

On the use of entropy power for threshold selection

Franck Luthon^{a,*}, Marc Liévin^b, Francis Faux^a

^a*Computer Science Dept, University of Pau & Adour Province, LIUPPA, IUT Bayonne, France*

^b*Center of Advanced European Studies and Research, CAESAR, Bonn, Germany*

Abstract

This paper deals with an entropic approach as unsupervised thresholding technique for image processing, in order to extract a relevant binary information from noisy data. It is dedicated to situations where a signal of relatively high energy is localized in the image whereas the noise is spread over the entire frame. The method is based on the computation of the entropy power of the information source, as defined by Shannon. The threshold used for binarization is proportional to the entropic deviation of the observation source. The performance of the approach is illustrated by two classical image preprocessing tasks, namely motion detection and edge detection. The evaluation set contains both synthetic data and real-world image sequences.

Résumé Cet article propose une approche entropique comme technique de seuillage automatique en traitement d'images, dans le but d'extraire une information binaire pertinente à partir de données bruitées. La méthode est applicable dans le cas d'un signal de forte énergie localisé dans le plan image, tandis que le bruit est majoritaire et diffus. La technique est basée sur la notion de puissance entropique d'une source d'observations, telle que définie par Shannon. Le seuil de binarisation est proportionnel à l'écart-type entropique de la source. L'efficacité de la méthode est illustrée par deux applications classiques du traitement d'images: la détection de mouvement par les changements temporels et la détection de contours par les maxima du gradient spatial. L'évaluation des performances est réalisée à la fois sur des données synthétiques et sur des images réelles.

Key words: automatic thresholding, binarization, entropic deviation, motion detection, edge detection, Gaussian noise
PACS: 07.05.P

1. Introduction

The thresholding technique is of common use in image processing and video analysis, in order to bi-

narize noisy observations that are coded with n bits (typically $n = 8$), either in the spatial, temporal or frequency domain. Typical applications are motion detection or edge detection, where one aims at extracting significant temporal changes, or significant spatial gradients, in order to exhibit either the relevant mobile areas in a video scene, or the relevant contours in a static image. Other applications may concern spectral analysis or time-frequency segmentation, where one aims at extracting spec-

* Corresponding author. Tel.: +33 5 59574344; fax: +33 5 59574329.

Email address: Franck.Luthon@univ-pau.fr (Franck Luthon).

URL: <http://www.iutbayonne.univ-pau.fr/~luthon/> (Franck Luthon).

tral patterns of high energy that are localized in the frequency domain but embedded in widespread low-level noise.

The key point is then the choice of a proper threshold, in order to get rid of the noise that corrupts the data, without cutting significant information. We propose here an unsupervised method in order to automatically and adaptively estimate the threshold based on the computation of the entropy power of the observations, under the assumption of an additive Gaussian noise that is spread all-over the grid, whereas the signal is localized. Stated in other words, it means that the global SNR is low whereas the local SNR is high in some informative parts of the image grid. The optimal threshold (in the sense of information theory) is proportional to the square root of the entropy power (a quantity that we call *entropic deviation*).

Simulations with synthetic data are shown to illustrate our idea and validate the approach. The method is then tested with real-world images both for the detection of significant temporal changes, which is a classical preprocessing step before estimation, segmentation or compression of motion, and for the detection of the maxima of spatial gradients, which is the first step towards edge detection.

The paper is organised as follows: after an overview of other thresholding techniques in section 2, we recall what is the entropy power as defined by Shannon and present our threshold setting method with simulations on synthetic data in section 3. Section 4 deals with applications in motion detection and edge detection respectively. Finally, we conclude in section 5 with a brief discussion.

2. Related works

For images with distinct objects and background, where the histogram is clearly bimodal or multimodal, the mode method is classically used: thresholds are determined from peaks and valleys in the histogram. The method is simple (computation of zero-crossings on the derivatives) but it is difficult to apply with very unequal peaks or broad and flat valleys. An example of such method for

color segmentation is used in [1].

Other methods use second or higher order statistics. The threshold selection method proposed by Otsu is non-parametric and unsupervised. It dichotomizes the pixels into two classes by maximizing the interclass variance (computation of cumulative moments of the grey-level histogram and discriminant analysis through the optimization of an objective function) [2]. A multithreshold extension is also feasible. Reddi et al. proposed a faster implementation by assuming a continuous probability function [3]. Their method was used by [4] for change detection in face video sequences.

Threshold selection based on entropy of the histogram was first proposed by Pun [5,6], and then developed in various papers [7–11]. They maximize an evaluation function involving the a posteriori entropy or use an anisotropy coefficient that reacts to the shape of the histogram. Extensions to 2-D histograms, local versus global threshold estimations, or multithresholding are also proposed. Recently, the use of fuzzy logic was also introduced for image thresholding with application in medical imaging [12].

Our own entropic approach, based on the notion of entropy power, is quite different from those approaches since it does not rely on any kind of maximization of the entropy, but exploits the connection between the entropy of a signal and its power [13]. We use only the a priori entropy in contrary to the other techniques that usually maximize the a posteriori entropy. The condition of validity is also different from most of other methods (apart from the work by Pun on histogram anisotropy and the work by Lee on truncated distributions [14]) since it corresponds to the specific case of quick decaying histograms (*i.e.*, almost unimodal) like those arising in high-pass filtered images with two very unequal peaks, and not to the case of linear bimodal histograms.

A typical illustration of the kind of situation we want to address is shown in Fig. 1. It corresponds to the detection of temporal changes in consecutive frames of a video sequence for motion detection application like road surveillance. Most of the image of observations is black (no temporal change in the background) whereas a few areas that undergo significant temporal changes (due to moving

vehicles on the highway) appear as bright regions.

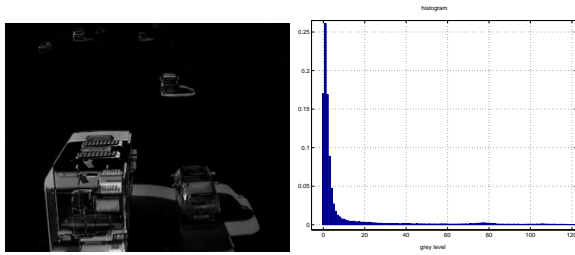


Fig. 1. Typical case of application: a) observations (frame difference); b) histogram.

3. The proposed method

3.1. Entropy power of a discrete source

As defined by Shannon [15], the entropy of an information source is a measure of its mean information. In the case of a discrete source O of observations taking values in the set of integers $\{i = i_0 \cdots i_M\}$ (typically $i_0 = 0$ and $i_M = M$ with $M = 255$), the entropy H is given by:

$$H = - \sum_{i=i_0}^{i_M} p_i \log p_i \quad (1)$$

where p_i is the probability that the observation at any site $s = (x, y) \in S$ takes the value $o(s) = i$ (S being the grid supporting the observations, typically an image of size $T = L \times C$, where L and C are the numbers of lines and columns respectively).

Note that the choice of the logarithmic base is arbitrary and corresponds to the choice of the unit of measure: natural unit (*nat*) with base e , binary unit (*bit*) with base 2 ; decimal unit (*digit*) with base 10 ; and we have:

$$H_{nat} = H_{bit} \log 2. \quad (2)$$

Shannon proved that the entropy of a Gaussian source $G(0, \sigma)$ with zero-mean and standard deviation σ is given by:

$$H(G) = \log(\sigma\sqrt{2\pi e}) \quad (3)$$

It is well known that this corresponds to the maximum value of the entropy for a continuous distribution subject to the condition that the standard deviation is fixed at σ (and this maximum is only achieved by the Gaussian distribution).

Note that Eq. (3) is also valid in the discrete case and even for non centered data, under a few weak assumptions on M and T (see proof in Appendix).

For an arbitrary source with a given entropy H , Shannon also defines the *entropy power* N , which represents the power of the white noise equivalent to source O , in the sense it has the same entropy and is limited to the same band:

$$N(O) = \frac{1}{2\pi e} \exp(2H) \quad (4)$$

The important properties of the entropy power are the following:

- (i) the entropy power of any source is always less than or equal to its actual power,
- (ii) the entropy power of a Gaussian source $G(0, \sigma)$ equals its power: $N(G) = \sigma^2$.
- (iii) the entropy power of the sum of two signals is lower-bounded by the sum of their respective entropy powers, and upper-bounded by the sum of their actual average powers.
- (iv) as stated by Shannon, “white Gaussian noise has the peculiar property that it can absorb any other signal which may be added to it, provided the signal power is small, in a certain sense, compared to noise”. In that case, the resultant entropy power is approximately equal to the sum of the white noise power and the signal power [15].

In the following, we will use this concept of entropy power and apply it to the case of a discrete source since we deal with numerical images (discrete approximation of the continuous case, similarly to what was done between [2] and [3]).

3.2. Threshold setting

Consider an observation source $O = \{o(s), s \in S\}$ consisting of a useful signal X corrupted by an additive Gaussian noise: $O = X + G$. Assuming that the noise level is low but the noise is present all-over the grid, whereas the signal is of high amplitude but remains localized on the grid, property

(iv) holds. One can estimate the equivalent *entropic deviation* σ_e by computing the entropy H of the observation source and then taking the square root of the entropy power given by Eq. (4):

$$\sigma_e = \sqrt{N(O)} = \frac{\exp(H)}{\sqrt{2\pi e}} \quad (5)$$

The threshold value θ may then be fixed as a quantity proportional to σ_e :

$$\theta = \kappa \cdot \sigma_e \quad (6)$$

The choice of the multiplicative parameter κ is based on the table of probability of the normal distribution which gives the correspondance between κ values and the percentage of thresholded distribution (Tab. 1 and Fig. 2).

Table 1
Percentage of standard distribution vs. κ .

| | | | | | | | | |
|----------|-----|-----|-----|------|------|------|------|-----|
| κ | .43 | .67 | .97 | 1.28 | 1.65 | 1.96 | 2.57 | 3.9 |
| % | 33 | 50 | 66 | 80 | 90 | 95 | 99 | 100 |

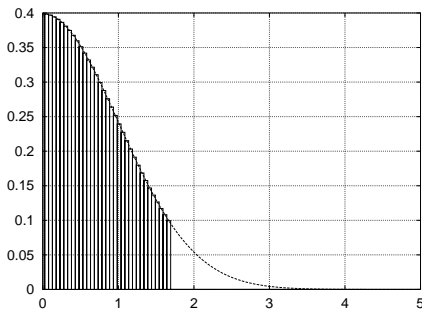


Fig. 2. Surface corresponding to 90 % of the positive part of a Gaussian distribution ($\kappa = 1.65$).

For $\kappa \approx 4$, 100% of the Gaussian distribution is taken into account [16]. Therefore, in order to properly cancel the noise contribution, and also for simplicity, we take $\kappa = 4$ for all video sequences shown here.

For convenience when dealing with numerical images, if we take the bit as entropy unit, we get from Eq. (2) and (5):

$$\sigma_e = \frac{2^{H_{bit}}}{\sqrt{2\pi e}} \quad (7)$$

Note the connection of this approach with a bit-plane slicing technique [17] if we consider the typical case of discrete positive observations coded with n bits (*i.e.*, an image of observations composed of n bit-planes). Indeed, by choosing $\kappa \approx 4$ to set the threshold, we simply obtain (cf. $\sqrt{2\pi e} = 4.13$):

$$\kappa \approx 4 \Rightarrow \theta = \kappa \frac{2^{H_{bit}}}{\sqrt{2\pi e}} \approx 2^{H_{bit}} \quad (8)$$

Hence, the method reduces to keeping the most significant bit-planes. The cutting is done at a bit-level specified by the value of 2^H . Only higher order bit-planes are kept. The lowest order planes that contribute to more subtle details or noise are discarded (Fig. 3).

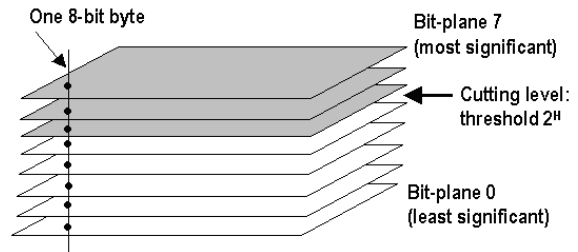


Fig. 3. Bit-plane slicing.

Fig. 4 illustrates our threshold setting method in two ideal cases: a symmetrical Gaussian source and an asymmetric one, obtained when considering as realizations absolute values instead of signed values. Actually, the asymmetric case corresponds to a realistic situation that typically arises in motion detection when we take as observations the absolute value of the frame differences (cf. Eq. (10) below).

The relationship between the entropy computed in the symmetrical case and the one obtained in the asymmetric case is derived in Appendix.

3.3. Algorithm and Conditions of validity

The proposed method relies on the following assumptions:

- the object pixels (signal) occupy a small part of the whole image, typically a few percent.

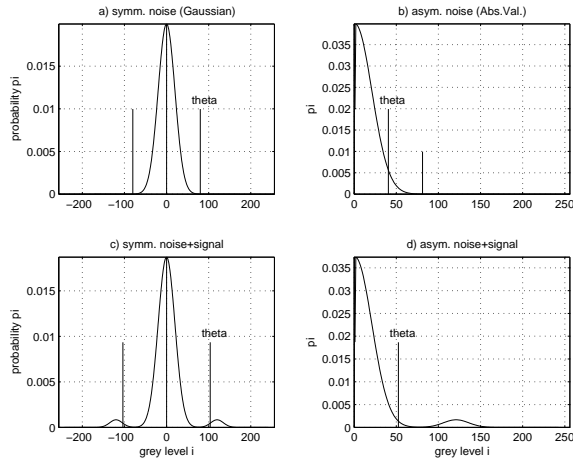


Fig. 4. Histograms and estimated thresholds θ for $\kappa = 4$. a) Symmetrical Gaussian histogram; b) Asymmetric Gaussian noise (absolute value); c) Symmetrical noise with signal added; d) Realistic situation of an asymmetric histogram resulting from noisy observations.

- the background pixels (noise) are prominent in the histogram.
- the additive noise is uncorrelated and of low-level compared to the signal: typically if the noise is Gaussian $G(m, \sigma)$, one should have a signal with grey levels greater than $m + 4\sigma$ (cf. Fig. 4c,d). It means that there should exist a grey level limit l that establishes an upper bound for the noise and a lower bound for the signal.
- the independance of the grey level probabilities is a priori assumed.

Note that those assumptions correspond to realistic situations for motion or edge detection.

Then, the following steps describe the proposed algorithm (only step 5 is context dependent).

- 1: Compute observations o
- 2: Estimate probabilities $pi = P[o=i]$
- 3: Compute entropy H in bits
- 4: Compute entropic deviation Sigma_e
- 5: Select K depending on application (typ. $K=4$)
- 6: Set Threshold according to Eq.6

3.4. Simulation results

To illustrate the principle of the method, let consider the sample image of Fig. 5 made of 7 grey

levels $\{i = 0 \dots 6\}$ with respective probabilities: $\{p_i = \frac{1}{2}, \frac{1}{4}, \frac{1}{16}, \frac{1}{32}, \frac{1}{8}, \frac{1}{32}, 0\}$. This image simulates

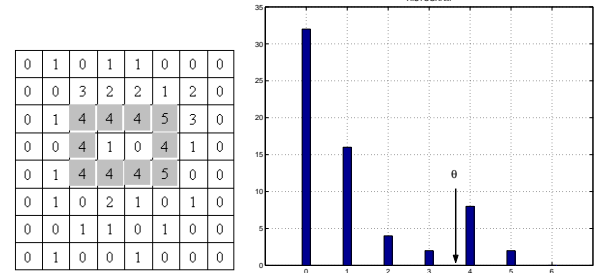


Fig. 5. Sample image of size 8×8 with corresponding histogram and estimated threshold for $\kappa = 4$.

a noisy map with a rectangular pattern (shown in grey). Tab. 2 gives the parameters computed from this image. The estimated threshold θ is well positioned on the histogram. Note that σ_e is significantly different from the usual computation of the standard deviation σ .

Table 2

Estimated parameters for sample image of Fig. 5

| parameter | H_{bit} | $2^{H_{bit}}$ | σ_e | $\theta = 4\sigma_e$ | σ |
|-----------|-----------|---------------|------------|----------------------|----------|
| value | 1.94 | 3.83 | 0.93 | 3.70 | 1.52 |

Fig. 6 shows the result of the proposed entropic binarization with $\kappa = 4$ for change detection on a synthetic sequence. Two mobile objects (a black square and a dark grey rectangle) are moving on a noisy background. They are well detected by the proposed method.

4. Some Applications

4.1. Application to motion detection

Motion detection consists in labelling each pixel or site s of the image at time t to get a binary map of temporal changes. The label at site s should ideally take the following logical value:

$$l(s) = l(x, y, t) = \begin{cases} "1" & \text{if } s \in \text{moving area,} \\ "0" & \text{if } s \in \text{static background.} \end{cases} \quad (9)$$

Assuming a static camera and a constant lighting of the scene, the basic observation computed from

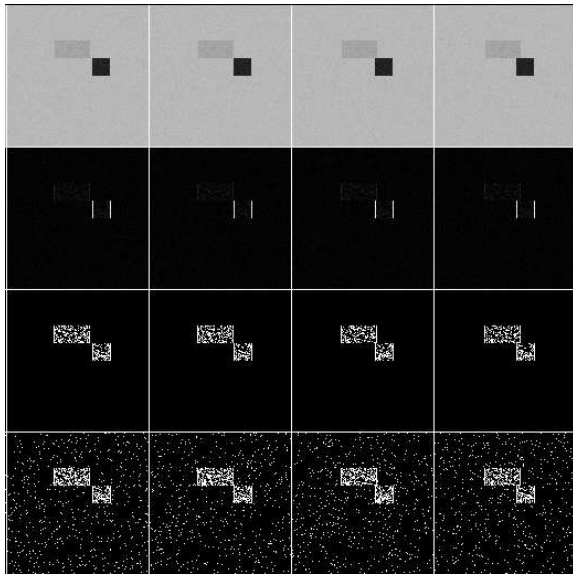


Fig. 6. From top to bottom: **Synthetic** sequence with two mobiles (frames number 12 to 15 in Fig. 7); Temporal observations (frame differences); Automatic binarization with $\kappa = 4$ ($H \approx 3\text{bits}$, $\sigma_e \approx 1.95 \Rightarrow \theta \approx 7.8$); Binarization with $\theta = 7$ for comparison purpose.

the data sequence is the absolute value of the temporal intensity difference between two consecutive images (frame difference):

$$o(s) = |I_t(s) - I_{t-1}(s)|. \quad (10)$$

Then, the label may be set by comparing the observation with a threshold θ :

$$l(s) = \begin{cases} \text{"1"} & \text{if } o(s) > \theta, \\ \text{"0"} & \text{otherwise.} \end{cases} \quad (11)$$

Note that the absolute value is taken in order to be insensitive to the sign of contrast between moving objects and background: dark (resp. bright) object moving on a bright (resp. dark) background.

This observation is noisy since the frame difference is sensitive not only to actual motion in the scene, but also to lighting variation (slight illumination changes) and to acquisition noise (due to camera and quantization). Therefore, an adequate thresholding technique is required to detect significant temporal changes.

The hypothesis of an additive Gaussian noise is commonly assumed. Various techniques based on

maximum likelihood tests have been proposed for the estimation of the noise level and the automatic setting of the threshold value [18–20]. Here, we show that the entropic approach described in section 3.2 is an alternative solution in order to automatically estimate the motion threshold.

Since the frame difference O is classically modeled as the sum of the relevant motion signal X plus an additive Gaussian noise $G(0, \sigma)$, property (iii) yields:

$$N(X) + N(G) \leq N(O) \leq P_X + \sigma^2. \quad (12)$$

where P_X is the average power of the motion signal.

Considering that the useful signal remains localized in the image (relevant temporal changes arise for a limited amount of pixels) whereas the random noise is present everywhere in the frame, the entropic contribution due to the useful signal is actually small in the sense understood by Shannon. Indeed P_X is small compared to the noise power σ^2 (*i.e.*, the average SNR over the entire frame difference is low, although the local SNR may be high). Therefore, property (iv) holds and we have:

$$\sigma_e^2 = N(O) \approx P_X + \sigma^2.$$

Whereas the noise power σ^2 is almost constant over time, the signal power P_X varies over time, since it depends on the actual motion present in the scene. Hence, $N(O)$ is sensitive both to the noise level σ and to the amplitude variations of actual motion in the scene. Its measure provides a means to set a motion threshold as given by Eq. (6).

One can see (Fig. 7) that the threshold adapts over time, depending on the amount of motion present in a scene. For sequence **Street 1**, the entropic deviation grows when a car enters the camera field of view, since more pixels undergo a temporal variation in intensity. On the opposite, when there is little motion in the scene, σ_e falls since there are very few temporal changes.

Fig. 8 shows the influence of κ for the detection of mobile pixels in the case of a static camera. The scene contains two cars and four pedestrians. As κ increases from 2 to 4, the background noise is better eliminated.

The sequence **Street 1** (Fig. 9) corresponds to a scene acquired also with a static camera, but

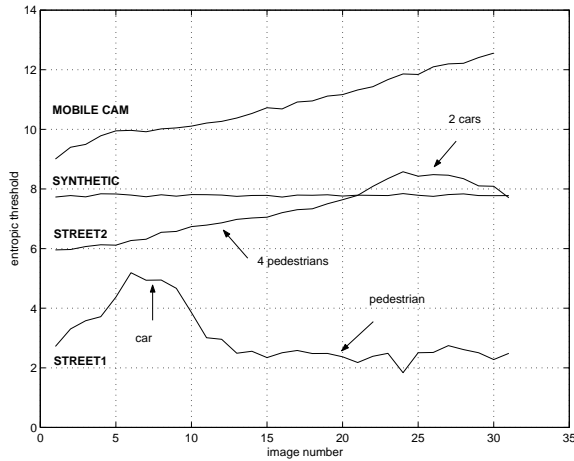


Fig. 7. Evolution with time of the threshold $\theta = \kappa \cdot \sigma_e$, on four sequences for $\kappa = 4$.

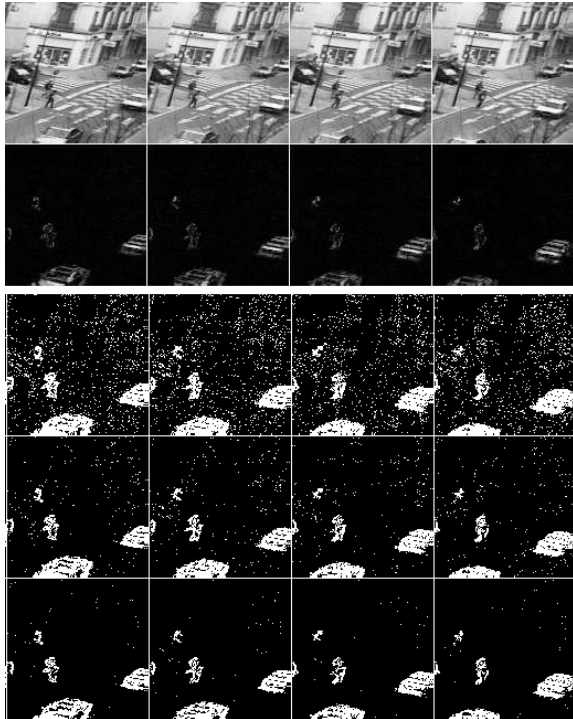


Fig. 8. From top to bottom : Four frames of sequence **Street 2** (images number 28 to 31); Observations as computed by Eq. (10) which are typical of a motion detection application; Binary maps obtained by entropic thresholding for $\kappa = 2; 3; 4$ respectively. The estimated entropic deviation (in average for the four frames) is: $\sigma_e \approx 2$. The corresponding thresholds are $\theta \approx 4; 6; 8$ respectively.

with low-pass filtering. Therefore the noise power is filtered, yielding a lower value for σ_e and hence a lower threshold compared to the other sequences (Fig. 7).

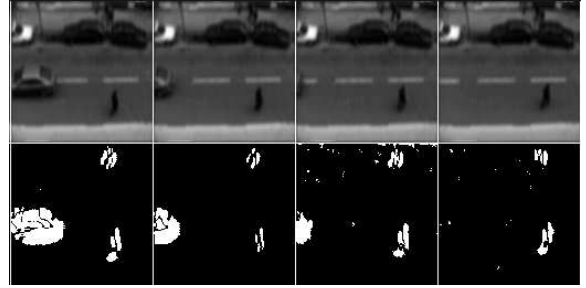


Fig. 9. Sequence **Street 1** that was low-pass filtered; change detection with $\kappa = 4$ ($\Rightarrow \theta \approx 3$) for frames 10 to 13 containing two pedestrians and a moving car.

Fig. 10 corresponds to the case of a mobile camera translating from right to left. The mobile contours are detected properly.

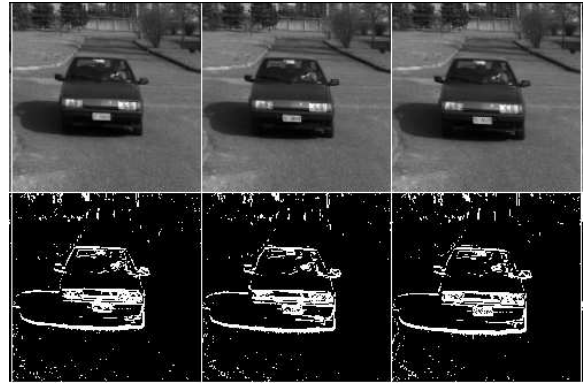


Fig. 10. Sequence **Mobile cam**: car approaching towards a mobile camera ; change detection with $\kappa = 4$ for frames 28 to 30 ($\theta \approx 12$).

The estimation of the entropy power is thus an adequate tool for setting automatically the threshold to be applied on temporal observations.

The efficiency and robustness of this thresholding technique has also been tested for face analysis application (Fig. 11) within a Markov random field framework integrating motion and color information. The white points correspond to pixels detected as mobile, informative of facial feature motion (especially lips). For more details about this application, the reader is referred to [21,22].



Fig. 11. **Lip** sequence: change detection with $\kappa = 4$. Pixels detected as mobile are marked in white.

For comparison purpose, we show on Fig. 12 the result of our thresholding technique for motion detection in correspondence with results obtained by Otsu and Pun methods. As can be seen, Otsu and

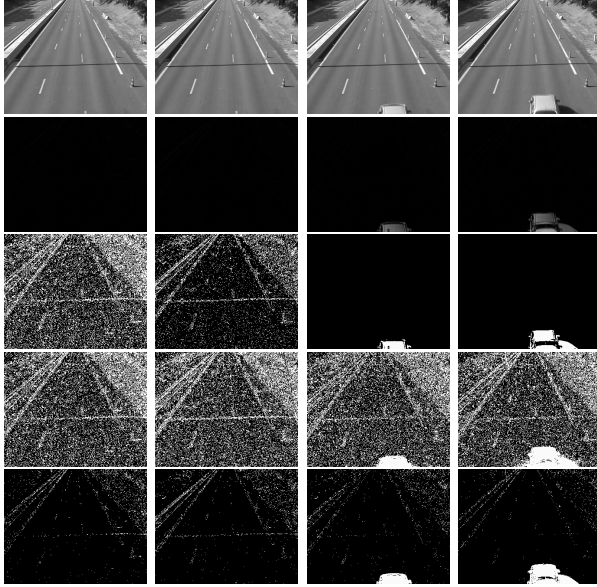


Fig. 12. Comparison of three thresholding techniques applied to motion detection. From top to bottom: Four consecutive frames of **Highway** sequence; Temporal observations (frame differences); Otsu thresholding; Pun thresholding; Proposed method.

Pun's methods are not well suited for motion detection. Depending on the presence or absence of moving objects (cars on a highway), the threshold variations are too important with Otsu method (too much noise detected in the background when there is no motion), whereas Pun systematically underestimates the threshold so that the maps are noisy (it tends to yield the same amount of black and white pixels). Our own method seems to be more robust for detecting temporal changes: the amount of noisy pixels in the background remains almost constant whatever the actual quantity of motion: it is little influenced by the amount of motion information present in the scene. Our threshold seems

to adapt better in that specific case. Fig. 13 compares the threshold variations over a long sequence of 100 frames.

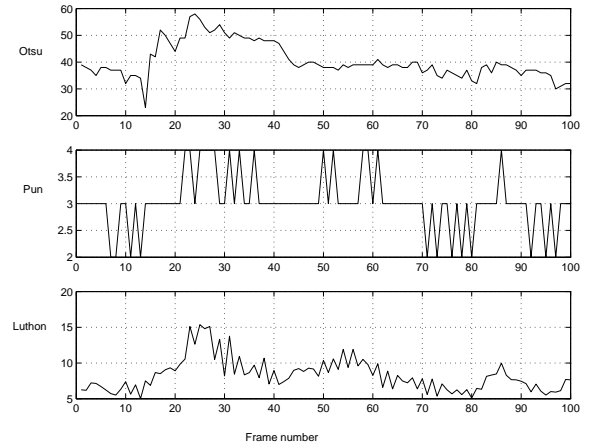


Fig. 13. Evolution with time of the thresholds for the three methods: Otsu, Pun, and proposed method.

4.2. Application to edge detection

Edge detection is classically based on thresholding the modulus of the spatial gradients in the image. The observation at site $s = (x, y)$ is computed as:

$$o(s) = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2} \quad (13)$$

where the spatial derivatives $\partial I/\partial x$ and $\partial I/\partial y$ may be estimated with various types of high-pass filters. Since the maximal values of $o(s)$ are located on the edges in the image plane, a simple detection criterion is the comparison to a threshold: if $o(s) > \theta$, then s is likely to be a contour pixel. The choice of the threshold is the key-point to extract the pixels that are good candidates for edges. Hysteresis thresholding for example requires the use of two thresholds (high and low) that are usually set manually.

Our approach allows an automatic determination of the threshold to be applied. Typical results of entropic thresholding are shown in Fig. 14 and 15 (outdoor and indoor images). Shown in the upper row are the real-world image, the vertical

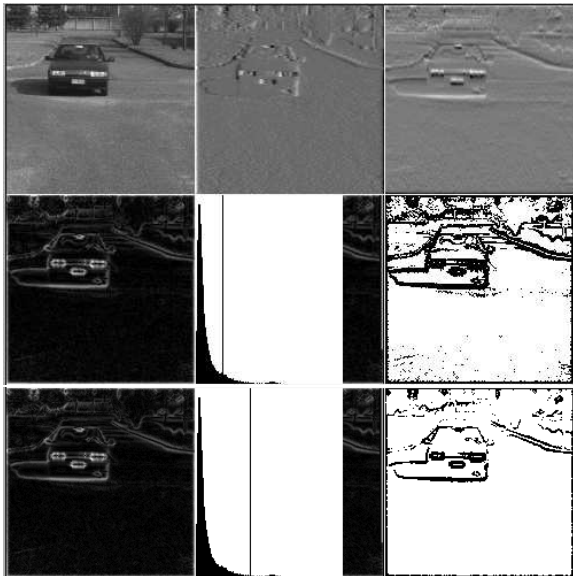


Fig. 14. Contour detection on outdoor image for $\kappa = 2$ and 4 respectively ($H_{bit} = 5.61$, $\sigma_e = 11.88$). The upper row shows the original image together with its horizontal and vertical spatial derivatives. The lower rows show the modulus of the spatial gradient, its histogram with estimated threshold positioning, and the final result of binarization exhibiting the candidate contour points.

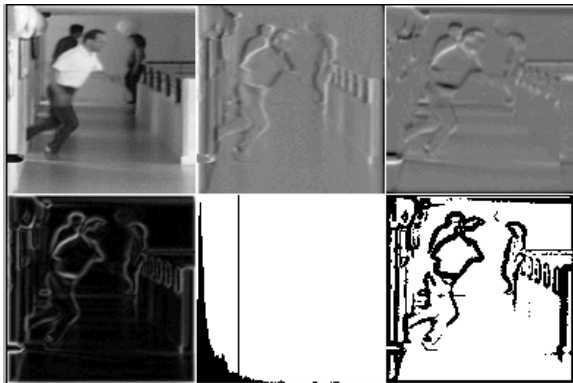


Fig. 15. Contour detection on indoor image ($H_{bit} = 5.88$, $\sigma_e = 14.34$, $\kappa = 2$). Upper row: original image with its horizontal and vertical derivatives. Lower row: spatial gradient in module with its histogram and the detected edge points.

and horizontal gradients computed with an exponential derivation filter [23]. The middle or lower rows show the modulus of gradient (which is the observation on which the entropic thresholding is applied), the histogram of observations with the

position of the estimated threshold, and the binary map obtained after thresholding showing the contour-candidates.

The histograms shown in Fig. 14 and 15 prove that we are in the same kind of situation as for motion detection: *i.e.*, a quick decaying histogram where the small contribution of the signal that is informative of contour points is confined to the trail of the distribution (higher grey levels). So that the proposed method may also be applied successfully.

5. Discussion

The entropic thresholding technique described here is both simple and efficient. It may be applied in many situations encountered in image preprocessing, where the data is corrupted by an additive noise spread all-over the observation grid. The binarization technique was used successfully for face analysis [21]. Another potential application for 2-D spectral analysis is described in [24]. We also tested the potential interest of this approach for the estimation of noise in the context of medical volume restoration [25]. In actual applications, a postprocessing is often applied after binarization (regularization by MRF, mathematical morphology like erosion, dilation, opening, closing...). This will of course influence the choice of κ . In such cases, one should lower the threshold by taking $\kappa = 2$ typically. Indeed, a nice postprocessing should hopefully remove the remaining noise that is spatially and/or temporally decorrelated.

It should be pointed out that this technique is costless when used in the context of image compression based on an entropic coding scheme (Huffman-like coding), since the entropy of the source is computed anyway for compression purpose. Further developments we are currently working on are the computation of a local entropic deviation, instead of the global computation over the whole frame, and the use for video indexing (automatic scene cutting).

As regards local computation, one strong limitation is that the signal should remain localized compared to noise. In areas with a lot of moving pixels, this is no longer true so the condition of validity

of our method does not hold anymore. Moreover, if the local area under investigation is too small, the statistics about the pixels may eventually become irrelevant, which is another limitation of any method based on probability and entropy computation (this remark is also true for other methods like the one of Pun).

As regards a possible extension to multi-thresholding, one possibility we are thinking about is the use of a low and a high threshold (as is usual for edge detection based on hysteresis thresholding [26]). One may for example take $\kappa = 2$ and $\kappa = 4$ in that case (see Fig.14). Concerning edge detection, another idea consists in applying our threshold selection method separately for the horizontal and vertical components of the spatial gradient instead of applying it on the modulus of the gradient as was done here, so that two specific thresholds can be estimated, one for each direction. This might be of interest for anisotropic gradient images or special textures.

Appendix

A.1. Proof of Eq. (5) in the discrete case

Let $O = \{o(s), s \in S\}$ be a random source of discrete observations on grid S with a Gaussian PDF denoted:

$$p[o(s) = i] = p_i = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{i-m}{\sigma}\right)^2}$$

where i is an integer value corresponding to a grey level: $i \in [i_0, i_M]$, and m stands for the mean value. Let consider the case of central symmetry: $i_0 = -M$ and $i_M = M$. The number of levels M is supposed to be big enough to get an acceptable approximation of the continuous distribution (typically $M = 255$). Of course, the distribution should not be truncated which implies the following conditions : $M > m + 4\sigma$ and $-M < m - 4\sigma$.

Note that in a practical situation, the probabilities p_i are estimated from the histogram: $p_i = f_i/T$ where f_i is the frequency of occurrence of level i and T is the total number of observations that is

supposed to be big enough (typ. $T = L \times C$ is the image size, asymptotically $T \rightarrow \infty$).

The a priori entropy is given by:

$$H = - \sum_i p_i \log p_i$$

Since $\log p_i = \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2} \left(\frac{i-m}{\sigma}\right)^2$, we have:

$$H = - \log \frac{1}{\sqrt{2\pi}\sigma} \sum_i p_i + \frac{1}{2\sigma^2} \sum_i p_i (i-m)^2$$

But $\sum_i p_i = 1$ and by definition: $\sum_i p_i (i-m)^2 = \sigma^2$, so that :

$$H = \log \sqrt{2\pi}\sigma + \frac{1}{2}$$

Hence $\sigma = \frac{\exp(H-0.5)}{\sqrt{2\pi}} = \frac{\exp(H)}{\sqrt{2\pi e}}$ where the entropy H is expressed in nat.

In that case, we see that $\sigma = \sigma_e$ where

$$\sigma_e = \frac{\exp(H)}{\sqrt{2\pi e}}$$

whereas $\sigma = \sqrt{\left(\sum_i p_i i^2\right) - m^2}$

A.2. Entropy in the asymmetric case

We establish here the entropy relationship between the central symmetry histogram and the asymmetric one, when one considers the absolute value of the Gaussian noise realizations. This is useful especially for motion detection, where observations are the absolute value of the temporal differences. In the central symmetry case (level i ranging from $-M$ to $+M$) we have $o(s) = i$ with probability p_i , where $i \in [-M, +M]$ and $p_{-i} = p_i$. Hence:

$$H_c = - \sum_{i=-M}^M p_i \log p_i = -p_0 \log p_0 - 2 \sum_{i=1}^M p_i \log p_i$$

In the asymmetric case, *i.e.* when we take as observations absolute values instead of signed values, we get new probabilities q_i for $i \in [0, M]$:

$$q_i = \begin{cases} p_i & \text{for } i = 0 \\ 2p_i & \text{for } i > 0 \end{cases} \quad (14)$$

Then, the entropy in the asymmetric case becomes:

$$H_a = -\sum_{i=0}^M q_i \log q_i = -p_0 \log p_0 - \sum_{i=1}^M 2p_i \log 2p_i$$

Hence:

$$H_a = -p_0 \log p_0 - 2 \sum_{i=1}^M p_i (\log p_i + \log 2)$$

$$H_a = H_c - \sum_{i=1}^M 2p_i \log 2 = H_c - (1 - p_0) \log 2$$

If we use base 2 for the logarithm, i.e. compute the entropy in bits, we get:

$$H_{abit} = H_{cbit} - 1 + p_0$$

Since p_0 is typically a few percents, we see that there is about 1 bit of information difference between H_a and H_c :

$$H_a < H_c \text{ with } : H_a \approx H_c - 1 \text{ bit.}$$

So that we finally have a factor of about 2 for the threshold positioning with $\kappa = 4$ in the ideal case of a pure asymmetric Gaussian noise (without any signal):

$$\theta \approx 2^{H_a} \approx \frac{2^{H_c}}{2}$$

which is confirmed by our simulations (see Fig. 4b).

Acknowledgments

We thank Drs. Yannick Berthoumieu and David Izquierdo from ENSEIRB, Bordeaux, for providing us with the road sequence of Fig. 12.

References

- [1] Y. Lim, S. Lee, On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques, *Pattern Recognition* 23 (9) (1990) 935–952.
- [2] N. Otsu, A threshold selection method from gray-level histograms, *IEEE Trans. on Systems, Man and Cybernetics* 9 (1) (1979) 62–66.
- [3] S. Reddi, S. Rudin, H. Keshavan, An optimal multiple threshold scheme for image segmentation, *IEEE Trans. on Systems, Man and Cybernetics* 14 (4) (1984) 661–665.
- [4] K. Aas, Detection and recognition of faces in video sequences, in: *Proc. Norwegian Conf. on Image Proc. Pattern Recogn. (NOBIM-konferansen)*, Oslo, Norway, 1998, pp. 1–6.
- [5] T. Pun, A new method for gray-level picture thresholding using the entropy of histogram, *Signal Processing* 2 (1980) 223–237.
- [6] T. Pun, Entropic thresholding, A new approach, *Computer Graphics and Image Processing* 16 (1981) 210–239.
- [7] J. Kapur, P. Sahoo, A. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, *Computer Vision, Graphics, and Image Processing* 29 (1985) 273–285.
- [8] P. Sahoo, S. Soltani, A. Wong, Y. Chen, A survey of thresholding techniques, *Computer Vision, Graphics, and Image Processing* 41 (1988) 233–260.
- [9] A. Wong, P. Sahoo, A gray-level threshold selection method based on maximum entropy principle, *IEEE Trans. on Systems, Man and Cybernetics* 19 (4) (1989) 866–871.
- [10] E. Jansing, T. Albert, D. Chenoweth, Two-dimensional entropic segmentation, *Pattern Recognition Letters* 20 (1999) 329–336.
- [11] S. Lee, S. Horng, H. Tsai, Entropy thresholding and its parallel algorithm on the reconfigurable array of processors with wider bus networks, *IEEE Trans. on Image Processing* 8 (9) (1999) 1229–1242.
- [12] P. Saha, J. Udupa, Optimum image thresholding via class uncertainty and region homogeneity, *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23 (7) (2001) 689–706.
- [13] F. Luthon, M. Liévin, Entropy power for thresholding technique in image processing, in: *XI European Signal Processing Conf. (EUSIPCO'02)*, Toulouse, France, 2002, pp. 605–608, vol. I.
- [14] J. Lee, M. Yang, Threshold selection using estimates from truncated normal distribution, *IEEE Trans. on Systems, Man and Cybernetics* 19 (2) (1989) 422–429.
- [15] C. E. Shannon, A mathematical theory of communication, *The Bell System Technical Journal* 27 (1948) 379–423, 623–656.
- [16] F. Reza, An introduction to information theory, McGraw-Hill, New York, 1961.

- [17] R. Gonzalez, R. Woods, Digital image processing, Addison-Wesley Publishing Company, Reading, MA., 1993.
- [18] T. Aach, A. Kaup, R. Mester, Statistical model-based change detection in moving video, *Signal Processing* 31 (2) (1993) 165–180.
- [19] Y. Z. Hsu, H. H. Nagel, G. Reckers, New likelihood test methods for change detection in image sequences, *Computer Vision, Graphics, and Image Processing* 26 (1984) 73–106.
- [20] K. Skifstad, R. Jain, Illumination independent change detection for real world image sequences, *Computer Vision, Graphics, and Image Processing* 46 (1989) 387–399.
- [21] M. Liévin, Analyse entropico-logarithmique de séquences vidéo couleur. Application à la segmentation markovienne et au suivi de visages parlants, Ph.D. thesis, National Polytechnic Institute, Grenoble, France (Sep. 2000).
- [22] M. Liévin, F. Luthon, Nonlinear color space and spatiotemporal MRF for hierarchical segmentation of face features in video, *IEEE Trans. on Image Processing* 13 (1) (2004) 63–71.
- [23] S. Castan, J. Zhao, J. Shen, Une famille de détecteurs de contours basée sur le filtre exponentiel optimal, in: 7e Congrès AFCET-INRIA-RFIA, Paris, 1989, pp. 23–36, vol. 1.
- [24] F. Luthon, X. Navarro, M. Liévin, Seuillage entropique en traitement d’images, in: 18e Colloque sur le traitement du signal et des images (GRETSI’01), Toulouse, France, 2001, pp. 353–356, vol.2.
- [25] M. Liévin, F. Luthon, E. Keeve, Entropic estimation of noise for medical volume restoration, in: 16th Int. Conf. on Pattern Recognition (ICPR’2002), Québec City, Canada, 2002, pp. 871–874, vol.3.
- [26] Y. Rungsunseri, A. Chéhikian, A real time system for extracting edges and lines in images, in: Proc. of 7th Scandinavian Conf. on Image Analysis, Aalborg, Sweden, 1991, pp. 839–846, vol. 2.