity

analysis is conducted on the encoded sequence using four- or eight-connectivity, converting it into a graph data structure, followed by a depth-first search (DFS) to identify connected regions and their vertex sequences. If the graph contains cycles, inner contours are present; otherwise, only outer contours exist. The contours are then smoothed using a mean-based filter to reduce noise, and uniform sampling is applied to reduce the number of points while preserving the contour's shape. This process helps in obtaining accurate, well-defined contours for further analysis.

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.

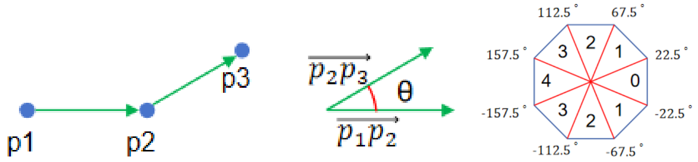


Fig. 1: Correspondence between differential chain codes and angles

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.

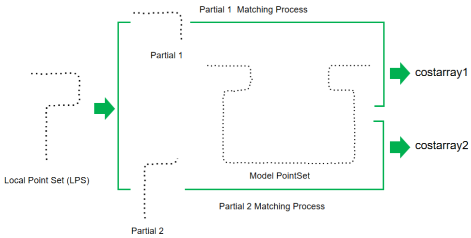


Fig. 2: Segmented fuzzy matching

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.

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

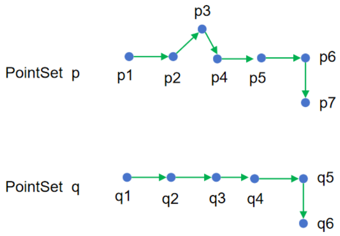


Fig. 3: Point set with noise

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:

$$C_i = \sum_{0}^{n} |b_i - a_i| \tag{2.1}$$

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. where a is the chain code of the template contour, b is the chain code of the contour to be matched, and c is the final computed representative value

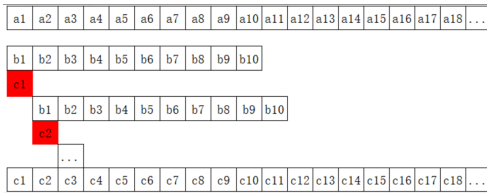


Fig. 4: Matching cost calculation

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:$B^l = \{\{b_1, b_2, \ldots, b_l\}, \ldots, \{b_{n-l+1}, b_{n-l+2}, \ldots, b_n\}\} = \{B_1^l, B_2^l \ldots B_k^l\}$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

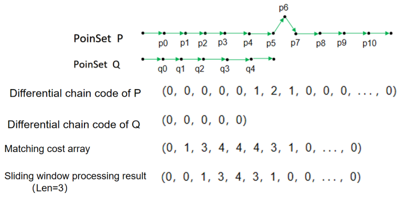


Fig. 5: Sub-contour cost calculation process

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.

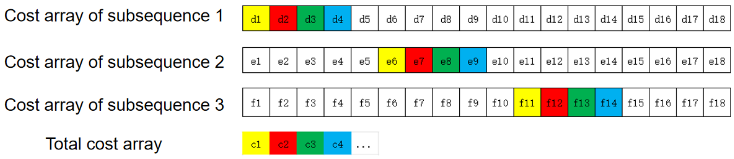


Fig. 6: Statistical cost value

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.

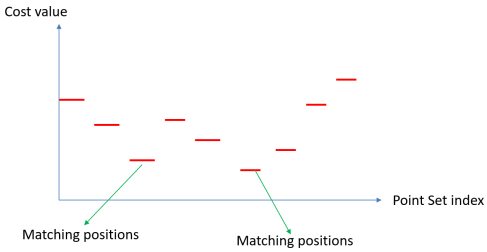


Fig. 7: Cost distribution map

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.

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 9. 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.

Fig. 8: SmartFlash-2020 Measuring Instrument

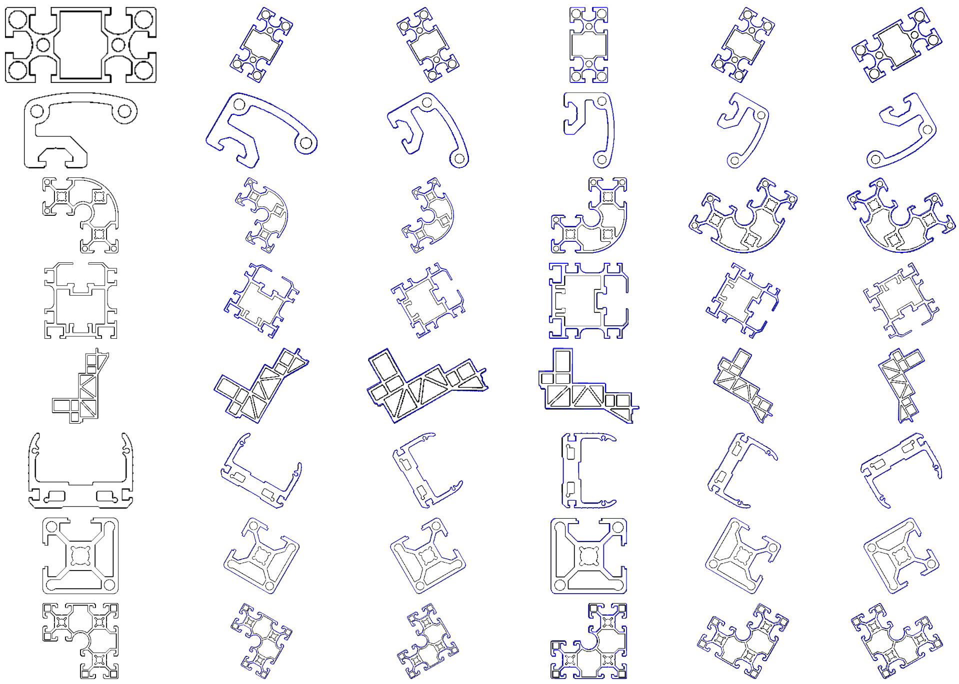

Fig. 9: Matching experimental results

Conclusions:

Proceedings of CAD'25, Shenzhen, China, June 23-25, 2025, 300-305

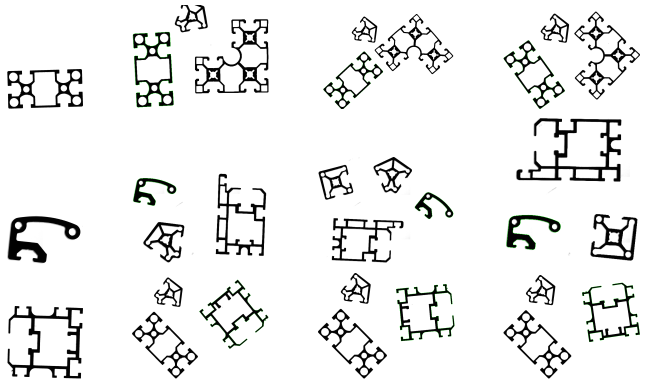

Fig. 10: Multi-Target Matching Experimental Results

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.

References:


[1] Lowe, David G. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60:91–110, 2004.Springer.http://dx.doi.org/10.1023/b:visi.0000029664.99615.94

[2] Bay, Herbert; Ess, Andreas; Tuytelaars, Tinne; Van Gool, Luc. Speeded-up robust features (SURF). Computer vision and image understanding, 110(3):346–359, 2008. Elsevier.http://dx.doi.org/10.1016/j.cviu.2007.09.014

[3] Rublee, Ethan; Rabaud, Vincent; Konolige, Kurt; Bradski, Gary. ORB: An efficient alternative to SIFT or SURF. In: 2011 International conference on computer vision, pp. 2564–2571, 2011. IEEE.http://dx.doi.org/10.1109/iccv.2011.6126544

[4] Korman, Simon; Reichman, Daniel; Tsur, Gilad; Avidan, Shai. Fast-match: Fast affine template matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2331–2338, 2013.http://dx.doi.org/10.1109/cvpr.2013.302

[5] Lan, Xu; Zhu, Xiatian; Gong, Shaogang. Person search by multi-scale matching. In: Proceedings of the European conference on computer vision (ECCV), pp. 536–552, 2018.http://dx.doi.org/10.1007/978-3-030-01246-5_33

[6] Koch, Gregory; Zemel, Richard; Salakhutdinov, Ruslan. Siamese neural networks for one-shot image recognition. In: ICML deep learning workshop, 2(1):1–30, 2015. Lille.http://dx.doi.org/10.1109/icacite57410.2023.10182466

[7] Leong, Oscar F.; Sakla, Wesam. Low shot learning with untrained neural networks for imaging inverse problems. In: LatinX in AI at Neural Information Processing Systems Conference, LXAI at NeurIPS 2019, Dec. 2019. http://dx.doi.org/10.52591/lxai2019120832


Proceedings of CAD'25, Shenzhen, China, June 23-25, 2025, 300-305