Interaction and improvement of characteristic-number-line-matching and D-nets

Zhixing Li, Yujie Lei

School of software, Dalian University of technology

jt_project1@163.com, NewtYunyee@163.com

1.Abstract

Robust line matching is a challenging task in computer vision. The team at Dalian University of Technology proposed a new method for constructing projective invariants. **[1]**This method improves the matching precision and robustness of the projective transformation. The D-nets algorithm builds the network through classical sparse or dense feature points, while constructing descriptors by sampling the information on each side of the network. The D-Nets algorithm**[2]** not only has invariance to cropping, translation, scale, rotation, etc., but D-Nets is more robust to nonlinear distortion and severe transmission transformation than Patch-Based. This method also has good performance. Our research is based on the characteristic number linear matching algorithm, combined with the D-nets algorithm for optimization and improvement.

Keywords: line matching, projective invariance, characteristic number, D-nets, intersection algorithm.

2. Introduction

2.1 Characteristic number

In the field of computer vision, the characteristics of images have always been one of the most basic researches. Among them, the most studied ones are the point features of images, so there are many algorithms for detecting and matching image feature points. However, there is very little matching algorithm for the line in the image, and the conventional line matching algorithm obtains the texture feature as the feature descriptor of the line near the line, and then matches the descriptor. This method may have errors when the texture features of the image are not obvious. To this end, we draw on the straight line matching algorithm of Qi JIa and others from Dalian University of Technology. The algorithm constructs a projective invariant called a characteristic number(CN), which is bound to a straight line to form a straight line descriptor and used for matching.

2.1.1 Introduction of characteristic number and algorithm flow

The characteristic number is a new projective invariant that is extended based on the intersection ratio of the line. It combines the point and the line steadily, and then only needs to be matched based on it, which is more flexible than directly using the texture feature descriptor of the line. It is also more invariant under homography. Figure 1 is a definition of a characteristic number.

Definition: In one plane, there are n points P which can be connected in sequence to form a closed loop ($P_{n+1}=P_1$), and there are m points $Q_i^{(m)}$ on the line segment P_iP_{i+1} formed by two adjacent points. For each $Q_i^{(m)}$ point, we can use the formula (1) to linearly represent:

$$Q_i^{(j)} = a_i^{(j)} P_i + b_i^{\{j\}} P_{i+1} \qquad (1)$$

Then we integrate all the points $P=\{P_i\}_{i-1}^{n}$ and points $Q=\{Q_i^{(j)}\}(i=1,\ldots,n;\ j=1,\ldots,m)$. The numbers obtained by this formula (2) are called the characteristic number of P and Q.

$$CN(\mathcal{P},\mathcal{Q}) = \prod_{i=1}^{r}\left(\prod_{j=1}^{n}\frac{a_i^{(j)}}{b_i^{(j)}}\right) \qquad (2)$$

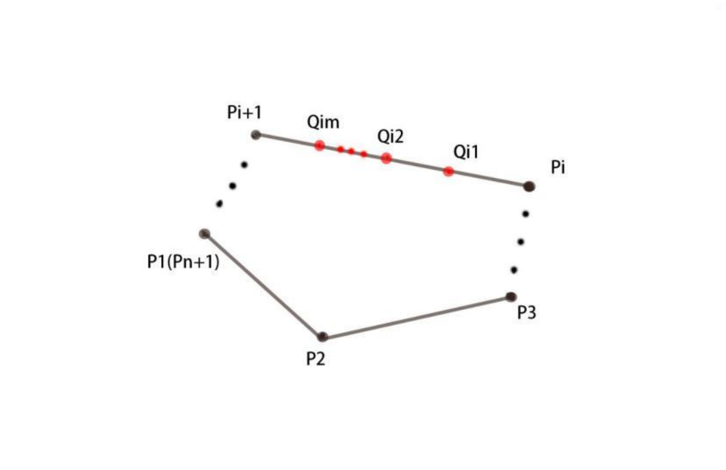
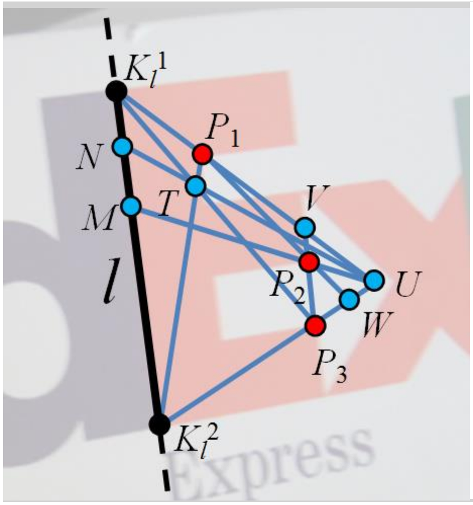


Figure 1

In the line matching algorithm of Qi Jia et al., five points are used to construct this characteristic number. As shown in Fig. 2, in one image, Kl1 and Kl2 are key points on the straight line l, and P1, P2, and P3 are three detected feature points on the same side of the straight line l. The intersection of Kl1P1 and Kl2P3 is defined as U; Kl1P3 and Kl2P1 intersect at point T; the intersection of UT, UP2 and line l are N and M respectively; P1P2 and Kl2P3 intersect at point W; P3P2 and Kl1P1 intersect at point V. So far, we have newly generated six points of M, N, T, V, U, W from the original five points, but according to the definition of the characteristic number, in the closed loop Kl1UKl2, we take M, N, P1, V, W P3 six points to calculate the features number of this closed loop (the original five points). In this way, we successfully combine feature points with straight lines to form a feature invariant.



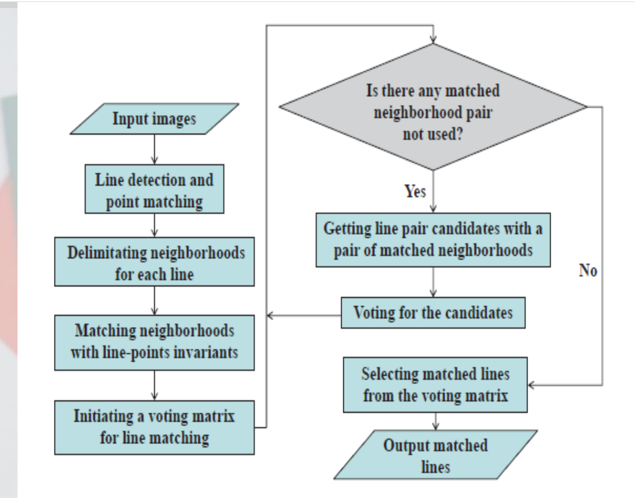


Figure 2 Figure 3

2.1.2Characteristic number line matching algorithm flow:

For the two images input, we use the famous line detection algorithm LSD to detect the lines in the two images and generate corresponding neighborhoods for each line. The feature points in the two images are detected by the SIFT feature point detection algorithm proposed by David Lowe, and then we take three points from the feature points in the neighborhood of each line and construct the characteristic number with the line, and Use characteristic numbers to match linear neighborhoods. Finally, the voting model is used to filter out the correct matching neighborhood and output the matching straight line.

2.2 D-nets

The D-nets algorithm builds the network through classic Sparse features or Dense features, while constructing descriptors by sampling the information on each side of the network. The D-Nets algorithm not only has invariance to cropping, translation, scale, rotation, etc., but D-Nets is more robust to nonlinear distortion and severe transmission transformation than Patch-Based. The D-Nets algorithm compares the results of the SIFT and ORB algorithms on the Oxford standard dataset, it greatly improves the matching accuracy and matching probability while maintaining computational efficiency.

In general, D-nets feature point matching is divided into three parts: constructing a stripe, coding a hash, and matching the most appropriate hash by a ticketing algorithm.

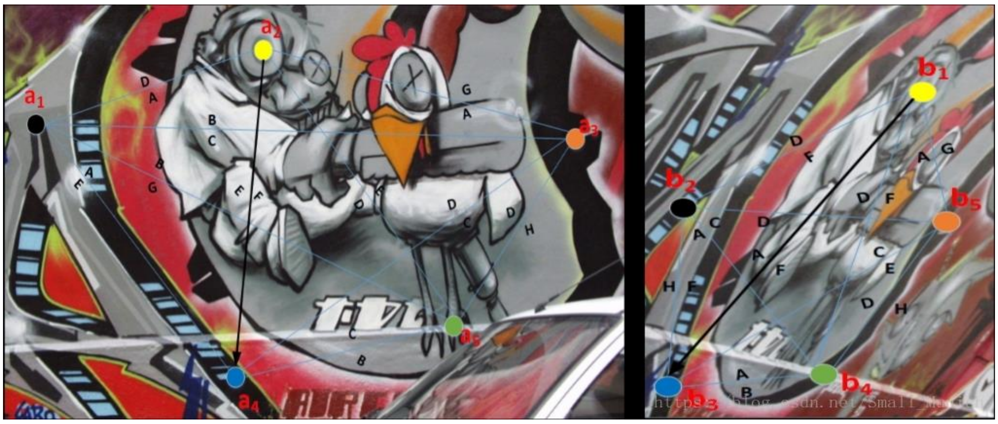


Figure 4

2.2.1 Building strips

We refer to the image area under this connection (consisting of the original pixels) as a "strip". The simplest formula for a strip is a straight line segment between two points of interest in the image. In the D-Nets approach, image matching is established by matching connections (rather than nodes) between images. Two connections are defined as correct matches if and only if their start and end nodes correspond to the same physical point in both images. In Fig. 4, the connections (a2, a4) (a2, a4) and (b1, b3) (b1, b3) are correct matching points because the nodes a2 $\boxed{\leftrightarrow}$ b1 a2 $\boxed{\leftrightarrow}$ b1 match a4 $\boxed{\leftrightarrow}$ b3 a4 $\boxed{\leftrightarrow}$ b3.

The traditional block-based approach will infer the match between the corresponding key points independently, and optionally perform geometric verification on the feature point set. In D-Nets, we use the image content in the corresponding strip to determine the match and use hash and voting algorithms to aggregate this information directly at the image level.

2.2.2 Coded Hash

The coded hash is for the pixel information of the image, and the peripheral pixel information is sampled by a specific algorithm at regular intervals on the strip, thereby converting the pixel information on the strip into digital information.

Detailed operation

$\bar{a}_i + 0.1 * (\bar{a}_j - \bar{a}_i)$ $\bar{a}_i + 0.8(\bar{a}_j - \bar{a}_i)$

1. On this layer, from to along the 10% of the strip The 80% portion samples the intensity of the pixel. In short, the reason behind this unusual operation is that our algorithm is insensitive to the pixel information of the start and end nodes. Our experiments have shown that it is beneficial to omit the tail of the strip, especially when the feature points are positioned with noise, and anti-symmetry is also desirable.

2.Group this sequence of values into a set of smaller s uniform blocks, averaging the pixel intensities in each block to reduce noise and generate s-dimensional vectors.

$$min_i s_i = 0.1 \qquad max_i s_i = 1$$

3.Normalize the vector using scaling and translation, etc., and In the special case $\forall_{(i,j)}(s_i = s_j)$, we set $\forall_{(i,j)} s_i = 0.5$

4.Discretely normalize each value in the s vector using b-bits.

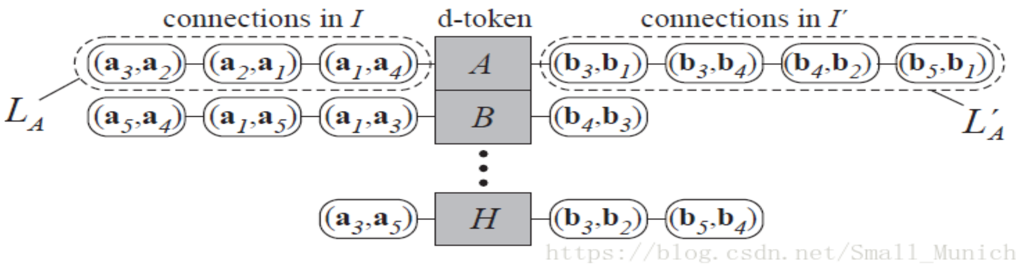


Figure 5

2.2.3 voter algorithm matching

In reality, |D||D| can be very large, and many bins in the hash table contain at least one empty list. Since these goals do not contain valid assumptions, they are skipped without casting any votes. The same happens if the two images are very different, so a small amount of d-tokens will appear in both images. Therefore, the run time of the algorithm is highly dependent on the difficulty of matching the task. Image pairs that are impossible to match (most of them) are quickly eliminated, and computation time is mainly concentrated in difficult situations.

In order to evaluate D-Nets based on the precision recall rate, we must correlate each matched quality metric to sort the matches. The value in the i∗i∗ row in the corresponding grid is expressed as g(j)=G[i∗,j]g(j)=G[i∗,j], we choose $j^* := arg_j(g^* := maxg(j))$ is the best match. We define a quality metric as the selection consistency correspondence i∗↔j∗ as its entropy normalized vote $q := \frac{g^*}{-\sum_{\forall_j} p_j log_2 p_j}$ where $p_j := \frac{g(j)}{\sum_{\forall_k} g(k)}$.

Algorithm 1 D-Nets Voter(G, L_i, L'_i)

clear G
for all $d \in D$ **do**
  $\mathcal{B}_d := \{(\mathbf{e}, \mathbf{e}') | \mathbf{e} \in L_d, \mathbf{e}' \in L'_d\}$
  $v \leftarrow 1/|\mathcal{B}_d|$
  for all $((\mathbf{a}_i, \mathbf{a}_j), (\mathbf{b}_k, \mathbf{b}_l)) \in \mathcal{B}_d$ **do**
    $G[i, k] \leftarrow G[i, k] + v$
    $G[j, l] \leftarrow G[j, l] + v$
  end for
end for

Figure 6

3 Line matching based on D-nets

After understanding the original method, we found that an important factor affecting the accuracy of the original method is the accuracy of matching feature point pairs, and D-nets just improves the accuracy of the traditional SIFT matching algorithm**[3]**. Therefore, we replace the feature points obtained by the original SIFT algorithm with the matching feature points obtained by D-nets. Before the replacement, we can see that the accuracy of the points matched by D-nets will increase, but the number of matched points will be reduced. Generally speaking, only half of the original SIFT algorithm.

However, this phenomenon has two-sides. In some pictures with large scale changes, because the matching feature points are too small, the number of matching straight lines is greatly reduced. This is because there are many straight lines in the neighborhood. The feature points are caused by pairs. However, in the picture where the scale change is not very large, D-nets reduces the matching point logarithm, but these matched feature points are evenly distributed in the picture, which can satisfy the point logarithm requirement of most lines. Therefore, in this case, since the accuracy of the point pair is improved, the accuracy of the original method is improved, and since the number of pairs of feature points used is reduced, the time taken for the line matching is much reduced.

Since the feature points are the core of the original method, we then combined and compared the feature points generated by D-nets and SIFT. The experimental results are shown in the experimental part.

3.Experiment

3.1 D-nets combined with the improvement of the original method

We export the matched-point information matched by the D-nets program from the data structure to the text document, so that it can replace the feature points matched by the SIFT method in the original project. After that, we can run the feature-based script. The number of lines matches the script. At this time, the feature point pairs of the matching lines in the script are completely the feature point pairs matched by D-nets. The difference between the D-nets method and the SIFT method is that the accuracy of the points is Increased, but the number of points becomes very small. The effect of this is two-sided. On the one hand, some large spatial scale transformations may result in fewer matching lines and lower precision due to insufficient points, but in the case where a small number of points can cover the entire picture, the drop in points brings about a dramatic increase in speed. Moreover, the accuracy has been improved, which can be regarded as a good experimental result.

Results

Boat-1,2   Boat-1,5   Graf-1,2

Experiment	Line number	Origin method	Point pairs number	D-Nets improved	Point pairs number
Boat1-2	(552/445)	95.33% (306/321)	4320	98.47% (323/328)	1164
Boat1-5	(548/270)	86.73% (85/98)	2903	84.53% (82/97)	304
Graf1-2	(730/311)	94.91% (392/413)	1957	96.92% (410/423)	431

Figure 7

For the combined algorithm, we can clearly see that the number of extracted feature points is much lower than the sift feature point extraction algorithm used in the original characteristic number line matching algorithm. Too few points mean that many feature points that should be detected are not detected. To this end, we have designed an algorithm to solve this situation. Because the accuracy of the feature points that D-nets can extract is very high and is absolutely stable after detection, theoretically we believe that the extra points generated by these stable points are also stable. Figure 8 shows the extra point generation algorithm we designed.

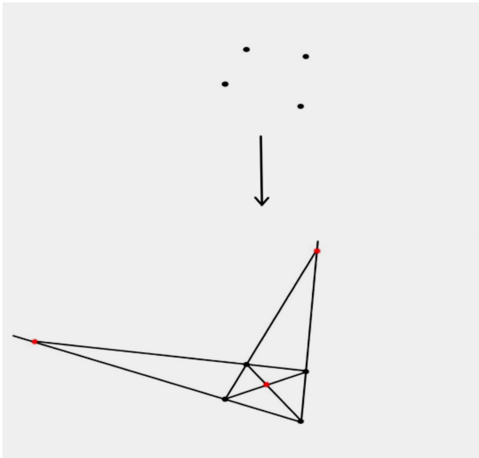


Figure 8

We found that every four points can generate three additional points. Of course, there are special cases. For example, when three points are col-linear, the three additional points generated by these four points coincide with three of the existing four points. But this has no effect on our experiments. After one such expansion, we can generate new points and the number is 75% of the original points. Moreover, the experimental results also show that these new points can be stably corresponding to the position of the feature points in the image that should be generated but not detected. By analogy, we can make a second expansion, three expansions... until we meet our specific needs, so the point problem is solved perfectly.

Experiment	Line number	D-nets	Point pairs number	Increase points	Point pairs number
Boat1-5	(548/270)	84.53% (82/97)	304	86.6% (78/90)	416
Graf1-2	(730/311)	96.92% (410/423)	431	96.94% (318/331)	535

Figure 9

After solving the problem of missing D-nets points, we will perform the experiment of straight line matching, and Figure 5 is the experimental result. In the figure, we can clearly see that after the algorithm is improved, our matching accuracy is higher than that of the original characteristic number straight line matching algorithm with the same number of points; and when the same accuracy is achieved, we use Fewer points.

It can be seen that the intersection expansion method improves the number of point pairs and makes the point pair distribution more uniform. In the case of Boat1-5 with a large scale change, the matching line accuracy is improved.

4.Conclusion

In this paper, we have improved a new line-point projective invariant, which improves the performance of the line-matching algorithm based on this invariant. At the same time, it also provides several good ideas for improving this method. Experimental results show that the proposed method is robust to many distortions and can achieve better performance than some of the most advanced methods available, especially in images with low texture and large viewpoint changes. But for large scale transformations, we still have a lot of work to do.

5.Reference

1. Qi Jia, Xinkai Gao, Xin Fan*, Zhongxuan Luo, Haojie Li, Ziyao Chen:***Novel Coplanar Line-points Invariants for Robust Line Matching Across Views***

2. Felix von Hundelshausen Rahul Sukthankar Institute for Autonomous Systems Technology (TAS), University of the Bundeswehr Munich Google Research and The Robotics Institute, Carnegie Mellon University:***D-Nets: BeyondPatch-BasedImageDescriptors:CVPR 2012***

3. D. G. Lowe. :***Distinctive image features from scale-invariant keypoints. IJCV, 60(2), 2004.***

4.Al-Shahri, M., Yilmaz, : ***Line matching in wide-baseline stereo: A top-down approach. Image Processing, IEEE Transactions on 23(9), 4199 – 4210 (2014)***

5. Fan, B., Wu, F., Hu, Z.: ***Robust line matching through line – point invariants. Pattern Recognition 45(2), 794 – 805 (2012)***