16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

A NEW OPTIMIZATION CRITERION FOR EXTRACTING SEVERAL STRAIGHT LINES IN A BINARY IMAGE

Cécile Bidot, Nicolas Paul, Luc Féty and Michel Terré

Conservatoire National des Arts et Métiers
292 rue Saint Martin, 75003, Paris, France
phone: + (33) 0140272767, fax: + (33) 0140272481, email: terre@.cnam.fr
web: www.cnam.fr

ABSTRACT

This paper introduces a new algorithm for extracting several straight lines in a binary image. This algorithm is based on the optimization of a new criterion called "Line-K-product" (LKP) which is derived from the K-product algorithm we introduced in a previous work [1] for unsupervised mixture estimation. A relaxation algorithm is proposed to find the minimum of LKP. Contrary to the classical Hough transform (HT) based methods our approach does not require the tricky configuration of extra parameters. Simulations finally illustrate the superiority of our approach compared to the HT-based algorithm.

1. INTRODUCTION

In this paper, we focus on straight line extraction algorithms. This topic has been deeply studied in the past and Hough transform (HT) is a well-known technique for extracting lines [2]. Implementation of HT was analysed in [3] while its performance for images with additive noise was addressed in [5]. It is nevertheless well-known that HT involves a threshold process and a clustering stage and that detecting peaks in the accumulator array is not always a reliable process [6]. This clustering stage was deeply analysed in [4][9][12][13].

A new approach is proposed to extract several straight lines from a noisy binary image. It is mainly based on the minimization of a new optimization criterion called "Line K-Product" (LKP) derived from the K-Product (KP) algorithm we recently introduced in [1] for unsupervised mixture estimation. The paper is organized as follows. In section II we recall the definition of the KP criterion and derive the LKP criterion for straight line extraction. An iterative relaxation algorithm is then proposed to find the minimum of LKP. In section III we present simulations results which illustrate the superiority of our approach compared with the Hough Transform. Section IV finally concludes the paper.

2. LK PRODUCT ALGORITHM

Given a set of univariate observations $\{x_n\}_{n\in[1,N]}$ of a K-component mixture, the K-product (KP) algorithm was introduced in [1] [10] to estimate the mixture component expectations $\boldsymbol{m}=(m_1,...,m_K)$. The algorithm is based on the minimization of the KP criterion $J_{KP}(\boldsymbol{u})$ defined by:

$$J_{KP}(\boldsymbol{u})=\sum_{n=1}^{N}\prod_{k=1}^{K}(x_n-u_k)^2 . \qquad (1.)$$

The first intuitive motivation for defining this criterion is its behavior in the limit case, when the variances of the components are null. In this particular case, all the observations are equal to one of the m_k so $J_{KP}(\boldsymbol{m})=0$ and $J_{KP}(\boldsymbol{u})$ is minimal in $\boldsymbol{u}=\boldsymbol{m}$. The second motivation to define the KP criterion is that, in the general case, it does not have any local non-global minima. The global minimum of J_{KP} can be for instance reached by the simple relaxation algorithm described in [10] or by a non-iterative algorithm described in [1]. It provides a biased but useful estimation of the mixture component expectation which can be used to cluster the data observation (nearest neighbor classification) and separate the K mixture components.

In this paper we extend the KP principle to the extraction of straight lines in binary image. The threshold process used to obtain the binary image is not detailed in this paper even if it has a strong impact of the performance. The observation is then a set of couples $\{x_n, y_n\}_{n\in[1,N]}$ which indicate the coordinates of the black pixels in the binary image.

Among this set of (potentially noisy) observations we are looking for K Cartesian straight line equations defined by parameters $\{a_k, b_k\}_{k\in[1,K]}$: $y=a_k x+b_k$. In other word we assume that for all observation n there is one k such as $y=a_k x+b_k+\varepsilon_n$, where ε_n is an error (noise) random process.

The extended KP criterion, called "Line K-Product" (LKP) criterion $J_{LKP}(\boldsymbol{a},\boldsymbol{b})$, is defined by:

$$J_{LKP}(\boldsymbol{a},\boldsymbol{b})=\sum_{n=1}^{N}\prod_{k=1}^{K}\frac{(y_n-a_k x_n-b_k)^2}{(1+a_k^2)}, \qquad (2.)$$

where $\boldsymbol{a}=(a_1,...,a_K)$ and $\boldsymbol{b}=(b_1,...,b_K)$ are the set of tested parameters of the straight line equations, and $\frac{(y_n-a_kx_n-b_k)^2}{(1+a_k^2)}$ is therefore the exact (thanks to the denominator) quadratic distance between the observation $\{x_n,y_n\}$ and the tested straight line $\{a_k,b_k\}$.

This criterion can be minimized using a relaxation algorithm. The main idea is to freeze all couples $\{a_k,b_k\}$ except one, for instance $\{a_j,b_j\}$, and to cancel partial derivatives with respect to this "free" couple.
In such a case $\{a_j,b_j\}$ is then obtained by the minimisation given hereafter:

$$\underset{a_j,b_j}{Min}\sum_{n=1}^{N}\frac{(y_n-a_jx_n-b_j)^2}{(1+a_j^2)}p_{j,n}, \qquad (3.)$$

where the weight $p_{j,n}$ is defined by:

$$p_{j,n}=\prod_{\substack{k=1\\k\neq j}}^{N}\frac{(y_n-a_kx_n-b_k)^2}{(1+a_k^2)}. \qquad (4.)$$

Once $\{a_j,b_j\}$ obtained, we freeze this couple and process to the optimisation described in equation (3) with respect to another couple. Once all couples optimised, we iterate the algorithm. We stop the algorithm either when a certain number of iterations is reached or when the $J_{LKP}(\boldsymbol{a},\boldsymbol{b})$ decrease between two iterations is less than a given threshold.

In the two following section we focus on the cancellation of partial derivative of $J_{LKP}(\boldsymbol{a},\boldsymbol{b})$ with respect to $\{a_j,b_j\}$.

2.1 First partial derivative

We first cancel the derivative with respect to a_j:

$$\frac{\partial J_{LKP}(\boldsymbol{a},\boldsymbol{b})}{\partial a_j}=0$$

We obtain:

$$\sum_{n=1}^{N}\Big[-x_np_{j,n}(y_n-a_jx_n-b_j)(1+a_j^2)\\ -a_jp_{j,n}(y_n-a_jx_n-b_j)^2\Big]=0$$

Developing this equation gives:

$$\sum_{n=1}^{N}\Big[-x_np_{j,n}y_n+x_n^2p_{j,n}a_j+x_np_{j,n}b_j-x_na_j^2p_{j,n}y_n\\ +x_n^2a_j^3p_{j,n}+x_na_j^2p_{j,n}b_j-a_jp_{j,n}y_n^2-p_{j,n}a_j^3x_n^2\\ -a_jp_{j,n}b_j^2+2a_j^2p_{j,n}y_nx_n+2a_jp_{j,n}y_nb_j\\ -2a_j^2p_{j,n}b_jx_n\Big]=0$$

Then:

$$\sum_{n=1}^{N}\Big[a_j^2(p_{j,n}y_nx_n-x_np_{j,n}b_j)+x_np_{j,n}b_j-x_np_{j,n}y_n\\ +a_j(x_n^2p_{j,n}-p_{j,n}y_n^2-p_{j,n}b_j^2+2p_{j,n}y_nb_j)\Big]=0 \qquad (5.)$$

We then introduce the following intermediate terms:

$$P=\sum_{n=1}^{N}p_{j,n}\ ,\ P_x=\sum_{n=1}^{N}p_{j,n}x_n\ ,\ P_y=\sum_{n=1}^{N}p_{j,n}y_n$$

$$P_{xy}=\sum_{n=1}^{N}p_{j,n}x_ny_n\ ,\ P_{xx}=\sum_{n=1}^{N}p_{j,n}x_n^2\ ,\ P_{yy}=\sum_{n=1}^{N}p_{j,n}y_n^2\ ,$$

and $Q=P^{-1}$,

where we omit the index j for the sake of simplicity.

Equation (5) becomes:

$$a_j^2(P_{xy}-b_jP_x)+a_j(P_{xx}-P_{yy}-b_j^2P+2b_jP_y)\\ +b_jP_x-P_{xy}=0 \qquad (6.)$$

2.2 Second partial derivative

The second step of the algorithm concerns the partial derivative with respect to b_1: $\frac{\partial J_{LKP}(\boldsymbol{a},\boldsymbol{b})}{\partial b_j}=0$. We obtain:

$$\sum_{n=1}^{N}-(y_n-a_jx_n-b_j)p_{j,n}=0$$

Using the variables defined in 2.1:

$$-P_y+a_jP_x+b_jP=0$$

Then:

$$b_j=Q(P_y-a_jP_x) \qquad (7.)$$

Merging (7) with (6), yields to:

$$
\begin{aligned}
&a_j^2\left(P_{xy}-Q\left(P_y-a_jP_x\right)P_x\right)\\
&+a_j\left(P_{xx}-P_{yy}-Q^2\left(P_y-a_jP_x\right)^2P+2Q\left(P_y-a_jP_x\right)P_y\right)\\
&+Q\left(P_y-a_jP_x\right)P_x-P_{xy}=0
\end{aligned}
$$

We obtain then directly:

$$
\begin{aligned}
&a_j^3\left(QP_x^2-Q^2P_x^2P\right)\\
&+a_j^2\left(P_{xy}-QP_yP_x+2Q^2P_xP_yP-2QP_xP_y\right)\\
&+a_j\left(P_{xx}-P_{yy}-Q^2PP_y^2-QP_x^2+2QP_y^2\right)\\
&+QP_yP_x-P_{xy}=0
\end{aligned}
\qquad (8.)
$$

By definition we have $QP=1$, the polynomial of order three of equation (8) is then reduced to a polynomial of order two given by:

$$
\begin{aligned}
&a_j^2\left(P_{xy}-QP_yP_x\right)+a_j\left(P_{xx}-P_{yy}-QP_y^2-QP_x^2+2QP_y^2\right)\\
&+QP_yP_x-P_{xy}=0
\end{aligned}
\qquad (9.)
$$

So it appears that we just have to calculate the roots of this polynomial. After this stage b_j is then directly given by (7).

Depending on the roots of the polynomial, we obtain then one or two couples (a_j, b_j). We keep the solution that minimize (3)

The complete algorithm is summarized in table 1 hereafter:

for l =1 to the maximal number of iterations
 for j =1 to K
  Freeze (a_k, b_k) for $k \in [1,N]-\{j\}$,
  Calculate, $P, P_x, P_{xy}, P_{xx}, P_{yy}$
  Find a_j that are roots of equation (9)
  Find corresponding b_j parameters with equation (7)
  Keep the couple (a_j, b_j) that minimizes equation (3)
 end for j
end for l

Table 1 – LK product Algorithm for N lines

3. SIMULATION RESULTS

In this section we consider binary images with pixels belonging to straight lines. If we consider ideal coordinates of a pixel $\{x_n^i, y_n^i\}$, we simulate the observation noise as an additive two dimensional Gaussian process. We obtain then the noisy observation $\{x_n = x_n^i + v_n, y_n = y_n^i + w_n\}$ where v_n and w_n are two iid Gaussian variables with the same variance σ^2.

In figure 1 we represent pixels (crosses) and straight lines obtained with HT and LKP algorithms. In this simulation, we have 3 lines, 100 pixels per line and the observation noise is weak: $\sigma^2 = 1$. It appears that the two algorithms provide similar estimation performances.

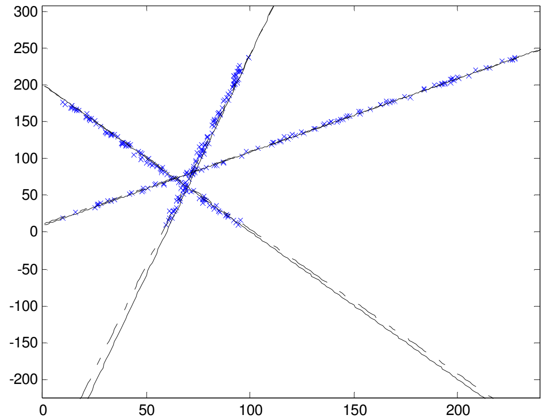

Figure 1 – 3 lines, σ²=1, HT solid, LKP dash

Values obtained for lines parameters are given in table 2 hereafter:

(a ; b)	6 ; -350	1 ; 10	-2 ; 200
HT	5.97 ; -351.4	1 ; 9.9	-2 ; 199.5
LKP	5.73 ; -326.1	0.99 ; 11.9	-1.96 ; 198.9

Table 2 – Estimation of lines, parameters 3 lines, σ²=1

In figure 2, we analyse the same configuration but we increase the noise variance: $\sigma^2 = 16$.

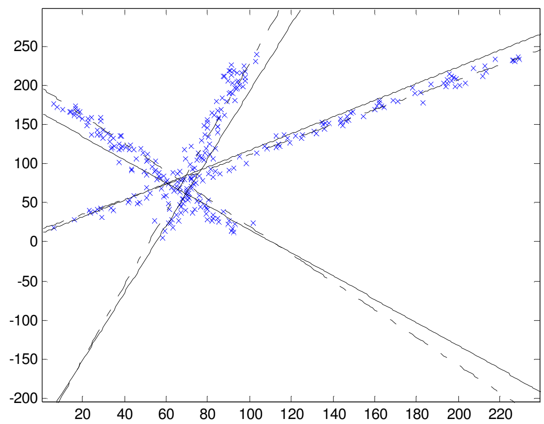

Figure 2 – 3 lines, σ²=16, HT solid, LKP dash

It appears that HT algorithm is then less accurate than LKP. This assumption is confirmed by results presented in table 3.

(a ; b)	6 ; -350	1 ; 10	-2 ; 200
HT	4.30 ; -233.4	1.06 ; 11.6	-1.48 ; 162.8
LKP	4.73 ; -240.3	0.96 ; 16.9	-1.77 ; 195.1

Table 3 – Estimation of lines, parameters 3 lines, σ²=16

In order to confirm this first result more than 100 simulations have been performed with the same straight lines equations, but with different pixels coordinates and noise realisations.
For the ***a*** parameter the variance obtained with the HT algorithm was equal to 2.74 while it was only equal to 1.26 with the LKP algorithm.
For the ***b*** parameter, the variance was equal to $1.4\ 10^4$ with the HT algorithm and $9.7\ 10^3$ with the LKP algorithm.
This scattering of estimations is illustrated in figure 3 where accumulation of results obtained with 4 simulations is presented.

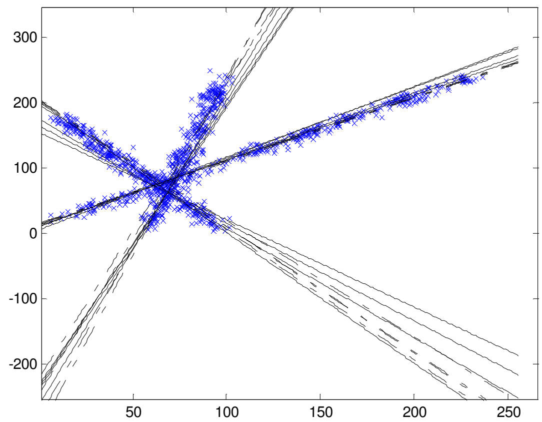

Figure 3 – 3 lines, σ²=16, HT solid, LKP dash, 4 simulations

In figure 5, we present results obtained always with the same configuration, but we increased once more the noise variance, reaching $\sigma^2 = 24$. In this simulation case we observe an error for the HT algorithm where one estimated straight line is between two true lines. Estimations given by LKP algorithm exhibit, in this case, better performances.

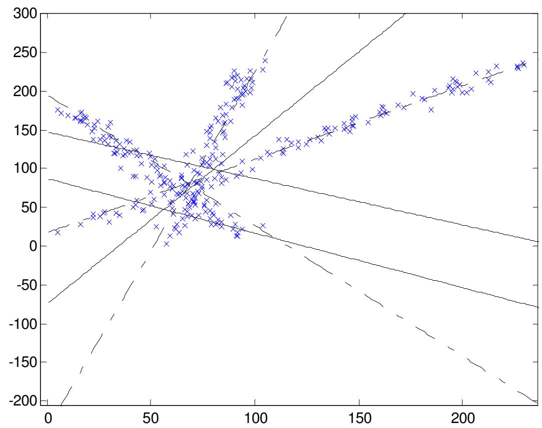

Figure 4 – 3 lines, σ²=24, HT solid, LKP dash

(a ; b)	6 ; -350	1 ; 10	-2 ; 200
HT	2.16 ; -72.6	-0.6 ; 146.6	-0.7 ; 86.5
LKP	4.61 ; -232.2	0.95 ; 18.2	-1.68 ; 193.6

Table 4 – Estimation of lines parameters, 3 lines, σ²=24

In Figure 5 we analyse a configuration with 5 lines. It appears in this case that HT algorithm falls in a local minimum. For HT, we observe one straight line estimation between two true lines and two estimations taking into account the same line.

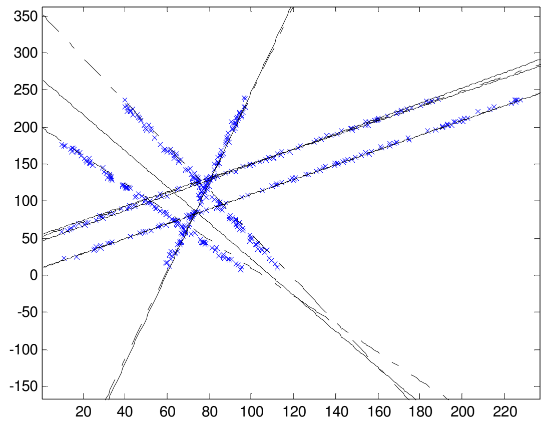

Figure 5 – 5 lines, σ²=1, HT solid, LKP dash

(a ; b)	6;-350	1;10	-2;200	-3;350	1;50
HT	6.1; -357.8	0.99; 10.7	-2.43; 263.0		1.03; 46.83 0.96; 55.87
LKP	5.95; -345.08	0.99; 10.37	-1.90; 197.91	-2.98; 350.74	0.98; 53.09

Table 5 – Estimation of lines parameters, 5 lines, σ²=1

In Figure 6 we analyse the same configuration with an increase of the noise variance. Once more we observe some problems with HT algorithm, having one estimation between two true lines and two estimation going from one line to the other..

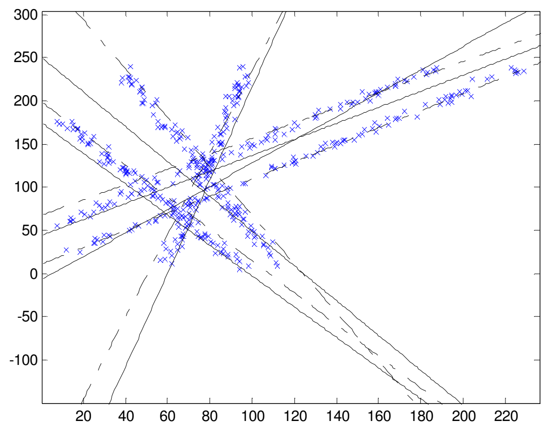

Figure 6 – 5 lines, σ²=7, HT solid, LKP dash

(a ; b)	6;-350	1;10	-2;200	-3;350	1;50
HT	5.40; -318.8	1.35; -5.9	-1.77; 173.2	-2.01; 248.8	0.93; 44.88
LKP	4.75 ; -235.9	0.99; 11.56	-1.84 ; 197.4	-2.82 ; 347.4	0.99; 11.56

Table 6 – Estimation of lines parameters, 5 lines, $\sigma^2=7$

4. CONCLUSION

In this paper we introduced a new algorithm for extracting several straight lines in a binary image. We compared this algorithm with the well-known Hough Transform and we showed that, in many simulation cases, performances obtained were greater than those obtained with the Hough Transform. The main advantage of this new algorithm is the absence of threshold process or clustering stages. The only parameter of the algorithm is a number of iterations for a relaxation process. All simulations presented have been obtained after 40 iterations but this number can easily be reduced thanks to a stop criterion.

The straight line extraction in binary images is involved in many algorithms [8] and the new proposed algorithm offers finally an interesting non-parametric approach to this problem.

The LKP algorithm can be extended to other parametric functions (e.g. circles), like the HT. For the case of K circles with centers (a_k, b_k) and radius r_k, the new criterion J_{CKP} is given hereafter :

$$J_{CKP}(\boldsymbol{a},\boldsymbol{b},\boldsymbol{r}) = \sum_{n=1}^{N} \prod_{k=1}^{K} \left((x_n - a_k)^2 + (y_n - b_k)^2 - r_k^2 \right)^2 \tag{10.}$$

After straightforward derivations, we obtain :

$$a_k = \frac{1}{2} \frac{P_{y^2} P_x + P_{x^2} P_x - P_{xy^2} - P_{x^3}}{(P_x)^2 - P_{x^2}}$$

$$b_k = \frac{1}{2} \frac{P_{x^2} P_y + P_{y^2} P_y - P_{x^2 y} - P_{y^3}}{(P_y)^2 - P_{y^2}}$$

$$r_k^2 = P_{x^2} + P_{y^2} - 2a_k P_x - 2b_k P_y + a_k^2 + b_k^2$$

with $P_x = \sum_{n=1}^{N} D_{k,n} x_n$ and $P_{x^q} = \sum_{n=1}^{N} D_{k,n} x_n^q$

and with $C_{k,n} = \prod_{\substack{p=1 \\ p \neq k}}^{K} \left((x_n - a_p)^2 + (y_n - b_p)^2 - r_p^2 \right)^2$

REFERENCES

[1] Nicolas Paul, Michel Terré and Luc Féty, "A global algorithm to estimate the expectations of the components of an observed univariate mixture," *Advances in Data Analysis and Classification*, Volume 1, pp 201-219, 2007.

[2] Henri Maitre, "Contribution to the prediction of performances of the Hough transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , Volume 8 , Issue 5, pp. 669-674, September 1986

[3] J. Illingworth , J. Kittler, "A survey of the Hough transform, Computer Vision, Graphics, and Image Processing," *Computer Vision, Graphics and Image Processing*, Volume 44, n°1, pp.87-116, October 1988.

[4] Thomas Risse, "Hough transform for line recognition: Complexity of evidence accumulation and cluster detection," *Computer Vision, Graphics, and Image Processing*, Volume 46, Issue 3, pp. 327-345, June 1989.

[5] Douglas J. Hunt, Loren W. Nolte, Amy R. Reibman and W. Howard Ruedger, "Hough transform and signal detection theory performance for images with additive noise," *Computer Vision, Graphics, and Image Processing*, Volume 52, Issue 3, pp. 386-401, December 1990.

[6] Edward K. Wong, .Yu Chen; Syng-Yup Ohn, "Straight line extraction in binary drawings using morphological operations, ", *Proc. SPIE Nonlinear Image Processing IX*, Volume 3304, pp. 166-173, April 1998.

[7] N. Milisavljevic, I. Bloch and M. Acheroy, "Application of the Randomized Hough Transform to Humanitarian Mine Detection," *IASTED International Conference on Signal and Image Processing*, pp. 149-154, Hawaii, USA, August 2001.

[8] Yasutaka Furukawa and Yoshihisa Shinagawa, " Accurate and robust line segment extraction by analyzing distribution around peaks in Hough space," *Computer Vision, Graphics, and Image Processing*, Volume 92, Issue 1, pp. 1-25, October 2003.

[9] Ulrich Hillenbrand, "Consistent parameter clustering: Definition and analysis," *Pattern Recognition Letters*, Volume 28 , Issue 9, pp 1112-1122, July 2007

[10] Nicolas paul, Michel Terré and Luc Féty, "The K-product criterion for Gaussian mixture estimation," *in Proc. of the 7th Nordic Signal Processing Symposium*, Reykjavik, Iceland, 2006.