Grid Smoothing for Image Enhancement

Guillaume Noel, Karim Djouani, and Yskandar Hamam

French South African Institute of Technology,
Tshwane University of Technology, Pretoria, South Africa

Abstract. The present paper focuses on sharpness enhancement and noise removal in two dimensional gray scale images. In the grid smoothing approach, the image is represented by a graph in which the nodes represent the pixels and the edges reflect the connectivity. A cost function is defined using the spatial coordinates of the nodes and the gray levels present in the image. The minimisation of the cost function leads to new spatial coordinates for each node. Using an adequate cost function, the grid is compressed in the regions with large gradient values and relaxed in the other regions. The result is a grid which fits accurately the objects in the image. In the presented framework, the noise in the initial image is removed using a mesh smoothing approach. The edges are then enhanced using the grid smoothing. If the level of noise is low, the grid smoothing is applied directly to the image. The mathematical framework of the method is introduced in the paper. The processing chain is tested on natural images.

Keywords: Grid smoothing, Sharpness enhancement, Mesh smoothing, Non-linear optimisation, Graph-based image.

1 Introduction

Image enhancement techniques seek at improving the appearance of an image without referring to a specific model for the degradation process, while *image restauration* relies on the knowledge of a degradation model [1]. The framework presented in the paper belongs to the *image enhancement* domain. The two main forms of image quality degradation are blur (loss of sharpness) and noise. Methods have been developed and address both types of degradation, either in a pixel-representation of the image or a mesh-representation. The pixel-representation considers an image to be a matrix of pixels. The edge enhancement methods modify the gray level of the pixels to improve the quality of the image. In a mesh representation, the image is represented by nodes (or vertices) and edges. Using the pixel-representation of an image, adaptive bilateral filters and quadratic-weighted median filters were applied with success for edge enhancement [1],[2]. These methods are based on filtering and may induce overshooting of the edges. Moreover, the parameters of the filters, in the case of the adaptive bilateral filter, are tuned according to a training dataset, narrowing the scalability to various applications. In any case, the performance of these methods is bounded by the use of the pixel-representation of the image. In low resolution images, the shape

T.-h. Kim et al. (Eds.): FGIT 2010, LNCS 6485, pp. 125–135, 2010.

of the object does not systematically match the matrix and may lead to severe distortion of the original shape. For example, a clear straight line whose orientation is 45 degrees is represented by a staircase-like line. Image enhancement techniques, such as super-resolution ([3]) tackle the issue of the misrepresentation of an image. However, the square (even with a smaller size than the original pixel) is used as the construction brick of the image leading to the same misrepresentation of a shape ([3],[4] and [5]). The paradigm image and matrix may be overcome by the use of a mesh or graph-based representation of an image. The mesh defined on the image may be feature-sensitive or insensitive. In the feature-insensitive mesh, the definition of the mesh does not take into account the gray levels. The edge enhancement relies on the filtering of the gray levels in the image [6] which may lead to the same issues as in the pixel-representation (overshooting,...). In the feature-sensitive mesh, the graph is defined according to the distribution of the gray levels in the image and the position of the mesh is adjusted to the objects present in the image to enhance the image [7]. The usual problems of this class of techniques are the computation cost (interpolation) and the large number of edges in the graph. Mesh creation techniques may be found in [8], [9] and [10]. The present paper presents a novel combination of feature-insensitive and feature-sensitive mesh approaches, which reduces the complexity of the definition of the mesh and improves the representation of the information in the image. In the presented framework, an image is represented by a graph in which the nodes represent the pixels and the edges reflect the connectivity. The original graph (or grid) is a uniform grid composed by squares or triangles, depending on the connectivity chosen. The grid smoothing process modifies the coordinates of the nodes in the *(x,y)* plane while keeping the gray scale levels associated to the node unchanged. The grid smoothing relies on the minimisation of a cost function leading to a compression of the grid in the regions with large gradient values and a relaxation in the other regions. As a consequence, the new grid fits the objects in the image. The grid smoothing enhances the edges in the original image and does not modify the number of nodes. Noise removal techniques may be applied before the grid smoothing, depending on the properties of the original images. A new type of mesh smoothing is being used [11]. The mesh smoothing approach in [11] modifies the gray levels of the image while preserving the *(x,y)* coordinates of the nodes.

Section 2 of this paper presents the graph-based representation of an image while section 3 exposes the mathematical framework of the grid smoothing as well as the convergence properties. Noise removal using mesh smoothing is presented in section 4 and adapted to the grid smoothing context. Simulation results and examples of grid smoothing on real images may be found in Section 5. Conclusion and recommendations are underlined in section 6.

2 Graph-Based Image Representation

Our input data is a graph $G = (V, E)$, embedded in the 3D Euclidian space. Each edge e in E is an ordered pair (s, r) of vertices, where s (resp. r) is the

sending (resp. receiving) end vertex of e [11]. To each vertex v is associated a triplet of real coordinates x_v, y_v, z_v. Let C_{ve} be the node-edge incidence matrix of the graph G, defined as:

$$C_{ve} = \begin{cases} 1 \text{ if } v \text{ is the sending end of edge } e \\ -1 \text{ if } v \text{ is the receiving end of edge } e \\ 0 \text{ otherwise} \end{cases} \quad (1)$$

In the rest of the paper, the node-edge matrix C_{ve} will also be denoted C.

Table 1. Number of connections according to the connectivity

Number of points	Number of connections Connectivity 4	Number of connections Connectivity 8
2500	4900	9702
10000	19800	39402
90000	179400	358202

Table 2. Computation time according to the connectivity ($\theta = 0.005$)

	Computation time (s)			
	Connectivity 4		Connectivity 8	
	Per edge	Image	Per edge	Image
2500 points	1.2×10^{-5}	5.9×10^{-2}	1.5×10^{-5}	1.5×10^{-1}
10000 points	1.6×10^{-5}	3.3×10^{-1}	2.2×10^{-5}	8.9×10^{-1}
90000 points	2.8×10^{-5}	5.0	3.3×10^{-5}	1.2×10^{1}

Considering an image with M pixels, X, Y and Z repectively represent $[x_1, ..., x_M]^t$, $[y_1, ..., y_M]^t$ and $[z_1, ..., z_M]^t$. X and Y are at first uniformely distributed (coordinates of the pixels in the plane), while Z represents the gray level of the pixels. Each pixel in the image is numbered according to its column and then its rows. We define L as the number of edges in the graph. C is consequently a matrice with L rows and M columns.

The number of edges depends on the choice of the connectivity for the pixel. If the connectivity is equal to four, each pixel is connected to its four closest pixels. The initial grid is then composed of squares. If the connectivity is equal to eight, each pixel is connected to its eight closest pixels. The initial grid is then composed of triangles. The choice of the connectivity is important as it increases the size of the matrix C and consequently the computation time required for the grid smoothing (Table 1,2). An evaluation of L may be derived using the dimensions of the image. If L_x and L_y represent respectively the number of pixels along the x axis and y axis, we have $M = L_x \times L_y$. For a connectivity equals to 4, $L = 2M - L_x - L_y$ and $L = 4M - 3L_x - 2L_y - 2$ if the connectivity equals 8. Using the notation introduced before, it may be observed that the complexity

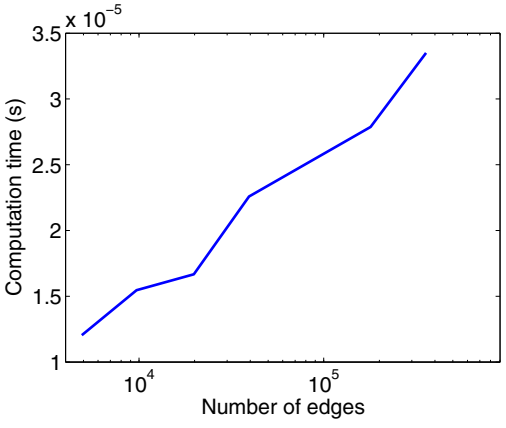
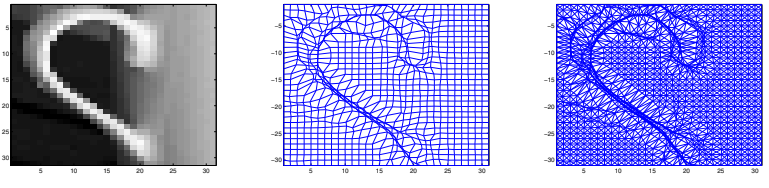


Fig. 1. Computation time per edge in seconds

of the algorithm is $L \times log(L)$(Fig.1). It may be explained by the complexity of the conjugate gradient with a stopping criterion ϵ in which the maximal number of iteration is bounded by $\alpha log(L/\epsilon)$, α being a constant. When using the high connectivity, the number of connections doubles as well as the computation time. The choice should be made according to the applications and the characteristics of the images. If an image includes thin lines which have to be preserved, the high connectivity should be used. For the other cases, the low connectivity gives satisfactory results.



(a) Original image (detail) (b) Grid (connectivity 4) (c) Grid (connectivity 8)

Fig. 2. Result of the grid smoothing according to the connectivity

3 Optimisation-Based Approach to Grid Smoothing

The present section introduces the framework for the grid smoothing. An extensive study of the convergence of the method as well as its application to satellite images may be found in [12], [14].

3.1 General Framework

A cost function is introduced to fit the content of the image with the grid. The main idea is that the regions where the variance is small (low gradient) require less points than the regions with a large variance (large gradient). The grid smoothing techniques will move the points of the grid from small variance regions to large variance regions. To achieve this goal, a cost function J is defined as follows:

$$J = J_X + J_Y \tag{2}$$

where

$$J_X = \frac{1}{2}\left[\left(X - \hat{X}\right)^t Q\left(X - \hat{X}\right) + \theta\left(X^t A X\right)\right] \tag{3}$$

and

$$J_Y = \frac{1}{2}\left[\left(Y - \hat{Y}\right)^t Q\left(Y - \hat{Y}\right) + \theta\left(Y^t A Y\right)\right] \tag{4}$$

with $A = C^t \Omega C$ and $\hat{X}$ (resp. $\hat{Y}$) represents the initial coordinates of X (resp. Y). Ω is a diagonal matrix. The first term in the expression of the cost function is called the *attachment* as it penalises the value of the cost function if the coordinates are too far from the original values. It is introduced to avoid large movement in the grid [11]. θ is a real number and acts as a weighing factor between the terms of the cost function. The matrix Ω is defined as follows:

$$\Omega_{k,k} = (z_i - z_j)^2 \tag{5}$$

where node i is the sending end of the vertex k and node j the receiving end. Ω and Q are square diagonal matrices which dimensions are respectively $L \times L$ and $M \times M$.

As a result of the definition of Ω, the minimisation of J leads to the reduction of the areas of the triangle formed by two connected points and the projection of one of the point on the Z-axis. The edges in the image act as attractors for the points in the grid. As a consequence, the edges are better defined in terms of location and steepness in the smoothed grid.

3.2 Convergence of the Cost Function with Fixed Points and Attachment

The cost function with attachment results in a grid whose size might differ from the original grid size. A solution to conserve the original size is to fix the coordinates of the outer points of the grid. Let the X coordinates be partitioned into two parts, variable coordinates 'x' and fixed coordinates 'a' giving

$$X = \begin{bmatrix} x \\ a \end{bmatrix} \tag{6}$$

Then the first order cost function without attachment is

$$J_x = \frac{1}{2}\left(\left[\,(x - \hat{x})^t \; 0\,\right] Q \begin{bmatrix} (x - \hat{x}) \\ 0 \end{bmatrix} + \theta\left[\,x^t \; a^t\,\right] \begin{bmatrix} C_x^t \\ C_a^t \end{bmatrix} \Omega \left[\,C_x \; C_a\,\right] \begin{bmatrix} x \\ a \end{bmatrix}\right) \tag{7}$$

Expanding the above equation gives

$$J_x = \frac{1}{2}\left[(x-\hat{x})^t\, Q_x\,(x-\hat{x}) + \theta x^t C_x^t \Omega C_x x + 2\theta x^t C_x^t \Omega C_a a + \theta a^t C_a^t \Omega C_a a\right] \quad (8)$$

The gradient of J_x with respect to x is

$$\nabla_x J_x = Q_x\,(x-\hat{x}) + \theta C_x^t \Omega C_x x + \theta C_x^t \Omega C_a a \quad (9)$$

Setting the gradient to zero gives

$$x = -\left[Q_x + \theta C_x^t \Omega C_x\right]^{-1}\left[Q_x\hat{x} - \theta C_x^t \Omega C_a a\right] \quad (10)$$

This gives the exact solution for the coordinates x.

Let x_{n+1} and x_n be x at iteration $n+1$ and n then

$$x_{n+1} = x_n - \alpha_n \nabla_x J_x \quad (11)$$

The gradient of J_x at the point x_{n+1} is equal to

$$\nabla_x J_{x_{n+1}} = \nabla_{x_n} J_x - \alpha_n Q_x \nabla_x J_{x_n} - \alpha_n \theta C_x^t \Omega C_x \nabla_x J_{x_n} \quad (12)$$

The optimal step condition may by expressed by $\nabla_x J_{x_n}^t . \nabla_x J_{x_{n+1}} = 0$

This leads to:

$$\alpha_n = \frac{\nabla J^t \nabla J}{\nabla J^t\,(Q_x + \theta C_x^t \Omega C_x)\,\nabla J} \quad (13)$$

The experience shows that the convergence is quicker using the conjugate gradient descent with optimal step. A quadratic function may be expressed by:

$$J(x) = \frac{1}{2}x^t A x + b^t x + c \quad (14)$$

where A is a definite positive matrix [13]. At each iteration, $x_{n+1} = x_n - \alpha_n d_n$, where α_n is the step and d_n is the direction of descent. The direction and the step are calculated at each iteration. By assimilation with the cost function with fixed points and attachment, we have $A = Q_x x + \theta C_x^t \Omega C_x$ and $b = \hat{x}^t Q_x - \theta a^t C_a^t \Omega C_x$. The step at the iteration n may be computed by:

$$\alpha_n = \frac{(b - Q_x\hat{x})^t\,(b - Q_x\hat{x})}{d_n^t Q_x d_n} \quad (15)$$

and the direction at iteration $n+1$ is equal to:

$$d_{n+1} = e_{n+1} + \frac{e_{n+1}^t e_{n+1}}{e_n^t e_n} d_n \quad (16)$$

where $e_{n+1} = e_n - \alpha_n Q_x d_n$ and $e_1 = b - Q_x\hat{x}$.

3.3 Stopping Criterion

As mentionned earlier, for large scale problem, the minimisation uses a gradient descent algorithm as it is computationally expensive to inverse very large matrices. Three gradient methods are used for the simulation, namely the steepest descent gradient with fixed step, the steepest descent gradient with optimal step and the conjugate gradient with optimal step. The descent gradient methods are iterative process and require a stopping criterion ϵ to stop the iterations. The chosen criterion is the simulation is the norm of the gradient ∇J. The iterative process continues while $\nabla J^t \nabla J \geq \epsilon$. When it is possible, the comparison between the exact coordinates given by the inversion of the matrix and the result of the gradient descent algorithm is small and is of the order of ϵ. For example, if $\epsilon = 10^{-3}$, the difference between the exact coordinates (matrix inversion) and the coordinates obtained through the gradient descent is 10^{-3} of the width of a pixel. The conjugate gradient descent is faster for any ϵ.

4 Noise Removal Using Mesh Smoothing

As mentionned in the sections above, the noise removal process is unrelated to the edge enhancement method in the grid smoothing framework. As there is no link between the two operations, the connectivity may be chosen differently. The following paragraph presents the noise removal process using a connectivity equal to four. The same operations may easily be derived for another connectivity. If the connectivity is equal to four, the grid on which the image is composed by quadrilaterals. Let's denote the quadrilateral Q_{ijkl}, the direct quadrilateral composed by the points i,j,k and l. Without noise removal, each quadrilateral represents a facet and the color of the facet is chosen equal to the graylevel of one point of the quadrilateral.

In the mesh smoothing framework, the image is represented using the matrix C_{ve} introduced in second section [11].The mesh smoothing techniques rely on the minimisation of a cost function J_Z, Z being the gray levels of the vertices. The result is a new vector $\bar{Z}$ containing the filtered gray levels of the image. The general form of the cost function is:

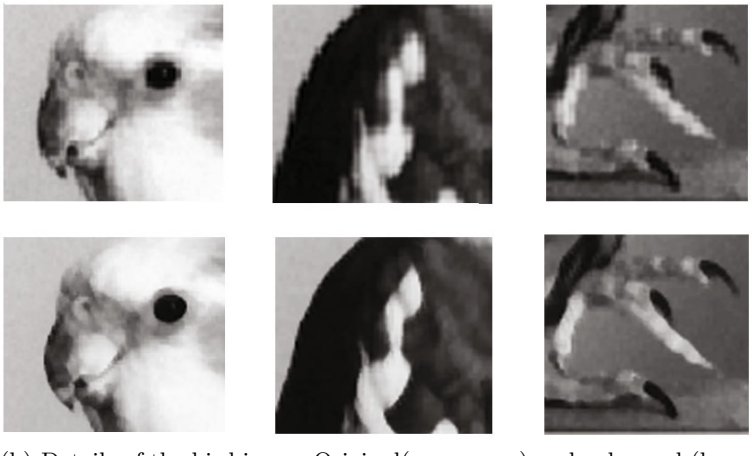
$$J_Z = \frac{1}{2}\left[\left(Z - \hat{Z}\right)^t Q \left(Z - \hat{Z}\right) + \theta_0 Z^t Z + \theta_1 Z^t \bar{A} Z + \theta_2 Z^t \bar{A}^2 Z\right] \tag{17}$$

The cost function is composed of various terms whose respective weigths might be tuned by the values of the θ_i. The first term represents an attachment to the initial coordinates of the point to avoid large variations in graylevels [11]. The minimisation of the other terms of the cost function brings each point to the center of gravity of its neighbourhood. It may be shown that the second order term ($\theta_2 Z^t \bar{A}^2 Z$) smoothes the curve of the objects. For more information and the notations, please refer to [11]. The mesh smoothing used in the simulation takes into acount the attachment term and the second order term. It is refered as SOWA (Second Order With Attach) in [11].

5 Simulations

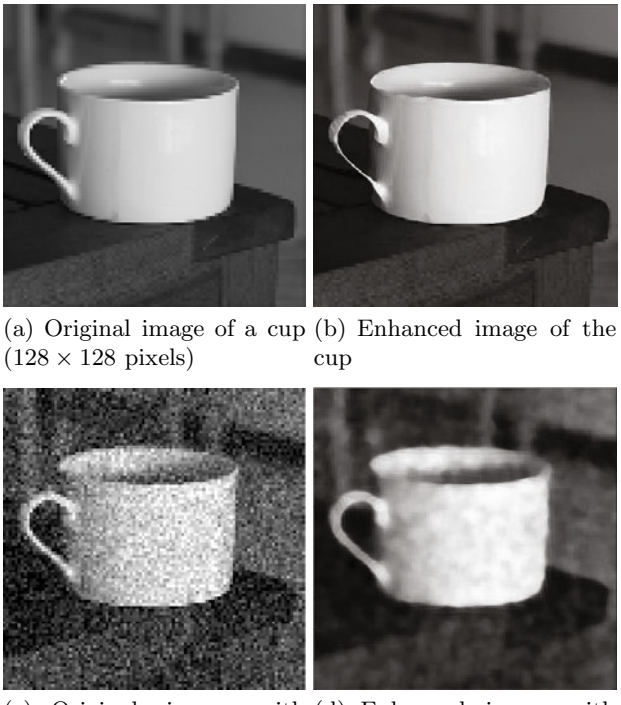
The simulations were performed using a standard laptop (1.87 GHz processor, 2GB RAM and *Windows Vista SP1* as operating system) and *Matlab R14 Service Pack 2*. Figure 2 shows the results of the grid smoothing process on a detail of the original image (Fig. 4(a)). Figure 2(b) shows the results of the grid smoothing with a connectivity is equal to 4 while Figure 2(c) presents the result when the connectivity is equal to 8. The results where obtained with a conjugate gradient descent. The regions with high variations in the graylevels needs more points than the other regions leading to a distorsion of the original grid. The

(a) Original image ($256 pixels$) of a bird (left) and the result of the enhancement (right)



(b) Details of the bird image-Original(upper row) and enhanced (lower row)

Fig. 3.



(a) Original image of a cup (128×128 pixels)

(b) Enhanced image of the cup

(c) Original image with added noise

(d) Enhanced image with $\theta = 0.0005$

Fig. 4. Image restoration with noise removal

distorsions present in the two grids are similar. However, a higher connectivity leads to a more accurate and robust fitting of the shapes.

Figure 3 presents the results of the edge enhancement. The size of the original image (Figure 3(a)) is 256×256 pixels. It may be observed that the level of noise is low but that the edges are not well defined (pixelisation). The enhanced image (Figure 3(a)) exposes a good restauration on the edges while not compromising the quality of the image. The edges are smooth and continuous (the pixels which may be seen are due to the pdf compression of the image and are not present in the original simulation results). The texture of the bird is recovered while the dimension are slightly altered. A closer look at the improvment may be found in the details presented in Figure 3(b). Fig. 4 presents the results on a noisy version of the image(Fig. 4(a)). A white gaussian noise ($\mu = 0$ and $\sigma^2 = 0.1$) is added to the image. To avoid random movements of the points during the grid smoothing process, the image is firstly filtered using the mesh smoothing technique (SOWA). The filtered version of the image is then fed into the edge enhancement process. Fig. 4(d) depicts the results obtained with respectively $\theta = 0.0005$. It may be seen that the global shapes of the cup and the table are recovered in both cases. A greater θ leads to a better sharpening of the image

while introducing distorsion in the shape represented. The choice of θ depends on the applications.

6 Conclusions

In conclusion, a new framework to enhanced images without a model for the degradation is presented in the paper. The method relies on the smoothing of the coordinates of the pixels in the image. The results of the image enhancement are promising. Combined with the mesh smoothing approach, the method performs well on noisy images. The output of the process is not a pixel-representation of the image. It leads to the main drawback of the technique which is the computational cost of the display of the facets. To overcome these limitations, further studies will involve redefinition of the connections in the grid to limit the number of facets. Another direction for future research will be to combine the mesh and the grid smoothing in a single operation, the objectif being to define a single cost function performing the same operations.

References

1. Zhang, B., Allebach, J.P.: Adaptative bilateral filter for sharpness enhancement and noise removal. IEEE Trans. on Image Processing 17(5), 664–678 (2008)
2. Aysal, T.C., Barner, K.E.: Quadratic weighted median filters for edge enhancement of noisy images. IEEE Trans. on Image Processing 13(5), 825–938 (2007)
3. Liyakathunisa Kumar, C.N.R., Ananthashayana, V.K.: Super Resolution Reconstruction of Compressed Low Resolution Images Using Wavelet Lifting Schemes. In: Second International Conference on Computer and Electrical Engineering ICCEE 2009, vol. 2, pp. 629–633 (2009)
4. Caramelo, F.J., Almeida, G., Mendes, L., Ferreira, N.C.: Study of an iterative super-resolution algorithm and its feasibility in high-resolution animal imaging with low-resolution SPECT cameras. In: Nuclear Science Symposium Conference Record NSS 2007, October 26-November 3, vol. 6, pp. 4452–4456. IEEE, Los Alamitos (2007)
5. Toyran, M., Kayran, A.H.: Super resolution image reconstruction from low resolution aliased images. In: IEEE 16th Signal Processing, Communication and Applications Conference, SIU 2008, April 20-22, pp. 1–5 (2008)
6. Wang, C.C.L.: Bilateral recovering of sharp edges on feature-insensitive sampled meshes. IEEE Trans. on Visualization and Computer Graphics 12(4), 629–639 (2006)
7. Xu, D., Adams, M.D.: An improved normal-meshed-based image coder. Can. J. Elect. Comput. Eng. 33(1) (Winter 2008)
8. Feijun, J., Shi, B.E.: The memristive grid outperforms the resistive grid for edge preserving smoothing, Circuit Theory and Design. In: Circuit Theory and Design ECCTD 2009, pp. 181–184 (2009)
9. Shuhui, B., Shiina, T., Yamakawa, M., Takizawa, H.: Adaptive dynamic grid interpolation: A robust, high-performance displacement smoothing filter for myocardial strain imaging. In: IEEE Ultrasonics Symposium, IUS 2008, November 2-5, pp. 753–756 (2008)

10. Huang, C.L., Chao-Yuen Hsu, C.Y.: A new motion compensation method for image sequence coding using hierarchical grid interpolation. IEEE Transactions on Circuits and Systems for Video Technology 4(1), 42–52 (1994)
11. Hamam, Y., Couprie, M.: An Optimisation-Based Approach to Mesh Smoothing: Reformulation and Extension. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 31–41. Springer, Heidelberg (2009)
12. Noel, G., Djouani, K., Hamam, Y.: Optimisation-based Image Grid Smoothing for Sea Surface Temperature Images. In: Advanced Concepts for Intelligent Vision Systems, ACIVS 2010, Sydney, Australia (2010)
13. Fletcher, R., Reeves, C.M.: Function Minimization by Conjugate Gradient. The Computer Journal, British Computer Society (1964)
14. Noel, G., Djouani, K., Hamam, Y.: Grid Smoothing: A graph-based Approach. In: 15th Iberoamerican Congress on Pattern Recognition, CIARP 2010, Sao Paulo, Brasil (2010)