ACCURATE AND RELIABLE FRAMEWORK FOR FAST PARAMETRIC CURVES DETECTION[1]

A. Levashov [2], D. Yurin [3]

[2,3] Laboratory of Mathematical Methods of Image Processing
Faculty of Computational Mathematics and Cybernetics
Lomonosov Moscow State University
Leninskie Gory, Moscow 119991 Russia
alexeylevashov89@gmail.com [2], yurin@cs.msu.su [3]

We propose a multi-stage framework for parametric curves detection in images. Firstly, we search edge curves, then, after several preparatory procedures, they are stored as chains of connected pixels. These chains are analyzed both as whole and as piecemeal. Analysis of each fragment firstly uses randomized methods to reject certainly inappropriate models. Only for remaining hypothesis we perform accurate model parameter estimation by least squares method and check reliability of a hypothesis by chi-square criterion. Such two stage approach allows us achieve high performance, search parametric curves of different types simultaneously, including cases when different fragments of one curve correspond to different models. The measurement results of parameters estimation accuracy and algorithm operation time are given.

Introduction

One of the key problems in image processing and computer vision is detection of objects of interest on images. Usually such objects are artificial objects, so the task of finding of geometric primitives such as straight lines, circles, ellipses, etc. is very important.

There are many methods that solve this problem. Approaches based on Hough transform [4] are slow, use a large amount of memory and focus on finding only one type of curves. Significant improvement of these approaches is use of random sampling [8], [1], [6]. Approach [1] requires very small amount of memory. However, a common disadvantage for randomize methods are missing of curves in complex images due to low probability of that a random sample set of points belongs to a single curve. Another serious disadvantage of both approaches is missing of small length curves. It is due to two main reasons. The first is a small maxima value in parametric space for such curves. And the second reason is that when transform to parametric space we lose information about points positions, so two situation becomes indistinguishable: a set of connected points and a set of same number and brightness sparse points.

The approach based on beamlets [3] is also focused on finding single type of curves.

We proposed a framework for detection of parametric curves of different types simultaneously, including cases when different fragments of one curve correspond to different models. Our approach based on analyzes of chains of connected pixels and uses random methods only for primary analysis to exclude certainly inappropriate hypothesis.

Algorithm overview

The general scheme of our algorithm (Fig. 1) consists of several steps. First step is edge detector. We use Canny detector for grayscale images and Di Zenzo/Cumani detector for color images [7], both with subpixel accuracy [2]. These detectors give edge curves with small number of breaks and almost everywhere 1 pixel thickness. The next step is edge curves preprocessing. We thin edge

[1] The research is done under support of Federal Target Program "Scientific and Scientific Pedagogical Personnel of Innovative Russia" in 2009-2013 and the Russian Foundation for Basic Research, project no. 09-07-92000-HHC_a.

curves with thickness more than 1 pixel [9]. Then we remove branch points from edges. It simplifies our algorithm and allows us to collect curves points into lists on the next step instead of graph. On the other hand, edge detection accuracy in the neighbourhood of branch points much lower, so excluding such points from consideration looks like disturbance removal. In the lists each point is represented by its coordinates, direction and magnitude of (color-) gradient. All further analysis use only vectorized data instead of pixel representation. Final decision on curve detection is accepted based on χ^2-criterion.

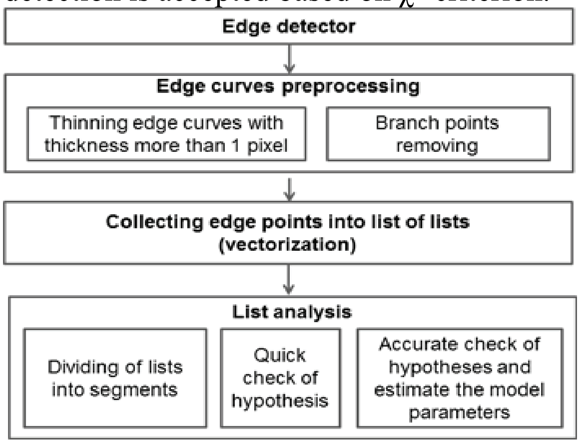

Fig 1. General scheme

Removing of branch points

We remove branch points after edge line thinning, so all edge curves have 1 pixel thickness at this step. Our branch detection algorithm use ideas and notations similar to [2].

Let P is pixel on image. If P is edge pixel than $P = 1$, otherwise $P = 0$. For each edge point with (i, j) coordinates take a window of size 3×3 and enumerate pixels in it in the following order

$P_1(i,j), P_2(i-1,j), P_3(i-1,j+1), P_4(i,j+1), \ldots, P_9(i+1,j-1)$.

Calculate two values: $J(P_1) = \sum_{i=2}^{9} P_i$ and $K(P_1)$ is number of patterns 01 in the sequence $P_2 \ldots P_9 P_2$. If $J(P_1) \geq 5 \vee K(P_1) > 2$ than P_1 is branch point.

Curves analysis algorithm

General scheme of curves set analysis is represented by ***Algorithm 1*** - ***Algorithm 4***. More formal and detailed description of curves analysis algorithm is given in [5].

Algorithm 1. `FindParamCurves()` *Search for parametric curves in a list of curve points*

```
1.  for each edge curve do
2.   for each model do
3.     if QuickTest then do
4.      if FindParamCurve then do
         Add points from adjacent
         segments to current curve
5.       and refine the model
         parameters. Store
         found model.
6.    if model not found then do
7.       Separation
```

Algorithm 2. `QuickTest()`: *Quick test current hypothesis on current curve fragment*

```
1. Randomly take N points from
   curve. ( N - is minimal
   number of points required for
   model parameters estimation)
2. Estimate parameters of model
   using these N points.
3. Randomly take one more control
   point from curve.
4. If control point lies nearby
   model curve quick check is
   passed.
5. Repeat steps 1-4 several times
   and if count of quick check
   is reasonable, suppose that
   QuickTest is passed.
```

Algorithm 3. `FindParamCurve()`: *Estimate parameters of model using all curve points.*

```
1. Estimate model parameters by
   Least Square Method.
2. Reliability of each acceptable
   model for this curve is
   estimated with χ2-criterion.
3. Chosen a most reliable and
   simple model.
4. If no any model satisfied to
   χ2, model is not found.
```

Algorithm 4. `Separation()`: *Recursive separation of curve into 2 parts.*

```
1. Edge points of the curve
   divided in half parts.
2. For each part invoke
   FindParamCurves
```

[1] The research is done under support of Federal Target Program "Scientific and Scientific Pedagogical Personnel of Innovative Russia" in 2009-2013 and the Russian Foundation for Basic Research, project no. 09-07-92000-HHC_a.

Results

Timing experiments were made on computer with CPU Intel Core 2 Duo 2.00 GHz, RAM 2 Gb. In our C++ realization we search for two type of parametric curves: lines and circles. Times of our algorithm and comparison with algorithm [1] are presented in table 1. Dashes indicate tests for which algorithm [1] is not computed. Example of results shows on fig. 2

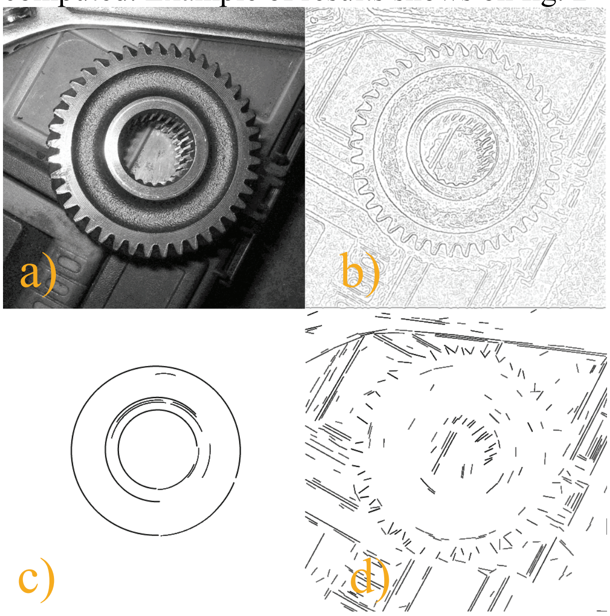

Fig 2. a) Source image; b) image of gradient magnitude after NMS [2]; c) arcs of circles that our algorithm found; d) straight lines that our algorithm found.

We compute accuracy of our method by follows way:

1) Generate test circles with different radius and with different visible parts of arcs on images. Add normal noise with different σ_{noise} to the whole image (Fig. 3g).
2) Detect edges on test images (Fig. 3e).
3) Find parametric curves with proposed framework.
4) Compare found parameters with those for initially generated circle.

For each radius, visible part of arc and σ_{noise} we generate 1000 test circles and then we calculate mean error of center offset (in pixels), radius error (in pixels) and circle missing probability (0...1).

There are 6 dependencies are shown in fig.3. 1st row shows error of center offset, 2nd row shows radius error, 3rd row shows missing probability. Columns represent experiments with different radius. On each graph x-axis represents visible part of arc (in degrees).

Fig. 3 shows that small arcs of circles with small radius are more resistant to miss detecting. This is because small arcs under influence of noise have less chance to break during edge detection.

Conclusion

Framework for parametric curves detection on grayscale and color images with reliability control is developed. Comparison with algorithm [1] shows similar time, but our method finds simultaneously different types of curves, has no restrictions on complexity of images and on length of edge curves. Proposed algorithm has low probability of missing curve. Another curve types detection can be easy incorporated in the proposed framework.

References

1. Chen T.C., Chung K.L., An Efficient Randomized Algorithm or Detecting Circles. // CVIU 2001, V.83, P.172–191.
2. Devernay F. A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy // INRIA techreport RR-2724 -1995. -20 p.
3. Donoho D.L., Huo X., Jermyn I., Jones P., Lerman G., Levi O., Natterer F. Beamlets and Multiscale Image Analysis //In Multiscale and Multiresolution Methods, Springer 2001, P.149-196.
4. Duda R., Hart P. Use of the Hough Transformation to Detect Lines and Curves in Pictures. Comm. ACM 15: 1 – 15, 1972.
5. Levashov A.E., Yurin D.V. Fast framework for parametric curves detection on grayscale and color images with reliability control // 21-th International Conference on Computer Graphics GraphiCon'2011. Moscow, Russia, 2011, pp. 212–215. (In Russian).
6. McLaughlin R.A. Randomized Hough Transform: better ellipse detection // IEEE TENCON - Digital Signal Processing Applications, V. 1, P. 409-414, Nov. 1996.
7. Pratt W.K. Digital Image Processing: PIKS Scientific inside (4th ed.) // Wiley-Interscience, John Wiley & Sons, Inc., Los Altos, California. -2007. P. 782.
8. Xu L., Oja E. Randomized Hough Transform // in Encyc. of Artif. Intell., Ed.By: J.Ramón, R.Dopico; J.Dorado; A.Pazos, IGI Global publishing comp., 2008, P.1354—1361.
9. Zhang T.Y., Suen C.Y. A fast parallel algorithm for thinning digital patterns // Communications of the ACM, V. 27, No. 3, P. 236--239, 1984.

[1] The research is done under support of Federal Target Program "Scientific and Scientific Pedagogical Personnel of Innovative Russia" in 2009-2013 and the Russian Foundation for Basic Research, project no. 09-07-92000-HHC_a.

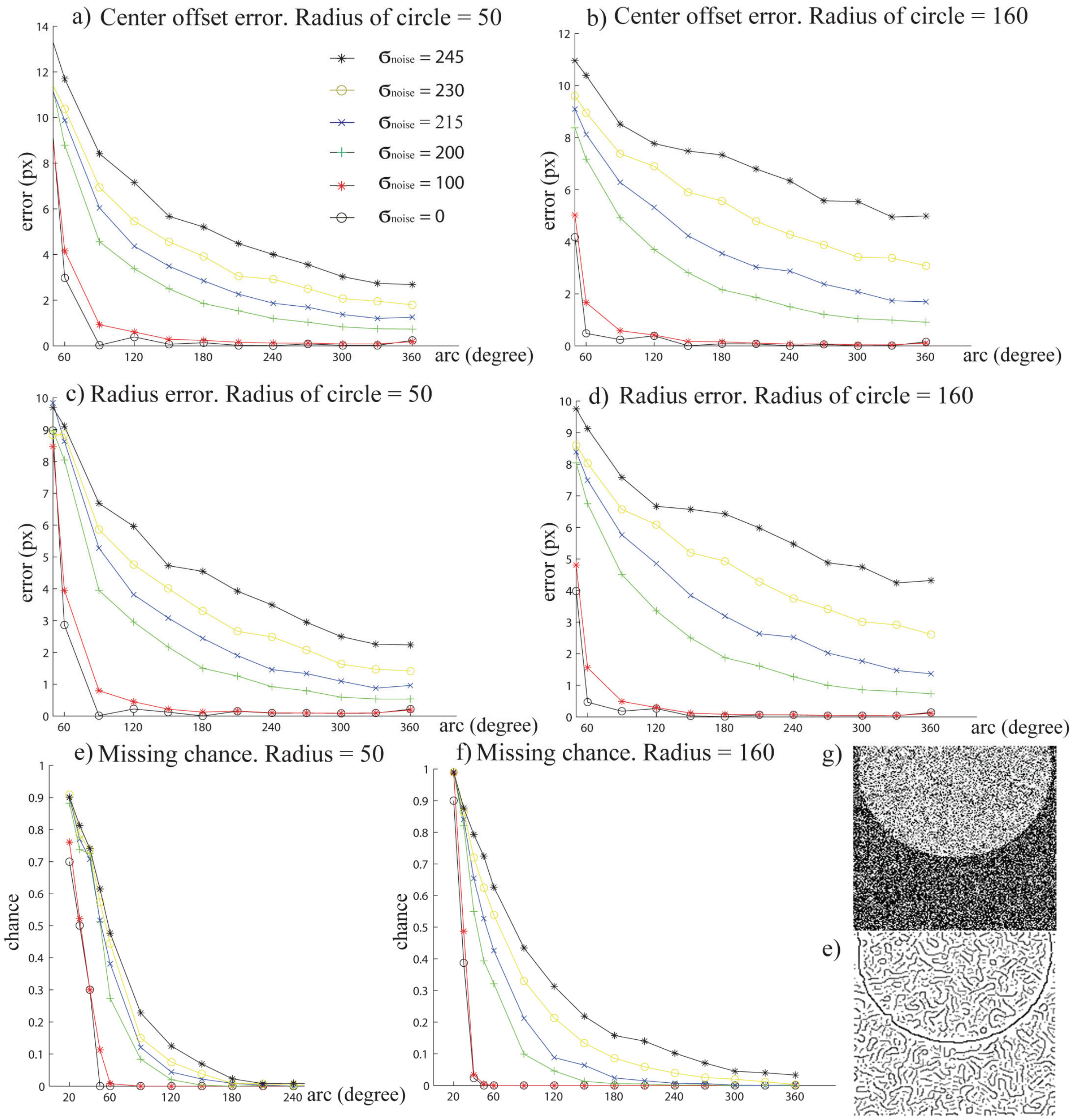

Fig. 3. Accuracy results, g) is one of test images, e) is edges founded on g)

	Image size	Edge pixels	Edge lines	Elapsed times, ms									Test images
				1	2	3	4	5	6	7	8	9	
1	500×500	35776	3409	147	10	10	11	15	46	3	45	1	
2	381×381	5279	388	74	3	3	3	2	11	1	18	1	
3	256×256	13063	1511	11	4	3	6	8	21	5	54	3	
4	256×256	11839	1473	11	3	3	5	7	18	7	55	4	
5	256×256	9167	1000	11	2	3	4	5	14	6	20	0	
6	256×192	7938	689	9	2	3	8	5	18	5	20	2	
7	1024×1024	228526	30932	372	106	88	85	128	407	3	-	-	
8	945×1155	136800	9849	625	50	40	50	70	210	0	-	-	

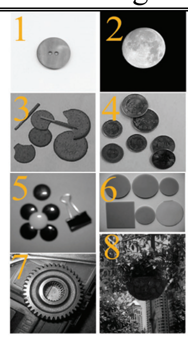

Fig. 4. Test images

Table 1. Execution times. Numbered columns contain times in milliseconds of (1) edge detector (2) thinning curves, (3) removing nodes, (4) vectorization, (5) finding parametric curves, (6) total execution time of our algorithm without edge detection (7) number of detected circles with our algorithm, (8) total execution time of algorithm [1] without edge detection, (9) number of detected circles with algorithm [1].

[1] The research is done under support of Federal Target Program "Scientific and Scientific Pedagogical Personnel of Innovative Russia" in 2009-2013 and the Russian Foundation for Basic Research, project no. 09-07-92000-HHC_a.