HOUGH TRANSFORM: UNDERESTIMATED TOOL IN THE COMPUTER VISION FIELD

Dmitry P. Nikolaev
Institute for Information Transmission Problems, RAS
127994, Moscow, Russia
E-mail: dimonstr@iitp.ru

Simon M. Karpenko
Institute for Information Transmission Problems, RAS
127994, Moscow, Russia
E-mail: simon.karpenko@gmail.com

Ilia P. Nikolaev
Department of Physics
Moscow State University
119991, Moscow, Russia
E-mail: nikolaev@optics.ru

Petr P. Nikolayev
Institute for Information Transmission Problems, RAS
127994, Moscow, Russia
E-mail: nikol@iitp.ru

KEYWORDS

Hough transform, fast Hough transform, SIMD-optimized algorithms, document layout analysis, vanishing point detection, ICR

ABSTRACT

We discuss Hough Transform, some of its key properties, a scheme of fast and complete calculation of the Hough Transform (similar to the Fast Fourier Transform), and an efficient implementation of this scheme on SIMD processors. We also demonstrate an application of the Fast Hough Transform in computer vision by the example of an automatic page orientation detection unit incorporated in an intelligent character recognition system. Both 2D (scanner) and 3D (camera) cases of page acquisition are considered.

INTRODUCTION

Almost every modern textbook on image processing and computer vision (Pratt 2001; Gonzalez and Woods 2002; Ballard and Brown 1982; Forsyth and Ponce 2003) pays attention to Hough Transform (HT) (Hough 1959), the transformation of summation along various lines. In those textbooks containing no "HT" reference in the index, you will most likely find the description of Radon Transform, the transformation of integration along various lines in a continuous space. So, HT is a discretization of Radon Transform. This transform is traditionally used to solve the linear regression problem in the presence of a strong spike noise or the problem of decomposition of a distribution into linear submanifolds.

In Section 1, we discuss why HT is often not considered as an efficient tool to solve computer vision problems. In Section 2, a method for fast implementation of HT on modern computer platforms is suggested. In Section 3, we demonstrate how to use this method in an industrial ICR system for document page orientation detection. In Section 4, a more complex problem of vanishing point detection in perspective text views is considered.

1. WIDESPREAD MYTHS ABOUT THE HOUGH TRANSFORM

The HT is a well-known transform widely used in image processing and machine vision (Gonzalez and Woods 2002; Forsyth and Ponce 2003). Though many results have been obtained regarding its properties, these results remain surprisingly unknown to a variety of researchers. Let us discuss those myths about the HT, which are most frequently adopted in literature.

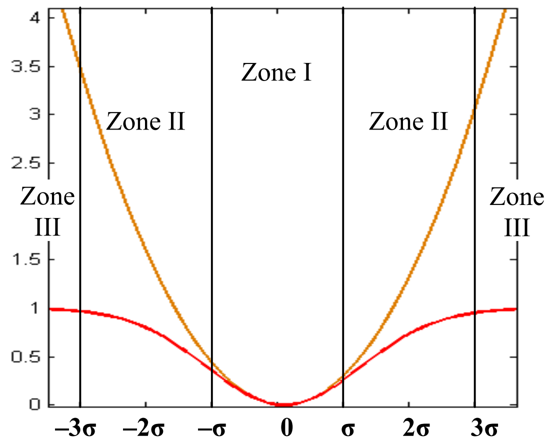

Figure 1: Quadric and Gaussian penalties

1) The HT is unstable against a normally distributed additive coordinate noise (Kiryati and Bruckstein, 2000).

This is not true. Consider the HT of data array pre-smoothed with a Gaussian filter. Evidently, the maximum in the Hough space corresponds to the line minimizing the Gaussian penalty (see Fig. 1). Such a penalty function can be conventionally split into three zones. Within Zone I, the Gaussian curve fits a parabola with a deviation not exceeding 20%, while within Zone III it is almost constant. Thus, if all the voting points belong to Zone I relatively to ideal line, solving the regression problem with the HT method yields almost the same results as obtained with the least squares method (LSM), which is known to be optimal in the case of a normal noise. At the same time, the presence of voting points within Zone III does not affect the solution until these points form an additional line with greater effective weight (as opposed to the LSM being quite sensitive to a non-correlated spike noise – see Fig. 2a). Those nodes belonging to transient Zone II do affect the solution but weaker than those nodes belonging to central Zone I. Thus, if the smoothing parameter sigma is properly chosen, the HT method is suitable for solving the linear regression problem in the case of simultaneous presence of a normal additive and a spike coordinate noise.

Figure 2 demonstrates robustness of HT in the presence of spike noise. Fig 2a shows set of six data points (2-7) and a noise one (1). Grey line marks LSM solution. Fig 2b shows HT results for this data set, and grey line on the Fig 2c marks Hough solution.

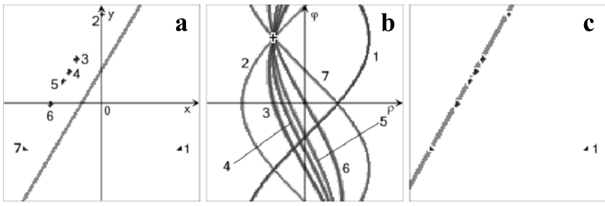

Figure 2: LSM and Hough Solutions for Linear Regression Problem

2) When applying the HT, it is important to guess the discretization scale (Forsyth and Ponce 2003).
Though this is not absolutely wrong, the problem is overstated. As it follows from the considerations of paragraph 1, the discretization scale can be fixed while the smoothing parameter is adjusted. Nevertheless, this does not erase the problem. Note that it is not too unusual for computer science that an algorithm contains a freely adjustable parameter. In such a case, it is only desirable that when there is an *a posteriori* quality criterion, the parameter adjustment process can be implemented fast enough. An iterative variation of the smoothing parameter followed by recalculation of HT does not belong to high-speed algorithms. However, it is known (Gindikin et al. 2003) that the HT is "permutable" with the Gaussian convolution! This means that after a single calculation of HT we become able to adjust the smoothing parameter in the initial space by smoothing the transformation result along ρ axis with 1D Gaussian kernel.

3) The HT is slow. Or: the HT can be accelerated only by approximate methods, including stochastic ones (Song et al. 2002).
This is not true. For instance, a method for calculating the fast HT (FHT) using the fast Fourier transform (FFT) was proposed almost 20 years ago (Lawton 1988).

4) The right way of calculating the FHT implies using the FFT (Lawton 1988).
This is not true as well. There are FHT calculation algorithms having the same asymptotic as FFT ($O\left(n^2 \cdot \log n\right)$, where n is the linear size of the array) but using only integer addition with no complex arithmetic (Brady 1998; Donoho and Hue 2000; Karpenko et al. 2004). In the next section, we shall demonstrate how to improve the complexity multiplier for one of these methods, using contemporary processor systems.

Thus, the HT can be calculated with $O\left(n^2 \cdot \log n\right)$ integer additions. To be precise, full FHT requires $4 \cdot n^2 \cdot \log n$ operations comparing to n^3 for brute-force HT. That means more than 25 times speed-up for 1024x1024 images. This allows one, provided that the smoothing parameter is properly adjusted, to suboptimally solve the linear regression problem with simultaneous presence of a normal additive and a spike coordinate noise. It is notable that the parameter adjustment iteration is reduced to a 1D Gaussian convolution of the transformation results.

2. SIMD-OPTIMIZED FHT CALCULATION ALGORITHM

Now consider a method for fast calculation of FHT. Apparently, it was first suggested by Brady (Brady 1998), and a few years later an in-place (buffer-free) version of this algorithm was published (Karpenko et al. 2004; Frederick et al. 2005).

To describe the FHT algorithm, let us introduce the following definitions. Let φ_l be the angle between the line l and the X-axis. Those lines for which $\left|tg\varphi_l\right| \le 1$ will be called "mostly horizontal". Correspondingly, all the lines with $\left|ctg\varphi_l\right| \le 1$ will be called "mostly vertical". Evidently, those lines parallel to quadrant diagonals belong to both classes.

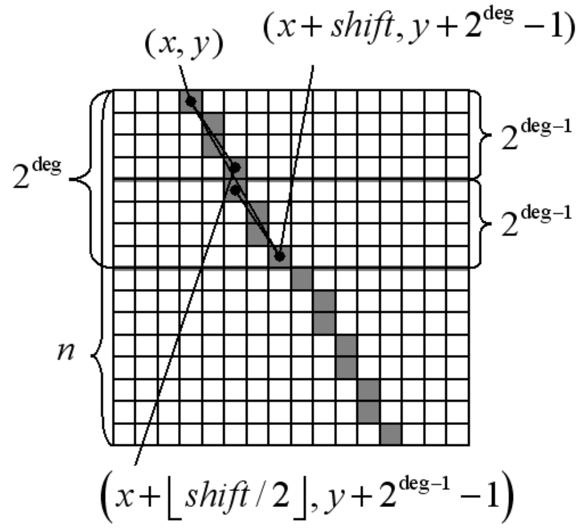

Figure 2: FHT Calculation Scheme

A discrete 8-connected line in a data array will be called an 8-connected chain of elements of this array, which approximates this line with a certain accuracy (i.e., the line passes through each element of the chain, while any 3 successive elements of the chain make an angle of at least 135˚). Examples of discrete 8-connected lines are chains built by the algorithm of Bresenham.

Consider the HT calculation algorithm for the case of "mostly vertical" discrete 8-connected lines, the case of "mostly horizontal" lines being completely analogous. Consider an input array with linear dimensions $m \times n = m \times 2^D$. A version of the algorithm, dealing with height different from a power of two, is described in the next section. Consider a horizontal strip, $m \times 2^{\deg}$ ($0 < \deg \le D$), with a vertical shift, $y = k \cdot 2^{\deg}$ ($0 \le k < 2^{D-\deg}$). Let us introduce the following recurrent definition of the 8-fold chain connecting the elements (x, y) and $\left(x + shift, y + 2^{\deg} - 1\right)$:

$$\begin{aligned} &\left[(x,y),\left(x+shift,y+2^{\deg}-1\right)\right]= \\ &=\left[(x,y),\left(x+\left\lfloor\frac{shift}{2}\right\rfloor,y+2^{\deg-1}-1\right)\right]\cup \\ &\cup\left[\left(x+\left\lceil\frac{shift}{2}\right\rceil,y+2^{\deg-1}\right),\left(x+shift,y+2^{\deg}-1\right)\right] \end{aligned} \quad (1)$$

It is quite evident that the 8-connected chain defined by Eq. (1), is a discrete 8-fold line passing through the terminal points (Fig. 3), although it is neither Bresenham chain nor Donoho chain. Indeed, expression (1) is a method for building a line by dichotomy along the "mostly longitudinal" axis.

Instead of the classical HT parameterization (ρ, φ), we shall use the space $(x_0, shift)$, where *shift* is the horizontal shift acquired by a "mostly vertical" line all along the array ($shift = \tan(\varphi) \cdot n$), and x_0 is the coordinate of the intersection of the parameterized line and the top row. To calculate HT, let us use Eq. (1) and fix the sign of *shift* (not greater or not less than zero). At the iteration with number *deg* ($0 < deg \le D$), "mostly vertical" HT is calculated for each of $2^{D-\deg}$ horizontal strips of height $2^{\deg}$ on the basis of the previous iteration results. Calculation of a single horizontal strip of the current iteration will be referred to as a "sub-iteration". Each sub-iteration involves into calculations two strips of the previous iteration (let us call these "sub-strips"). In each sub-strip, HT rows with $0 \le |shift| < 2^{\deg}$ are already calculated. As it follows from Eq. (1), the $|shift| = 2 \cdot s$ and $|shift| = 2 \cdot s + 1$ rows of the current strip may be calculated on the basis of the $|shift| = s$ rows of the sub-strip. Let us call this action a row operation of FHT. Since the input data of the row operation are no longer required for further calculations, the current result can be stored in-place. Unfortunately, in such a case HT rows become "shuffled". To obtain the index of the $|shift| = s$ row in a strip of height $2^{\deg}$, it is necessary to write down the *deg* bits of the number *s* from right to left.

The described algorithm allows one to calculate HT for the chosen "primarily direction" of lines (horizontal or vertical) and the sign of *shift*, using $n \cdot \log(n)$ row operations ($4 \cdot n \cdot \log(n)$ operations for the complete HT). Now consider in more detail an elementary row operation of FHT. Let it process row A_U of the upper sub-strip and row A_D of the lower sub-strip. The simplest implementation of the FHT row operation uses a buffer array, A_T , of length $m+1$. Let us copy row A_D into the buffer array, cyclically left-shifting it by *s* elements. This requires two commands of block copying. The last element of the buffer array is filled in with the zero element. Then, for each $0 \le i < m$, we can calculate $A_D[i] = A_U[i] + A_T[i+1]$ and $A_U[i] = A_U[i] + A_T[i+1]$.

Now discuss a possibility of accelerating the suggested algorithm with the use of modern processor systems. Most general-purpose processors presently support the technology called SIMD (single instruction, multiple data). For x86-family processors, this is MMX (for integers), for Power PC – AltiVec (also known as Velocity Engine or VMX), for Cell/B.E. – SPE. All the above-mentioned technologies use 128-bit registers, which allows processing in parallel four 32-bit integers. (Legacy MMX instructions supports only 64-bit registers, but SSE2 technology includes MMX128 subset). However, inevitable vectorization overheads make the overall profit of this approach not very high. Using 16-bit integers is only possible when processing images with linear dimensions not exceeding 256 pixels (with 8-bit brightness). Otherwise, either saturation or overflow of a register may occur during summation.

Nevertheless, we have succeeded in increasing the degree of parallelism up to 16 operations per command with a minimum loss of accuracy. In the scheme described above, at each summation the summands are sub-sums of an equal number of elements. This feature allows us to replace the summation with averaging. All the above-mentioned SIMD technologies have corresponding commands for simultaneously averaging 16 pairs of 8-bit integers (MMX: *pagvb*, VMX: *vavgub*, SPE: *avgb*). With such a calculation scheme, we obtain, as a result, an estimate of the 8 higher bits of the exact HT result. Of course, such accuracy is not always enough. Nevertheless, this approach is worth paying careful attention to, because it demonstrates almost 10x acceleration of the HT calculation and 4x reduction of the required memory size. All the results mentioned in the next section are obtained with the use of the 8-bit calculation scheme.

3. AUTOMATIC ORIENTATION OF SCANNED DOCUMENT PAGE USING FHT

The classical image preprocessing sequence in OCR and ICR is the following: binarization, page orientation and layout analysis. However, for a mixed document flow it is often impossible to find a universal binarization algorithm. In such a case, it is desirable to perform layout analysis first. In turn, the layout analysis task becomes greatly simplified if you precisely know the internal coordinate system of the document. This fact makes it very important to robustly estimate the angle of page orientation on the basis of the original image of the page.

Consider various 1D projections of the brightness of text page pixels. It is quite evident that the projection taken along the text lines contains strongly pronounced maxima (projections of text gaps) and minima (projections of text lines). With any deviation from this particular direction, the extrema get blurred. Thus, a simple but efficient functional for detecting the optimum page orientation is the dispersion of brightness of projection pixels. Taking into account that the sum of the values contained in each row of HT is the same (and equals the brightness integrated over the image), let us reformulate the criterion: the angle of orientation of the text page is determined by the HT row showing the maximum sum of squared values.

Now consider some details of algorithm implementation. Suppose that the maximum page orientation angle is 30˚ (limited by the scanner used). Let us now skew the image vertically by 30˚ clockwise. Correspondingly, the expected range of text orientation angles is from 0˚ to 45˚ clockwise. For the image obtained, it is enough to calculate FHT within a single quadrant only: the direction is "mostly horizontal" and the shift is positive. To rescale the image width to the form 2^D, we used bilinear up-scaling of the image to the nearest power of 2. We should note that up-scaling and skewing were performed simultaneously.

The algorithm performance is illustrated in Fig. 4. Fig 4a shows source image, Fig 2b shows result of applying FHT to the skewed image and Fig 2c shows corrected image. White lines on the Fig 2b marks zero angle and angle of projection with maximal dispersion. The algorithm execution time is 60 msec for an A4 page scanned with a resolution of 200 dpi (1660 x 2340 pixels) and Pentium IV 2 GHz processor. For the sake of backward compatibility, we used FHT implementation with 8x parallelism (MMX technology, not MMX128). An additional acceleration might be obtained by preliminarily downscaling the image, but this was not supposed worthwhile. In such a form, the algorithm has been exploited for more than one year in the industrial document recognition system "Cognitive Forms", showing robust results for images of passports, driving licenses, financial documents, and forms for surveys and tests.

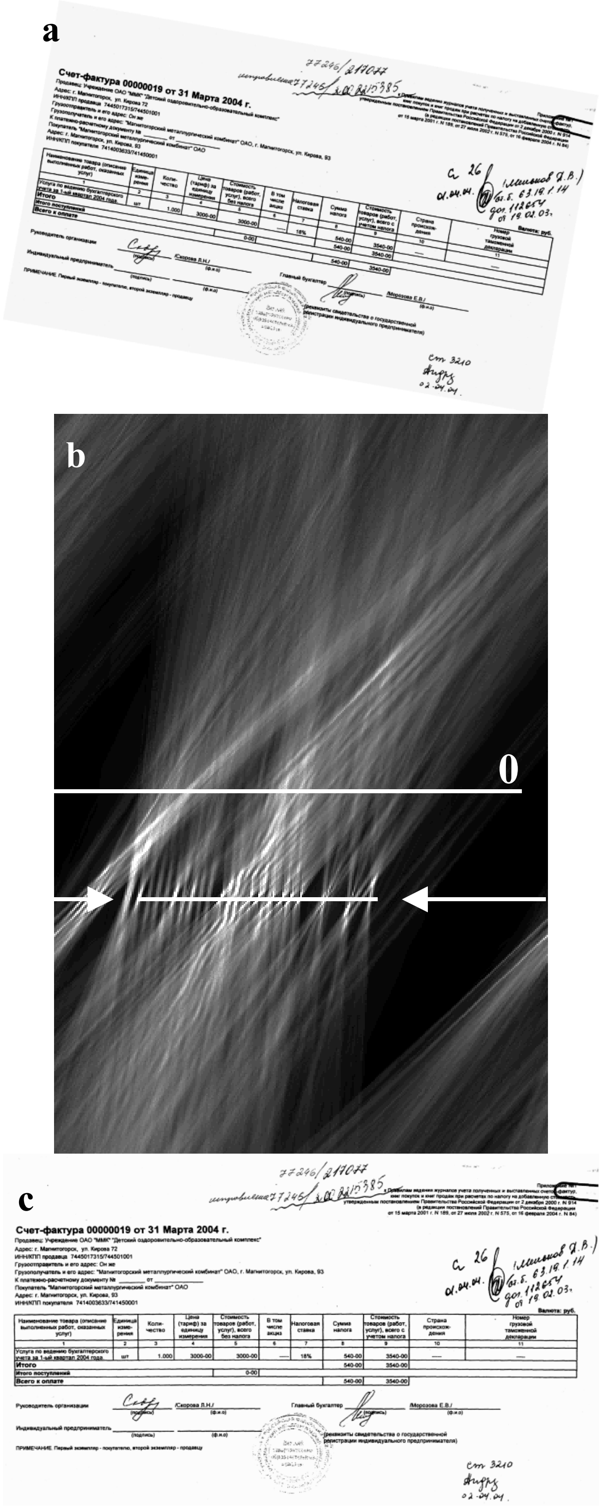

Figure 4: Automatic Page Orientation Using FHT

4. VANISHING POINT DETECTION FOR CAMERA ACQUIRED DOCUMENTS

Now consider a more complex case when the page to process is acquired in a perspective view. In this case, text strings are no longer parallel on the acquired image.

To normalize their orientation, it is necessary to detect the so-called "vanishing point". The use of HT for vanishing point detection is not a novel idea (Cantoni et al. 2001). A pencil of lines generates in the Hough space a series of maxima belonging to a 1D two-parametric manifold, such that the manifold parameters are unambiguously determined by the coordinates of the pencil center. In the space (ρ, φ), the pencil generates a sinusoid, which leads to computational difficulties on the stage of looking for the manifold that maximizes the quality functional.

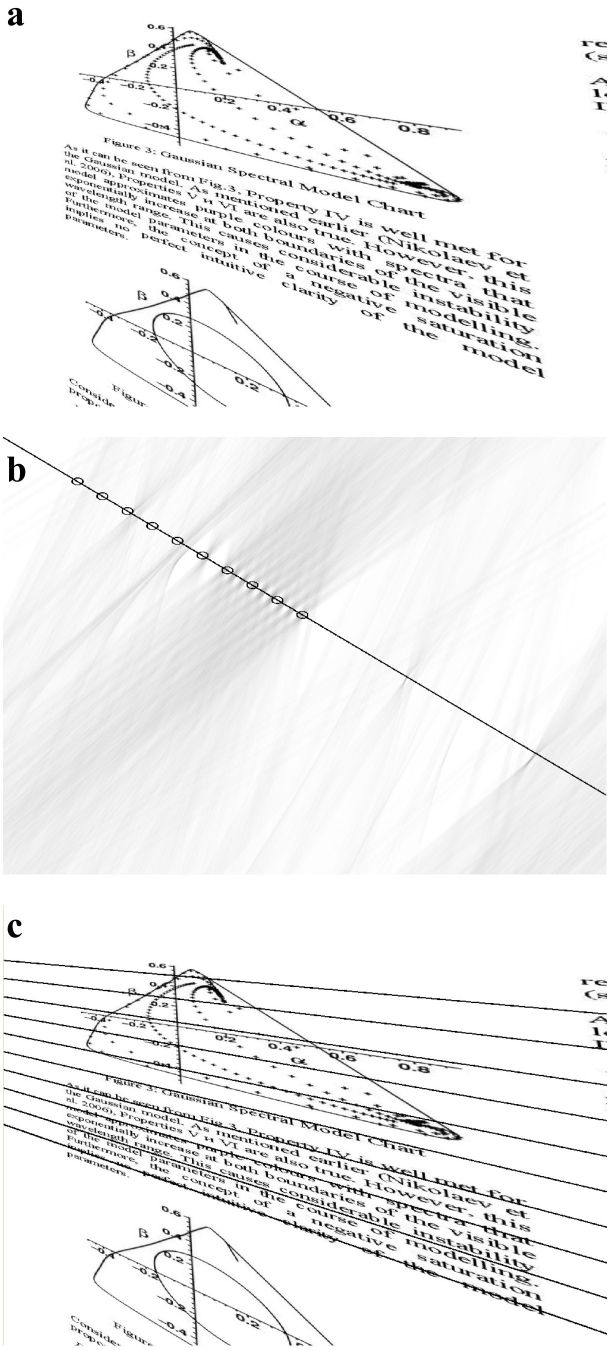

Figure 5: Vanishing Point Detection Using FHT

At the same time, in the space $(x_0, shift)$ the maxima corresponding to the lines forming the pencil lie on the same line. Using the results of the previous section, we can formulate the following criterion for searching for the vanishing point in a text page: the coordinates of the vanishing point are determined in the Hough space by the parameters of the line for which the dispersion of the values lying on it is maximum.

For lines arbitrarily oriented in the Hough space, the sum of the values is not a constant. However, numerical simulations show that the differences between such sums are negligible. We propose the following algorithm of determining the text orientation, based on using the SIMD-optimized FHT. First, apply HT to the original image. Then, calculate the value $\overline{v}$ averaged along a HT row. Then, build up a look up table (LUT) for the squared deviations: $LUT(v) = \min\left(255, \left[\alpha \cdot (v - \overline{v})^2\right]\right)$, where α is a scaling factor. Transform the result of the first HT using the LUT and perform FHT once more. The coordinates of the maximum in the resultant array unambiguously determine the coordinates of the vanishing point. The performance of the proposed algorithm is illustrated in Fig. 5. Fig 5a shows source image, Fig5b shows results of first FHT and line of maximal dispersion. Fig 5c visualizes pencil of lines with center in estimated vanishing point. As it can be seen from the figure, the dispersion maximum criterion is still efficient in the presence of pictures.

CONCLUSION

In our opinion, despite a large number of relevant papers are published, the potential of the Hough transform is not yet fully adopted in image processing and computer vision. Probably, this is due to a lack of efficient program realizations. In future, we are going to continue our research on using FHT as a tool for solving some problems of color segmentation (Nikolaev and Nikolayev, 2004) and color constancy (Finlayson and Schaefer, 2001; Nikolaev and Nikolayev, 2007).

ACKNOWLEDGEMENTS

We thank Dr. Lars Nolle for his presentation on the ECMS-2007 (Nolle 2007) that inspire us to this work. This research is partly financially supported by Russian Foundation for Basic Research (grants №06-01-22001, №06-02-16117 and №08-01-00879) and Russian Science Support Foundation.

REFERENCES

Ballard, D. and Brown, C. 1982. *Computer Vision.* Prentice-Hall.

Brady, M.L. 1998. “A fast discrete approximation algorithm for the Radon transform”. In *SIAM J. Computing*, V. 27(1), 107-119.

Cantoni, V.; Lombardi, L.; Porta, M. and Sicard, N. 2001. “Vanishing point detection: representation analysis and newapproaches.” In *Proceedings of 11^{th} Int. Conference on Image Analysis and Processing*, 90-94.

Donoho, D. and Huo, X. 2000. “Beamlet pyramids: A new form of multiresolution analysis, suited for extracting lines, curves, and objects from very noisy image data”. In *Proc. SPIE 2000*. V. 4119.

Finlayson, G. and Schaefer, G. 2001. “Solving for colour constancy using a constrained dichromatic reflection model”. In *Int. J. Com. Vision*. V. 42, 127-144.

Forsyth, D.A. and Ponce, J. 2003. *Computer Vision: A Modern Approach.* Prentice Hall.

Frederick, M.T.; VanderHorn, N.A. and Somani, A.K. 2005. “Real-time H/W Implementation of the Approximate Discrete Radon Transform”. In *Proc. of IEEE International Conference on Application-Specific Systems, Architecture Processors (ASAP'05).* 399-404.

Gindikin, S.G.; Gelfand, I.M. and Graev, M.I. 2003. *Selected Topics in Integral Geometry.* Amer. Mathematical Society.

Gonzalez, R.C. and Woods, R.E. 2002. *Digital image processing (2^{nd} ed.)* Prentice Hall.

Hough, P.V.C. 1959. “Machine Analysis of Bubble Chamber Pictures”, In *Proc. of Int. Conf. High Energy Accelerators and Instrumentation*.

Karpenko, S.M.; Nikolaev, D.P.; Nikolayev, P.P.; Postnikov V.V. 2004. “Fast Hough transform with controllable robustness”. In *Proc. of IEEE AIS'04 and CAD-2004*, V. 2, 303-309 (In Russian).

Kiryati, N and Bruckstein, A.M. 2000. “Heteroscedastic Hough Transform (HtHT): An Efficient Method for Robust Line Fitting in the 'Errors in the Variables' Problem”. In *Computer Vision and Image Understanding.* V.78(1), 69-83.

Lawton, W.A 1988. “New polar Fourier transform for computer-aided tomography and spotlight synthetic aperture radar”. In *IEEE Trans. Acoustics Speech Signal Process*. V. 36(6).

Nikolaev, D.P. and Nikolayev, P.P. 2004. “Linear color segmentation and its implementation”. In *Comp. Vision and Image Understand.* V. 94, 115-139.

Nikolaev, D.P. and Nikolayev, P.P. 2007. “On spectral models and colour constancy clues”. In *Proc. of 21^{st} European Conference on Modelling and Simulation*. 318-323.

Nolle, L. 2007. “Application of computational intelligence to target tracking”. In *Proc. of 21^{st} European Conference on Modelling and Simulation*. 289-293.

Pratt W., 2001. *Digital image processing (3^{rd} ed.)* Wiley.

Song, J.; Cai, M.; Lyu, M.R. and Cai, S. 2002. “A New Approach for Line Recognition in Large-size Images Using Hough Transform”. In *Proc. of 16^{th} International Conference on Pattern Recognition (ICPR'02).* V1, 33-36.

AUTHOR BIOGRAPHIES

DMITRY P. NIKOLAEV was born in Moscow, Russia. He studied physics, obtained his Master degree in 2000 and Ph.D. degree in 2004 from Moscow State University. Since 2007 he has been a head of the sector at the Institute for Information Transmission Problems, RAS. His research activities are in the areas of computer vision with primary application to colour image understanding. His e-mail address is dimonstr@iitp.ru and his Web page can be found at http://www.ddt.ru/chaddt/Lace/pub.html.

SIMON M. KARPENKO was born in Moscow, Russia. He studied mathmatics, obtained his Master degree in 2003 from Moscow State University. Since 2007 he has been a research scientist at the Institute for Information Transmission Problems, RAS. His research activities are in the areas of computer vision and machine learning.

ILIA P. NIKOLAYEV was born in Moscow, Russia.

He studied physics and obtained his Master degree in 1994 and Ph.D. degree in 1997 from Moscow State University. He is an associate professor at Moscow State University, his research activities are in the areas of non-linear optics.

PETR P. NIKOLAYEV was born in Moscow, Russia. He studied physics and obtained his Master degree in

1966 from Moscow State University. He obtained a Ph.D. degree in biology in 1975 from the Institute for Information Transmission Problems, RAS and a Dr. of Science degree in physics in 1993 from the Institute for Biophysics, RAS. Since 1966, he has been a research scientist at the Institute for Information Transmission Problems, RAS, where he is now leading a research group in the field of computer colour vision. His e-mail address is nikol@iitp.ru.