

MASTER THESIS, 2016, MATHEMATICS SECTION

Sky Images Segmentation and Motion Prediction via Geodesic Active Contours

Author:

Rim Miryam CHAABOUNI

A thesis conducted within

ABB Corporate Research, Baden-Dättwil

Supervisor: Dr. Jan POLAND

in collaboration with

Ecole Polytechnique Fédérale de Lausanne

Supervisor: Pr. Jean-Philippe THIRAN

January 29, 2016



ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Abstract

School of Basic Sciences
Computational Sciences and Engineering

Master of Science

Sky Images Segmentation and Motion Prediction via Geodesic Active Contours

by Rim Miryam CHAABOUNI

Photo-voltaic (PV) cells are subject to regular fluctuations of solar power input caused by the passage of clouds. In order to manage correctly the electrical grid it is important to be able to predict the sky coverage and thus avoid ramp events. The goal of this thesis is to track the cloud motion and provide short term predictions of the cloud cover estimation from a sequence of pre-segmented sky images. Geodesic active contours are a suitable way of detecting and tracking non-rigid objects in a sequence of images. They rely on an implicit representation of the object borders as the zero level set of a moving surface. This surface is often a signed distance function which can be computed by minimizing an energy function. We developed an energy function which can segment the full sequence of images simultaneously while keeping a trace of the inter-frames motion. This information is reused to produce a short term prediction in continuation of the motion during the sequence. We formulate our optimization as quadratic programming problem and solve it using the open source software IPOPT. We tested our model on synthetic and real sky images, and the results are very satisfying when the clouds' motion is dominated by deformations and size variations. However, we observe some problems in case of predominantly lateral movement which is why we combined it with a pre-existing optical flow based model. In this report we provide the details of our model, examples of predictions and suggestions for improvement.

Acknowledgements

This work would not have been feasible without the clever ideas and the patient explanations of my supervisor Dr. Poland. I am also very grateful to Pr. Thiran for his insight into Active Contours and his help to maintain my focus. Finally I would like to thank ABB for providing me with the infrastructure, data, and human support which allowed me to conduct my thesis in the best conditions.

Contents

Abstract	iii
Acknowledgements	v
1 Introduction	1
2 Related work	3
2.1 Introduction to active contours	3
2.2 Fast solvers to the Hamilton-Jacobi equation	4
2.3 The variational level set method	6
2.4 Solving the optimization problem	7
2.5 Motion tracking with active contours	9
2.6 Summary	9
3 Cloud tracking with Active Contours	11
3.1 Input images	11
3.2 The representation	14
3.3 The cost function	15
3.3.1 Conformation	16
3.3.2 Motion fit	17
3.3.3 Regularization	18
3.4 The solver	19
3.5 Including horizontal movement	20
3.6 Summary	22
4 Results	23
4.1 Experiment on a sequence of synthetic images	23
4.1.1 Vertically moving surface	23
4.1.2 Laterally moving surface	25
4.2 Tracking and prediction on real sky images	28
4.2.1 Sequences with slowly deforming clouds	28
4.2.2 Sequence with lateral shift and topological changes	33
5 Discussion and future work	35
6 Conclusion	37
Bibliography	39
Declaration of Authorship	43

List of Figures

2.1	Point snake	4
2.2	Geodesic snake. The blue contour results from the intersection between the surface ϕ and the plane $z = 0$	5
2.3	Narrow band	5
2.4	Example of disconnected regions separated by an interface	6
3.1	Data collected on July 20 th 2015 from ABB's site in Cavriglia, Italy. Left : example of sky image taken at 15:24:01. Top right : Power input to the PV plant. Bottom right : irradiance measured (blue) and clear sky model (red).	12
3.2	Process to obtain a signed distance function from a regular color image. The last step (Active Contours) is the main contribution of this thesis.	13
3.3	Pre-segmentation of a sky image taken on July 20 th 2015 at 15:24:01	14
3.4	Grayscale image and surface corresponding to a sky image.	15
3.5	Signed distance function from the cloud boundary. The shrinking cloud is easily modelled by shifting down the function.	16
3.6	Difference between a linear cost function versus a saturated one for $\rho = 1$	17
3.7	Scheme of the cost function for $\rho = 1$. The slopes ω_{lo} (blue curve) and ω_{hi} (red curve) are exaggerated for clarity	18
3.8	Example of linear regression. The data corresponds to the image intensity at a single pixel through time	18
3.9	Pixel intensity variation (red arrows) for purely vertical and purely horizontal movement. This variation is used to predict the next state of the curve.	21
3.10	In strategy 2, all the contours of the sequence are shifted horizontally to match the last one.	21
4.1	Surface representation of a shrinking cloud. The corresponding surface (red) is shifted downwards, thus decreasing the surface of the intersection with the zero level.	24
4.2	Surface representation of a laterally moving cloud. The corresponding surface (red) is shifted along the horizontal axis.	25
4.3	Sequence of synthetic images representing a sinking surface seen from the top. Time increasing from upper left to bottom right.	26

4.4	Segmentations of the first (black) and last (red) frame from the synthetic sequence of a shrinking object and future prediction (green). The color levels in the background show the motion matrix, where yellow corresponds to a stable intensities and blue to an intensity decrease.	26
4.5	Sequence of synthetic images representing a laterally moving surface seen from the top. Time increasing from upper left to bottom right. Generated with a vertical shift constant $s = 0.3$	27
4.6	Segmentations of the first (black) and last (red) frame from the synthetic sequence of a laterally moving object and future prediction (green). The color levels in the background show the movement matrix, where yellow corresponds to an intensity increase and blue to a decrease.	27
4.7	Sequence of sky images taken in ABB's Cavriglia site on July 20 th 2015	28
4.8	Segmentations of the first frame (black) and last frame (red) and predicted segmentation (green) of the sky image for the sequence taken on July 20 th . The colors in the background show the movement matrix m , where the positive values indicate an increase and the negative values a decrease of the intensity levels.	29
4.9	First and last images of a the sequence taken on October 19 th 2015	30
4.10	Segmentations of the first frame (black) and last frame (red) and predicted segmentation (green) of the sky image for the sequence taken on October 19 th 2015. The colors in the background show the movement matrix m , where the positive values indicate an increase and the negative values a decrease of the intensity levels.	31
4.11	First and last images of a the sequence taken on December 9 th 2015	31
4.12	Segmentations of the first frame (black) and last frame (red) and predicted segmentation (green) of the sky image for the sequence taken on December 9 th 2015. The colors in the background show the movement matrix m , where the positive values indicate an increase and the negative values a decrease of the intensity levels.	32
4.13	First and last images of a the second sequence taken on July 29 th 2015	33
4.14	Segmentations (black and red) and predicted segmentation (green) of the sky image for the sequence taken on on July 29 th 2015	34
4.15	Example of shift elimination (with additional frames). Contours of a sequence before (left) and after (right) applying the shift. The color of the contour is dark brown for the first image of the sequence and goes to bright red as the time increases.	34

Chapter 1

Introduction

Renewable energy like wind and solar power fluctuates in a hard to predict way and thus poses substantial challenges to the electrical grid. To mitigate this, an important research direction is on short term prediction methods. The focus of this project is cloud tracking for photovoltaic power generation. The basis is a fish-eye camera system, which is presently among the most favoured camera types for cloud tracking. The aim of this project is to apply advanced image processing methods in order to improve the predictability of cloud motion and formation. Specifically, the cases that clouds change in size, appear, disappear, grow together, or uncouple, are difficult to handle with present algorithms. Another challenge arises when different layers of clouds move with different speeds or directions. As a data source for the project, at least one running prototype installation of the camera system is accessible.

Since the 1980s, the Atmospheric Radiation Measurement Climate Research has been developing a sky-imaging system for military purposes, which is able to detect clouds day and night with a high accuracy [25]. Such a system is of course very expensive, and commercial solutions exist such as the *Total Sky Imager* developed by Yankee Environmental Systems, the *SRF-02* from EKO instruments, and *VISJ1006* from Schreder CMS. However, these systems suffer from low camera resolution and are generally not adapted to power plants generation. For this reason, several research laboratories began to develop new systems with the purpose of improving the cloud coverage estimation for solar irradiance forecasting. These systems rely on weatherproof cameras with a fish-eye or a wide-angle lens [21, 6]. Though the results are encouraging, there are still improvements to be made which motivated ABB Corporate Research to create its own system for solar irradiance measurement and short term forecasting.

The data images that were analyzed in this project were obtained from a fish-eye camera placed near a photovoltaic power plant in Caviglia, Italy. The location of this site is particularly advantageous because of the low precipitation rate compared to the rest of Europe. Moreover, strong winds are rare which makes the movement of the clouds generally slow and thus more easy to predict. A master thesis submitted to EPFL in 2014 describes in details the physical setting [31]. It also includes a first model to perform cloud tracking which is translated thereafter into irradiance prediction. In this model, each cloud is represented by a rectangular box and the tracking is performed by detecting and following the centroid of the cloud throughout a sequence of frames. With such a representation, issues occur when the

clouds split, merge, appear or disappear. It also doesn't take into account the shape of the cloud and its possible variations, which can be decisive in predicting a sun coverage.

In this project we explored a new method to estimate and predict the sky coverage, which accounts for the size and shape variations of the clouds. This method is based on the Geodesic Active Contours model, which was created in the 1990s in the context of solid-liquid interface propagation [18] and quickly became popular in the field of computer vision to perform image segmentation and motion tracking [3, 8, 2]. The key concept of this theory is based on an implicit representation of the contours of the target object, which is defined by the zero level set of a higher dimensional surface. This representation, though less intuitive and which might be more computationally demanding than a parametric one, has the advantage that the topological variations such as merger and split, are naturally handled. The large success of this approach during the last two decades provide us with a large literature on which we based ourselves in order to develop a model specifically adapted for cloud tracking in sky images.

While other works on sky-imagery for PV plants focused on improving the optical system, the cloud coverage estimation or the solar irradiance model, this thesis is mainly concerned by developing a mathematical model to capture and make a short-term prediction of the clouds' movement. The image acquisition system as well as the data pre-processing algorithms were developed at other stages of the global project and only a general overview of the models will be provided here.

The next chapter gives an introduction to active contours and an overview of the major developments since the original model by Osher and Sethian with a focus on the variational approach that we will adapt to the needs of our problem.

The chapter 3 goes more into details about the specifics of our data and the information that we want to extract from them. Then we will explicit our model and how it was implemented.

In chapter 4 we show and analyze the results that we obtain using our model both for synthetic and real sky images.

Finally a critique of the model and possible improvement will be discussed in chapter 5.

Chapter 2

Related work

In this thesis we developed a model for image segmentation and motion tracking based on the theory of active contours, also known as *snakes*. First we will start by a general overview with different types of snakes. Then we will move to geodesic active contours which is the variant that we chose for our model. We will see how image segmentation can be formulated as an optimization problem. In section 2.4 we present some possibilities on how to solve the equations. Finally we shift our focus to the motion tracking techniques that can be found in the literature.

2.1 Introduction to active contours

Active contours, are a popular way of performing image segmentation. The general concept is to start from a closed curve on the plane and evolve it until it fits the contours of a target object in the image. The evolution of this contour is performed iteratively by minimizing an energy function. The snake representation and this energy function constitute the two fundamental characteristics of a snake. The goal of this project is to find a representation and an energy model that are best suited for the detection and tracking of clouds in sky images.

Snakes were originally introduced by Kass *et al.* in 1988 [14] and were used for object tracking in image sequences first in [3] and profusely during the last two decades. A recent and detailed review of various types of snakes can be found in [10], with a focus on B-spline parametric snakes. Here we are mainly interested in geodesic snakes, therefore we will briefly go through the other representations.

There are three main categories of snakes representations

Point snakes which are a simple collection of 2D points on the plane (figure 2.1). Though very easy to grasp, this representation has the disadvantage of being computationally costly.

Parametric snakes where the contour is represented by a parametric curve $C : [a, b] \rightarrow \mathcal{R}^2, p \rightarrow C(p)$. Once a type of parametrization is selected, e.g., polynomial, B-splines, etc., the goal of the optimization is to determine the coefficients of the parametric function. This representation can be very efficient and fast, but works better with *a priori* knowledge of the shape of the target, for example ellipse shaped cells in biomedical images [11]. Besides, the topology of the curve is fixed by the parametrization, which is a major drawback for our application. Indeed, as we are trying to track clouds in a sequence of images,

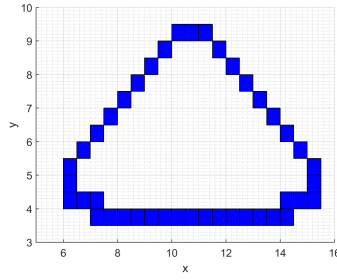


FIGURE 2.1: Point snake

it often happens that these clouds will split or merge, appear or disappear, inducing topological changes of the contour.

Geodesic snakes where the contour is implicitly represented as the zero level set of a three dimensional surface ϕ , i.e., $C = \{\mathbf{x} \in \mathbb{R}^2 | \phi(\mathbf{x}) = 0\}$, where C can be a single contour or a set of contours (figure 2.2). Note that for the same contour C there are several possible solutions for ϕ . Usually we use a *signed distance function* which is positive inside C , negative outside, and zero on the contour. Starting with an arbitrary initialization ϕ_0 (for example a checkerboard), the surface is evolved according to the so called *level set equation*

$$\frac{\partial \phi}{\partial t} + v|\nabla \phi| = 0 \quad (2.1)$$

until a convergence criterion is reached. Here v is the normal velocity of the surface and is determined by the choice of the energy function, and t designates the time steps of the iteration. Usually, the energy contains one or several regularization terms imposing smoothness on the contour or the surface, and other forces which attract the contour to the boundaries in the image. By taking the zero level set of this surface, it is easy to obtain several contours which split or merge. This natural handling of topological changes is the main advantage of geodesic snakes which makes it so suitable for the cloud tracking project.

2.2 Fast solvers to the Hamilton-Jacobi equation

The idea of using level sets to represent contours was originally introduced by Osher and Sethian [18] who were working on the propagation of solid-liquid interfaces. The interface Γ was bounding a two or three dimensional region Ω and its motion was dictated by a velocity field \mathbf{v} , following the level set equation 2.1. Finding Γ was equivalent to solving a PDE (partial differential equation) of type Hamilton-Jacobi using an iterative method. This could be rather slow and fast implementations made their appearance such as the Narrow Band [1] or the Fast Marching method [23, 24].

The idea of the Narrow Band algorithm is to reduce the computations by finding the signed distance function only around a tubular domain (or a band) around the propagating front or the curve. This method cannot be used in our case, as we need to obtain a indication of the sky cloudiness

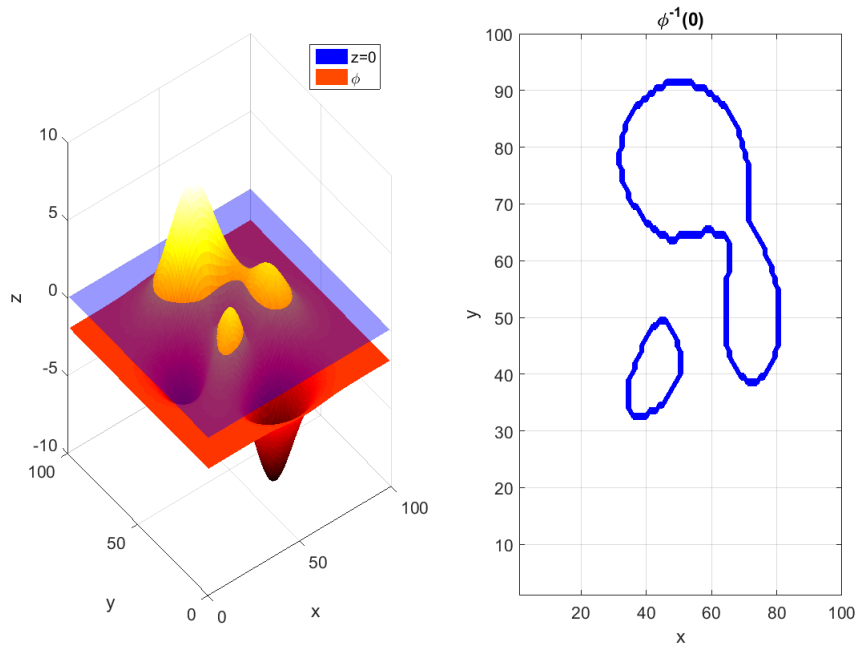


FIGURE 2.2: Geodesic snake. The blue contour results from the intersection between the surface ϕ and the plane $z = 0$.

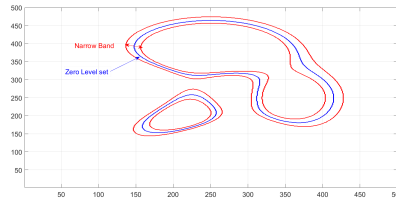


FIGURE 2.3: Narrow band

at each pixel of the image in order to compute an estimation of the power input. Moreover, by using a narrow band we discard the influence of distant pixels which can be determinant in predicting the appearance of a new cloud for example.

The Fast Marching Method is closely related to Dijkstra's algorithm, which is a famous method for computing the shortest path in a graph. The idea was introduced by Tsitsiklis [26] who developed a fast method to solve the Eikonal equation.

$$|\nabla \phi| = F(\mathbf{x}) > 0 \quad (2.2)$$

where F is a given speed function. The fast marching method as devised by Sethian [23] finds an *arrival time surface* which gives the arrival time of the propagating front at each point of a grid given the surface speed. It is fast because it starts from the lowest point of the surface and builds the rest of it *slice by slice*, i.e., one level at a time. However, as indicated by equation 2.2, this method assumes that the speed function never changes sign, so that the front is always propagating either outwards or inwards. This of course is not what happens with deforming clouds, as we can observe simultaneously growing and shrinking parts in the same object.

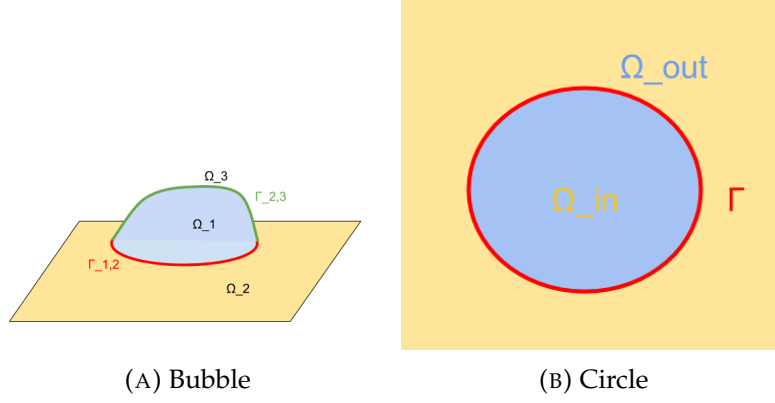


FIGURE 2.4: Example of disconnected regions separated by an interface

2.3 The variational level set method

The variational approach [32] was introduced in 1996 as an elegant mathematical solution to the problem of Multiphase Motion, i.e., when the studied material has several phases and thus several disconnected interfaces. It restated the problem of equation 2.1 as an optimization problem. In the following, we see how this approach was formulated by Osher and Fedkiw in their level set methods overview from 2000 [17].

First let's start with a graphical example. A soap bubble on a solid support can be viewed as three regions (or phases) Ω_1 , Ω_2 and Ω_3 separated by a solid and a liquid interface $\Gamma_{1,2}$ and $\Gamma_{1,3}$ respectively (figure 2.4a). The area of the bubble is linked to the internal and external pressure via the *surface tension*, which is the Gibbs energy per unit area [7]. The interfaces take the shape which will minimize this energy. Similarly, a circle on a planar image can be represented as the interface between the foreground object, which is a disc, and the background (figure 2.4b). We can define an energy term whose minimum gives us the right shape and size of this interface.

We now move to the general case. Let us consider a disjoint family of n regions Ω_i with Γ_{ij} being the boundary between the region Ω_i and Ω_j , and the associated energy

$$E = E_1 + E_2 \quad (2.3)$$

where

$$E_1 = \sum_{1 \leq i < j \leq n} f_{i,j} \text{length}(\Gamma_{i,j}) \quad (2.4)$$

and

$$E_2 = \sum_{1 \leq i \leq n} e_i \text{area}(\Omega_i) \quad (2.5)$$

The interface is captured as the zero level set of a smooth function ϕ such that $\Gamma(t) = \{\mathbf{x} | \phi(\mathbf{x}, t) = 0\}$. These energies can be re-written as

$$E_1 = \sum_{i=1}^n \gamma_i \int \int \delta(\phi_i(x, y, t)) |\nabla \phi_i(x, y, t)| dx dy \quad (2.6)$$

$$E_2 = \sum_{i=1}^n e_i \int \int H(\phi_i(x, y, t)) dx dy \quad (2.7)$$

Where δ is the Dirac delta "function"

$$\delta(x) = \begin{cases} +\infty & \text{if } x = 0, \\ 0 & \text{if } x \neq 0. \end{cases}$$

and H is the Heaviside step function

$$H(x) := \int_{-\infty}^x \delta(s) ds \quad (2.8)$$

$f_{i,j}$, e_i and γ_i are constants.

Now, instead of solving equation 2.1 with an iterative scheme, ϕ is found by minimizing E under the constraint

$$\sum_i^n H(\phi_i(x, y, t)) - 1 \equiv 0 \quad (2.9)$$

Soon, the level set method found several applications including computer vision and image processing where the variational approach could be easily used [4, 3, 2, 15, 19, 20, 27]. In this context, the image is represented by a function $\rho(x, y)$ which maps the coordinates on a square to the intensity value at that position. The surface, or the curve moves with a velocity which vanishes when the target object in the image is surrounded. A famous model of variational level set method for image segmentation was created by Mumford and Shah [16] in 1989 where they approximated an image ρ by a piecewise smooth function f . In their formulation, they used as energy function

$$E(f, C) = \mu^2 \int \int_{\mathbb{R}} (f - \rho)^2 dx dy + \int \int_{\mathbb{R} \setminus |C|} \|\nabla f\|^2 dx dy + \nu |C| \quad (2.10)$$

where C is the contour on which f is allowed to be discontinuous, and $|C|$ stands for the length of this contour. μ and ν are simply weights, used to tune the optimization. The first term asks that f approximates the image ρ . Then we have a smoothness term which imposes that f doesn't vary much on each region, and finally another smoothness term which limits the length of the contour C .

In our context, this approach has the advantage of allowing to segment a full sequence of images in a single optimization. By including an appropriate link between the different frames in the energy function, we can also impose that the signed distance function does not vary much from an image to the next.

2.4 Solving the optimization problem

After formulating the segmentation as an optimization problem, the standard method to solve it is to use a gradient descent. In broad terms, the

algorithm starts with an initial guess ϕ_0 and moves each level in the direction of its normal until a local minima is found. The mathematical development using Lagrange multipliers was made in [32]. To accelerate the computations, they replaced the constraint 2.9 by

$$\int \int \frac{(\sum H(\phi_i(x, y, t)) - 1)^2}{2} dx dy = \epsilon \quad (2.11)$$

with $\epsilon > 0$ as small as possible. Then they found

Theorem 1 *The solution to the minimization problem :*

M : Minimize $E_1 + E_2$ of equations 2.6 and 2.7 in a fixed domain D to the integral constraint 2.11

satisfy for $i = 1, \dots, n$

$$\delta(\phi_i) \left(\gamma_i \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - e_i - \lambda_L \left(\sum_{i=1}^n H(\phi_i) - 1 \right) \right) = 0 \quad (2.12)$$

with boundary conditions

$$\frac{\delta \phi_i}{|\nabla \phi|} \frac{\delta \phi}{\delta n} = 0 \text{ on } \delta D \quad (2.13)$$

where λ_L is a Lagrange multiplier.

The advantage of the gradient descent is that it can be applied to any type of energy functional. However, it suffers many shortcomings including sensitivity to local minima caused by noise in real images. To overcome this, Chan and Vese created a model without edges [5] based on the Mumford-Shah energy functional 2.10 where the function f was only allowed two values

$$f(\mathbf{x}) = \begin{cases} c_1 & \text{if } \mathbf{x} \text{ inside } C, \\ c_2 & \text{if } \mathbf{x} \text{ outside } C. \end{cases}$$

with an additional term to limit the area inside the contour

$$E(c_1, c_2, C) = \mu \text{ length}(C) + \nu \text{ Area}(\text{inside}(C)) + \lambda_1 \int_{\text{inside}(C)} |f(x) - c_1|^2 dx + \lambda_2 \int_{\text{outside}(C)} |f(x) - c_2|^2 dx \quad (2.14)$$

The evolution equation of the surface ϕ became

$$\frac{\delta \phi}{\delta t} = |\nabla \phi| \left[\mu \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} - \nu - \lambda(f - c_1)^2 + \lambda(f - c_2)^2 \right] \quad (2.15)$$

for parameters $\mu, \nu, \lambda \geq 0$. This model became a staple in image segmentation via level sets thanks to its remarkable robustness to noise.

Other attempts to improve the performance of the gradient descent also involved a modification of the energy functional, such as using a probabilistic framework [8]. In our project, though we were largely inspired by the Chan-Vese expression of the energy functional, we preferred to rely on an external solver [28] which gave us more flexibility for the motion tracking. More details will be provided in the next chapter.

2.5 Motion tracking with active contours

In addition to image segmentation, active contours are also an ideal tool for motion tracking in sequences of images. Although parametric snakes can be used [3, 13] geodesic snakes are a better choice for non-rigid motion, because the topological changes, such as splitting and merging are naturally handled, and because the parametrisation tends to constraint the shape of the target.

Several methods combined statistical frameworks to geodesic active contours [8]. In 1998, Isard and Blake proposed the Condensation algorithm [13] which relied on parametric B-Spline snakes and particle filtering. This idea was transposed later to geometric active contours [22].

Many of the statistical methods relied on a Bayesian maximum *a posteriori* (MAP) estimation [8] which was linked to the Mumford-Shah functional by Zhu and Yuille in 1996 [33]. The general idea is to compute an optimal partition $P(\Omega)$ of the image plane Ω by maximizing the a posteriori probability $p(P(\Omega)|\rho)$ for a given image ρ . This was applied by Paragios and Deriche [19] for the detection and tracking of moving objects in video sequences. They alternated motion detection and boundaries tracking using a statistical edge detector and transformed it into a geodesic active contours problem.

The statistical approach allows to detect more easily the target object in the next frame. Our goal however is to capture the motion during the sequence in order to translate it into a prediction.

With this goal in mind, it is also worth mentioning works which were interested in image registration rather than object tracking on a long sequence of images [30, 27, 12] also using an active contour framework. The level set equation 2.1 only gives the evolution of the surface. In order to obtain the geometric transformation between two images, they explicitly tracked the deformation field vector $u : \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that

$$\phi(\mathbf{x}, t) = \phi(\mathbf{x} + u(\mathbf{x}, t), 0) \quad (2.16)$$

However, this approach was developed with the aim of tracking a single object in "sequence" of two frames.

2.6 Summary

The literature devoted to image segmentation and motion tracking via level sets is huge, and the methods above are by no means exhaustive. They consisted however a solid basis on which to build our model.

We have seen that we can represent active contours in different ways, and previous works have shown that an implicit representation was best suited for contours with high variability. This approach requires finding a signed distance function to the borders of the contours which can be formulated as an optimization problem. The Chan-Vese model is a very popular level set formulation which does not rely on the images edges making it particularly robust to noise.

Building on this knowledge, we have developed a model specifically designed to detect and track non-rigid object and simultaneously be able to generate a prediction in the future.

Chapter 3

Cloud tracking with Active Contours

In this chapter we want to define our problem more precisely and see how we can solve it using geodesic active contours. First we will describe the input data that we need to analyze with the specific challenges that they present and what pre-processing techniques were used to overcome these challenges. After the practical aspects, we will move to the theoretical part starting by choosing an appropriate representation for the moving surface in section 3.2. In section 3.3 we will pose the segmentation and tracking as an optimization problem and go through each term of the energy function individually. Afterwards, we will derive the matrix formulation of the problem in section 3.4. Finally, in section 3.5, we will combine our model with a pre-existing optical flow algorithm which can be useful for sequences with a strong lateral movement component.

3.1 Input images

The data images that we will process were collected by a fish-eye camera in Cavriglia, Italy, as part of ABB's research project on cloud tracking. This camera takes several pictures with different exposures which are merged into a High Dynamic Range Image (HDRI) using the Debevec algorithm [9]. Then the software in charge of the image acquisition saves this HDR image to the data base. Thus, we obtain a data image every six seconds.

The camera is placed near a PV power plant of which we can also obtain the data. In figure 3.1, we can see data collected on July 20th 2015. The two subplots on the right are measurements obtained directly by captors in the power plant. On the bottom one, we can see the irradiance measurement for that day (blue points) compared to the irradiance on a clear day (red curve). The clear sky model has a bell shape with a peak in the middle when the sun is at the highest point. The clouds appearing in the afternoon resulted in a dramatic drop of the power input (top right). In order to avoid such ramp events to the electrical grid, we want to predict a few minutes in advance when the clouds get too close to the sun, i.e., when they enter the green circle.

Apart from precipitations and the occasional bug, there are some challenges that come with these data images. The image in figure 3.1 is a good example. Instead of the original color image, we take the grayscale image from the red to blue ratio on which we perform the segmentation. When the sun is visible in the sky image, this ratio will be high not only for the

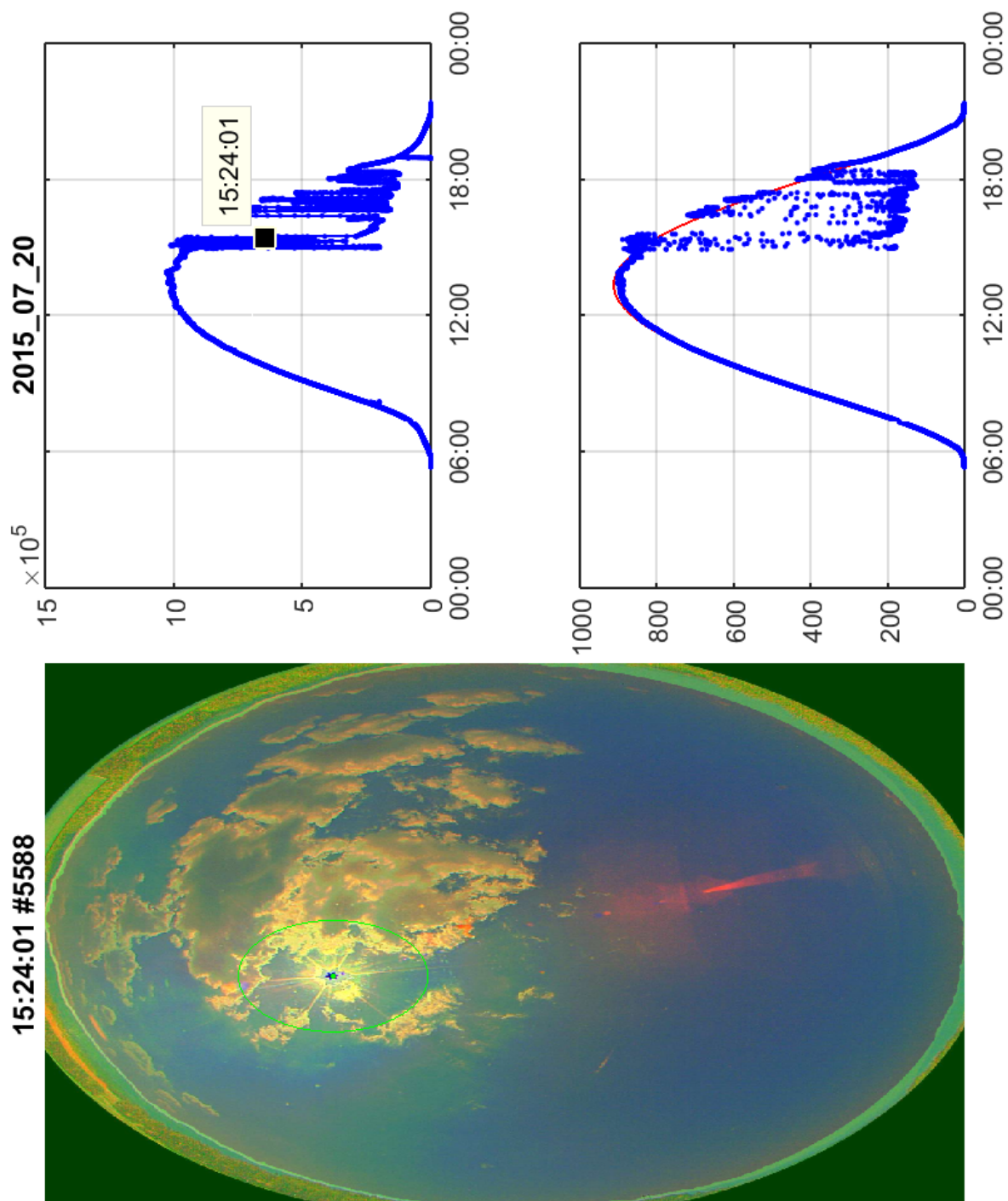


FIGURE 3.1: Data collected on July 20th 2015 from ABB's site in Cavriglia, Italy. Left : example of sky image taken at 15:24:01. Top right : Power input to the PV plant. Bottom right : irradiance measured (blue) and clear sky model (red).

clouds, but also for the sun glare, the red spots resulting from the diffusion on the camera surface, and the diffusion by the clouds themselves. The clouds themselves don't necessarily have a uniform red to blue ratio, and some dark gray areas can be detected as sky pixels. Moreover, the HDRI technique performed on each image individually very often results in a tone difference between the frames of a sequence, leading to a significant variation in the general intensities level. Therefore, two frames taken a few seconds apart where the cloudiness hasn't changed significantly may have a completely different red to blue map if the sun is visible when they were taken.

In order to overcome these difficulties, a pre-processing step is necessary to perform a coherent cloud segmentation. The technique employed for the pre-processing is outside the scope of this Master thesis. The main steps are listed in a flowchart in figure 3.2. First we increase the range of the images using the Debevec algorithm. Then we move to grayscale by taking the red-to-blue ratio. Then perform a *local thresholding* step to eliminate the noise due to the sun glare or reflections. From this local threshold, we obtain a binary image where we separate cloud pixels from sky pixels. Therefore the raw data that we will feed our optimization is a pre-segmentation of the sky image, which already gives a fairly good estimation of the cloud cover. As an example, we have represented the pre-segmentation of the image 3.1 in figure 3.3. We can see that the sun area was simply truncated, and the glare and other reflections have disappeared. Moreover, the clouds are very well detected regardless of the sun reflections.

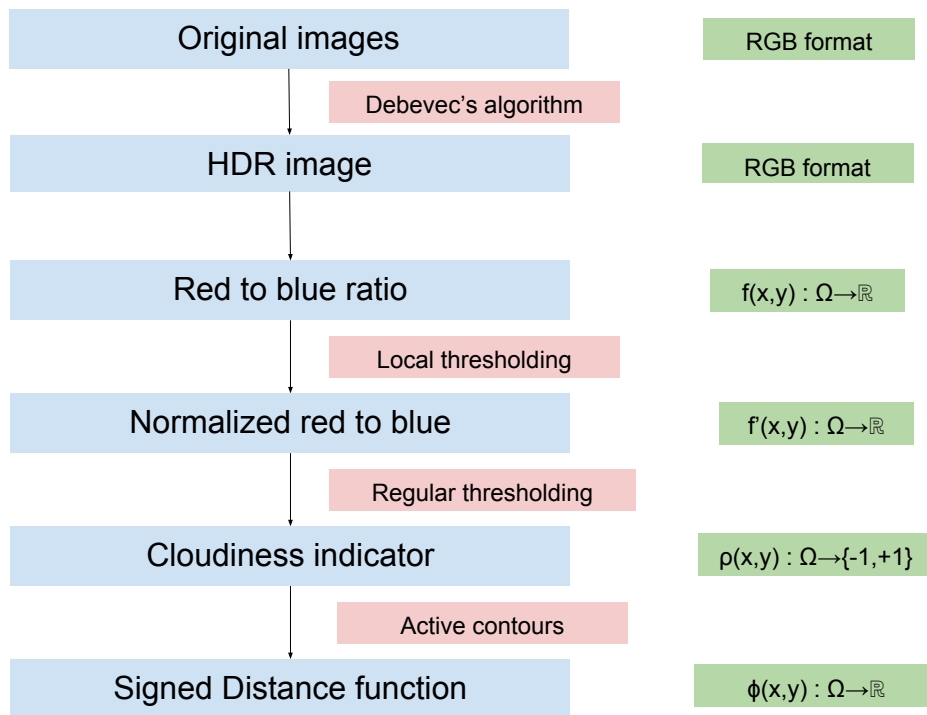


FIGURE 3.2: Process to obtain a signed distance function from a regular color image. The last step (Active Contours) is the main contribution of this thesis.

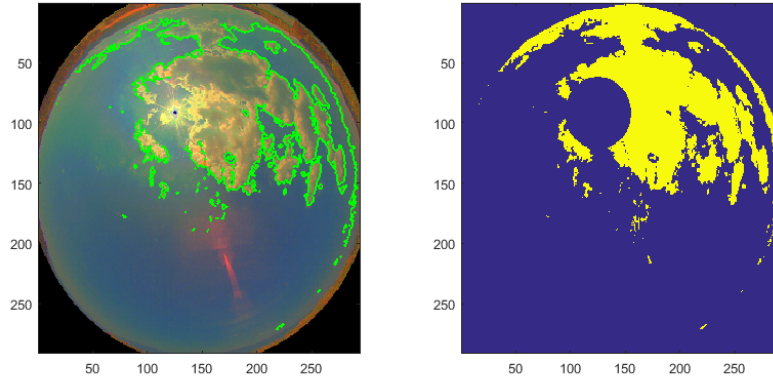


FIGURE 3.3: Pre-segmentation of a sky image taken on July 20th 2015 at 15:24:01

Note that we will use the pre-segmented image minus the circular area around the sun because the power predictions obtained from the segmentation of this area is too error prone. If a cloud comes exactly between the sun and the camera, this doesn't necessarily translate into a coverage of the PV plant surface. Reversely, the PV cells can be partly covered while the sun is still visible to the camera. It is sufficient for the purpose of this project to know when the clouds are approaching the sun within a certain margin.

The last step of figure 3.2 is the main contribution of this thesis. We are interested in getting from a grayscale image to a signed distance function. The The figure 3.4 give an idea on how to link an image and a surface.

3.2 The representation

The goal of this project is to find a way to determine and predict the cloudiness of the sky precisely enough in order to compute the expected power input of a PV plant. In this context, rather than following a particular contour, we want to build a function which we can use as an indicator of the cloudiness state of each pixel. In addition to segmenting each frame individually, we need to track the moving objects from an image to the next, and keep a trace of this movement throughout the sequence in order to predict the future position and shape of these objects.

To accomplish that, we want to find a *signed distance function from the cloud boundary* which satisfies

$$\begin{cases} \phi_i = 0 & \text{at the cloud boundary,} \\ \phi_i > 0 & \text{inside the cloud ,} \\ \phi_i < 0 & \text{outside the cloud .} \end{cases} \quad (3.1)$$

Additionally we rely on a *motion matrix*¹ m as a link between the surfaces fitting each frame of the sequence, which will also be used to perform the

¹We do not use the term *velocity* or *speed* in order not to confuse this term with the surface velocity field v in equation 2.1 which is zero when a solution is reached

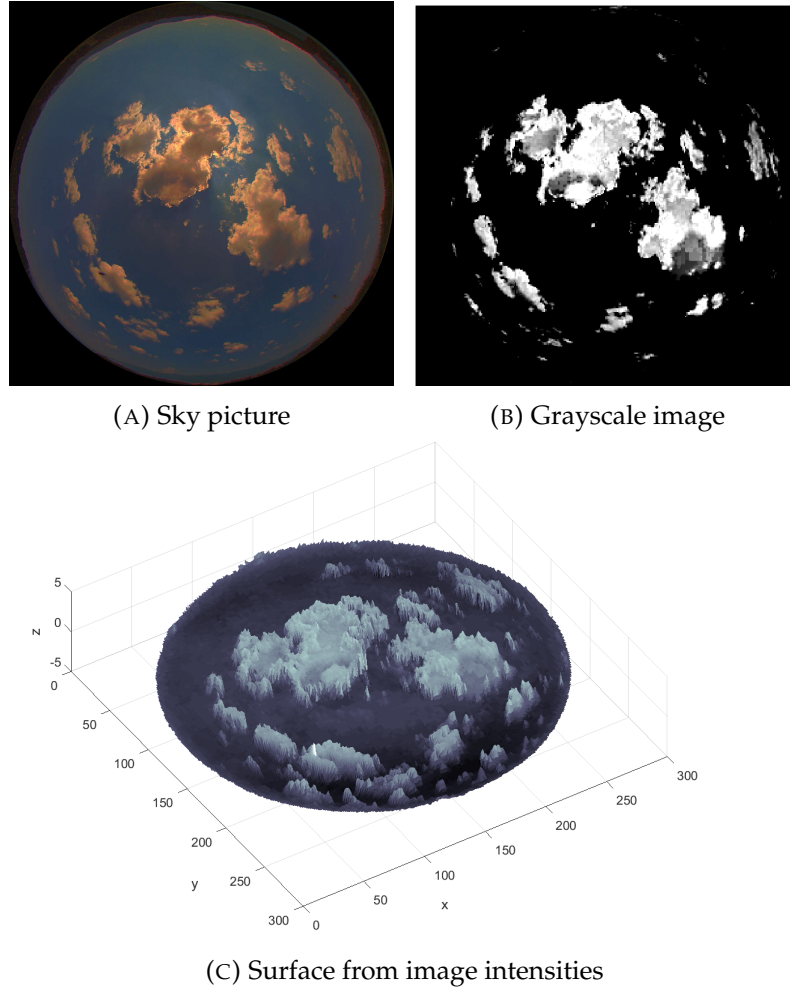


FIGURE 3.4: Grayscale image and surface corresponding to a sky image.

prediction.

In figure 3.5 we have sketched the function ϕ that we would like to build. The advantage of this signed distance function to us is that a growth of the cloud corresponds exactly to a shift upwards of the interpolating function. Similarly a shrinking cloud is well captured by a downward shift of the function.

As an input to our program, we will be using a pre-segmentation $\rho : \Omega \rightarrow \{+1, -1\}$ which is positive for a cloud pixel and negative for a sky pixel. This pre-segmentation is the *cloudiness indicator* that we obtained following the procedure of the diagram in figure 3.2.

3.3 The cost function

In order to obtain this result, we have formulated the problem as follows. Given a sequence of binary images $\rho_i(\mathbf{x}, t) : \Omega \rightarrow \{-1, +1\}$, $i = 1, \dots, \tau$ we want to find the family of surfaces $\phi_i : \Omega \rightarrow \mathbb{R}$ and the motion matrix $m \in \Omega \rightarrow \mathbb{R}$ which minimize the following cost function

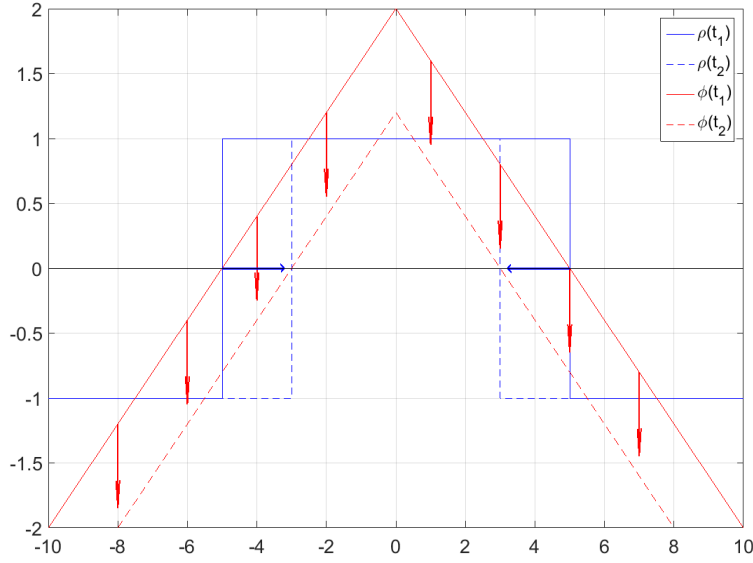


FIGURE 3.5: Signed distance function from the cloud boundary. The shrinking cloud is easily modelled by shifting down the function.

$$\operatorname{argmin}_{(\phi_i)_{i=1}^{\tau}, m} E = \sum_{i=1}^{\tau} \int E_{\rho}(\rho_i, \phi_i) + \quad \text{conformation} \quad (3.2)$$

$$\omega_{\phi} \sum_{i=1}^{\tau} \int \|\nabla \phi_i\|^2 + \quad \text{surface smoothing} \quad (3.3)$$

$$\omega_m \sum_{i=-1}^{t-1} \int \|\phi_{i+1} - \phi_i - (t_{i+1} - t_i)m\|^2 + \quad \text{motion fit} \quad (3.4)$$

$$\omega_{\nabla m} \int \|\nabla m\|^2 \quad \text{motion smoothing} \quad (3.5)$$

where

$$E_{\rho}(\rho, \phi) = \omega_{lo} \rho \cdot \phi + \omega_{hi} \max(0, -\rho \cdot \phi + c_{sat}) \quad (3.6)$$

under the constraint

$$\|\nabla \phi_i\|_2 \leq 1 \quad \forall 1 \leq i \leq \tau \quad (3.7)$$

We will now go term by term to explain the effect of this cost function.

3.3.1 Conformation

It is the constraint 3.7 along with the saturated cost function 3.6 that will give us the desired signed distance from the cloud boundary of figure 3.5. In order to understand this cost function, we will rely on a different formulation of the Chan-Vese model (section 2.3), where the conformation term

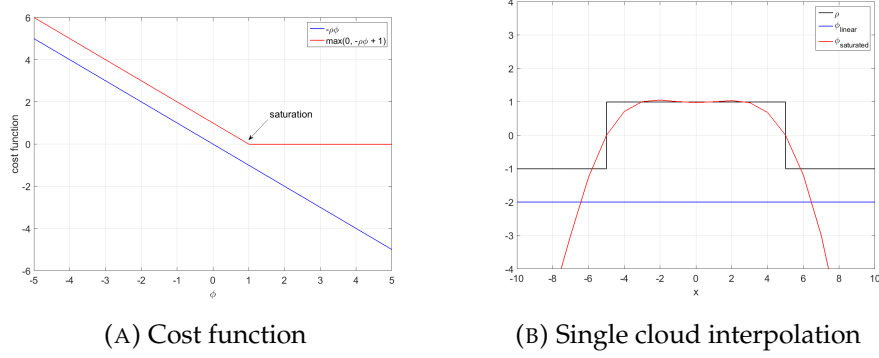


FIGURE 3.6: Difference between a linear cost function versus a saturated one for $\rho = 1$

is linear rather than quadratic. Let us suppose that we want to segment a grayscale image into background and foreground given a pixel preference function ρ which is

- $\rho(\mathbf{x}) > 0$ if the pixel \mathbf{x} is rather cloud
- $\rho(\mathbf{x}) < 0$ if the pixel \mathbf{x} is rather sky
- $\rho(\mathbf{x}) = 0$ if the pixel \mathbf{x} is undecided

We need to find the function $f : \Omega \rightarrow \{0, 1\}$ which minimizes the energy

$$E(f) = \int_{\Omega} \|\nabla f\| - \rho f dx \quad (3.8)$$

As before, we have that f is allowed two values depending if the pixel is inside ($c_1 = 1$) or outside ($c_2 = 0$) the desired contour. If we used such a simple linear cost in our model, many of the individual small clouds would be ignored. In figure 3.6a, we can see in blue the cost of this linear term for $\rho = 1$, i.e., in presence of a cloud. In this case, we observe that the cost function keeps decreasing with ϕ without limit. This might give rise to artificially high values of ϕ and because of the smoothing term, the total energy would be more favourable with a flat surface (figure 3.6b). To avoid this, we saturate the cost function so that it cannot be lower than a certain threshold (red curve in figure 3.6a). This is expressed in the second term of equation 3.6 and is accompanied with a high weight ω_{hi} .

In figure 3.6b, we can see how the saturation has for effect to pull the interpolated function to isolated cloud, but this function is flattened at the top. If we want to obtain a triangular function as in figure 3.5 we can add a linear term so that the cost function may continue to decrease but very slowly because we set it with a lower weight ω_{lo} . In figure 3.7 resulting cost function is sketched by a black solid line. The low weight term is shown in blue with an exaggerated slope ω_{lo} .

3.3.2 Motion fit

The third term of the cost function 3.4 defines the motion matrix m in a way similar to a linear regression model. We want a single matrix to describe

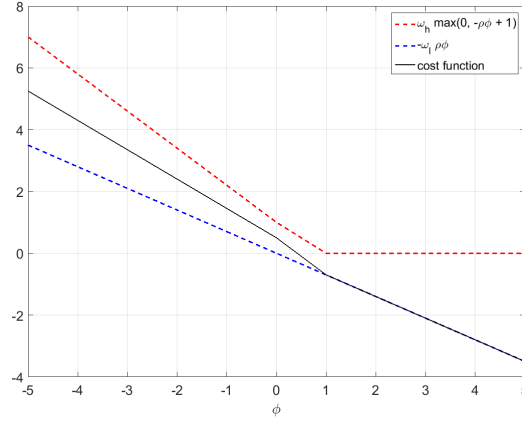


FIGURE 3.7: Scheme of the cost function for $\rho = 1$. The slopes ω_{lo} (blue curve) and ω_{hi} (red curve) are exaggerated for clarity

the movement of the full sequence. The assumption of a linear progression is safe within a reasonably short time range. Let us consider the intensities $\phi_{i=1,\dots,\tau}(\mathbf{x}_p) = z(t)$, $t \in T$, at a single pixel from the image. We suppose a linear evolution of $z(t)$ through time, and we are looking for the *slope* $m(\mathbf{x}_p) = \alpha$ which best approximates (in the least square sense) the relationship

$$z(t) = \alpha \cdot t + \epsilon_t, \quad t \in T \quad (3.9)$$

where ϵ_t is the error.

By finding α for each pixel, i.e., the matrix $m(\mathbf{x})$ we can make a prediction for the intensities at a following time step

$$\phi_{\text{next}} = \phi(\mathbf{x}, t + \Delta t) = m(\mathbf{x}) \cdot \Delta t + \phi(\mathbf{x}, t) \quad (3.10)$$

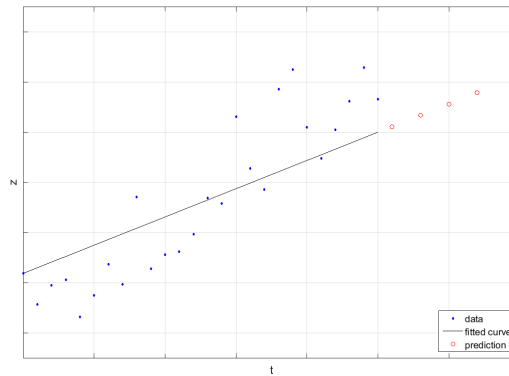


FIGURE 3.8: Example of linear regression. The data corresponds to the image intensity at a single pixel through time

3.3.3 Regularization

The second and last term of the cost function (equations 3.3 and 3.5) are regular smoothing terms. The former one imposes the smoothness of the

surface ϕ while the latter imposes the smoothness of m , so that a group of neighbouring pixels will vary as much as possible with a similar intensity and in the same direction.

3.4 The solver

In order to solve the problem of section 3.3 we rely on the open source software IPOPT (Interior Point OPTimizer) [28] which was designed to find solution to non-linear problems (NLP) of the form

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{such that} \quad & g_L \leq g(\mathbf{x}) \leq g_U \\ \text{with} \quad & \mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \end{aligned} \quad (3.11)$$

where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, and $g(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ are the constraint functions. The vectors g_L and g_U denote the lower and upper bounds on the constraints, and the vectors \mathbf{x}_L and \mathbf{x}_U are the bounds on the variables \mathbf{x} .

More details on this method can be found in [28, 29].

There is a special type of NLPs called quadratic programming (QP) which can be written as

$$\begin{aligned} \text{minimize} \quad & \mathbf{u}^T Q \mathbf{u} + c^T \mathbf{u} \\ \text{such that} \quad & A \mathbf{u} \leq b \end{aligned} \quad (3.12)$$

We can write our optimization problem as a QP model and feed it to the software. First, we need to define the *variable* \mathbf{u} , which is the solution of our optimization as a 1D vector. This vector includes the members of ϕ , m to which we add a *slack variable* s that we will use to obtain the conformation term. So we have

$$\begin{aligned} u = [\phi_{11}, \dots, \phi_{1n}, \dots, \phi_{n1}, \dots, \phi_{nn}, \\ m_{11}, \dots, m_{1n}, \dots, m_{n1}, \dots, m_{nn}, \\ s_{11}, \dots, s_{1n}, \dots, s_{n1}, \dots, s_{nn}]^T \end{aligned} \quad (3.13)$$

We used a quadratic term to approximate the gradient. Let a be an $n \times n$ vector. The gradient can be written $(a_j - a_{j-1})^2, j = 1, \dots, n$. Using the fact that

$$\begin{aligned} (a_j - a_{j-1})^2 &= a_j^2 - 2a_j a_{j-1} + a_{j-1}^2 \\ &= [a_j \quad a_{j-1}] \tilde{Q} \begin{bmatrix} a_j \\ a_{j-1} \end{bmatrix} \end{aligned} \quad (3.14)$$

where $\tilde{Q} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ and taking into account the double use of each mid-element of the vector u :

$$\begin{aligned} (a_2 - a_1)^2 &= a_2^2 - 2a_2 a_1 + a_1^2 \\ (a_3 - a_2)^2 &= a_3^2 - 2a_3 a_2 + a_2^2 \\ &\dots \end{aligned} \quad (3.15)$$

we obtain the tri-diagonal matrix

$$Q_{\nabla} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & & \\ 0 & -1 & 2 & -1 & \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & & & -1 & 2 & -1 \\ 0 & & & 0 & -1 & 1 \end{bmatrix} \quad (3.16)$$

such that $\mathbf{a}^T Q_{\nabla} \mathbf{a} \approx ||\mathbf{a}||^2$. Now we can use this matrix for the surface smoothness, the motion fit and motion smoothness. We need an extra matrix for the motion fit

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \\ 0 & & & -1 & 1 \end{bmatrix} \quad (3.17)$$

Finally we can write

$$Q = \omega_{\phi} \begin{bmatrix} Q_{\nabla} & & \\ & 0 & \\ & & 0 \end{bmatrix} + \omega_m Q_{\nabla} \begin{bmatrix} D & & \\ & -D \cdot t & \\ & & 0 \end{bmatrix} + \omega_{\nabla m} \begin{bmatrix} 0 & & \\ & Q_{\nabla} & \\ & & 0 \end{bmatrix} \quad (3.18)$$

Now let's move to the conformation term. For the low weight term, we pose

$$c = w_{lo} \rho \quad (3.19)$$

For the high weight term, we use a slack variable. Let $s = \max(0, -\rho \cdot \phi + c_{sat})$. We ask the optimizer to find $s = -\rho \cdot \phi + c_{sat}$ under the constraint $-\rho \cdot \phi - s \leq -c_{sat}$. This can be expressed using the matrix

$$A_s = \begin{bmatrix} \rho & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ 0 & \rho & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & & \ddots & \ddots & \ddots & \\ 0 & \cdots & & \rho & 0 & & \cdots & 0 & 1 \end{bmatrix} \quad (3.20)$$

Now all is left is the constraint 3.7 for which we can reuse the matrix Q_{∇} . So we finally have

$$A = \omega_{hi} A_s + \begin{bmatrix} Q_{\nabla} & & \\ & 0 & \\ & & 0 \end{bmatrix} \quad (3.21)$$

and

$$\mathbf{b} = [-c_{sat}, \cdots, -c_{sat}, 0, \cdots, 0, 1, \cdots, 1]^T \quad (3.22)$$

3.5 Including horizontal movement

As our model is pixel based, it only considers the vertical movement of the surface at each point. This view works very well for the detection and prediction of growing or shrinking clouds which translate into the increase

or decrease of the intensity at that pixel (figure 3.9a). However, it is difficult with this representation to describe a purely lateral movement. In figure 3.9b, we see the surface intensity variation per pixel in such a case indicated by red arrows. The change captured in the movement matrix will be that one half of the surface is sinking while the other half is growing. If we use this variation to predict the next state, we will obtain a deformation of the surface rather than shift. An example of this behaviour will be shown in the results section 4.1.2.

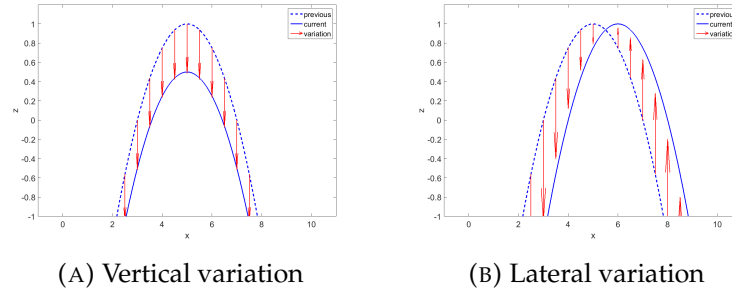


FIGURE 3.9: Pixel intensity variation (red arrows) for purely vertical and purely horizontal movement. This variation is used to predict the next state of the curve.

As our model works best with a purely vertical movement of the surface, one possibility is to eliminate movement in other directions by shifting the surfaces of the full sequence so that they overlap (see figure 3.10). We will try to emulate a three dimensional movement building on the optical flow model previously implemented in the cloud tracking project, which computes the lateral movement of the clouds. Then we apply the AC-model to take care of the vertical component. The resulting prediction needs to be shifted again, in order to take into account the lateral movement. The main steps are detailed in algorithm 1.

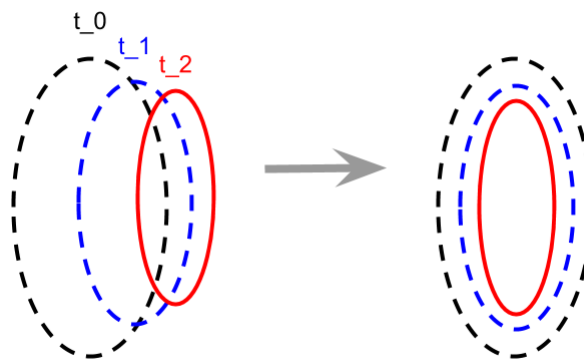


FIGURE 3.10: In strategy 2, all the contours of the sequence are shifted horizontally to match the last one.

We now have three different ways of predicting the clouds motion

strategy 0 where we rely on the lateral movement of the clouds using optical flow

strategy 1 where we only take into account the vertical variation of the pixel intensities by using active contours

strategy 2 where the lateral movement is eliminated before the vertical variation is analyzed

Algorithm 1: Strategy 2

Input: A collection S of grayscale images $p_i, i = 1, \dots, n$, a timestep Δt

Output: A matrix ϕ_{next} predicting the segmentation of the future image

```

1 Let  $S' = \{\}$  be a new empty sequence
2 for  $i \leftarrow 1$  to  $n$  do
3    $v \leftarrow \text{lateralMotionVector}(p_i)$ 
4    $p' \leftarrow \text{shift}(p_i, v)$ 
5    $S' \leftarrow S' \cup \{p'\}$ 
6  $(\phi_i)_{i=1}^n, m \leftarrow \text{motionTracking}(S')$ 
7  $\phi_{\text{next}} \leftarrow \phi_n + m \cdot \Delta t$ 
8  $\phi_{\text{next}} \leftarrow \text{shift}(\phi_{\text{next}}, v)$ 
9 return  $\phi_{\text{next}}$ 

```

3.6 Summary

In this chapter, we have seen how the original color images collected by the camera are transformed and pre-processed to obtain our input data, which is already a segmentation of the image into sky and cloud pixels. We determined a target signed distance function with a hat shape which facilitates the tracking of size variation in the clouds. We developed an energy function with a data fit, a motion fit, and two regularization terms. The output of the optimization is a sequence of signed distance functions, one for each frame of the sequence, and a motion matrix which regulates the transition between them. We wrote a matrix formulation of our problem so that we can rely on the powerful IPOPT solver. Finally we combined our program with a pre-existing algorithm to include the lateral motion in our model.

Chapter 4

Results

Now that we developed a theoretical model and selected a way to solve it we will test it on some sequences. Real images are often noisy and contain too much information to allow an objective assessment. Therefore we will start in section 4.1 by testing our model on two synthetic image sequences which contain each a single object with a one directional movement. Afterwards we will move to real sky images in section 4.2 starting with the common case of slowly deforming clouds, then testing our combined strategy from section 3.5 where the clouds have a stronger lateral movement. For each sequence we will represent in the same figure the segmented contours of the first and last frames along with the prediction for the future. We will also show the motion matrix to observe the areas of growing and decreasing intensities. This should allow us to assess whether the direction of the direction of motion is properly captured and predicted.

All the results shown in this chapter were obtained with the weights set to $\omega_\phi = 0.01$, $\omega_m = 1$, $\omega_{\nabla m} = 1$, $\omega_{hi} = 1$, $\omega_{lo} = 0.01$, and $c_{sat} = 0.5$.

4.1 Experiment on a sequence of synthetic images

First, we start by applying the program on a sequence of synthetic images in order to assess its results as objectively as possible.

To simplify their representation, we divide the possible movements of the clouds into two types

- The shrinking or growing of a cloud translating into a vertical shift in the corresponding surface respectively downwards, or upwards (see figure 4.1).
- The lateral movement of a constant size cloud, translating into the lateral shift of the corresponding surface (see figure 4.2).

Usually the clouds motion is a combination of both with one predominating over the other.

4.1.1 Vertically moving surface

We first take the case of a shrinking cloud, which is represented by a sinking surface (figure 4.1). The sequence of synthetic images is generated using a two dimensional Gaussian function with a decreasing base

$$z_t = \exp(-x^2 - y^2) - \epsilon t \quad (4.1)$$

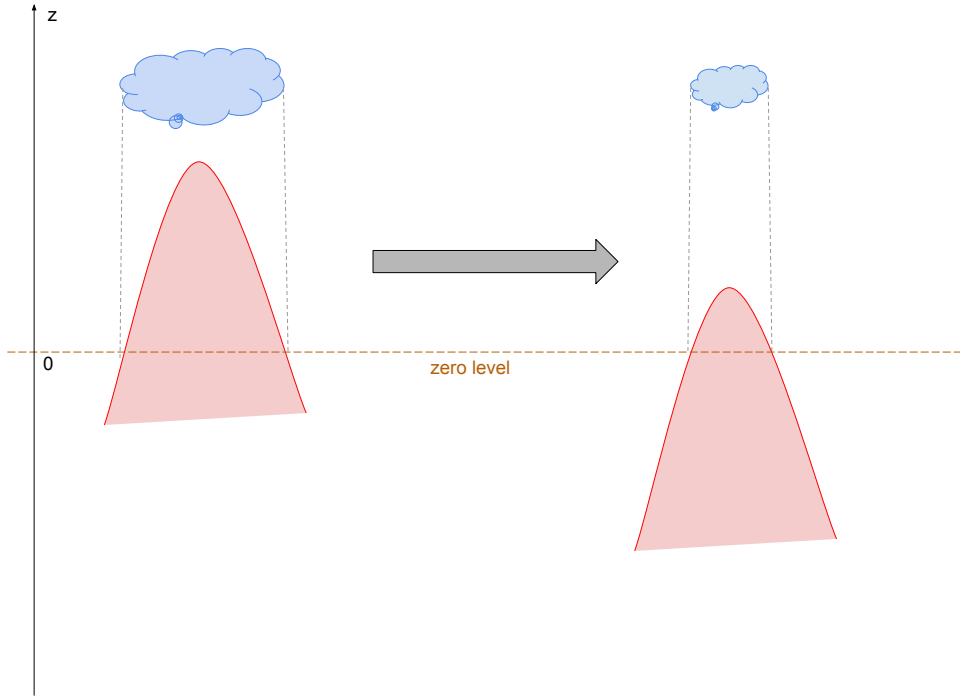


FIGURE 4.1: Surface representation of a shrinking cloud. The corresponding surface (red) is shifted downwards, thus decreasing the surface of the intersection with the zero level.

Were $t \in \{1, \dots, 9\}$ and ϵ is a parameter. The generated sequence of 9 images can be viewed at figure 4.3, where the intensity of the graylevels gradually decreases from the first image (top left) until the last (bottom right). By applying the program on such a sequence, we expect to predict a further shrinking of the cloud, i.e., a further sinking of the corresponding surface until the cloud completely disappears.

In figure 4.4, we see the results obtained for the shrinking cloud synthetic sequence. The black contour corresponds to the segmentation of the first frame ϕ_0 , the red contour the segmentation of the last frame ϕ_τ , and the green one is the predicted segmentation for the next 9th frame. The contours (blue to yellow) correspond the levels of the motion matrix m . We observe that the first and last images of the sequence are successfully segmented. Furthermore, we can see that the movement matrix m has negative values between the initial and last contours of the object, and gradually grows to zero as it leaves this area, which corresponds to the movement we generated. Consequently, the prediction shows a further shrinking of the cloud. Moreover, the decrease in size seems proportional to the difference in size between the first and last segmented object. This shows that our model performs extremely well in the case of a synthetic sequence of shrinking object. The time to solve the optimization for this sequence of 9 grayscale images of size 51×51 is approximately 150 seconds. In the case of a growing object, the program also shows a similar performance.

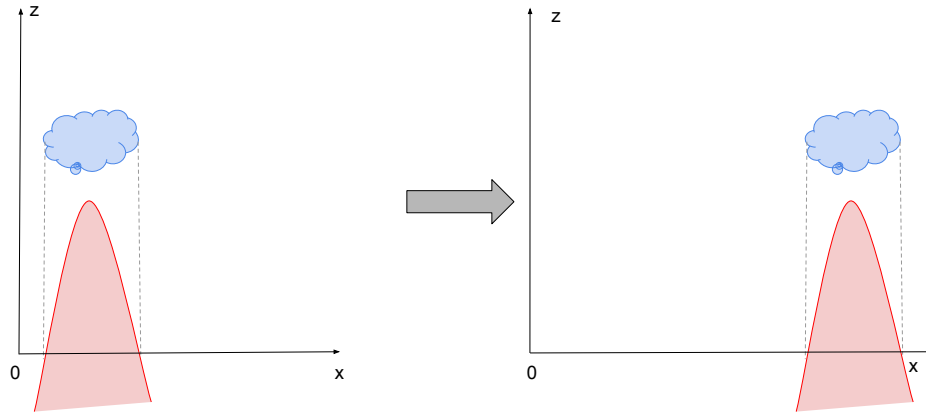


FIGURE 4.2: Surface representation of a laterally moving cloud. The corresponding surface (red) is shifted along the horizontal axis.

4.1.2 Laterally moving surface

Let us now look at what we obtain with a laterally moving synthetic cloud. A sequence of synthetic images was generated using the formula

$$z_t = \exp(-(x + 0.1t)^2 - y^2) - s, \quad t \in \{1, \dots, 9\} \quad (4.2)$$

which is a Gaussian surface with an average (center) moving along the x axis. The last term s represents a constant downward shift to obtain negative values. The resulting sequence is shown at figure 4.5.

By applying the program on this sequence we would like to predict a further shift along the x axis in the same direction followed during the sequence, i.e. from right to left. The result of the optimization is shown at figure 4.6. We do observe a general motion to the left, as the center of the predicted object has shifted in the same direction as the red contour. However, we also observe a deformation in the shape of the object, specifically a growth along the y axis, although there is no difference in size between the first and last contour. Moreover, the right side of the contour seems to be compressed slightly. This situation corresponds well to figure 3.9b, and is caused by the formulation of our problem, which only considers the vertical displacement of each point of the surface as explained in section 3.5.

We conclude from these basic experiments that the program is capable of detecting the contours of a moving object in a sequence of images. Moreover, it is capable to capture and predict the change of size of the target object throughout the sequence. However, the predicted deformation can be altered in case the target object has a movement with a lateral component. This is why we combined it with a second tracking method (section 3.5).

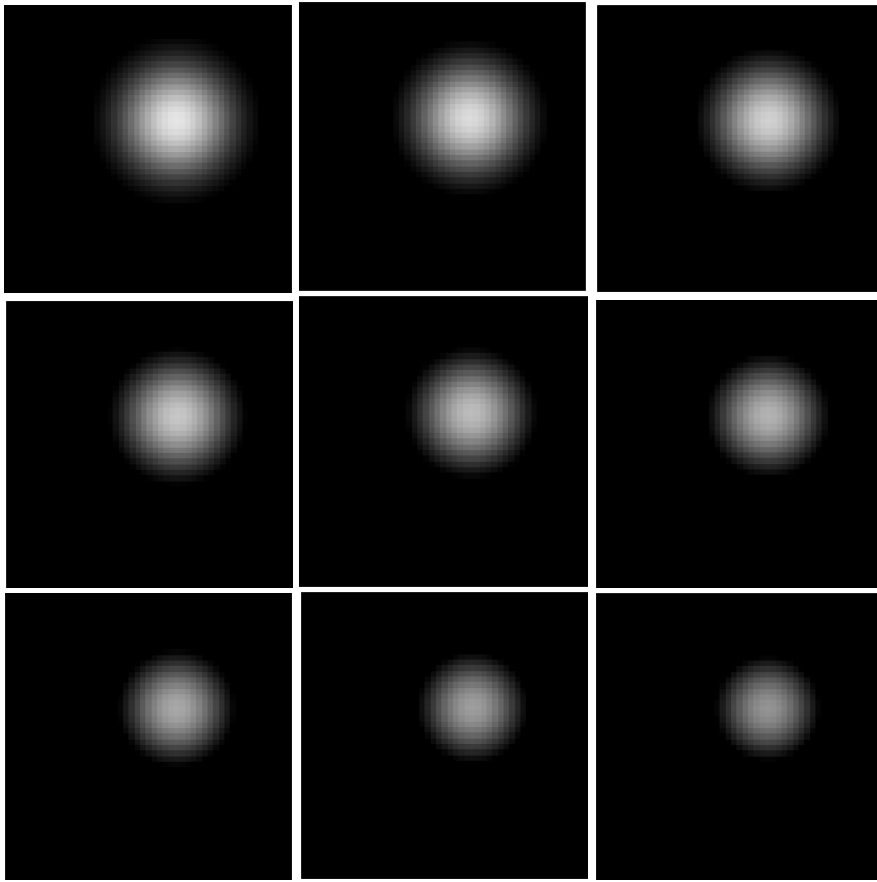


FIGURE 4.3: Sequence of synthetic images representing a sinking surface seen from the top. Time increasing from upper left to bottom right.

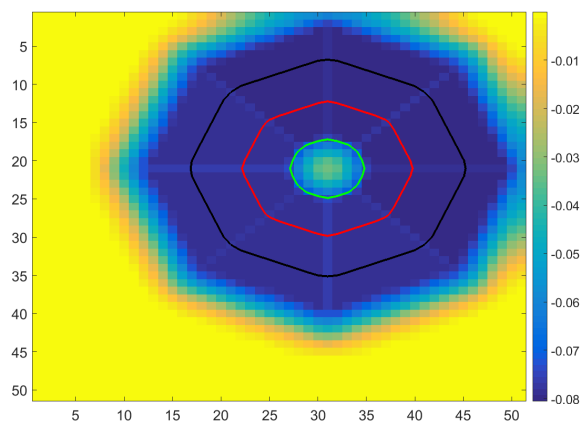


FIGURE 4.4: Segmentations of the first (black) and last (red) frame from the synthetic sequence of a shrinking object and future prediction (green). The color levels in the background show the motion matrix, where yellow corresponds to a stable intensities and blue to an intensity decrease.

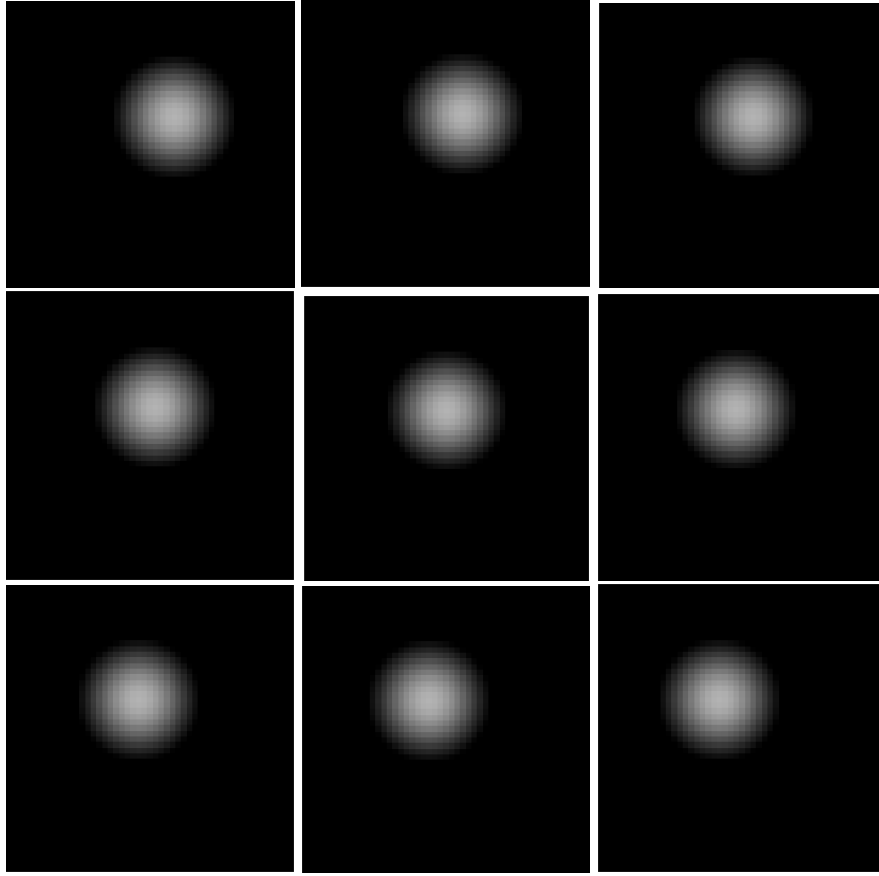


FIGURE 4.5: Sequence of synthetic images representing a laterally moving surface seen from the top. Time increasing from upper left to bottom right. Generated with a vertical shift constant $s = 0.3$.

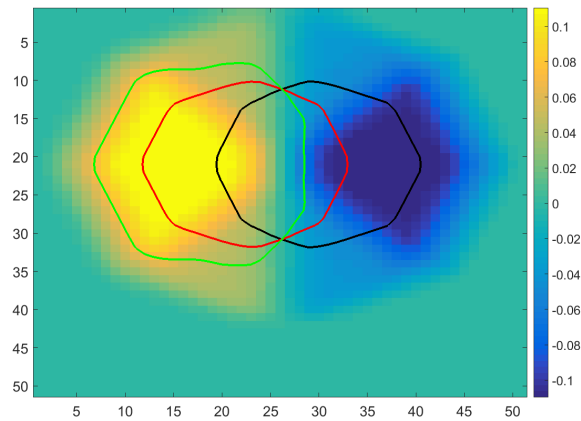


FIGURE 4.6: Segmentations of the first (black) and last (red) frame from the synthetic sequence of a laterally moving object and future prediction (green). The color levels in the background show the movement matrix, where yellow corresponds to an intensity increase and blue to a decrease.

4.2 Tracking and prediction on real sky images

4.2.1 Sequences with slowly deforming clouds

For this section, we selected three sequences taken at different seasons, thus different climatic conditions, in which the sky cloudiness is dominated by topological variations. After the pre-processing step (see section 3.1), we applied our model in order to segment the frames and generate a prediction in the near future. The time for which we make the prediction is proportional to the time step between the frames, around one or two minutes.

For the first experiment, we selected a sequence of three images from July 20th 2015 at 16:01:07, 16:02:13 and 16:03:19 (figure 4.7). In this sequence, the sun is completely covered by the clouds, and is about to re-appear. The clouds are slightly moving to the south of the image while deforming.

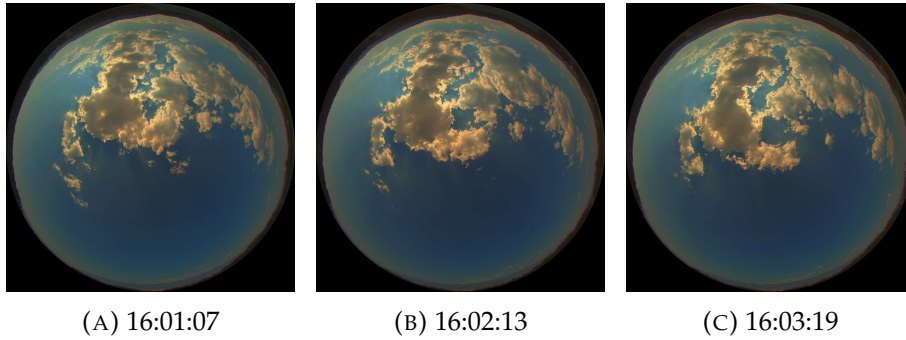


FIGURE 4.7: Sequence of sky images taken in ABB's Cavriglia site on July 20th 2015

The result for this sequence is shown at figure 4.8. As before, we show the distribution of the movement in the background. The negative values (blue) indicate a decrease of the intensity level, and the positive values (yellow) an increase, where a growth of the cloud should be observed. The black contour shows the segmentation of the first frame (figure 4.7a), the red contour the segmentation of the last frame (figure 4.7c), and the green is the prediction in the future.

In general we can notice that the segmentation seems to correspond to what we see in the original images. The contours are smooth enough while allowing for the detection of smaller objects. Moreover, the prediction generally corresponds to the evolution from the black to the red contour. In the center of the image for example, we have a large yellow area surrounding a front propagation from black to red. The predicted contour, in green, continues in the same direction with similar speed.

For this sequence the optimizer converged in ≈ 20 seconds.

For the second sequence of the test, we used three frames taken on October 19th, 2015 between 12:45:09 and 12:47:37. The first and last frames of the sequence are shown at figure 4.9. Here the contours of the clouds are more diffuse than in the previous sequence, and they show a stronger shape variation. We can note the appearance of a *hole* in the main upper group, and a smaller cloud becoming thinner and more flat at the bottom.

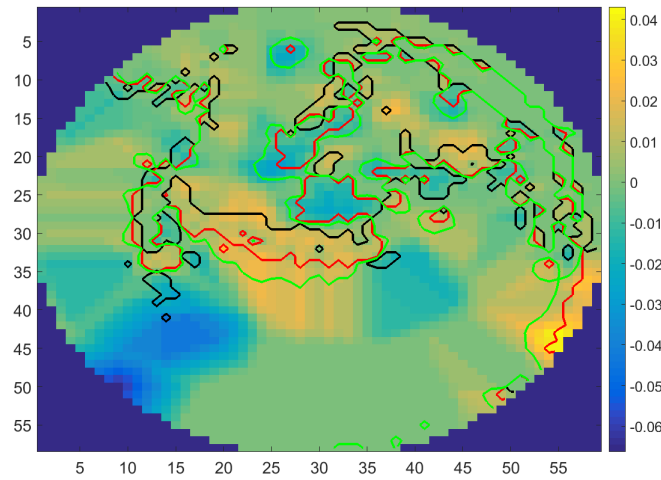


FIGURE 4.8: Segmentations of the first frame (black) and last frame (red) and predicted segmentation (green) of the sky image for the sequence taken on July 20th. The colors in the background show the movement matrix m , where the positive values indicate an increase and the negative values a decrease of the intensity levels.

In figure 4.10 we have the results of the optimization for this sequence. First we observe that even the diffuse clouds are successfully detected, owing to the pre-segmentation step. As before, the predicted contour in green expands further from the red one in the growth areas signalled in yellow, and pulls away from it in the blue areas, reflecting the variation between the first frame (black) and the last frame (red). We can also observe that the large whole keeps its expansion in the prediction. We notice that the prediction (green contour) follows more the shape and position of the last segmentation (red) than the first (black). This is particularly remarkable for the isolated cloud at the bottom. Thus, even for a short sequence with strong modifications, the influence of the past frames is not as strong that it would distort the prediction. This shows that the linear regression is sufficiently accurate even in the case of large transformations.

For this sequence the optimization also took around 20 seconds.

The last sequence of this section was taken on December 9th 2015 between 13:01:40 and 13:03:34. We represent the first and last frame at figure 4.11.

Here the sun is not entirely covered by the clouds, but as explained in section 3.1, this region is taken out of the optimization and the result that we see in figure 4.12 around that area results from the smoothing. This sequence is an example of why it difficult to predict the power input accurately from a segmentation of the sun region. As we can see in the last frame (figure 4.11b), the sun light is still somewhat perceivable although the area coverage is important. However, the power input could be completely null depending on the position of the PV cells.

Let us now focus on the other regions of the segmentation. Again, we have an important topological change here where a cloud is detaching itself from

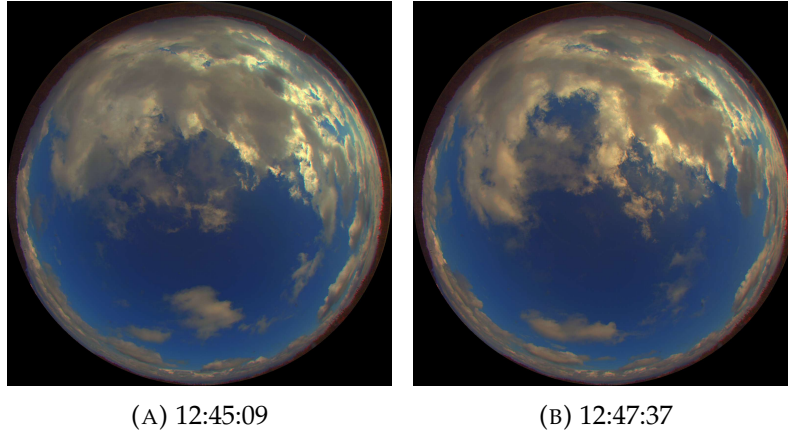


FIGURE 4.9: First and last images of a the sequence taken on October 19th 2015

the main group at the top of the image. This is well reflected in the movement matrix where the blue color dominates the space between them, and consequently this trend continues in the prediction.

As before, the time to perform the optimization on this sequence was around 20 seconds.

So far, the performance of the model has proved to be satisfyingly accurate and robust. In the sequences that we have seen, the movement of the clouds was dominated by shape and size variations. Some of them did show a lateral displacement, but it was slow enough to be accurately captured by our model.

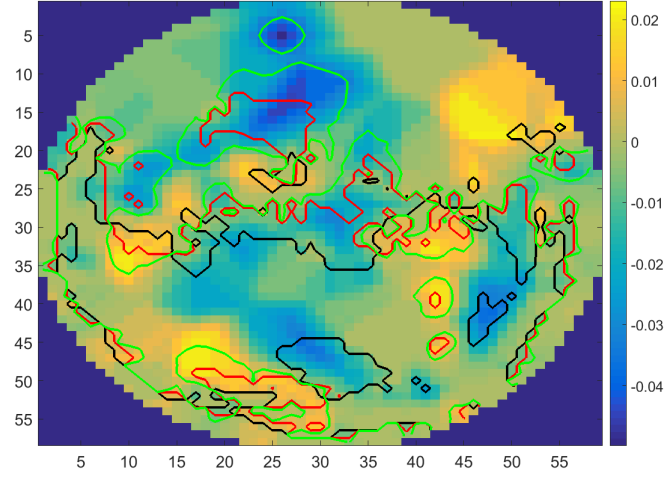


FIGURE 4.10: Segmentations of the first frame (black) and last frame (red) and predicted segmentation (green) of the sky image for the sequence taken on October 19th 2015. The colors in the background show the movement matrix m , where the positive values indicate an increase and the negative values a decrease of the intensity levels.

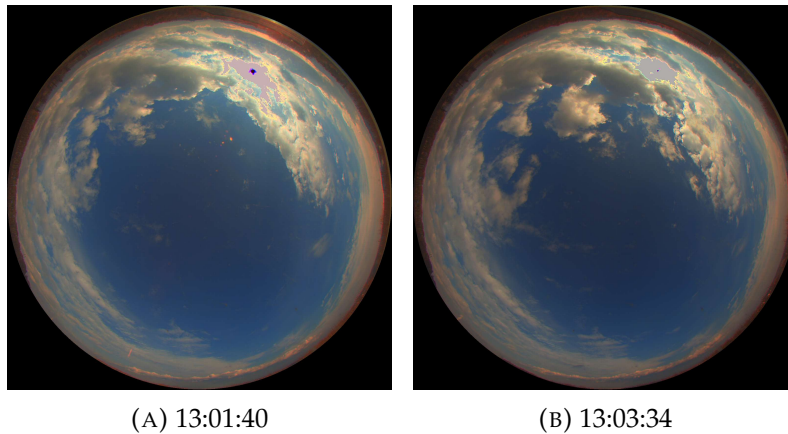


FIGURE 4.11: First and last images of a the sequence taken on December 9th 2015

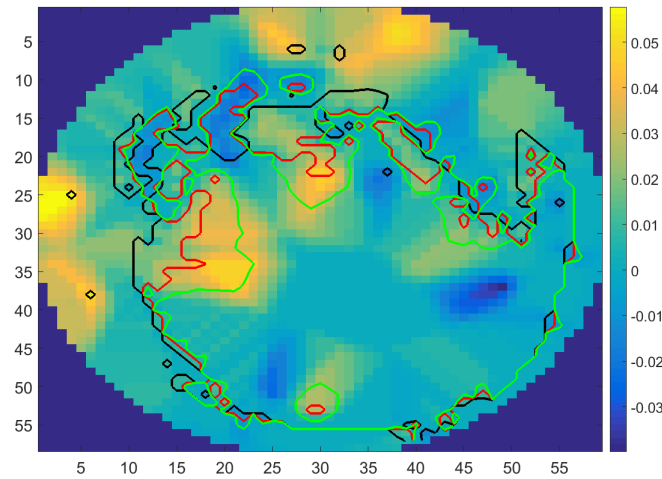


FIGURE 4.12: Segmentations of the first frame (black) and last frame (red) and predicted segmentation (green) of the sky image for the sequence taken on December 9th 2015. The colors in the background show the movement matrix m , where the positive values indicate an increase and the negative values a decrease of the intensity levels.

4.2.2 Sequence with lateral shift and topological changes

For this section, we selected a sequence taken during a strong wind, resulting in a rapid shift along with topological changes. It was taken from the same site in Caviglia on July 29th between 11:40:29 and 11:41:38. The input sequence contains 4 pre-segmented images of size 58×59 . In figure 4.13 we can see the first and last images of this sequence.

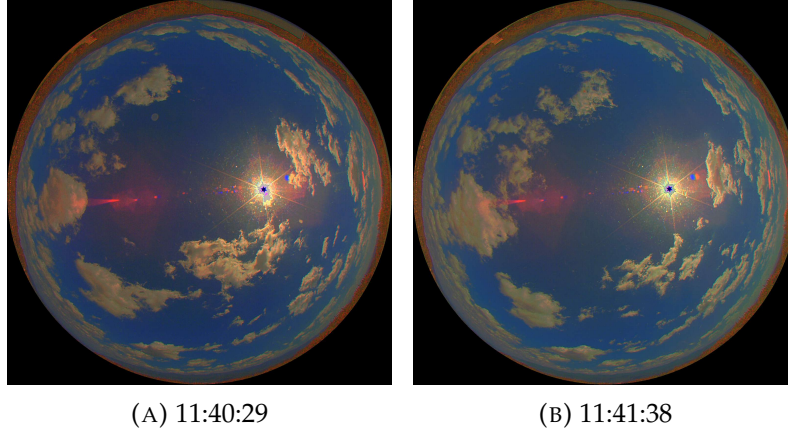


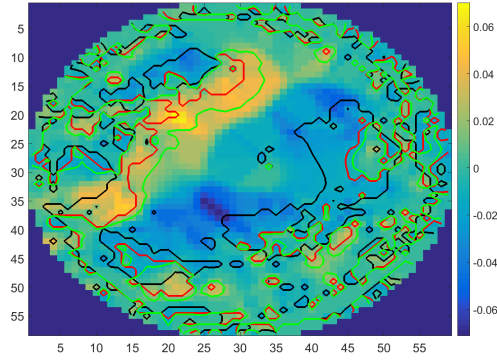
FIGURE 4.13: First and last images of a the second sequence taken on July 29th 2015

In figure 4.14a, we see the results of the optimization if we only use the AC-model. As before, we see that the prediction tends to grow the contour in the yellow areas where the intensities have increased, and to shrink at the blue areas where the intensities have increased. The problem is that these yellow areas are not necessarily due to the growth but to the shift of a cloud. This confusion causes here a general tendency to grow and merge the predicted contours located around these yellow areas, even if they were originally splitting. This is a case where the strategy 2 explained in section 3.5 can be tested. We start by eliminating the shift (figure 4.15) before we apply our optimization. The result is shown at figure 4.14b.

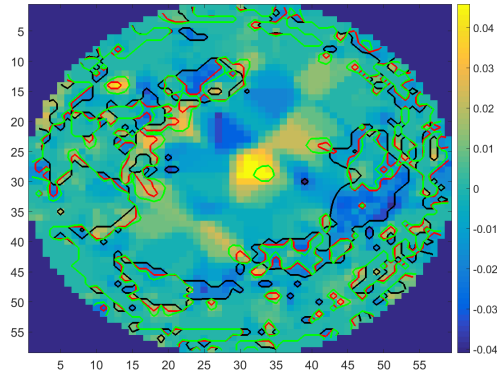
First we notice that the segmentation of the first frame (in black) has radically changed as we have applied the shift to the full sequence. The first and last frame (in red) seem more tightly packed, and now the movement matrix corresponds to the difference between these two superimposed contours. We omitted the last shift (line 8 of algorithm 1) for a better comparison between the last segmentation and the prediction of the AC-model. The prediction is more correct than in the previous result (figure 4.14a) with respect to the real sequence as the large propagating front in the upper left area has disappeared. However, a new growth area has appeared near the sun region, possibly due to noise accumulation.

For this sequence, the optimization converged in ≈ 27 seconds.

It is worth noting that such a case is relatively rare in the data collected in Caviglia, and that the predictions of the AC-model alone are usually accurate for most of the sequences. However, this case demonstrates the necessity of adding an additional dimension to our program.



(A) Strategy 1



(B) Strategy 2

FIGURE 4.14: Segmentations (black and red) and predicted segmentation (green) of the sky image for the sequence taken on July 29th 2015

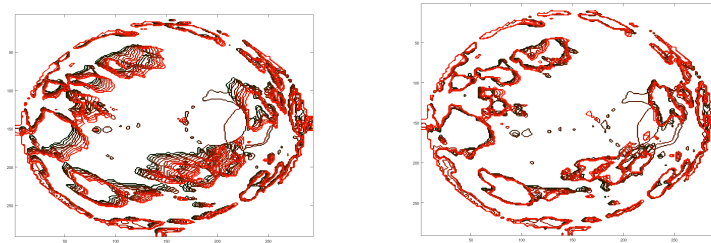


FIGURE 4.15: Example of shift elimination (with additional frames). Contours of a sequence before (left) and after (right) applying the shift. The color of the contour is dark brown for the first image of the sequence and goes to bright red as the time increases.

Chapter 5

Discussion and future work

In this thesis, we created a new model relying on a variational approach of the level set method in order to segment a sequence of sky images and to perform a short term prediction of the cloud coverage. The segmentation is performed in a similar way to the Chan-Vese model, however, we adapted the energy functional for a better detection and prediction of the object size variation by saturating the conformation term of the cost function. Moreover, our model is capable of segmenting the full sequence in a single optimization using a motion matrix to link each frame to the next. By doing so, we force the subsequent frames' segmentations to be similar to one another which reduces the effect of noise. This matrix is also used to make a short term prediction based on the variation of the intensity levels during the sequence.

The tests that we performed on both synthetic and real sky images showed very satisfying results regarding the segmentation, the description of the motion and prediction when the variation of the clouds consisted mainly in topological changes. The motion matrix that we obtained showed positive values in areas where the clouds were growing which was due to the level increase of the signed distance function to the clouds borders. This was translated into an expansion of the predicted contour with respect to the last frame of the sequence. Similarly, the areas where clouds were shrinking or moving away resulted into a decrease of the signed distance function, negative values in the motion matrix, and in the prediction pulling away from the contour of the last frame. In the cases where a cloud was showing both types of deformations simultaneously, this was correctly reflected in the prediction as well because the model considers individual pixels rather than clouds as single objects. This reinforces our choice of geodesic active contours over a parametric representation.

However, this representation based on pixel intensities has its shortcomings. Although the model is very good at tracking topological variation of the clouds, which is what it was designed for, some problems appear when the sequence shows a preponderant lateral movement due to a strong wind. To counter that, we combined our active contours model to an optical flow method which computes the lateral shift separately. However, this further slows down the full computation because it requires to solve two optimization problems separately. Another option would be to compute the lateral component directly from our segmented sequence by subtracting the vertical variation from the normal velocity field. In this way we would fully use

the information captured by our model and make the algorithm self contained.

In general, we observed that the convergence of the optimization is rather slow. It took around twenty seconds for sequences of only three images which were largely downscaled. In the context of the project, we need the program to execute in real-time or nearly. This may not only increase the accuracy of the predictions by producing higher precision cloud maps, but it could also allow to use longer sequences which would reduce the noise produced by singularities in the data such as in figure 4.14b. One option to speed up the calculations, would be to use a parallel approach with block processing. This can be easily implemented for GPUs which are available on any modern computer. Another option to reduce the noise would be to replace the linear regression in the motion fit component by another model which eliminates the outliers.

Chapter 6

Conclusion

Solar power generation is subject to frequent fluctuations in climates with a high precipitation and cloudiness rate. Developing short term prediction systems for the cloud cover estimation is paramount for an optimal management of the next generation electrical grid. Many researches took steps to accomplish this using sky images collected by weatherproof fish-eye cameras. The distinction between a sky pixel and a cloud pixel is in itself a major challenge. In this thesis we build on the prior acquisition system and denoising techniques developed within ABB's research project on sky imaging in order to develop a new model for the cloud tracking and motion prediction.

This thesis is mainly concerned with the motion tracking and prediction of cloud coverage based on pre-segmented sky images. Previous models relying on a parametric representation of individual clouds have proven lacking in the often occurring cases of clouds merger, split and shape and size variation. Geodesic active contours are a popular method in computer vision for the detection and tracking of non-rigid object in sequences of images. The vast literature which boomed during the two last decades provides numerous variants aiming at increasing the accuracy, accelerating the performance or adapting to various types of input.

The variational formulation of the level set method poses the image segmentation as an optimization problem. It represents the edges of the an object as the intersection between a 3D surface and the plane $z = 0$. The surface which is often a signed distance function to the borders of the object contours can be determined by minimizing an energy function which depends on the image intensities. The Chan-Vese model is a particular variant of the level set method which does not rely on edge descriptors, making it more robust to noise than other methods.

In our model, we devised a new energy functional inspired by the Chan-Vese model, where an entire sequence of images is segmented simultaneously, and where each frame segmentation is related to the previous one using a linear regression model. The optimization also outputs a motion matrix which describes the transition between the frames and allows us to produce a prediction in the future based on the general motion in the sequence.

The tests performed on both synthetic and real sky images showed a correct segmentation and description of the motion, as well as a prediction coherent with the past. The results are particularly consistent in case of topological variation of the contour shapes, owing to the active contours

representation.

A major concern for our model is the case of predominantly lateral movement of the clouds, which may occur in case of strong winds. We have successfully coupled our program with an optical flow algorithm in order to account for this lateral shift.

The research on cloud tracking still continues within ABB in order to improve the image segmentation and motion prediction, and produce short term estimation of the PV plant power input.

Bibliography

- [1] David Adalsteinsson and James A Sethian. "A fast level set method for propagating interfaces". In: *Journal of computational physics* 118.2 (1995), pp. 269–277.
- [2] A. Blake and M. Isard. *Active contours: the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion*. Springer Science & Business Media, 2012.
- [3] V Caselles and B Coll. "Snakes in movement". In: *SIAM Journal on Numerical Analysis* 33.6 (1996), pp. 2445–2456.
- [4] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. "Geodesic active contours". In: *Computer Vision, 1995. Proceedings., Fifth International Conference on*. IEEE. 1995, pp. 694–699.
- [5] Tony F Chan, Luminita Vese, et al. "Active contours without edges". In: *Image processing, IEEE transactions on* 10.2 (2001), pp. 266–277.
- [6] R Chauvin et al. "Cloud Detection Methodology Based on a Sky-imaging System". In: *Energy Procedia* 69 (2015), pp. 1970–1980.
- [7] John A Clark. "The thermodynamics of bubbles". In: *Cambridge, Mass. Massachusetts Institute of Technology, Division of Industrial Cooperation, [1956]* 15186908.7 (1956).
- [8] Daniel Cremers, Mikael Rousson, and Rachid Deriche. "A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape". In: *International journal of computer vision* 72.2 (2007), pp. 195–215.
- [9] Paul E Debevec and Jitendra Malik. "Recovering high dynamic range radiance maps from photographs". In: *ACM SIGGRAPH 2008 classes*. ACM. 2008, p. 31.
- [10] Ricard Delgado-Gonzalo et al. "Snakes on a Plane: A perfect snap for bioimage analysis". In: *Signal Processing Magazine, IEEE* 32.1 (2015), pp. 41–48.
- [11] Ricard Delgado-Gonzalo et al. "Snakes with an ellipse-reproducing property". In: *Image Processing, IEEE Transactions on* 21.3 (2012), pp. 1258–1271.
- [12] Subrahmanyam Gorthi et al. "Active deformation fields: Dense deformation field estimation for atlas-based segmentation using the active contour framework". In: *Medical Image Analysis* 15.6 (2011), pp. 787–800.
- [13] M. Isard and A. Blake. "CONDENSATION—Conditional Density Propagation for Visual Tracking". In: *International Journal of Computer Vision* 29 (1998), pp. 5–28.

- [14] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. "Snakes: Active contour models". In: *International journal of computer vision* 1.4 (1988), pp. 321–331.
- [15] Frederic Leymarie and Martin D Levine. "Tracking deformable objects in the plane using an active contour model". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 15.6 (1993), pp. 617–634.
- [16] David Mumford and Jayant Shah. "Optimal approximations by piecewise smooth functions and associated variational problems". In: *Communications on pure and applied mathematics* 42.5 (1989), pp. 577–685.
- [17] Stanley Osher and Ronald P Fedkiw. "Level set methods: an overview and some recent results". In: *Journal of Computational physics* 169.2 (2001), pp. 463–502.
- [18] Stanley Osher and James A Sethian. "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations". In: *Journal of computational physics* 79.1 (1988), pp. 12–49.
- [19] Nikos Paragios and Rachid Deriche. "Geodesic active contours and level sets for the detection and tracking of moving objects". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.3 (2000), pp. 266–280.
- [20] Nikos Paragios and George Tziritas. "Adaptive detection and localization of moving objects in image sequences". In: *Signal Processing: Image Communication* 14.4 (1999), pp. 277–296.
- [21] G Pfister et al. "Cloud coverage based on all-sky imaging and its impact on surface solar irradiance". In: *Journal of Applied Meteorology* 42.10 (2003), pp. 1421–1434.
- [22] Yogesh Rathi and Allen Tannenbaum. "Tracking deforming objects using particle filter for geometric active contours". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.8 (Aug. 2007).
- [23] J. A. Sethian. "A fast marching level set method for monotonically advancing fronts". In: *Proceedings of the National Academy of Sciences* 93 (Feb. 1996), pp. 1591–1595.
- [24] James A Sethian et al. "Level set methods and fast marching methods". In: *Journal of Computing and Information Technology* 11.1 (2003), pp. 1–2.
- [25] Janet E Shields et al. "Day/night whole sky imagers for 24-h cloud and sky assessment: history and overview". In: *Applied optics* 52.8 (2013), pp. 1605–1616.
- [26] John N Tsitsiklis. "Efficient algorithms for globally optimal trajectories". In: *Automatic Control, IEEE Transactions on* 40.9 (1995), pp. 1528–1538.
- [27] B.C. Vemuri et al. "Image registration via level-set motion: Applications to atlas-based segmentation". In: *Medical Image Analysis* 7 (2003), pp. 1–20.
- [28] A. Wächter and L. T. Biegler. "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Non-linear Programming". In: *Mathematical Programming* 106.1 (2006), pp. 25–57.

- [29] A. Wächter and L. T. Biegler. *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT*. [Online; Retrieved on January 27, 2016]. COIN-OR (COmputational INfrastructure for Operations Research). URL: <http://www.coin-or.org/Ipopt/documentation/>.
- [30] *A Variational Framework for Joint Segmentation and Registration*. Georgia Institute of Technology Institute of Electrical and Electronics Engineers, 2001.
- [31] Burak Zeydan. “Real Time Cloud Tracking via Sky Imagery for Power Output Prediction of Solar Plants”. MA thesis. Ecole Polytechnique Fédérale de Lausanne, 2014.
- [32] Hong-Kai Zhao et al. “A variational level set approach to multiphase motion”. In: *Journal of computational physics* 127.1 (1996), pp. 179–195.
- [33] S. C Zhu and A. Yuille. “Region Competition : Unifying Snakes, Region Growing, and Bayes/MDL for Multiband Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18.9 (Sept. 1996).

Declaration of Authorship

I, Rim Miryam CHAABOUNI, declare that this thesis titled, “Sky Images Segmentation and Motion Prediction via Geodesic Active Contours” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:
