

Definition of line formula on images

Mamatov Narzillo¹, Samijonov Abdurashid², Niyozmatova Nilufar³, Dadakhanov Musokhon⁴, Rahmonov Erkin⁵

^{1,3}Tashkent University of Information Technologies named after Al-Kharezmi, Tashkent, Uzbekistan

²Bauman Moscow state technical university, Moscow, Russia

⁴Namangan State University, Namangan, Uzbekistan

⁵Tashkent Pharmaceutical Institute, Tashkent, Uzbekistan

Abstract. The article considers the problem of determining line formulas in a monochrome image. To solve this problem, a scheme, method and algorithm for determining the line coefficients using a finite difference are proposed. Based on the proposed method, it is also possible to solve the problems of approximation and interpolation.

1. Introduction

Currently, in various fields of human activity, systems that work with images are widely used. For example, seismology, military science, medicine, radar, astronomy and others. The foundations of these systems are methods and algorithms for analysis, processing and recognition of images. The problem of determining line formulas in images is one of the important problems in creating image processing and recognition systems.

Typically, the lines in the images have a complex graphic construction and in practice they cannot be expressed by formulas. But, with the help of smart algorithms, most of these complex lines can be divided into lines that can be expressed by mathematical formulas. This process is called the definition of image primitives. For example, a line, line segment, circle, ellipse, etc.

Basically, when determining line formulas, methods such as interpolation, approximation, and extrapolation are used. But these methods cannot be used directly in computer graphics and image processing. Since, when entering the image through the device, there may be noise, blur, different line thicknesses or additional points.

Given the discrete properties of images, in addition to finding the exact mathematical formula, the article also considers the search for approximate functions of lines that cannot be expressed through formulas.

Usually in a real image, the lines have different thicknesses. This requires solving the pre-processing problem, that is, reducing the image to certain normalized conditions. In this paper, it is believed that the lines are binarized and have a thickness of 1 pixel. Since the primary processing algorithms are sufficiently developed [1]. Therefore, this stage is not considered in this paper.

The problem of determining the main features of an object based on its contour lines is solved. For example, when identifying a person based on the auricle, the auricle region is distinguished in the first stage. At the second stage, contour lines and signs of the auricle are distinguished. And the contour lines are preliminary approximated to the curves, which can be estimated by the parameters of the analytical curve. These parameters can be used to create the database and specific indicators of the object, they can also be used as independent indicators of the characteristic features of the image, or as an intermediate element for highlighting more complex features.

Currently, work is underway to recognize lines in the image. For example, the Radon transform is a well-known integral transform that maps functions f in the original space to a function on the set of planes defined by the integrals of f along these planes [1]. Numerous variants of integral transformations are known, which are modifications or generalizations of this approach [2, 3, 4]. A



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

common way to search for parametric curves in images is the Hough transform [5]. Its essence is that from the set of image points, subsets of points are selected, by which the parameters of the curve passing through these points are calculated. A counter is associated with each point in the parametric space, showing for how many subsets this set of parameters was calculated. The parameters with the maximum counter value determine the curves found in the image. The number of subsets to be sorted exponentially increases with the number of curve parameters, which leads to a large number of calculations. In this regard, many approximate methods have been developed for calculating the Hough transform, based on enumerating not all subsets of points, but only some of them [6-8]. In [9], an algorithm for calculating the Radon transform through the two-dimensional Fourier transform was proposed. In [10], an algorithm for calculating the Radon transform through the two-dimensional Hartley transform was proposed. The memory requirements are two times less than in [9]. The advantage of the Hartley transform over Fourier is that the former is real and requires less memory for its calculation and, as a result, processing less data, less computation time. In [11, 12], an algorithm for calculating the integrals of an image along straight lines is given. This can be used to calculate the Radon transform of an image.

Despite the existence of various line recognition methods, they are aimed at recognizing lines of one subclass. And in other cases, interpolation and approximation are used. In many cases, the lines in the image are graphs of some functions. In such cases, it is required to develop new methods and algorithms for determining line formulas.

Next, we consider the problem of finding formulas based on graphs of the functions of a line, a parabola, a cubic parabola, and n-order polynomials.

2. Statement of a problem and the concept of the problem decision

Suppose we are given a binary image with size $N \times M$ and let the lines on it be localized. We take one of the localized lines (Fig. 1) and compile the table below.

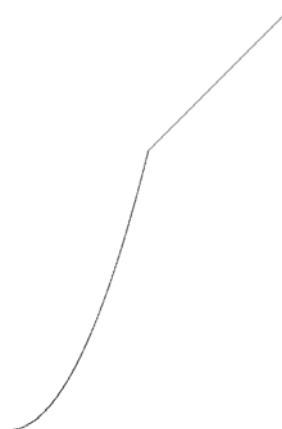


Figure 1. Parabola and a straight line.

Table 1. Initial data

x_0	x_1	x_N
y_0	y_1	y_N

Using Table 1, you can build a maximum of an n-order polynomial, i.e. $y = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$.

The construction of the polynomial is reduced to determining the coefficients a_i . These coefficients are determined on the basis of finite differences.

Line formulas are defined in the following steps.

Stage 1 Definition of a line. At this stage, the finite differences of the first order are calculated, and Table 1 continues (Table 2).

Finite first-order differences are calculated by the following formula:

$$\Delta^1 y_i = \frac{(y_i - y_0)}{i}, i = \overline{1, N};$$

here y_i are the values of the function for the corresponding x_i .

Table 2. Initial data and finite differences of the first order.

x_0	x_1	x_2	x_N
y_0	y_1	y_2	y_N
0	$\Delta^1 y_1$	$\Delta^1 y_2$	$\Delta^1 y_n$

Next, a number of satisfying conditions is determined $\Delta_{ij} = |\Delta^1 y_i - \Delta^1 y_j| \leq \varepsilon, i = \overline{1, N-1}, j = \overline{i+1, N}$, where ε is the parameter of the algorithm. Points corresponding to a certain row make up a straight line. The coefficients of the line are determined as follows:

$$a_0 = y_0, a_1 = \frac{1}{N} \sum_{i=1}^N \Delta^1 y_i$$

Stage 2 Definition of a parabola of the second order. In this case, the finite differences of the second order are calculated, and Table 2 continues (Table 3). Finite second-order differences are calculated by the following formula:

$$\Delta^2 y_i = \frac{(\Delta^1 y_{i+1} - \Delta^1 y_i)}{i}, i = \overline{1, N-1};$$

Table 3. Initial data and finite differences of the first and second order.

x_0	x_1	x_2	x_3	x_N
y_0	y_1	y_2	y_3	y_N
0	$\Delta^1 y_1$	$\Delta^1 y_2$	$\Delta^1 y_3$	$\Delta^1 y_N$
0	0	$\Delta^2 y_1$	$\Delta^2 y_2$	$\Delta^2 y_{N-1}$

Next, a number of satisfying conditions is determined: $\Delta_{ij} = |\Delta^2 y_i - \Delta^2 y_j| \leq \varepsilon, i = \overline{1, N-1}, j = \overline{i+1, N}$. Points corresponding to a certain series constitute a second-order parabola. The second-order parabola coefficients are defined as follows:

$$a_0 = y_0, a_1 = \frac{1}{N} \sum_{i=1}^N \Delta^1 y_i, a_2 = \frac{1}{N-1} \sum_{i=1}^{N-1} \Delta^2 y_i$$

Stage 3 Definition of a cubic parabola. In this case, the finite differences of the third order are calculated, and table 3 continues.

Stage k Definition of an k-order algebraic polynomial. In this case, finite differences of the k-order are calculated, and the table continues similarly to the previous step.

Finite differences of higher orders are calculated by the following formula:

$$\Delta^k y_i = \frac{(\Delta^{k-1} y_{i+1} - \Delta^{k-1} y_i)}{i}, i = \overline{1, N-k};$$

Table 4. Initial data and finite k-order differences.

x_0	x_1	x_2	x_k	x_N
y_0	y_1	y_2	y_k	y_N
0	$\Delta^1 y_1$	$\Delta^1 y_2$	$\Delta^1 y_k$	$\Delta^1 y_N$
0	0	$\Delta^2 y_1$	$\Delta^2 y_k$	$\Delta^2 y_{N-1}$
....
0	0	0		$\Delta^k y_1$	$\Delta^k y_{N-k+1}$

Next, a number of satisfying conditions is determined: $\Delta_{ij} = |\Delta^k y_i - \Delta^k y_j| \leq \varepsilon, i = 1, N - k - 1, j = i + 1, N - k$. The points corresponding to a certain series constitute a k-order polynomial. The coefficients are determined as follows:

$$a_0 = y_0, a_k = \frac{1}{N - k + 1} \sum_{i=1}^{N-k+1} \Delta^k y_i$$

Results of computational experiments.

Evaluation of the effectiveness of the described algorithm was carried out on various images: ideal graphics, distorted and manual images. Vectorization was carried out on the basis of the Canny boundary detector with subsequent processing and localization of graphs. To calculate the effectiveness of the algorithm, the formula was used:

$$E = \frac{R}{N} \cdot 100\%$$

where R -is the correctly found line formula, N -is the total number of localized lines in the image.

In the first experiment, 1000 plots of linear and polynomials of various orders generated by the program were taken. All graphs were reduced to zero abscissa, one pixel was taken as a single segment and $\varepsilon = 0$. In this case, the algorithm gave a result of 100%.

In the second experiment, 100 manual images were taken that were close to linear and polynomials of different orders. In this case, the error of coincidence of the image pixels with the found function graph was no more than 15%, where $\varepsilon = 1$.

3. Conclusion

The article solves the problem of determining line formulas in a monochrome image. To solve this problem, a scheme, method and algorithm for determining the line coefficients using a finite difference are proposed. Here, the line formulas of the localized graphs are determined on the basis of the proposed advanced Newton method. Based on the proposed method, it is also possible to solve the problems of approximation and interpolation.

The proposed algorithm is based on finite differences; in existing methods, the difference between neighboring points is considered. In the present method, an approximate function is constructed at the starting point, and it may have some equal initial values y .

In the Lagrange interpolation method, each member of the interpolation depends on the entire nodes. If a new node is introduced, then it is necessary to reconstruct the interpolation polynomial. This problem is solved on the basis of Newton's interpolation method. The proposed method is based on the Newton method; therefore, the interpolation accuracy is estimated as in the Newton method.

References

- [1] Couprie M 2005 Note on fifteen 2D parallel thinning algorithms IGM2006-01 Universite de Marnela-Vallee
- [2] Toft P 1996 The Radon Transform PhD Thesis
- [3] Hough P V C 1962 Method and means for recognizing complex patterns US Patent 3,069,654
- [4] Kotlyar V and Kovalev A 2003 Ring Radon Transformation Computer Optics No 25 pp 126–133
- [5] Princen JP Illingworth J and Kittler JV 1992 A Journal of Mathematical Imaging and Vision vol 1 num 1 pp 153-168
- [6] Kiryati N Eldar Y and Bruckstein A 1991 Pattern Recognition vol 24 pp 303-316
- [7] Xu L Oja E and Kultanen P 1990 Pattern Recognition Letters vol 11 no 5 pp 331-338
- [8] Kalvianen H and Hirvonen P 1995 *Proc. 9th Scandinavian Conference on Image Analysis* (Sweden: Uppsala) pp 1029-1036
- [9] Cheyne Gaw Ho Rupert CD Young Chris D Bradfield Chris R Chatwin 2000 Real-Time Imaging vol 6 num 2 pp 113-127
- [10] Volegov D Gusev V and Yurin D 2006 *Proc. 16th International Conference on Computer*

Graphics and Application GraphiCon (Russia: Novosibirsk)

- [11] Donoho D and Huo X *Beamlets and multiscale image analysis* <http://www-stat.stanford.edu/donoho/Reports/> <http://citeseer.ist.psu.edu/donoho01beamlets.html>
- [12] David D and Xiaoming H *Applications of Beamlets to Detection and Extraction of Lines, Curves and Objects in Very Noisy Images* <http://citeseer.ist.psu.edu/446366.html>