A robust segmentation and tracking method for characterizing GNSS signals reception environment

A. Cohen, C. Meurie, Y. Ruichek, J. Marais

▶ **To cite this version:**

A. Cohen, C. Meurie, Y. Ruichek, J. Marais. A robust segmentation and tracking method for characterizing GNSS signals reception environment. IS&T/SPIE Electronic Imaging, Jan 2011, San Francisco, United States. pp.787704, 10.1117/12.876674 . hal-04757077

HAL Id: hal-04757077
https://hal.science/hal-04757077v1

Submitted on 28 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A robust segmentation and tracking method for characterizing GNSS signals reception environment

A. Cohen[1] and C. Meurie[1] and Y. Ruichek[1] and J. Marais[2]

[1]Systems and Transportation Laboratory, Université de Technologie de Belfort-Montbeliard, 13 rue Ernest Thierry Mieg, 90010 Belfort Cedex, France;
[2]Univ Lille Nord de France, Lille, France - INRETS, LEOST, Villeneuve d'Ascq, France, 20 rue Elise Reclus, 59650 Villeneuve d'Ascq, France

ABSTRACT

This paper is focused on the characterization of GNSS signals reception environment by estimation of the percentage of visible sky in real-time. On previous works ([1,2]), a new segmentation technique based on a color watershed using an adaptive combination of color and texture information was proposed. This information was represented by two morphological gradients, a classical color gradient and a morphological texture gradient based on mathematical morphology or co-occurrence matrices. The segmented images were then classified into two regions: sky and not-sky. However, this approach has high computational cost and thus, cannot be applied in real-time. On this paper, we present this adaptive segmentation method with a texture gradient calculated by the Gabor filter and a region-tracking method based on a block-matching estimation. This last step reduces the execution time of the application in order to respect the real-time conditions. Since the application works for fish-eye images, a calibration and rectification method is required before tracking and is also presented on this paper. The calibration method presented is based on the straight line condition and thus does not use real word coordinates. This prevents measurement errors. The tracking results are compared to the results of the classification method (which has already been evaluated on previous works). The evaluation shows that the proposed method has a very low error and decreases the execution time by ten times.

Keywords: image segmentation, adaptive combination, color, texture, rectification, tracking

1. INTRODUCTION

Most of ITS applications rely on positioning system. Most of them are based on GNSS (Global Navigation Satellite System) such as GPS. As other radio-navigation solutions, satellite-based systems use propagation time measurements for positioning. Indeed, each satellite of the constellation broadcasts continuously its own signal and the role of the receiver is to estimate the propagation time in order to translate it into a distance, called "pseudorange". By triangulation, three simultaneous pseudoranges allow the receiver to compute its position in a 3D absolute referential. However in constrained environment, such as urban area, signals can be reflected or blocked by obstacles (building, trees, ...). Reflections add a delay to the propagation time estimation, in particular when the direct path of the signal is not available. In order to ensure a safe and accurate position to the user, mitigation techniques can be applied.[3] The simplest one consists in excluding corrupted signals when detected but this policy enhances unavailability of the service.
A more efficient approach would be, if possible, to use all available signals, weighting or estimating noises to reduce the error induced.[4] At the receiver level, the detection of disturbed signal is not always easy. Some criteria can be used such as signal to noise ratio (SNR) but a threshold to distinguish direct signal or not has to be chosen.[5] The method developed in this paper uses a video record of the environment surrounding the GPS antenna. This system has been developed for the first version of the PREDISSAT tool[6] and has inspired other developments with infra-red cameras.[7,8] The goal of this paper is to benefit from new image processing developments in order to enhance satellite state reception and masking elements. In the context of our application, images present two important informations: color and texture. It is hence useful to use a segmentation method based on these informations. Angulo[9] proposes a segmentation method combining color and texture informations. However, this method involves many parameters, which are difficult to adjust according to the considered

application. Based on the texture gradient proposed by Angulo,[9] we have proposed in previous works a new segmentation method based on a non-parametric and adaptive combination of color and texture information.[1,2,10] In this paper, we develop a new texture gradient based on Gabor filter and a new segmentation technique using an original and adaptive combination (considering local image content) to take into account color and texture gradients. The segmented images are then used as input for a kmeans algorithm in order to classify the different regions of the fish eye images. In this way, we can identify the satellites that are situated in the sky region (direct satellite reception state). On a second stage, we use a region-tracking method in order to reduce computational time of our application. The aim consists on estimating a classification of the image at the instant t by using the classification of the precedent frame. This allows to apply the full segmentation method every N frames, which reduces drastically the total execution time. This stage, however, requires an image calibration and rectification step presented in the literature[11,12] and recalled on this paper, since the method is applied to fisheye images.

The paper is organized as follows: Section 2 presents the adaptive segmentation method based on the combination of a color morphological gradient and a texture gradient using Gabor filter. Section 3 presents the calibration and rectification of fisheyes images. A region tracking method with a management of unknown objects found in the new image is presented in section 4. Before concluding, some experimental results are illustrated in section 5.

2. AN ADAPTIVE SEGMENTATION METHOD COMBINING COLOR AND TEXTURE INFORMATIONS

The watershed algorithm is one of the principal mathematical morphology image processing operations.[13,14] It permits to segment an image into homogeneous regions from a seeds image (markers) and a potential image (color or/and texture gradient). Image segmentation based on the watershed algorithm has proved to be a powerful segmentation tool but, unfortunately, when directly applied to an image, this algorithm presents a strong over segmentation. One way to suppress this over segmentation is to use a non-parametric hierarchy of watershed, also known as the waterfall algorithm.[15] Several authors propose different types of gradients including several ordering of color vectors.[16,17] But in the context of our application, the processing time of this approach would be too important. That is why, we prefer to define an specific image of seeds positioned experimentally and adapted to our application (few germs located on the borders of the image, since objects tend to be found on this region of the image due to fish-eye lens distortion). On the other hand, images of our application present two important informations: color and texture, that is why, we propose a segmentation method based on a watershed algorithm which used an adaptive and local combination of these informations. The watershed algorithm makes regions grow from the initial seeds using the priority given by an adaptive potential image described below (structural gradient based on color and texture informations).

2.1 Texture gradient definition using Gabor filter

Many methods exist in the literature to extract texture information of an image. Our previous works[1,2] use the mathematical morphology and the co-occurrence matrices to obtain a texture gradient and propose a structural gradient combining color and texture informations. In the paper, we have chosen to use Gabor filter to extract the texture information.

A 2D Gabor function can be viewed as a sinusoidal plane modulated by a gaussian envelope. Mathematically, it is defined as:

$$G_{\lambda,\theta,\varphi,\sigma,\gamma}(x,y) = \exp -(\frac{x'^2 + \gamma^2 y'^2)}{2\sigma^2} \cos\left(2\pi\frac{x'}{\gamma} + \varphi\right) \tag{1}$$

where

$$\begin{aligned} x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{2}$$

The variance σ of the gaussian function determines the size of the receptive field, the wavelength λ of the sinusoidal function determines the spatial frequency $\frac{1}{\lambda}$, the orientation θ corresponds to the normal to the parallel stripes of the Gabor function, the spatial aspect ratio γ specifies the ellipticity of the support of the Gabor function (for $\gamma = 1$, the support is circular and for $\gamma < 1$ the support is elongated in the orientation of the parallel stripes of the function), the phase offset φ determines the symmetry of the Gabor function (for $\varphi = 0$ and $\varphi = \pi$ the function is symmetric with respect to the center of the plane, for $\varphi = -\frac{\pi}{2}$ and $\varphi = \frac{\pi}{2}$ the function is antisymmetric).

If we consider a filter, a Gabor filter can be defined as the convolution of a given image (input image) f and a 2D Gabor function G, also called Gabor kernel. This convolution results in a filtered image or Gabor feature image r, where D is the domain of the input image.

$$r_{\lambda,\theta,\varphi,\sigma,\gamma}(x,y) = \iint_D f(x,y) G_{\lambda,\theta,\varphi,\sigma,\gamma}(x-\varepsilon, y-\eta) d\varepsilon d\eta \tag{3}$$

This kind of filter acts as a local band-pass filter for the spatial frequency of the Gabor kernel (given by λ) in the direction of the orientation given by θ. In other words, by using different values for wavelength, variance of gaussian and orientation, one can design a filter that is responsive to bars of a given thickness and orientation. Gabor filters have been proved to be closely related to the response of simple cells in the primary visual cortex of primates, also called "bar cells". Thus it is widely used for texture feature extraction.[18–21]

In terms of parameters consideration, the variance σ of the gaussian function determines the size of the receptive field. The ratio $\frac{\sigma}{\lambda}$ determines the spatial frequency bandwidth b. For our studies, b is chosen to be one octave, which corresponds to the value 0.56 of the ratio $\frac{\sigma}{\lambda}$. Therefore, the parameter σ is determined by 0.56λ. The spatial aspect ratio γ is set to 0.5. Hence, the only parameters to be set are: λ, θ and φ.

The outputs of a symmetric and an antisymmetric filter can be combined for each point of the input image, yielding a quantity called Gabor energy. This response is associated to the so-called complex cells in the primary visual cortex which are very sensitive to orientation. The Gabor-energy quantity can be mathematically defined as:

$$e_{\lambda,\theta}(x,y) = \sqrt{r^2_{\lambda,\theta,0}(x,y) + r^2_{\lambda,\theta,-(\pi/2)}(x,y)} \tag{4}$$

where $r_{\lambda,\theta,0}(x,y)$ and $r_{\lambda,\theta,-(\pi/2)}(x,y)$ are the outputs of a symmetric and antisymmetric filters. The image that groups the Gabor-energy quantities associated to each pixel of the input image f is called the Gabor-energy image ($e_{\lambda,\theta}$). In a $\lambda-\theta$-Gabor-energy image, pixels associated to a texture with an orientation given by θ and a thickness given by λ are brighter. It is therefore very important to chose a set of spatial frequencies and orientations that cover the entire spatial-frequency domain and orientation domain of all textures that are present on the picture, since the textures associated to an orientation or a spatial-frequency that does not make part of the used set will not be detected. For our experiments we have chosen to use two different spatial frequencies (5 and 3) and eight different equidistant orientations ($\theta = k(\pi/8)$, $k = 0, 1, ..., 7$). It is important to notice that the Gabor-energy quantity can only be computed over a gray-level image, hence it is computed separately for each color component of the input image f in the RGB color space. This results in a total of 16 Gabor-energy images calculated for each color component which gives a total of 48 Gabor-energy images per input image.

Thus, a morphological texture gradient is defined using references to the concept of texture feature images. For the Gabor filter technique, this concept refers to the Gabor-energy images. A morphological gradient is defined, for an image g, as the residue of dilatation and erosion (computed usually with a structuring element of size 1): $Q(g(x)) = \delta_\beta(g(x)) - \epsilon_\beta(g(x))$. Each texture feature image will be refereed to as t_k. The texture gradient of the image is finally defined in reference to the morphological gradient of each t_k, where K is a set of ordered pairs (λ, θ) of sizes and orientations:

$$Q_{tex}(f_{tex}(x)) = \bigvee_{k \in K} [Q(t_k(x))] \tag{5}$$

2.2 Morphological color gradient definition

The classical definition of the morphological gradient for a gray scale image g is given by: $\nabla g = \delta(g) - \epsilon(g)$. The extension of gray scale image algorithms to color or vector valued images is not simple since there is no natural ordering on a set of color vectors, and more generally of multivariate data. According to Barnett,[22] there are several possible types of multidimensional vector ordering: marginal ordering, reduced ordering, partial ordering and conditional ordering, etc. In this paper, the conditional ordering is considered.

Let $x_1, x_2, \cdots, x_n$ denote a set of n p-dimensional vectors: $x_i = \{x_{1(i)}, x_{2(i)}, \cdots, x_{p(i)}\}, x_i \in R^p$. In the conditional (also called lexicographic) ordering, the vectors are ordered according to a hierarchical order of the component (Red then Green then Blue in RGB color space). For two vectors x_i and x_j, one has:

$$x_i \leq x_j \begin{cases} x_{1(i)} < x_{1(j)} \text{ or} \\ x_{1(i)} = x_{1(j)} \text{ and } x_{2(i)} < x_{2(j)} \text{ or} \cdots \\ x_{1(i)} = x_{1(j)} \text{ and } x_{2(i)} = x_{2(j)} \cdots \; x_{p(i)} < x_{p(j)} \end{cases}$$

If f is a color image, $\delta(f)$ and $\epsilon(f)$ are color vectors and the classical morphological color gradient ∇f is given by: $\nabla f = \delta(f) - \epsilon(f)$.

2.3 Structural gradient definition

In order to achieve a robust and reliable segmentation, it is very useful to use both texture and color information. The main idea is to produce a structural gradient by combining the texture and color gradients. The problem is the fact that the color gradient is a color image while the texture gradient is a gray level one. To solve this problem, the proposed method starts by decomposing the color gradient image Q_{col} into its three components, which are Q^R_{col}, Q^G_{col} and Q^B_{col}. In the next step, each component of the color gradient image is combined with the texture gradient image Q_{tex} (which can be $Q_{ASM}(f(x))$ or $Q_c(f(x))$). This operation produces three gray levels images Q^R, Q^G and Q^B:

$$\begin{cases} Q^R = Q^R_{col} \otimes Q_{tex} \\ Q^G = Q^G_{col} \otimes Q_{tex} \\ Q^B = Q^B_{col} \otimes Q_{tex} \end{cases} \tag{6}$$

where the operator $\otimes$ represents the combination of two gray-level images which will be described on the following sections.

Q^R, Q^G and Q^B can be interpreted as the color components of a new color image, which is proposed to define the needed structural gradient. In other words, the structural gradient Q is defined as a color image with Q^R as the red component, Q^G as the green component, and Q^B as the blue component. This combination approach is suitable because, not only color information is preserved, but also the texture information is added to each color component. Indeed, texture, which is not a color phenomenon, is supposed to affect all colors equally. To combine a component of the color gradient image q and the texture gradient image r, three techniques are used: fixed combination, adaptive combination, supremum combination. Let h be the output of the combination process (represented by $\otimes$ on the preceding equation), which is applied for each pixel.

2.3.1 Fixed combination

The fixed combination is defined as a barycentric sum of the color gradient image and the texture gradient image. It uses a global weighting coefficient referred to as α:

$$h(p) = \alpha q(p) + (1 - \alpha) \times r(p) \tag{7}$$

where α is a constant coefficient taking its value in $[0; 1]$. The combination technique is not generally suitable, due to the coefficient α, which is constant for the entire image. Indeed, one may need to give priority to color or texture according to their importance in the different zones of the image. This technique requires manual adjustment of the coefficient α according to the content of the image.

2.3.2 Adaptive combination

The proposed adaptive combination strategy uses a modular combination of texture and color gradients according to the content of the image. It implies two advantages: First, it gives priority to the most important information (the color or the texture) for a given pixel. Second, it constitutes an automatic method, which can perform for all types of images. The adaptive combination is expressed as follows:

$$h(p) = \alpha_p q(p) + (1 - \alpha_p) \times r(p) \tag{8}$$

α_p is a coefficient taking its value in $[0; 1]$. It is calculated for each pixel p in order to give a high weight to the image that provides the most important information for the pixel. In other words, α_p is high if the information is more important for q than for r (and vice-versa). It is computed as follows:

$$\alpha_p = \frac{q(p)}{q(p) + r(p)} \tag{9}$$

2.3.3 Supremum combination

Using the same principle as in the adaptive strategy, the supremum combination is sensibly different. There is actually no combination at pixel level. Indeed, for a given pixel, the modular gradient is either a copy of the color gradient or the texture gradient depending on which one of them provides the biggest amount of information (the supremum). This combination has the same advantages as the adaptive one.

$$h(p) = \begin{cases} q(p), & \text{si } q(p) \geq r(p) \\ r(p), & \text{si } r(p) > q(p) \end{cases}$$

Figure 1 illustrates some segmentation results with different values of α (that corresponds to the color coefficient) and the results with the use of adapative or supremum combination.

3. CALIBRATION AND RECTIFICATION OF FISHEYE IMAGES

The proposed segmentation method (using an adaptive/supremum combination) gives very satisfying results but uses a computational time which is too important. Thus, we propose to use a tracking step (on the classified image) in order to limit when it is possible (for n images) the use of the segmentation step. For that, it is necessary to use rectified image by the strategy described in this section.

(a) *Image* (b) $C = 0.2$ (c) $C = 0.7$ (d) $C = 1$ (e) *Adaptive* (f) *Supremum*

Figure 1. Segmentation results (left to right: initial images, segmented images (columns 2-6: with C=0.2/T=0.8; with C=0.7/T=0.3; with C=1/T=0; with adaptive combination, with supremum combination)

3.1 Distortion model

The process of conception and assembly of camera lenses is not perfect. This is why images (specially fisheye images) suffer from distortion. This distortion can be divided into three components: 1/ Shift of optical center, 2/ Radial distortion, 3/ Decentering distortion. For the following section, let q be the ideal projection of a scene point (with no distortion). With the distortion q is mapped on q'. We will note (x, y) (resp.(x', y')) the Cartesian coordinates of q (resp. q') and (r, ϕ) (resp (r', ϕ')) its polar coordinates. The equations recalled in this section were presented by.[11]

A shift of the optical center corresponds to a shift of the image detector in the imaging plan. For correcting this distortion, an estimation of the coordinates of the optical center $C = (x_c, y_c)$ is necessary. This affects the coordinates of a point in the image in the following way:

$$\overline{x} = x - x_c \quad , \quad \overline{y} = y - y_c \tag{10}$$

Radial distortion shifts the image points in the radial direction. In other words, points are shifted closer or further away from the optical center. This is due to the radial curvature of the lenses. This distortion can be modeled by:

$$\Delta r(q) = \sum_{i=1}^{\infty} C_{2i+1} r^{2i+1} \quad \text{o} \quad r = \sqrt{\overline{x}^2 + \overline{x}^2} \quad \text{et} \quad \tan(\phi) = \frac{\overline{y}}{\overline{x}} \tag{11}$$

where C_{2i+1} represent the radial distortion parameters. Terms over the fifth order can be ignored without an important loss of precision. The equation is then reduced to:

$$\Delta r(q) \approx C_3 r^3 + C_5 r^5 \tag{12}$$

Decentering distortions appear because the lens components as well as the image detector are not orthogonal with respect to the optical axis. This distortion is tangential to the radial direction and can be modeled by:

$$\begin{array}{l} \Delta T_x(q) = [P_1 r^2 (1 + \cos^2(\phi)) + 2 P_2 r^2 \sin(phi) \cos(phi)] \cdot [1 + \sum_{i=1}^{\infty} P_{i+2} r^{2i}] \\ \Delta T_y(q) = [P_2 r^2 (1 + \sin^2(\phi)) + 2 P_1 r^2 \sin(phi) \cos(phi)] \cdot [1 + \sum_{i=1}^{\infty} P_{i+2} r^{2i}] \end{array} \tag{13}$$

where P_1, P_2 et P_{i+2} are the distortion parameters and ΔT_x, ΔT_y represents distortions on x and y directions. Again, higher-order terms can be ignored. The approximated model is then given by:

$$\begin{aligned}\Delta T_x(q) &\approx [P_1 r^2(1+\cos^2(\phi)) + 2P_2 r^2 \sin(\phi)\cos(\phi)] \\ \Delta T_y(q) &\approx [P_2 r^2(1+\sin^2(\phi)) + 2P_1 r^2 \sin(\phi)\cos(\phi)]\end{aligned} \tag{14}$$

which determines the complete distortion model:

$$\begin{aligned}\Delta x(q) &\approx [\cos(\phi)\delta r(q) + \delta T_x(q) \\ \Delta y(q) &\approx [\sin(\phi)\delta r(q) + \delta T_y(q)\end{aligned} \tag{15}$$

The distorted point q' can be also calculated as:

$$x' = x + \Delta x(q) \quad , \quad y' = y + \Delta y(q) \tag{16}$$

3.2 Calibration

The calibration process estimates the parameters of the camera model. In other words, it computes the parameter vector $P = \{C_1, C_2, P_1, P_2, x_c, y_c\}$. These parameters cannot be exactly determined since the distortion is not linear. In order to be able to estimate these parameters, we will define an evaluation function that can be then minimized by any general iterative minimization function. This function will be defined as a consequence of the straight line constraint. This method is based on the works of.[11,12]

The main goal of the calibration process is to obtain the transformation that goes from the image plane to a perspective projected image. The following property is taken into account: an image is in perspective projective only if the projection of every straight line of the scene is a straight line in the image. However, due to distortions, straight lines project into curves in fisheye images. Therefore, it is necessary to detect the original straight lines from the scene and measure their distortion degree. This will allow us to define an evaluation function to minimize.

We will assume that we dispose of a selection of points appertaining to non-radial straight lines from the scene. A calibration grid with distinct points should be enough for this task. Then we perform a normalized sum of square distances from these points to the lines that approximates them to a straight line. This will be the evaluation function. More specifically, let $\{q' = (x', y')\}$ be the set of selected point and let $P = \{C_1, C_2, P_1, P_2, x_c, y_c\}$ be a candidate parameter vector. The application of this vector to the set of points results in a rectified set of points noted as $\{q = (x, y)\}$. These points are supposed to be aligned if a good vector P was used. On a second step, the least-square method is used of estimate the straight lines that approximate the points $\{q = (x, y)\}$ the best.

Let l be the best line for a set of points q of size n_l. l can be parameterized by $y = mx + n$. The normalized error, which will be used as a measure of the distortion, associated to one single point q is defined by:

$$e = \frac{(y - mx - n)^2}{n^2} \tag{17}$$

Let L be the number of straight lines defined, and n_l the number of points that make part of each one of these lines. The evaluation function ξ is defined bellow. This function can be then minimized by any general minimization algorithm. For our experiments we have chosen to use the simplex method presented by.[12] Some example of rectification results are illustrated in the figure 2.

$$\xi = \sum_{l=1}^{L} \sum_{i=1}^{n_l} \frac{(y_{i,l} - mx_{i,l} - n)2}{n^2} \tag{18}$$

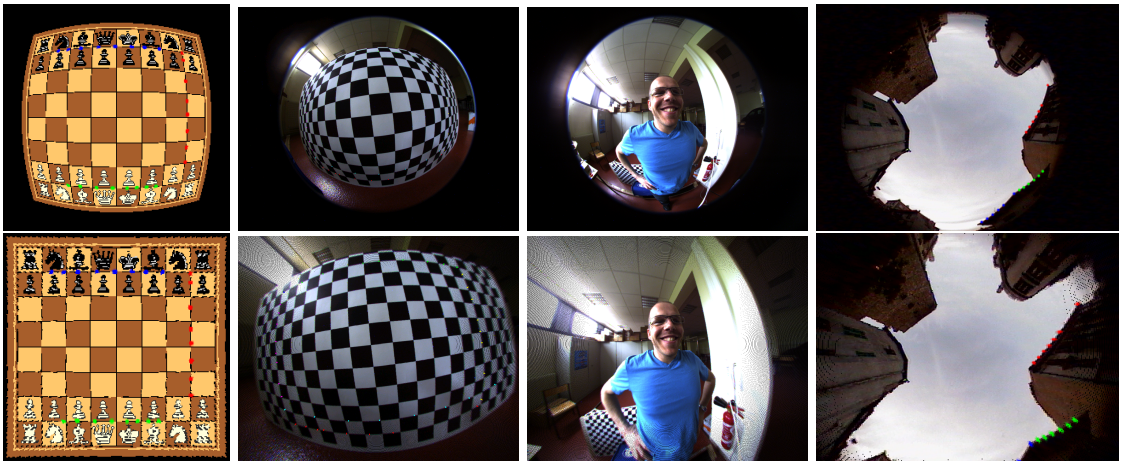


Figure 2. Rectification results obtained on one image of literature and on three images acquired in real conditions.

4. REGION TRACKING METHOD

Several space-time tracking methods are presented in the literature.[23,24] These methods are generally used for object tracking in video sequences. The method that we chose for our application is strongly inspired by the works presented on.[23]

Let I_t and I_{t+1} be two consecutive color images for a given image sequence. Let C_t be a classification that assigns a label to each pixel of I_t. The main goal of the tracking stage is to estimate a classification C_{t+1} of the image I_{t+1} by using C_t and taking advantage of the small amount of change between two consecutive frames. This process can be separated in two mains steps: 1/ Movement estimation and classification projection, 2/ Management of unknown elements found on the new image.

4.1 Movement estimation and classification projection

For our application, the movement estimation should be an easy task, since only the camera moves and not the different objects of the scene (and even when they do, they do not affect significantly the segmentation results). This means that, ideally, the entire image should move in the same way. However, since the fish-eye distortion cannot be rectified to a 100%, the movement of the image is not linear, which forces us to chose a method that can estimate different movement vectors for different parts of the image.

The chosen method was the block-matching algorithm. This algorithm estimates the movement between two consecutive images by performing a division of one of the images into regular sized blocks. To be more specific, the reverse mode of this algorithm is chosen, for which the image I_{t+1} is divided into regular blocks and then a research is performed across the image I_t in order to find the movement vector (u, v) corresponding to each block of the image I_{t+1}.

The matching criterion between two blocks that is generally used on this algorithm is the SAD (Sum of Absolute Differences). However, since we work with color images, each pixel is associated to a vector of dimension 3 that represents its color. This is why we redefine the SAD as the sum of the distance between the color vectors associated to the different pixels of each block. We will call this sum S, which is a function of a specific movement vector (u, v).

$$S(u,v) = \sum_{i=1}^{N}\sum_{j=1}^{N} \|I_{t+1}(i,j) - I_t(i+u, j+v)\| \tag{19}$$

where $I_{t+1}(i,j)$ represents the color vector at the coordinates (i,j) at the instant $t+1$.

The goal is to find the movement vector (u,v) corresponding to each block of I_{t+1} so that the sum S is minimal. This can be accomplished by several methods. The simplest one consists on performing an exhaustive research of the block of I_t that minimizes the sum S for every block of I_{t+1}. This method, although very precise, is also very slow, which interferes with our main motivation of performing a tracking algorithm in the first place. We have, instead, chosen to use the Successive Elimination Algorithm (SEA) presented by.[25] The SEA algorithm has the same results as the exhaustive research, but it performs a successive elimination of the research positions, which decreases the number of times that the sum S is computed, which is the most time consuming operation of the algorithm. This algorithm described in[25] and recalled below performs these successive eliminations by applying the following criterion for each block:

$$\begin{array}{l} R - M(u,v) \leq S(u,v) \\ M(u,v) - R \leq S(u,v) \end{array} \tag{20}$$

where $M(u,v) = \sum_{i=1}^{N} \sum_{j=1}^{N} \|I_t(i+u, j+v)\|$ represents the norm sum of a candidate block of I_t for vector (u,v) and $R = \sum_{i=1}^{N} \sum_{j=1}^{N} \|I_{t+1}(i,j)\|$ represents the norm sum of a block from I_{t+1}. $S(u,v)$ is the distance sum presented before.

This criterion lets us increase the speed of the algorithm in the following way: Let's assume that $S(m,n)$ has already be calculated for an initial vector (m,n). It is clear that for the following steps the only interesting vectors (u,v) will be those that can decrease these sum. According to the previous formula, this is only true if $R - M(m,n) \leq S(u,v) \leq R + S(m,n)$. The result is the ground of the SEA algorithm. It restrains the research among those blocks that satisfy the previous equation. However, the success of this algorithm depends on the fast computation for the norm sums for each possible block of I_{t+1}.

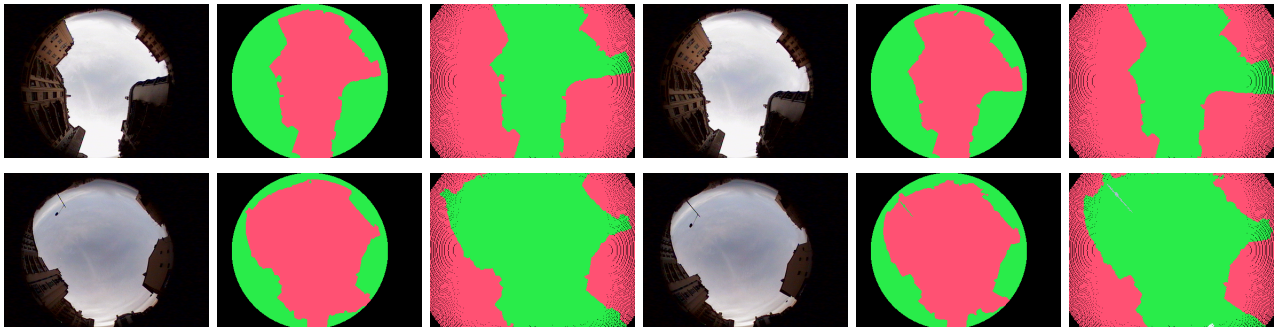
4.2 Management of unknown objects found in the new image

Once the projection of the classification is done (by moving each block of C_t according to the movement vector found by the SEA algorithm), it is expectable to find some new regions of I_{t+1} that are not taken into account by the estimation of C_{t+1} (which we will call C'_{t+1}). It is then necessary to do a classification of the non-labeled regions of C'_{t+1} in order to obtain a good estimation of C_{t+1}. This is done by going through every pixel of the non-labeled regions of C'_{t+1} and comparing its color with the mean color of each existing class of C'_{t+1}. Each one of these pixels is then labeled according to the class whose color is nearest to its own.

5. EXPERIMENTAL RESULTS

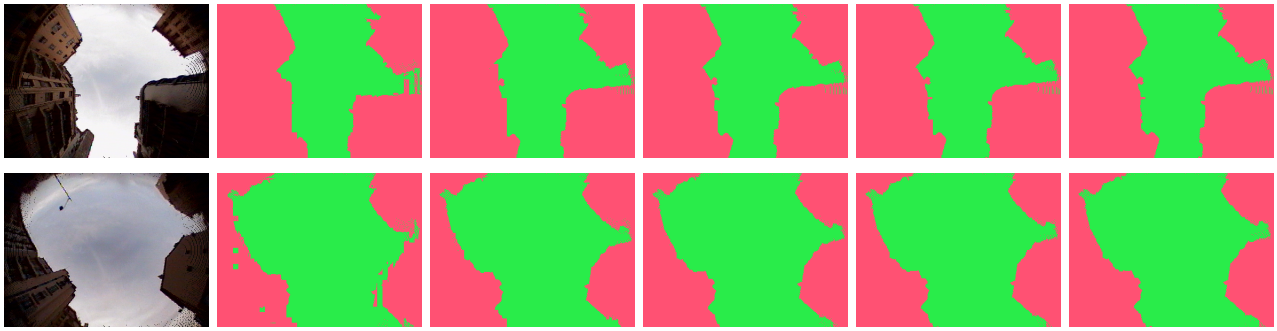
In this section, the discussion is focused on the classification results, which are based on the segmentation and tracking results presented in the previous sections. The classification process is performed using the K-means algorithm. The goal is to classify the image regions into two classes: sky region and not-sky region. For the evaluation, a classification reference image is created and an evaluation method which provides the percentage of pixels that are correctly classified is used.

The classification results obtained before the tracking step are shown in figures 5 and 6. One can notice that the classification results are different according to the color coefficient used. The segmentation results based only on the texture information (with C=0/T=1) are worse than the other combinations. In average, those obtained with C=0.7/T=0.3 or C=1/T=0 are satisfying expected one image of the database. For example, for the third image with the C=0.7/T=0.3 (column 3, line 3), the vegetation is classified as sky, and for the eighth image with a C=1/T=0 (column 8, line 4) the sky is not classified as such. The drawback of this fixed approach is that it requires a manual adjustment of the weighting coefficients which are global for all the pixels of the image, while the proposed method (adaptive or supremum combination) does not require any parameters setting. The adaptive or supremum combination takes into account local image content by automatically computing the



(a) I_t (b) C_t (c) C_t rectified (d) I_{t+1} (e) C_{t+1} (f) C_{t+1} rectified

Figure 3. Rectification results obtained for two real images.



(a) *Image* (b) $B = 10$ (c) $B = 16$ (d) $B = 20$ (e) $B = 24$ (f) $B = 26$

Figure 4. Tracking results for different bloc size B.

weighting coefficients for color and texture. Moreover, this method provides very good results, compared with a method based on a fixed combination of color and texture information.

Since the main goal of the application is to determine the visible percentage of sky present in fisheye images in real time, it is necessary to articulate the segmentation, rectification and tracking stages presented in the previous sections. Figure 3 illustrates the rectification results obtained for two real images of the database. The computation time of the segmentation and classification stage is 1min30, while the tracking and rectification stages can be executed in less than a second. Therefore, the tracking is executed in a loop by using the results from the segmentation and classification as original images. However, the tracking, while satisfying, is not as accurate as the segmentation stage. This is why the tracking can only be executed on a limited number of frames N and then the segmentation and classification process has to be used again. The number of frames N has to be determined in an experimental way ($N = 10$ in our case). The average process time per image is reduced approximately by a $1/N$ factor in respect to the original execution time. The good classification rate obtained by the tracking method for different block sizes B are presented on table 1. For a better visualization, the figure 4 illustrates the different tracking results according to different block sizes B for two images of the database. The block size must adapt to the image size. According to the obtained results, we can conclude that the optimal block size is 20×20. We can assume that this parameter does not change significantly from image to image for the same application since all of the images have the same size and the structures in them also have similar sizes. It is also important to observe that the choice of B has a very big influence on the classification results. Comparing to the previous results (without tracking step[1]), the good classification rate using the tracking step is sensibly equal but the computation time is decreased (9.7 s/image instead of 90 second/image).

6. CONCLUSION

An image segmentation method based on a combination of textural and color information for characterizing GNSS signals reception environment is proposed. This method fuses first the three morphological color gradient

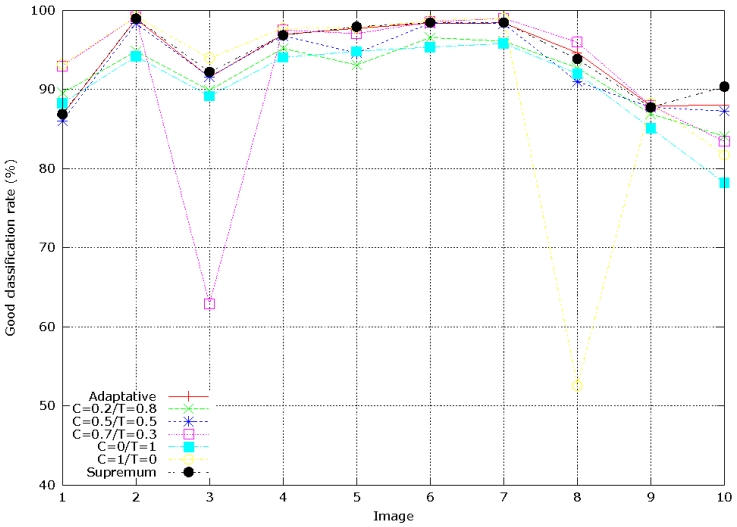


Figure 5. Good classification rate according to the weighting coefficients of color and texture.

Image	block size						
	10	12	16	20	24	26	without block
1	95.42	96.00	96.32	96.63	96.30	96.27	83.00
2	93.24	94.20	95.42	96.16	96.37	96.36	90.90

Table 1. Good classification rate of the tracking step for different bloc size B.

components (obtained with a lexicographic order) with the unique textural gradient component (obtained by using Gabor filters). The gradient mixtures are obtained using an adaptive technique for computing the weighting coefficients. The mixed gradient image is finally processed using the watershed based segmentation method. The proposed method is performed by considering different object classification applications and provides better results than the other fixed combinations. A region-tracking method is also presented in order to decrease the time required for the detection of the percentage of sky on fish-eye images. This method, based on the block-matching algorithm, takes advantage of the space-time continuity of the acquired images. A calibration and rectification is required for this procedure, which is also presented on this paper. The evaluation of the presented method is performed on a number of images acquired on real conditions. The obtained results are very promising, reaching a good classification rate of 96,4% and decreasing the execution time of the application by 10 times.

ACKNOWLEDGMENTS

The research works presented is a part of the ViLoc project, supported by the Regional Council of Franche-Comté (France) and are actually tested in the CAPLOC project developed by INRETS-LEOST and UTBM-SeT and supported by the PREDIT (France).

REFERENCES

[1] Meurie, C., Cohen, A., and Ruichek, Y., "An efficient combination of texture and color information for watershed segmentation.," in [*International Conference on Image and Signal Processing (ICISP), LNCS 6134*], 147–156 (2010).

[2] Meurie, C., Ruichek, Y., Cohen, A., and Marais, J., "An hybrid an adaptive segmentation method using color and textural information.," in [*IS&T/SPIE, Electronic Imaging 2010 - Image Processing: Machine Vision Applications III*], (2010).

[3] Lentmaier, M., Krach, B., and Robertson, P., "Bayesian time delay estimation of gnss signals in dynamic multipath environment," *International Journal of Navigation and Observation* (ID 372651), 11 (2008).

[4] Wang, J.-H. and Gao, Y., "High-sensitivity gps data classification based on signal degradation conditions," *IEEE Trans. On Vehicular Technology* **56**(2), 566–574 (2007).

[5] Viandier, N., Nahimana, F., Marais, J., and Duflos, E., "Gnss performance enhancement in urban environment based on pseudo-range error model," in [*IEEE/ION Position Location And Navigation System (PLANS)*], 6 (2008).

[6] Marais, J., Berbineau, M., and Heddebaut, M., "Land mobile gnss, availability and multipath evaluation tool," *IEEE Trans. On Vehicular Technology* **54**(5), 1697–1704 (2005).

[7] Meguro, J.-I., Murata, T., Takiguchi, J.-I., Amano, Y., and Hashizume, T., "Gps accuracy improvement by satellite selection using omnidirectional infrared camera," in [*EEE/RSJ International Conference on Intelligent Robots and Systems*], 22–26 (2008).

[8] Meguro, J.-I., Murata, T., Takiguchi, J.-I., Amano, Y., and Hashizume, T., "Gps multipath mitigation for urban area using omnidirectional infrared camera," *IEEE Trans. On Intelligent Transportation Systems* **10**(1), 22–30 (2009).

[9] Angulo, J., "Gradients morphologiques de texture. application à la segmentation couleur+texture par lpe," in [*COmpression et REprésentation des Siganux Audiovisuels*], 42–47 (2006).

[10] Cohen, A., Meurie, C., Ruichek, Y., and Marais, J., "Characterization of the reception environment of gnss signals using a texture and color based adaptive segmentation technique.," in [*IEEE Intelligent Vehicles Symposium (IV)*], 275–280 (2010).

[11] Swaminathan, R. and Nayar, S. K., "Nonmetric calibration of wide-angle lenses and polycameras," *On Pattern Analysis and Machine Intelligence (PAMI)* **22**(10), 1172–1178 (2000).

[12] Nelder, J. A. and Mead, R., "A simplex method for function minimization," *Computer Journal* **7**, 308–313 (1965).

[13] Vincent, L. and Soille, P., "Watersheds in digital spaces : an efficient algorithm based on immersions simulations," *IEEE Trans. On Pattern Analysis and Machine Intelligence (PAMI)* **13**(16), 583–598 (1991).

[14] Shafarenko, L., Petrou, M., and Kittler, J., "Automatic watershed segmentation of randomly textured color images," *IEEE Trans. On Image Processing* **6**(11), 1530–1543 (1997).

[15] Beucher, S., "Watershed, hierarchical segmentation and waterfall algorithm," *Mathematical morphology and its applications to image ans signal processing* , 69–76 (1994).

[16] Angulo, J. and Serra, J., "Color segmentation by ordered mergings," in [*International Conference on Image Processing (ICIP)*], **2**, 125–128 (2003).

[17] Lezoray, O., Meurie, C., and Elmoataz, A., "A graph approach to color mathematical morphology," in [*IEEE Symposium on Signal Processing and Information Technology (ISSPIT)*], 856–861 (2005).

[18] Turner, M. R., "Texture discrimination by gabor functions," *Biological Cybernetics* **55**, 71–82 (1986).

[19] Fogel, I. and Sagi, D., "Gabor filters as texture discriminator," *Biological Cybernetics* **61**, 102–112 (1989).

[20] Petkov, N., K. P., "Computational models of visual neurons specialized in the detection of periodic and aperiodic oriented visual stimuli: Bar and grating cells," *Biological Cybernetics* **76**, 83–96 (1997).

[21] Arivazhagan, S., G. L., "Texture classification using wavelet transform," *Pattern Recognition Letters* **24**, 1513–1521 (2003).

[22] Barnett, V., "The ordering of multivariate data," *Journal of the royal society of statistics* **A139**(3), 318–355 (1976).

[23] G. Foret, P. B. and Chassery, J.-M., "Suivi d'objets vido par propagation d'tiquettes et rtro-projection," *Traitement du signal* **22**(1), 4157 (2005).

[24] Lee, C.-H. and Chen, L.-H., "A fast motion estimation algorithm based on the block sum pyramid," *IEEE Trans. On Image Processing* **6**(11), 15871591 (1997).

[25] Li, W. and Salari, E., "Successive elimination algorithm for motion estimation," *IEEE Trans. On Image Processing* **40**(1), 105–107 (1995).

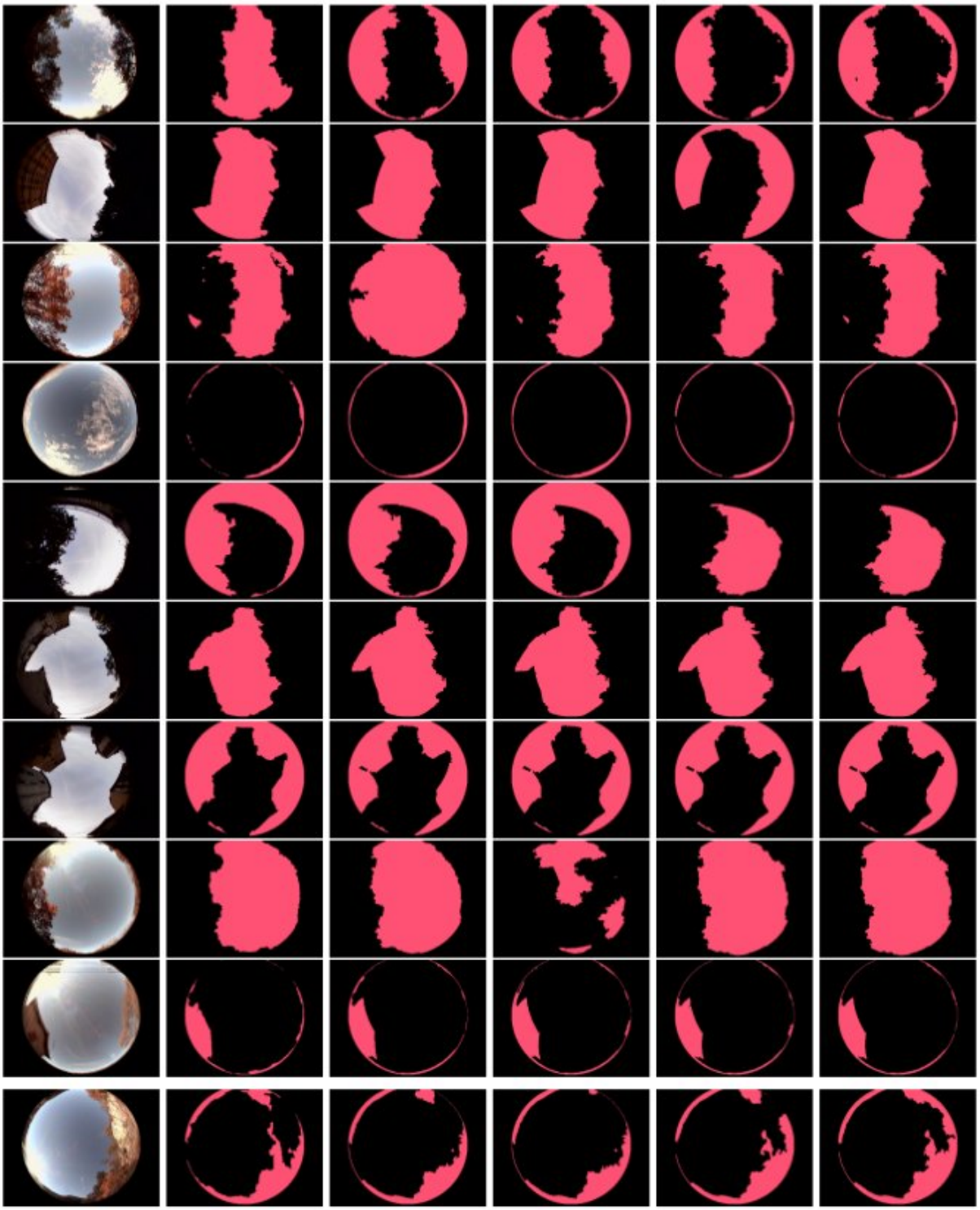


Figure 6. Classification results (left to right: initial images, classified images (columns 2-6: with C=0.2/T=0.8; with C=0.7/T=0.3; with C=1/T=0; with adaptive combination, with supremum combination)).