Several methods to reduce computational complexity of the Radon and Hough transforms applied to line segment detection

Kobasyar M., Rusyn B., Varetskyy Y.

Department of methods and systems of pattern processing, analysis and identification
Karpenko Physico-Mechanical Institute of the NAS of Ukraine
dep32@ipm.lviv.ua

Abstract

The different methods of the computational complexity reduction for estimation of the parameter domain (PD) are presented, in particular image division into segments, partial estimation of PD with its repeated calculation and simultaneous calculation of projection of 1D RT interpolation. These methods are based on linearity property of Radon and Hough transforms. They allow essentially to increase speed of line segment detection algorithms.

1. Introduction

The Radon transform (RT) is a well-know scientific technique, which is used in many branches of the science, in particular within tomography, seismic, microscopy etc [1].

The linear RT can be defined on a more general form [1] and [2] as

$$\breve{g}(\xi_0,\xi_1,\xi_2,)=\int_{-\infty}^{+\infty}\int_{-\infty}^{\infty} g(x,y)\delta(\xi_0-\xi_1 x-\xi_2 y)dxdy\,, \qquad (1)$$

where $g(x,y)$ - density distribution of the image brightness, $\delta(\cdot)$ - Dirac delta function.

In eq.1 line was described with three degrees of freedom. For line these three parameters should always have a link, which removes one degree of freedom. In the literature two forms are mentioned as the most common. They are (p,τ) (or slant stacking) and (ρ,θ) (or normal form) [3]. (p,τ) RT can be defined as

$$\breve{g}(p,\tau)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)\delta(y-px-\tau)dxdy\,, \qquad (2)$$

where $\xi_0=-\tau$, $\xi_1=p$, $\xi_2=-1$, τ is the offset from origin of the coordinate system to the line, respectively p is the line slope.

Slant stacking has some limitations, for instance in the case of vertical lines. That's why scientists usually use normal form of RT.

(ρ,θ) RT can be defined as integral of density distribution of the image brightness $g(x,y)$ along the integration line s [1]

$$\breve{g}(\rho,\theta)=\int_{-\infty}^{+\infty} g(x,y)ds=$$

$$=\int_{-\infty}^{+\infty} g(\rho\cos\theta-s\cdot\sin\theta,\rho\sin\theta+s\cdot\cos\theta)ds=$$

$$=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(x,y)\delta(\rho-x\cos\theta-y\cdot\sin\theta)dxdy\,, \qquad (3)$$

where ρ is a shortest distance from the origin of the coordinate system to the integration line s , θ is an angle corresponding to the angular orientation of the line s and abscissa axis.

The discrete value of $\breve{g}(\rho,\theta)$ for any combination of ρ and θ is named projection sum. The set of values $\breve{g}(\rho,\theta)$ for $\theta=const$ is named projection. The full system of equations for $\theta\in[0;\pi]$ forms parameter domain (PD).

The derivation of different forms of eq.3 gives the relationship between (p,τ) and (ρ,θ) RT

$$\breve{g}(\rho,\theta)=\frac{1}{|\sin\theta|}\int_{-\infty}^{+\infty} g\left(x,\frac{\rho}{\sin\theta}-xctg\theta\right)dx=$$

$$=\frac{1}{|\sin\theta|}\breve{g}(p,\tau)\,, \qquad (4)$$

where $p=-ctg\theta$, $\tau=\dfrac{\rho}{\sin\theta}$.

For very sparse images (for instance a binary image with few number of non-zero pixels) most of computer time is spent for summing up zeros (obviously, zeros do not contribute to the PD). In his world-know patent [4] P.V.C. Hough proposed approach to reduce computational cost using prior knowledge.

The relationship between Radon and Hough (HT) transforms is very simple. HT maps the individual pixels from the image domain (ID) into a shape in the PD. The RT transforms the line (shape in general) from the ID into a single pixel in the PD. The identical result for both of these transforms can be derived in the (p,τ) case only. For normal form the results will differ. In this paper [5] was considered different strategies of choosing sampling parameters. It also should be mentioned that HT is simple in discrete form only.

A huge number of algorithms for calculation of RT and HT were proposed. Some of them are: fuzzy RT [6], random RT [7], the gradient method of line segment detection, the hierarchical HT [3], a length invariant transform, trace transform [8] etc.

2. Computational complexity of RT and HT

The one way to calculate discrete (ρ,θ) RT is approximation of the eq.3

$$\breve{g}(\rho_\tau,\theta_t) \approx \Delta s \sum_{l=0}^{L-1} g(x_l,y_l), \qquad (5)$$

where s_l is a linear sampling of the s variable, $x_l = \rho_\tau \cos\theta_t - s_l \sin\theta_t$, $y_l = \rho_\tau \sin\theta_t + s_l \cos\theta_t$.

This approach has several limitations. The most serious is that for given value l the image points in eq.5, i.e. $(x,y)=(x_l,y_l)$, in principle never coincide with samples of the image $g(m,n)=g(x_m,y_n)$. Hence the interpolation of x and y should be done. We can use 1D interpolation instead of 2D one. Another problem is choosing of sampling parameter s.

The most used way of (ρ,θ) RT is a usage of 1D interpolation in y-direction, which uses eq.1 of (ρ,θ) RT and relationship between (p,τ) and (ρ,θ) RT of eq.4.

Due to [3] computational complexity of (ρ,θ) RT using nearest neighbour approximation, linear interpolation and sinc-interpolation can be approximately given by

$$\begin{aligned} O_{NN} &= O(RTN) \approx O(N^3), \\ O_{Linear} &= O(RTN) \approx O(N^3), \\ O_{Sinc} &= O(RTN^2) \approx O(N^4), \end{aligned} \qquad (6)$$

where $O(\cdot)$ is the order function (in this function terms of lower order have been skipped). For instance, for $N=100$ computational complexity of sinc-interpolation is about 100 times higher than other two methods of interpolation. The linear interpolation is slower than a nearest neighbour approximation.

Assume that only a few number of pixels are non-zero. The computational complexity of HT is

$$O_{HT} \approx O((MN)_r M), \qquad (7)$$

where index r indices that a limited pixel numbers have to be transformed.

If there is one non-zero pixel in the image ($(MN)_r = 1$), then HT is the faster than RT. Usually value of (MN) fits interval M and MN depending on the image structure. This indices that HT is the most applicable for binary images, often rather sparse.

The research of computational time depending on image size for different interpolations of (ρ,θ) RT and HT is carried out. The results of this research are presented on fig.1. The charts 1-4 correspond to different optimization methods of (ρ,θ) HT using a nearest neighbour approximation. The charts 5 and 6 correspond accordingly a nearest neighbour approximation and linear interpolation of (ρ,θ) RT.

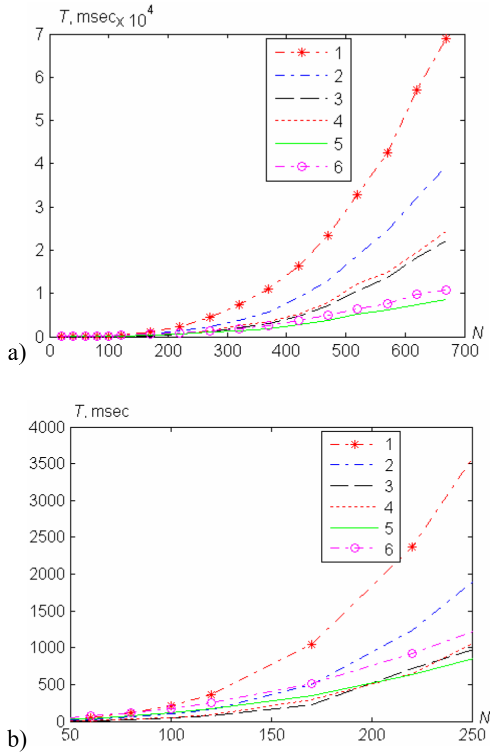

Fig.1. Time of PD's calculation T as a function of the image size $N \times N$ for different interpolation of (ρ,θ) HT (1-4) and (ρ,θ) RT (5-6). The value range of N is 0..700 (a) and 75..200 (b) pixels.

Using chart on fig.1 these conclusions can be done:

1. The initialization of the necessary vectors and arrays for (ρ,θ) HT requires a lot of time even if image contains a few number of non-zero image pixels.
2. In case of 10 and 10000 non-zero image pixels there is a huge time difference exists for (ρ,θ) HT . But even simple preprocessing can essentially increase time characteristics.
3. Time of PD's calculation for all interpolations of (ρ,θ) RT is roughly equal.
4. (ρ,θ) HT is much more faster than (ρ,θ) RT if image is presented by a few number of non-zero image pixels (up to 8-10% of total number of pixels, and image size is less 350×350 pixels). For larger images RT requires less computation time than HT. This is new and interesting fact.

The line segment detection usually requires large number of PD's calculations (using HT or RT). That's why the reduction of computational complexity of both transforms is very actual task.

3. Image division into segments

The simplest method of computational complexity reduction of RT and HT is an image division into equal square segments. In this case computational complexity of PD's estimation algorithm can be determined as

$$O_{HT/RT} = \sum_{k=1}^{N_S} O_{HT/RT}^{k}(N^w), \quad (8)$$

where N_S is a number of square image segments; w is a method calculation order coefficient ($w=3$ for RT, $w=2..3$ for HT depending on the image structure); $O_{HT/RT}^{k}$ is a computational complexity of estimation of PD for k -th segment using appropriate RT or HT. It is clear that computational complexity of RT doesn't depend on image structure. That's why eq.8 can be rewritten as

$$O_{RT} = N_S \cdot O_{RT}(N^3). \quad (9)$$

The method of image division into equal segments reduces total time of estimation of PD and essentially increases speed of the line segment detection algorithms.

We compared this approach for HT, 1D and 2D interpolations of RT. The research is carried out for line segment detection using different values of angle sampling $\Delta\theta$. Also RT without and with the usage of the transform symmetry properties is applied [8]. As an example satellite fragment of the real blocks of buildings is used. The image sizes are 256×256 pixels. After applying of Canny-Deriche operator [9] this image contains 3065 non-zero points. The space resolution of RT (sampling of ρ value) for all cases equals $\Delta\rho = 1$.

The results of research lead to these conclusions:

1. 1D and 2D RT with usage of the transform symmetry properties present respectively the best time and line segment detection results.
2. Decreasing values of angle sampling $\Delta\theta$ 2D RT with usage of the transform symmetry properties presents the best time and line segment detection results.

4. The partial estimation of PD with its repeated calculation using HT

For supplementary increasing of line segment detection algorithms computational cost approach of the partial estimation of PD with its repeated calculation using HT is proposed. It uses simple fact: after detection of the certain number of the line segments (even alone one) HT should be applied again and again to non-detected points. This procedure will be continued up to detection of all points or the most of the points (this depends on algorithm stopping criteria).

The proposed approach is based on the usage of linearity property of RT and HT [1]. It can be written such way: "RT of the weighted sum of functions equals weighted sum of transformations of each function". In the other words,

$$\breve{g}(\rho,\theta) = \sum_q \alpha_q \breve{g}_q(\rho,\theta), \quad (10)$$

where α_q , $g_q(x,y)$ is appropriately set of constants and functions. They allow to present image as $g(x,y) = \sum_q \alpha_q g_q(x,y)$.

Due to eq.10 PD of all points (line segments) is determined as the additive sum of all points (line segments).

The following algorithm of repeated calculation of PD for minimizing of the total operation number is proposed:

Step 1. Determine PD for all points in image domain (ID).
Step 2. Determine line segment (-s) from this PD.
Step 3. Remove points of detected line segment (-s) from ID.
Step 4. If $N_{ind} > N - N_{ind}$, i.e. number of points of detected line segment (-s) N_{ind} is large then number of the rest of points in the image ($N - N_{ind}$). Jump to the step 1.
Step 5. Determine PD only for points (from step 2) of detected line segment (-s).
Step 6. Determine PD only for the rest points in the image as a difference of full current PD (from step 1) and PD of detected line segment points (from step 2). Jump to the step 2.

The algorithm exit condition is checked on the step 2. To compensate time on step 4 it was experimentally determined that we should use condition $N_{ind} > k(N - N_{ind})$, where $k \in [1.05..1.1]$.

Now we'll determine number of operation needed for partial estimation of PD with its repeated calculation using HT. Suppose that image contains P non-zero points and these points can be presented with L line segments. In other words, $P = \sum_{l=1}^{L} P_l$, where P_l is a number of points in l -th line segment.

Suppose each iteration only alone line segment can be detected. Then we can easy estimate number of operation needed for PD's calculation of all possible Q line segments.

In a case of classic PD estimation approach. On the first iteration order of calculation, which is needed for transformation of points from into PD from ID, equals $O(P) = \sum_{l=1}^{L} P_l R_l$, on the second - $O(P) = \sum_{l=2}^{L} P_l R_l$, …, on the L -th - $O(P) = \sum_{l=L}^{L} P_l R_l$. In a case of classic approach the total order of calculation for estimation of all PDs can be written as

$$O(P) = \sum_{l=1}^{L} P_l R_l l, \quad (11)$$

where R_l is a number of operations needed for estimation of PD of a point on l -th operation.

In a case of proposed PD estimation approach first iteration requires to estimate full PD. That's why order of calculation needed for transformation of points into PD from ID equals $O(P) = \sum_{l=1}^{L} P_l R_l$. On the rest iteration points of detected line segments should be transformed, i.e. on the 2-d iteration $O(P) = P_1 R_1$, on the 3-th - $O(P) = P_2 R_2$ etc. On the l -th iteration $O(P) = P_{l-1} R_{l-1}$, where $l \in [2;L]$. In a case of proposed approach the total order of calculation for estimation of all PDs can be written as

$$O(P) = 2\sum_{l=1}^{L-1} P_l R_l + P_L R_L . \qquad (12)$$

The analyses of eqs.11 and 12 leads to this conclusion: usage of the partial estimation of PD with its repeated calculation using HT is effective in the case of large number non-zero points P in the image.

The experiments are carried out for different image fragments derived from satellites. In particular, for same image fragment used to compare interpolations of RT it is obtained these results: if $P = 6530$, time profit is 8.22%; if $P = 2703$ - 3.40%. But if $P = 2191$ this approach requires 0.84% of extra computational time.

5. Simultaneous calculation of projections for 1D RT

For 1D RT in y-direction i-th projection sum (see example in fig.2) is formed in the points $\breve{g}_{k,i,j}$, where k is the discrete angle value θ, i is the discrete offset value ρ (i-th projection sum), j is the number of image elements crossing by i-th integration line s. One of the main tasks is to determine the entrance and exit image elements for i-th integration line.

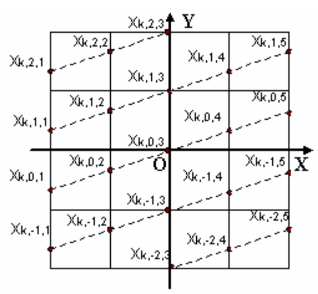

Fig.2. The estimation of PD in a case of 1D RT interpolation

x coordinate consecutively changes between its entrance and exit image elements. y coordinate can be determined from the line equation (see fig.2). Such calculation is a correct for $\theta \in [0^0;45^0]$ and $\theta \in [135^0;180^0]$. For $\theta \in [45^0;135^0]$ y changes consecutively and x coordinate can be determined from the line equation.

The computation time is mostly spent on estimation of the intermediate elements with the aid of the line equation. That's why it is proposed to determine y coordinates of alone projection simultaneously. x coordinate for all projection sums in point $x_{k,i,j}$ ($k = \theta = const$ i $j = const$) is equal. If $k = \theta = const$, then y coordinate of any adjacent projection sums $x_{k,i,j}$ and $x_{k,i+1,j}$ will differ on fixed offset (it can be determined from the line equation). Then y coordinate can be determined using previous y coordinate. So, y coordinate should be determined with the aim of the line equation only for $i = 0$.

The results of experiments are presented on the fig.3, where chart 1 is a classic approach, chart 2 is a minimal optimization of the classic approach and chart 3 is a proposed approach. We can see that proposed approach allows to reduce time of calculation of PD on ~13%.

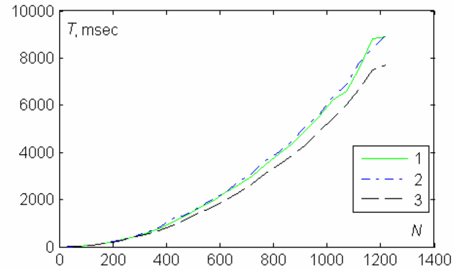

Fig.3. Time of PD T as a function of the image size $N \times N$ for 1D RT interpolation

6. Conclusions

The methods of image division into segments, partial estimation of parameter domain with its repeated calculation and simultaneous calculation of projection of 1D Radon transform interpolation are proposed. The experiments are carried out for different image fragments derived from satellites. All these methods allow to increase speed of parameter domain estimation.

7. References

[1] Deans S. R. *The Radon transform and some of its applications,* Florida, 1993.

[2] Ballard D., Brown C. *Computer Vision,* Prentice-Hall, 1982.

[3] Toft P. *The Radon Transform – Theory and Implementation*. Ph.D Thesis, Department of Mathematical Modelling, Technical University of Denmark, 1996.

[4] Hough P. V. C. *Methods and Means for Recognizing: Complex Patterns,* U.S. Patent 3069654, 1962.

[5] LamW. et al. "An Analysis on Quantization the Hough space", *Pattern Recognition Letters, Vol.15, 1994, p.1127-1135.*

[6] Kalviainen H., Hirvonen P., Xu L., Oja E. "Probabilistic and non-probabilistic Hough transforms - overview and comparisons", *Image and Vision Computing, Vol.13, № 4, 1995, p.239-252.*

[7] Kadyrov A., Petrou M. "The Trace Transform and Its Applications", *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.23, №8, 2001, p.811-828.*

[8] Kobasyar M., Rusyn B. "The general method of the fast Radon transform calculation with the usage of scheme calculation symmetry properties", *Radioelektronika I Informatika, №4(39), 2007, p.104-111*. (In Ukrainian)

[9] Canny J. "A Computational Approach to Edge-Detection", *IEEE Transactions* On Pattern Analysis and Machine Intelligence, 8:679-698, 1986.