Chapter 10

Clouds Motion Estimation from Ground-Based Sky Camera and Satellite Images

Ali Youssef Zaher and Afraa Ghanem

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.71263

Abstract

Estimation of cloud motion is a challenging task due to the non-linear phenomena of cloud formation and deformation. Satellite images processing is a popular tool used to study the characteristics of clouds which constitute major factors in forecasting the meteorological parameters. Due to the low resolution of satellite images, researchers have turned towards analyzing the high-resolution images captured by ground-based sky cameras. The first objective of this chapter is to demonstrate the different techniques used to estimate clouds motion and to compare them with respect to the accuracy and the computational time. The second aim is to propose a fast and efficient block matching technique based on combining the two types of images. The first idea of our approach is to analyze the low-resolution satellite images to detect the direction of motion. Then, the direction is used to orient the search process to estimate the optimal motion vectors from the high-resolution ground-based sky images. The second idea of our method is to use the entropy technique to find the optimal block sizes. The third idea is to imply an adaptive cost function to perform the matching process. The comparative study demonstrates the high performance of the proposed method with regards to the robustness, the accuracy and the computation time.

Keywords: block matching, exhaust search, three step search, four step search, optical flow, Lucas-Kanade, Horn-Schunck

1. Introduction

Clouds constitute major factors in determining the radiation budget of the Earth, and consequently, play crucial roles in modulating the climate for both local and global scales. Satellite images processing is an essential tool for both meteorologists and scientists to collect large-scale cloud information. Most of meteorological satellites are placed on geostationary orbit providing successive images in different spectral channels such as infrared, water vapor and visible channels [1]. Chiefly, the sequence resulting by collecting successive images have been useful to study the characteristics and the spatial distribution of clouds. Many satellite image based studies have focused on detecting the movement of clouds to forecast the solar irradiance which is essential for the effective operation of various solar applications such as solar thermal systems, photovoltaic systems, and grid regulation [2]. However, the low resolution of satellite images with respect to space and time is not adequate to satisfy the requirements of many real-time applications. Thus, researchers have used the images captured by ground-based cameras in order to make up the deficiency of satellite cloud observations [3].

In the context of motion estimation, block matching and optical flow are the most popular of all motion estimation techniques due to their effectiveness and simplicity. Optical flow methods are based on brightness constancy of matching pixels in consecutive images. Ideally, optical flow would be the same as the motion field, but the apparent motion can be caused by lighting changes without any motion. Thus, smoothness constraints are used by researchers to overcome the limitations of brightness constancy assumption [4].

Concerning block matching based approaches, the consecutive images are partitioned into macro blocks of pixels. And then, for each block at the current image, the algorithm seeks for the best matching block within a search area from the previous image according to a matching criterion. The exhaustive search ES provides the best performance by matching all possible blocks within the search area, but the search process takes a significant computation time [5]. Thus, fast algorithms have been proposed such as three step search 3SS, four step search 4SS, and diamond search DS algorithms [6].

In block based methods, block size has a large impact on the efficiency. All the approaches with fixed sized blocks have low performance because larger blocks may have some objects that move in different directions and at different speeds, on the other hand, smaller blocks are difficult to be recognized correctly [7].

Our first aim in this chapter is to provide an overview of the principal methods used for clouds motion estimation. The second purpose is to propose a fast and efficient block matching algorithm. The proposed algorithm has improved the estimation accuracy by using: two types of images, variable block sizes instead of fixed block sizes, and an adaptive cost function to perform the matching process.

Thus, this chapter is organized as follows: Section 2 describes the data analyzed in this study. In Sections 3 and 4, the popular block matching and optical flow approaches are presented. Section 5 is dedicated to the proposed algorithm. Section 6 shows the experimental results and a comparative study between the algorithms under different types of movements. Finally, Section 7 presents our conclusions.

2. Data collection

The development of our methodology for cloud motion detection has been performed on analyzing the images captured by METEOSAT second generation satellites MSG, and those taken by ground-based camera systems (**Figure 1**).

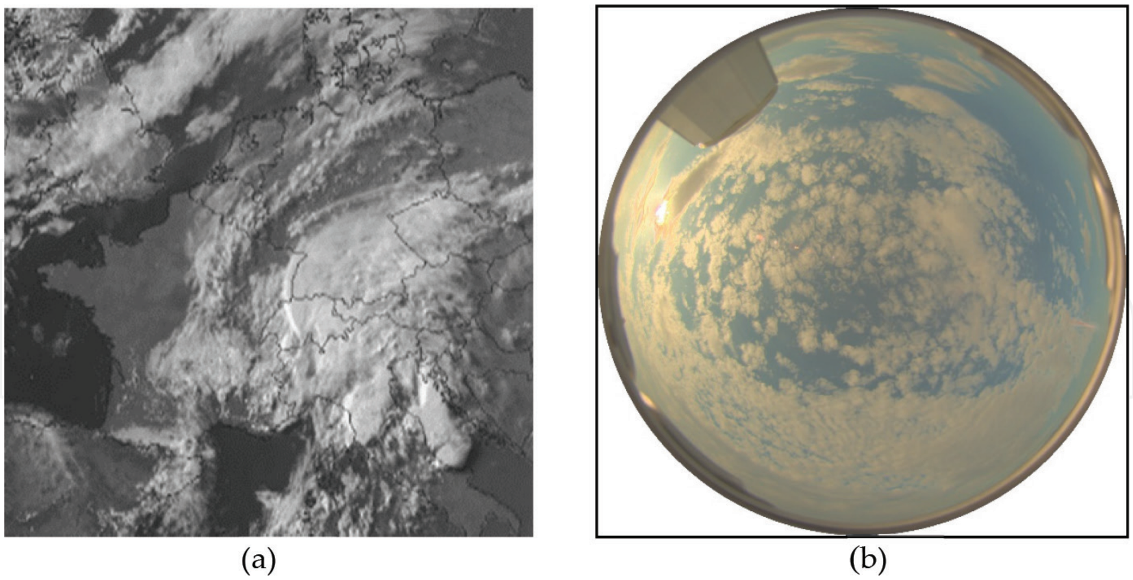

Figure 1. Sample of analyzed images: (a) gray scale satellite image, and (b) true color all-sky image.

The MSG satellite series are operated by the European organization for the exploration of meteorological satellites (EUMETSAT). They provide image data in visible and infrared channels that cover Europe, Africa and the Atlantic Ocean with a baseline repeat cycle of 15 minutes. We are interested in the visible images of the channel 0.6 μm which have a normal resolution of 3 × 3 km^2.

In fact, the METEOSAT sensor for the visible channel captures sunlight of the visible band reflected by an object back to the satellite. The fraction of solar radiation that is reflected back to space is called the albedo. The cloudy surfaces usually have a higher albedo than the other surfaces of the Earth. So, clouds usually appear white, while land and water surfaces appear in darker gray scale levels.

The ground-based images are taken by diverse whole sky camera systems (WSC) distributed on different geographic locations in France, Italy and Switzerland. The data are acquired in several seasons; therefore, they cover a wide range of possible sky conditions.

The whole sky camera consists basically of a digital color video camera mounted with a fish-eye lens that forms circular fish-eye images with 180-degree field of view across the image circle. The time interval between two successive images varies from 15 s to 5 minutes according to the application requirements. Images have a resolution of 768 × 576 pixels, and are stored as JPEG images. In these images, when clouds are thin they appear as white because they scatter and reflect all the visible colors of light which are combined to give white light. If clouds are thick, they will absorb a significant amount of the sunlight, and consequently, they will appear grayish and darker.

3. Block-based motion estimation methods

Block matching algorithms for motion estimation play a key role in most of video and image processing applications such as objects tracking, video compression standards, medical image analysis, video telephony and real-time video conferencing, etc.

The underlying assumption behind block matching algorithms is that the visual contents of an object in a frame of video sequence are closely related to those of the same object on the subsequent frame. The main idea of these algorithms is to divide the two successive images into a matrix of macro blocks with size of $L \times M$, as presented in **Figure 2**. Then, each block of the current image is considered as a reference block. Afterwards, the reference block is compared to all blocks within a search window with size of $(2W + L) \times (2W + M)$ of the previous image to determine the block with a higher match to the reference block.

In the literature, various cost functions are used as similarity measures to match the reference macro block to the candidates blocks such as: mean absolute differences *MAD* [8], or sum of absolute differences *SAD* [9, 10]. These criteria are expressed by the following formulas:

$$MAD = \frac{1}{L \times M} \sum_{i=1}^{L} \sum_{j=1}^{M} \left| I_n(x_r + i, y_r + j) - I_{n-1}(x + i, y + j) \right| \tag{1}$$

$$SAD = \sum_{i=1}^{L} \sum_{j=1}^{M} \left| I_n(x_r + i, y_r + j) - I_{n-1}(x + i, y + j) \right| \tag{2}$$

Where:

- $I_n\ (x_r, y_r)$ is the reference block of the current frame (n) and (x_r, y_r) is the coordinate of its upper left corner.
- $I_{n-1}(x, y)$ is the candidate block of the previous frame (n−1) and (x, y) is the coordinate of its upper left corner.

Once the best matching block is located, the motion vectors *MVs* can be obtained by calculating the displacement between the block in the present image and the best matched block in the previous image in horizontal and vertical directions.

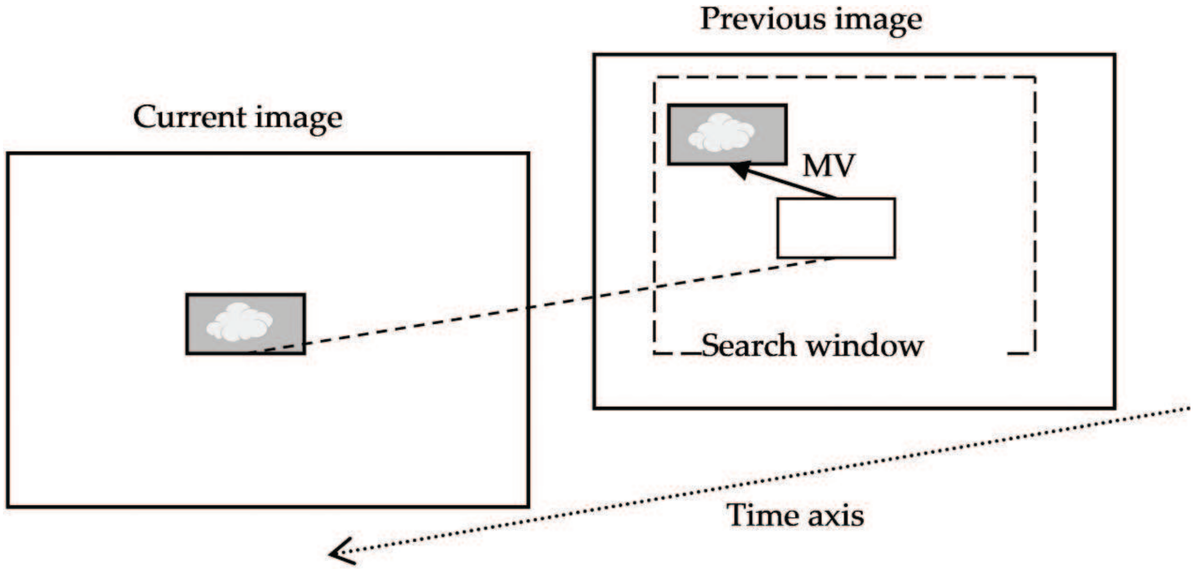

Figure 2. Block matching process.

The search-window size and the block size have a significant impact on the performance of block matching algorithms with regard to the accuracy and the computation time.

The search area size depends strongly on the object's velocity, if object's speed is high and small search windows are used, the block that matches best with the reference block will be outside the search region and the estimate will be unreliable. If the objects move at low speeds and the search window is too large, a large number of matching blocks can be found in the search zone and the estimate will be wrong. Further, the processing time is greatly increased.

Concerning the block size, the selection of this parameter is essential for any block-based motion estimation algorithm. If the blocks are too small, the number of matching blocks increases as well as the estimation error and the computation time. On other hand, if the blocks are very big, there is a high probability that the block contains different objects moving at different speeds, and consequently, the estimation error is increased as well.

In order to reduce the impacts of the above-mentioned parameters, dynamic search area and block sizes are preferred to find a reasonable balance between the processing time and the desired accuracy according the application.

3.1. Three step search approach

Three step search 3SS is one of the earliest methods used to optimize the search process with a performance close to that can be obtained from exhaustive search algorithm [11]. This method is based on three steps fine-coarse search mechanism. The basic idea is to conduct the research process in raster order. In other words, the block with minimum distortion in the previous step is used to determine the positions of the blocks that must be examined in the current step. **Figure 3** shows an example to illustrate the three-step search process for $W = 7$.

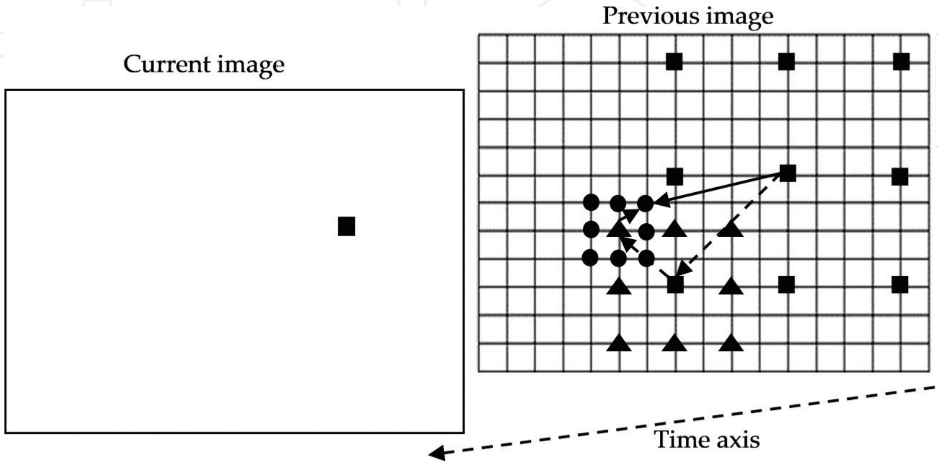

Figure 3. Three step search process with $W = 7$.

In the first step, eight blocks at a distance $d_1 = W/2$ from the center block (x_r, y_r) are determined in the previous image. Then the eight blocks in addition to the center block are compared to the reference block of the current image. The block with minimum distortion is considered as a center block in the next step. In the second step, a new eight blocks are determined at a distance $d_2 = d_1/2$ from the new center block. Then the matching operation is done to find the new center block as in the first step. In the third step, the distance from the center block becomes $d_3 = 1$ pixel and the best matched block resulted from this step is used to calculate the motion vectors.

As can be seen, the number of matches are significantly reduced to 27 operations comparing to 225 operations used in the exhaust search algorithm.

3.2. Four step search approach

The principle of four-step search algorithm is based on center biased searching process as well as in three-step search [12]. This method is composed of four steps. In the beginning, the size of the search window is fixed at W pixels. Afterwards, eight blocks around the center block are determined at a distance $d_1 = W/4$ pixels from the center block, and then, they are checked as in the three-step search process. The block with a higher match is kept to be the center in the next step unless it is the center of the search window then the search process jumps immediately to the fourth step. The following two steps are exactly the same as the first step. In the fourth step, the distance from the center block is a half of that used in the beginning $d_4 = d_1/2$, and the location of the best matched block is used to estimate the motion vectors of the reference block.

Contrary to three-step search algorithm, the number of matching operations to reach the best matched block is variable due to the condition imposed in the first three steps. With regard to the example presented in **Figure 4**, the number of checked blocks equals 36 blocks. This indicates that, the computation time of four-step search process is greater than that of three-step search, but it is still less than that of full search process.

3.3. Diamond search approach

As its name indicates, the diamond search method DS implies a diamond shaped search windows instead of square search windows which are used in the above-mentioned approaches [13]. In addition, the number of steps needed for the convergence is not limited.

Firstly, a large diamond-shaped window is used and eight blocks around the center block are examined. If the center of the search area has the best match then the procedure jumps to the last step, else the best matched block will be the center of the search window in the next step. The procedure is repeated until the match occurs with the center of the search window then, a smaller diamond-shaped search window is used and the block with the minimum distortion is used to determine the motion vectors.

The principle of diamond search process is presented in **Figure 5**. The number of examined blocks needed to reach the best match is equal to 9 + 9 + 9 + 9 blocks and this is equivalent to the number of operations used in four step search process.

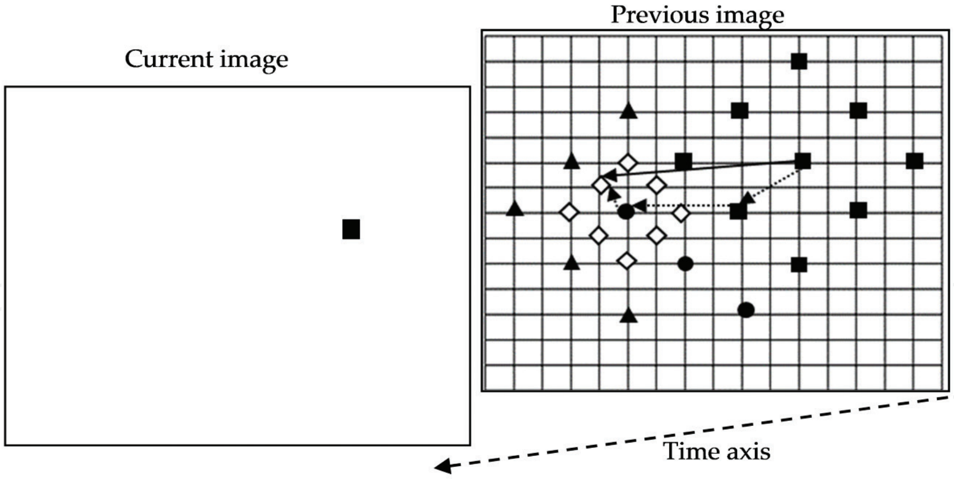

Figure 4. Four step search process.

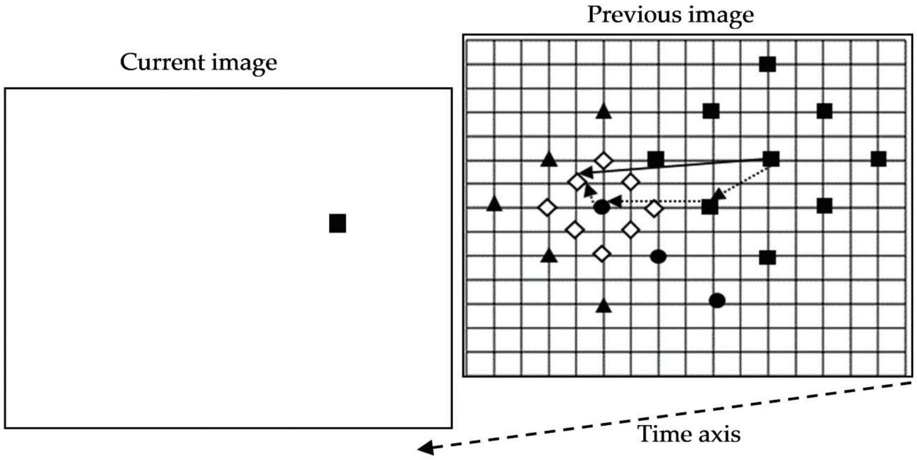

Figure 5. Diamond search process.

4. Optical flow methods

Block matching algorithms divide the consecutive images into blocks of pixels which move at the same speed and in the same direction. Whereas, optical flow algorithms are pixel-based approaches that determine motion vectors for every pixel in the image.

The last methods work on brightness constancy assumption, that is, two corresponding pixels in consecutive frames should have the same brightness. This assumption leads us to the optical flow equation [14, 15]:

$$\left(\frac{\partial I}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial I}{\partial y}\frac{\partial y}{\partial t}\right) + \frac{\partial I}{\partial t} = \left(\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v\right) + \frac{\partial I}{\partial t} = I_x u + I_y v + I_t = 0 \tag{3}$$

Where:

- I_x, I_y and I_t are the partial derivatives of image brightness with respect to x, y and t respectively.
- u is the horizontal velocity and v the vertical velocity.

Since the last equation involves two unknown variables u and v, there are infinite solutions that satisfy its constraints. In other words, many pixels in the previous frame could match with a given pixel in the current frame. Thus, the problem of motion estimation is ill-posed. It is for this reason that various methods are proposed to reduce the number of solutions by introducing additional constraints on the smoothness of the motion vector field. In this context, the following two methods are among the most common methods that use regularization constraints.

4.1. Horn and Schunk approach

Horn and Schunck added an additional regularization condition that reduces the solutions space of the optical flow equation. They transformed the last equation into an optimization problem that minimizes both the optical flow constraint and the magnitude of the variations of the flow field. In this context, the flow optic is formulated as a global energy function E which is then sought to be minimized as follows [16]:

$$E = \iint \left(I_x u + I_y v + I_t\right)^2 dxdy + \alpha \iint \left(\left(\frac{\partial u}{x}\right)^2 + \left(\frac{\partial u}{y}\right)^2 + \left(\frac{\partial v}{x}\right)^2 + \left(\frac{\partial v}{y}\right)^2 \right) dxdy \tag{4}$$

where: α is a regularization parameter which influences the smoothness of the motion vectors. It is usually adjusted empirically depending on the desired smoothness. Larger values of α give a smoother motion filed.

The minimization problem can be solved by using an iterative process through which we can estimate the new set of the motion vectors (u^{n+1}, v^{n+1}) from the estimated derivatives and the average of the previous velocity estimates (u^n, v^n) as follows:

$$u^{n+1} = u^{-n} - \frac{I_x\left(I_x u^{-n} + I_y v^{-n} + I_t\right)}{\alpha + I_x^2 + I_y^2} \tag{5}$$

$$v^{n+1} = v^{-n} - \frac{I_y\left(I_x u^{-n} + I_y v^{-n} + I_t\right)}{\alpha + I_x^2 + I_y^2} \tag{6}$$

Considering two consecutive images and a pixel of coordinates (i, j) the spatial and temporal derivatives I_x, I_y, I_t can be calculated using the following approximations:

$$I_x \approx \frac{1}{4}\left[I_{i,j+1,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i+1,j,k+1}\right] \tag{7}$$

$$I_y \approx \frac{1}{4}\left[I_{i+1,j,k} - I_{i,j,k} + I_{i+1,j+1,k} - I_{i,j+1,k} + I_{i+1,j,k+1} - I_{i,j,k+1} + I_{i+1,j+1,k+1} - I_{i,j+1,k+1}\right] \tag{8}$$

$$I_t \approx \frac{1}{4}\left[I_{i,j,k+1} - I_{i,j,k} + I_{i+1,j,k+1} - I_{i+1,j,k} + I_{i,j+1,k+1} - I_{i,j+1,k} + I_{i+1,j+1,k+1} - I_{i+1,j+1,k}\right] \tag{9}$$

4.2. Lucas and Kanade approach

Horn-Schunck approach gives an appropriate solution for the optical flow equation, but its iterative process is computationally expensive. Lucas and Kanade have overcome this problem by implying the least square method that works over an area of neighboring pixels to find the best matching pixels. This method assumes that the displacement of visual contents between two successive frames is small and approximately constant within a neighborhood of the pixel under consideration. Thus, the current image is divided into smaller zones of pixels in each of them the motion vectors are considered constant. Then, the basic optical flow equations for all pixels in a given zone Ω can be expressed mathematically as follows [17]:

$$\begin{aligned} I_x(x_1, y_1)u + I_y(x_1, y_1)v &= -I_t(x_1, y_1) \\ I_x(x_2, y_2)u + I_y(x_2, y_2)v &= -I_t(x_2, y_2) \\ &\vdots \\ I_x(x_n, y_n)u + I_y(x_n, y_n)v &= -I_t(x_n, y_n) \end{aligned} \tag{10}$$

Where x_n and y_n are pixel coordinates inside the zone Ω.

These equations can be written in the matricial form as:

$$\underbrace{\begin{bmatrix} I_x(x_1, y_1) & I_y(x_1, y_1) \\ I_x(x_2, y_2) & I_y(x_2, y_2) \\ \vdots & \vdots \\ I_x(x_n, y_n) & I_y(x_n, y_n) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u \\ v \end{bmatrix}}_{V} = -\underbrace{\begin{bmatrix} I_t(x_1, y_1) \\ I_t(x_2, y_2) \\ \vdots \\ I_t(x_n, y_n) \end{bmatrix}}_{B} \tag{11}$$

Consequently, the motion vectors can be expressed using the following equation:

$$V = (A^T A)^{-1} A^T B \tag{12}$$

The last equation gives the same weight to all pixels in the zone Ω. In fact, it is better to give more weight to the pixels close to the center of the zone Ω.

5. Proposed algorithm

The developed approach is a new block matching algorithm with oriented search process. This approach is based on two principal parts: the aim of first part is to estimate the direction of cloud motion using the low-resolution satellite imagery. In the second part, we introduce the obtained direction into the search process of a variable size block matching algorithm that must be applied to the high-resolution ground-based images.

In fact, detecting the direction of motion using satellite images is low expensive in terms of computation time. In other hand, the number of computations is also reduced by adopting an oriented search process in lieu of full search process in the second part. In addition, the high-resolution images and the use of variable block size play an important role to increase the accuracy.

Our method can be summarized by the following steps:

Step 1. Detection of motion direction:

To detect the direction of movement of cloudy objects, we determine the zone of interest on the satellite images. Then, the exhaust search block matching algorithm is applied to the two consecutive satellite images to obtain the motion vectors *u* and *v*. Finally, the direction of movement θ is computed by the following formula:

$$\theta = \tan^{-1}\left(\frac{u}{v}\right) \tag{13}$$

Step 2. Determining the block size

As mentioned in Section 3, the block must be large enough to contain pixels with varied details for reducing the number of matches in the search window. In other words, the block must be textured.

We have used the entropy of a given block to characterize the texture of this block. If the entropy is large enough, the block is considered as textured block.

To apply the entropy method, firstly, the image is divided into small blocks and the entropy *En* is calculated for each block as follows [18, 19]:

$$En = -\sum_{i=1}^{N} P_i \, log_2(P_i) \tag{14}$$

Where: *N* is the number of possible gray levels, and P_i is a scalar contains the histogram count for the gray level *i*.

If the block has an entropy larger than a threshold then it is considered as textured block otherwise the block is considered as a smooth block.

After classifying the blocks, the number of textured blocks in the whole image is compared to a threshold value if this number is bigger than the threshold then the block size can be accepted in the block matching algorithm, else, the block size must be increased and all the operations will be repeated until obtaining the optimal block size.

Step 3. Application of the oriented search process

In this step, the two consecutive images I_n and I_{n-1} are divided into macro blocks with the optimal size obtained in the previous step. Then, each reference block (x_r, y_r) in the image I_n is compared to the blocks (x, y) of the image I_{n-1} in the motion direction θ. The coordinates (x, y) of each examined block are given by:

$$x = x_r + w \times \sin\theta \tag{15}$$

$$y = y_r + w \times \sin\theta \tag{16}$$

Where w takes the values from 0 to W which is the size of the search window.

The criterion used to match the candidates to the reference block in the search window is a cost function of the sum of absolute differences SAD and the Euclidean distance D between the reference block and the block candidate. This function is given by:

$$Cost \;=\; (1-\alpha)SAD + \alpha D \tag{17}$$

The parameter α is a weighting parameter and takes values between 0 and 1. For the low displacements α is high and vice versa. The best match occurs when the cost function $Cost$ is minimum. The use of the distance in the cost function limits the number of matches in the search window and this minimizes the error in estimating the motion vectors.

6. Results and discussion

6.1. Data set description

The database used for implementation and validation of our method is made of two types of images:

- High resolution RGB images (HR) collected every minute of daylight by whole sky imager. Each of them consists of 768 × 576 pixels with 8 bits per channel.
- Low resolution METEOSAT images in the visible channel acquired at 15 minutes intervals. These images are 8-bit color images with a resolution of 745 × 1054 pixels. For each of them, only 600 × 700 pixels representing Europe are considered.

To cover the representative conditions of different types of sky conditions, we have screened the complete data acquired during the period from September 2016 to May 2017 and we have selected approximately 1500 whole sky images and 1500 satellite images taken at the same time. The selection procedure depends upon clouds fraction thresholds. In this procedure, images with cloud fraction lower than 10% are considered as clear sky images and they are discarded from the data base because they do not provide meaningful motion fields. Furthermore, the images representing overcast conditions (cloud fraction ≥ 95%) usually have texture-less cloudy objects and the error in the estimation is very big for all algorithms; therefore, this type of images is not included in our database. 300 images (20% of the data base) are used as training set to optimize the parameters of algorithms such as block size for block matching algorithms and the smoothness area and the number of iterations for optical flow based algorithms. The other images of the database (1300 images) are used in the test stage to show the performances of the implemented algorithms.

6.2. Experimental methodology

In this work, the experimental methodology includes two stages: the training stage and the test stage. In the training stage, 300 images (20% of the data base) are used as training set to optimize the parameters of algorithms such as block size for block matching algorithms and the smoothness area for optical flow based algorithms. The other images of the database (1300 images) are used in the test stage to show the performances of the implemented algorithms.

In order to select the optimal parameters for each algorithm, we have estimated the motion vectors using the images in the training data set for different values of the parameters and then, we have kept the parameters that provide the best accuracy.

The accuracy is characterized by the *PSNR* value (Peak-Signal-to-Noise-Ratio). Such parameter quantifies the differences between an original image and its distorted version that is reconstructed using the estimated motion vectors. The bigger values of *PSNR* indicate a better accuracy of the algorithm. The *PSNR* is defined as [20]:

$$PSNR = 10\,log_{10}\left(\frac{{I_{max}}^{2}}{MSE}\right) \tag{18}$$

Where, I_{max} is the maximum brightness and *MSE* is the mean square error between the original image and the reconstructed one.

The parameters of the block matching algorithm are selected experimentally by applying different values for each parameter and keeping those giving the better *PSNR*. Considering the block size, the best *PSNR* is obtained for a block size of 8 × 8 pixels as presented in **Figure 6**. Concerning the search area size, it depends on the maximum clouds displacement speed. For our experiments, the maximum displacement speed is 16 pixels, thus, the search range *W* is limited to 16 pixels.

For optical flow algorithms, the smoothness window Ω is fixed at 8 × 8 pixels as the block size in block matching algorithms.

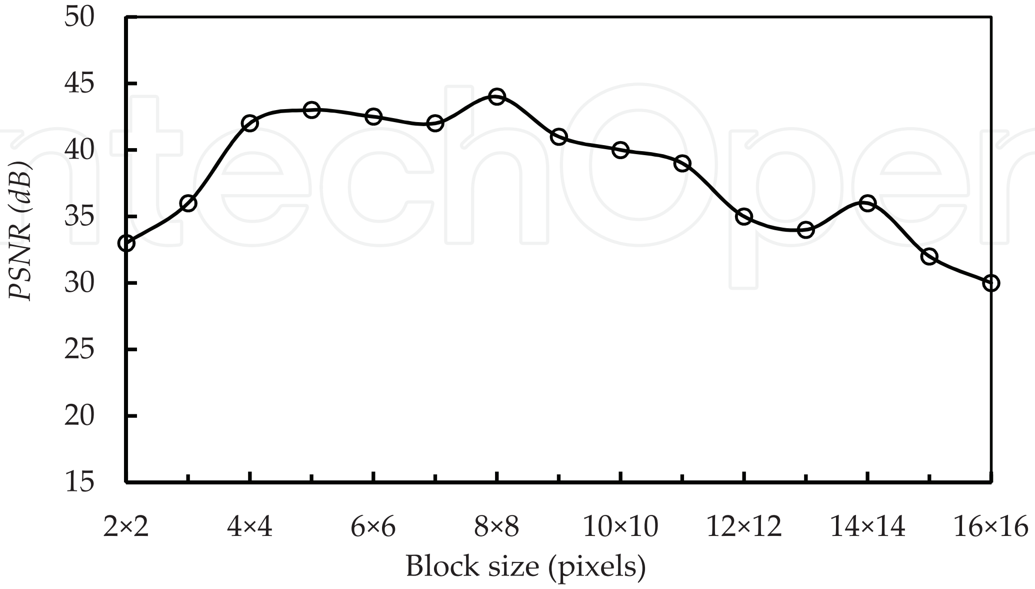

Figure 6. PSNR vs. block sizes.

6.3. Experimental results

All the above conventional algorithms have been implemented to estimate the displacement of clouds using sequences of high-resolution images captured by ground-based sky cameras. For our method, we have used the low-resolution METEOSAT satellite imagery and the high-resolution ground based images. Such sequences consist of different degrees and types of motion. Each image is divided into macro-blocks with the sizes of 8 × 8 pixels for all algorithms with fixed block size. But for our algorithm, the block size is variable and it takes values from varies from 4 × 4 to 12 × 12 pixels. The maximum displacement between two consecutive images is 16 × 16 pixels, thus, the optimal search window size *W* is ±16 pixels in both horizontal and vertical directions.

For evaluating the implemented algorithms, they are compared with regards to their accuracy and the computation time.

The accuracy is evaluated using the *PSNR* value. In addition to the accuracy, the standard deviation of the *PSNR* is calculated to qualify the robustness of the algorithms. A low standard deviation indicates that the *PSNR* values are close to the mean of the set and consequently the robustness is very high, while a high standard deviation indicates that the *PSNR* is spread out over a wide range of values and this makes the algorithm non-robust.

The *PSNR* comparison of the compensated images generated using the implemented algorithms is presented in **Figures 7** and **8**. **Table 1** indicates data for the standard deviation of the *PSNR* and the average computation time.

From these results, it is observed that the proposed method provided the maximum values of the *PSNR* for all types of motion and the minimum value of standard deviation, which means that our method has the best performance in terms of accuracy and robustness. On the other

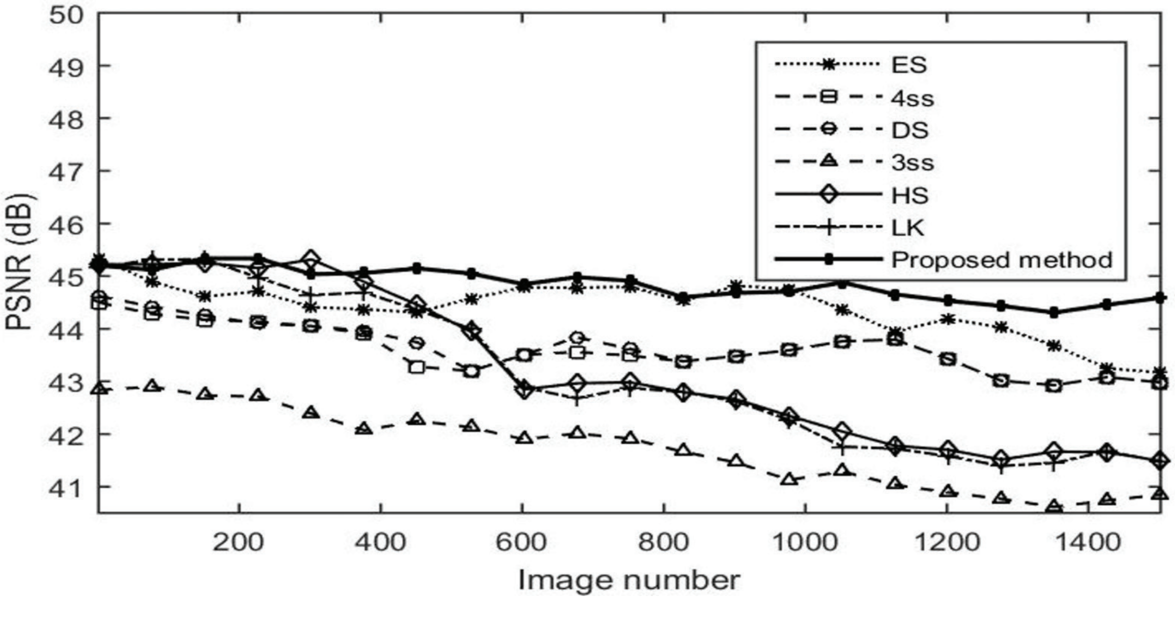

Figure 7. *PSNR* performance evaluation.

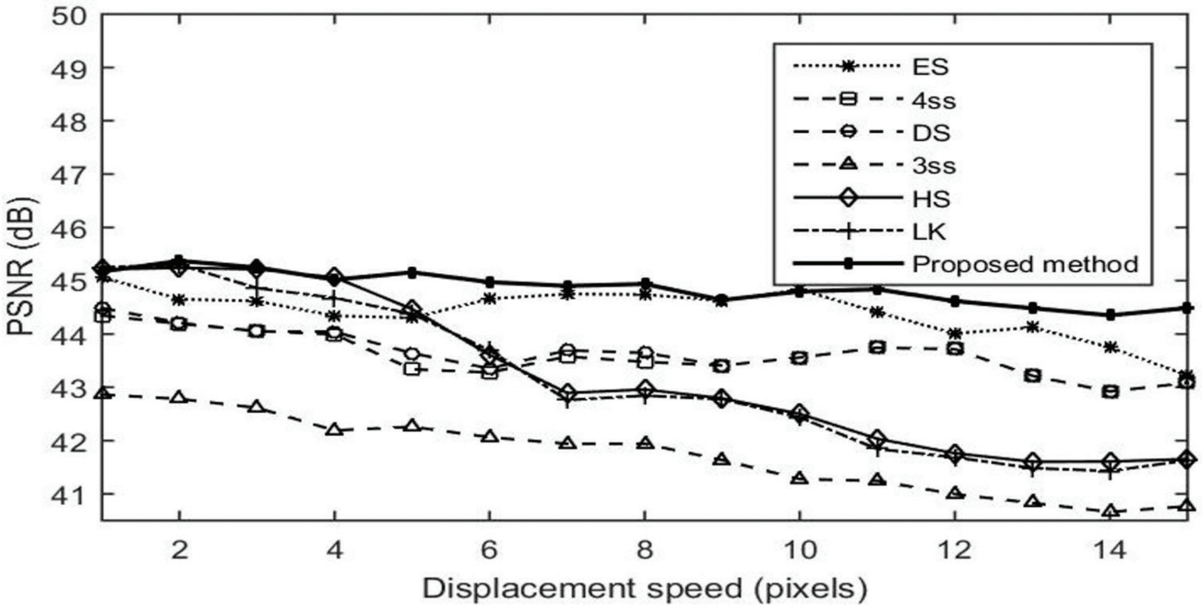

Figure 8. *PSNR* versus displacement speed.

Method	Average *PSNR* (dB)	Standard deviation	Computation time (s)
ES	44.3	0.49	13.2
3SS	41.7	0.73	1.8
4SS	43.6	0.41	2.5
DS	43.7	0.43	3.2
HS	43.22	1.41	10.4
LK	43.12	1.4	7.4
Proposed method	44.9	0.3	[4–6.8]

Table 1. Standard deviation and average values of computation time for all implemented algorithms.

hand, the computation time of our algorithm is highly decreased compared to the computation time of exhaust search algorithm and this is due to the use of an oriented search approach.

For the other algorithms, the exhaust search block matching method presents the better accuracy but it is the most expensive with regards to the computational time, and that is because of the great number of matching operations which equals to $(2W + 1) \times (2W + 1)$.

Furthermore, optical flow algorithms have a good accuracy for the low motions but this accuracy decreases significantly at high speeds. This is because moving objects do not preserve their intensity values from image to image in case of high motions.

The three-step search algorithm has the best computation time, but it also has the worst *PSNR* performance.

7. Conclusion

This chapter provides an overview of the popular techniques used for clouds motion estimation. In addition, it presents a fast and efficient block matching algorithm based on combining low-resolution satellite images and high-resolution ground based sky images.

In our algorithm, the use of an adaptive cost function and dynamic block sizes allowed us to obtain an accuracy better than that obtained by the conventional methods that treat one type of images and fixed block sizes. Furthermore, the oriented search process reduced the computation time of the proposed algorithm considerably comparing to the exhaust search and optical flow algorithms.

Author details

Ali Youssef Zaher[1]* and Afraa Ghanem[2]

*Address all correspondence to: ali.zaher@promes.cnrs.fr

1 University of Perpignan, PROMES-CNRS Laboratory, Perpignan, France

2 University of Perpignan, Perpignan, France

References

[1] Linares-Rodriguez A, Ruiz-Arias JA, Pozo-Vazquez D, Tovar-Pescador J. An artificial neural network ensemble model for estimating global solar radiation from Meteosat satellite images. Energy. 2013;**61**:636-645

[2] Alonso J, Ternero A, Batlles FJ, López G, Rodríguez J, Burgaleta JI. Prediction of cloudiness in short time periods using techniques of remote sensing and image processing. Energy Procedia. 2014;**49**:2280-2289

[3] Marquez R, Coimbra CFM. Intra-hour DNI forecasting based on cloud tracking image analysis. Solar Energy. 2013;**91**:327-336

[4] Chen W, Mied RP. Optical flow estimation for motion-compensated compression. Image and Vision Computing. 2013;**31**(31):275-289

[5] Liaw YC, Lai JZ, Hong ZC. Fast block matching using prediction and rejection criteria. Signal Processing. 2009;**89**(6):1115-1120

[6] Cuevas E, Zaldívar D, Pérez-Cisneros M, Sossa H, Osuna V. Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC). Applied Soft Computing. 2013;**13**(6):3047-3059

[7] Khan NA, Masud S, Ahmad A. A variable block size motion estimation algorithm for real-time H. 264 video encoding. Signal Processing: Image Communication. 2006; **21**(4):306-315

[8] Pandian SIA, Bala GJ, Anitha J. A pattern based PSO approach for block matching in motion estimation. Engineering Applications of Artificial Intelligence. 2013;**26**(8):1811-1817

[9] Saha A, Mukherjee J, Sural S. A neighborhood elimination approach for block matching in motion estimation. Signal Processing: Image Communication. 2011;**26**(8):438-454

[10] Basha SM, Kannan M. Design and implementation of low-power motion estimation based on modified full-search block motion estimation. Journal of Computational Science. 2017;**21**:327-332

[11] Kulkarni SM, Bormane DS, Nalbalwar SL. Coding of video sequences using three step search algorithm. Procedia Computer Science. 2015;**49**:42-49

[12] Po LM, Ma WC. A novel four-step search algorithm for fast block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology. 1996;**6**(3):313-317

[13] Zhu S, Ma K. A new diamond search algorithm for fast block-matching motion estimation. IEEE Transactions on Image Processing. 2000;**9**(2):287-290

[14] Wei SG, Yang L, Chen Z, Liu ZF. Motion detection based on optical flow and self-adaptive threshold segmentation. Procedia Engineering. 2011;**15**:3471-3476

[15] Sengar SS, Mukhopadhyay S. Motion detection using block based bi-directional optical flow method. Journal of Visual Communication and Image Representation. 2017;**49**: 89-103

[16] Horn BKP, Schunck BG. Determining optical flow. Artificial Intelligence. 1981;**17**(1-3): 185-203

[17] Lucas B, Kanade T. An iterative image registration technique with an application to stereo vision. In: Seventh International Joint Conference on Artificial Intelligence; 1981; Vancouver, Canada; 1981. pp. 674-679

[18] Kim SE, Jeon JJ, Eom IK. Image contrast enhancement using entropy scaling in wavelet domain. Signal Processing. 2016;**127**:1-11

[19] Wu H, Miao Z, Wang Y, Chen J, Ma C, Zhou T. Image completion with multi-image based on entropy reduction. Neurocomputing. 2015;**159**:157-171

[20] Jalloul MK, Al-Alaoui MA. A novel cooperative motion estimation algorithm based on particle swarm optimization and its multicore implementation. Signal Processing: Image Communication. 2015;**39**:121-140