Feature-Based Deformable Image Registration with RANSAC Based Search Correspondence

Thomas Colleu[1], Jian-Kun Shen[2],
Bogdan J. Matuszewski[2], Lik-Kwan Shark[2],
Claude Cariou[1]

ENSSAT
Ecole Nationale Superieure De Sciences Appliquees et de Technologies
6, Rue de Kerampont
BP 80518, 22305 Lannion cedex, FRANCE

Applied Digital Signal & Image Processing ADSIP Research Centre,
Department of Technology, University of Central Lancashire, PR1 2HE, UK
E-mail: {jkshen1; bmatuszewski1; lshark}@uclan.ac.uk

Abstract: The paper presents an algorithm for deformable image registration based on point features extracted from input images using the Harris corner detector. The correspondence between the points extracted from the different images is established using RANdom SAmple Consensus (RANSAC) method with the affine and perspective global transformation used to model the deformations. The initial correspondence is established using an enumerative search with rotation invariant cross-correlation measure. Based on the estimated corresponding set of points the further refinement of the displacement field is achieved through application of the multilevel B-spline technique applied to the overlapping image area. The performance of this method is demonstrated using few pairs of remote sensing images. The method is also evaluated against the global polynomial transformation and the registration results are compared with the results achieved using ENVI software.

1 Introduction

Image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. It geometrically aligns two images: the reference and sensed images. Typical applications of image registration lie in the field of medicine (monitoring tumor growth), computer vision (automatic quality control), and remote sensing (multispectral monitoring, integrating information into geographic information system (GIS)).

The basic methods used for registering two images are now shortly explained. The reader is referred to [**9**] for a comprehensive survey. Two methods are usually considered for image registration algorithms: *intensity-based methods* and *feature-based methods*. The first one looks at the structure of the image (via correlation metrics, fourier properties ...) whereas the second one is based on matching features like lines, curves, points. The use of feature-based methods is recommended if the images contain enough distinctive and easily detectable objects. This is usually the case of applications in remote sensing and computer vision where images contain a lot of details (towns, rivers, roads...). On the other hand, medical images are not so rich in such details and thus area-based methods are usually employed here. In this paper, a feature-based registration is used and tested with remote sensed images. This method usually consists of the following four steps:

- *Feature detection*: Salient and distinctive objects (closed-boundary regions, edges, contours, line intersections, corners, etc.) are manually or, preferably, automatically detected. For further processing, these features can be represented by their point representatives (centres of gravity, line endings, distinctive points), which are called control points in the literature.
- *Feature matching*: In this step, the correspondence between the features detected in the sensed image and those detected in the reference image is established. Various feature descriptors and similarity measures along with spatial relationships among the features are used for that purpose.
- *Transform model estimation*: The type and parameters of the so-called mapping functions are estimated. It must align the sensed image with the reference image. The parameters of the mapping functions are computed by means of the established feature correspondence.
- *Image re-sampling and transformation*: The sensed image is transformed by means of the mapping

functions. Image values at non-integer coordinates are computed by the appropriate interpolation technique.

The type of transform model must be chosen according to the context in which the images where acquired. It can be a global transform model or a local one. The global transform model can be used if the images only differ by a global transformation such as translation, rotation, scale, etc. On the other hand, a local transform model is needed when the images differ locally. A typical example is the use of airborne camera because there are a lot of local deformations in the acquired images due to the movement of the plane. This local transformation is also referred to as deformable registration.

Several methods based on features and deformable registrations have already been studied. A lot of the contributions lie in the field of medical images. Thus, in [**2**], radial basis functions with compact support are used for deformable registration of medical images; In [**7**], a landmark-based registration technique makes use of thin plate splines for medical image registration and biomechanical brain modeling. Fewer publications can be found in the field of remote sensing. Yet, the paper [**1**], written within the national center for geographic information and analysis, presents different image registration techniques for remote sensing applications such as multiquadratics functions, the finite element method, bivariate mapping polynomials, and thin plate spline. Finally, [**8**] describes an image registration method based on splines.

In the algorithm proposed here, the feature detection step is achieved with the well-known Harris corner detector. Then, an initial correspondence is given by enumerative search with rotation invariant cross-correlation. The final correspondence is obtained using RANdom SAmple Consensus (RANSAC) method which, also estimates the deformations with affine or perspective global transformation. Based on the estimated corresponding set of points, the further refinement of the displacement field is achieved through application of a local transformation using multilevel B-spline approximation.

This paper is organized as follows. Section 2 describes the algorithm in details. Section 3 presents three different experiments: It first compares the performances of the affine transformation and the perspective one, then evaluates the method against the global polynomial, and finally compares the results with those achieved using ENVI software. Conclusion and further work are discussed in section 4.

2 Methods

2.1 Feature detection and initial correspondence estimation

In order to find the same objects independently in both images, a detector with some invariant properties is needed. The Harris Corner Detector (see [3]) possesses strong invariance to rotation, illumination variation and image noise. It is based on the local similarity measure of a signal that gives the local changes of the signal thanks to a window shifted by a small amount in different directions. Figure 1 shows examples of shifted windows. On 1a, an edge is detected because changes appear in one direction only. On 1b, a corner is detected because significant changes appear in all directions.

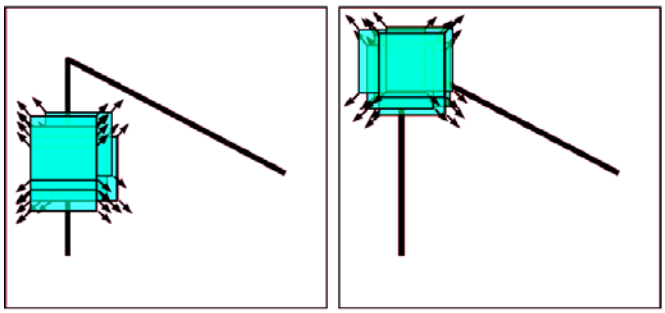

Figure 1: Examples of shifted windows

After the corners have been extracted, the next task is to find which corner in the sensed image corresponds to which corner in the reference image. To do this, circular normalized cross-correlation coefficient is used. The idea is to measure the local intensity similarity between each corner in both images thanks to the computation of the normalized cross-correlation coefficient defined as:

$$NCC = \frac{\sum_{i=1}^{n}(A_i - \bar{A})(B_i - \bar{B})}{\sqrt{\sum_{i=1}^{n}(A_i - \bar{A})^2 \sum_{i=1}^{n}(B_i - \bar{B})^2}}$$

where A and B are the n-by-n window surrounding a corner in reference and sensed images respectively. $\bar{A}$ and $\bar{B}$ are the mean values of A and B. Then, the correspondence between two corners is established only if their correlation coefficient is greater than a fixed threshold. Moreover, in order to improve the quality of match, the correlation is computed in polar coordinates where the angle that gives the maximum correlation is selected. Finally, a faster implementation of the circular correlation is achieved by using the 1-D Discrete Fourier Transform (DFT) during the computation of the coefficient.

The correspondence found is invariant to rotation only, this is a fast method but it doesn't guarantee that all the correspondences are correct.

2.2 Transform model estimation and RANSAC algorithm

At this stage, the wrong correspondences are identified and a global transform model (affine or perspective) is estimated. The use of a global model here is justified by its shape preserving property. Indeed, each image contains an overlapping area, i.e. a region which is common to both images, and a non-overlapping area whose shape must be preserved as well as possible since no information about the displacement is available in this area. Therefore, the global model provides a rough alignment and a good preservation

of the non-overlapping areas. The refinement of the alignment will be achieved with a local transformation in the next stage.

2.2.1 Affine transformation and perspective transformation

An affine transformation is defined by the equation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

In order to find the 6 unknowns, at least 3 data points are necessary. These unknowns are then approximated in the least-squares sense.

A perspective transformation, when defined in homogeneous coordinates, is defined by the equation:

$$\begin{bmatrix} xh \\ yh \\ k \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

The non-homogeneous coordinates are computed as following:

$$x' = \frac{xh}{k} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{yh}{k} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Since the equation is unique up to a scale factor, the problem is reduced to 8 unknowns and, therefore, the minimum number of data points necessary to solve the system of equations is 4. The explanations of how to compute the parameters of a perspective transformation are given in [4].

2.2.2 RANSAC algorithm

As explained in the previous section, the circular cross-correlation method doesn't guarantee that all the correspondences are correct, introducing outliers. A point is considered outlier if it follows a different model than the one estimated. These outliers can severely disturb the estimated transformation, and consequently must be identified.

RANSAC algorithm handles this problem by introducing a classification of the data into inliers (valid points) and outliers while estimating the optimal transformation for the inliers. A threshold *t* is used, it insures that none of the inliers deviates from the model by more than *t*. The procedure, described in [**4**], can be summarized as following:

1. Select as small data subset as possible. Basically, the number of data points necessary to solve the least-squares approximation is used. The subset must be selected randomly among the correspondence points.
2. Calculate the transformation model using the selected subset. Use the least-squares approximation here.
3. Transform all the other correspondence points with the calculated transformation and classify them into inliers and outliers using a threshold *t*.
4. Remember the transformation if the number of inliers is big enough.
5. Repeat this process a specified number of times.
6. When the iterations are finished, re-estimate the model using the inliers. This step provides the optimal model wrt the inliers found previously.
7. Re-classify the inliers as in step 3 with threshold *t*. Indeed, after step 6, additional points may be classified as inliers.
8. Repeat step 6 and 7 until the number of inliers converges.

Since the right number of inliers needed and the number of iterations are usually not known. An adaptive method is used that relieves the user from initialising those two parameters. The reader is referred to [**4**] for more details about this method.

Finally, the important parameter to be initialized is the distance threshold *t* for which a point is declared inlier or outlier. Without knowledge of the distance of the outliers, the only way to determine this parameter is by experimentation.

Figure 2 shows how the threshold is determined during the process. The graph gives the number of inliers as a function of the value of the threshold. As the threshold increases, the number of correspondence points within the threshold distance also increases and so does the number of inliers. One can easily observe a step behaviour on the graph, these steps enhance the fact that a big increase of the threshold is needed before a new inlier is found and it therefore gives the value when outliers begin to be considered as inliers. In conclusion, the user should select the threshold just before the first step appears as shown by the point in the graph.

Obviously, this method is subjective since few outliers could be close to the distance threshold, making the choice harder for the user. In that case, the best choice is to underestimate the number of inliers so that to make sure that no outliers would be present.

2.3 Deformable registration using multilevel B-Splines

As the image is scanned during the flight, it is highly probable that the deformation differs from the start of the scan to the end, resulting in local deformations throughout

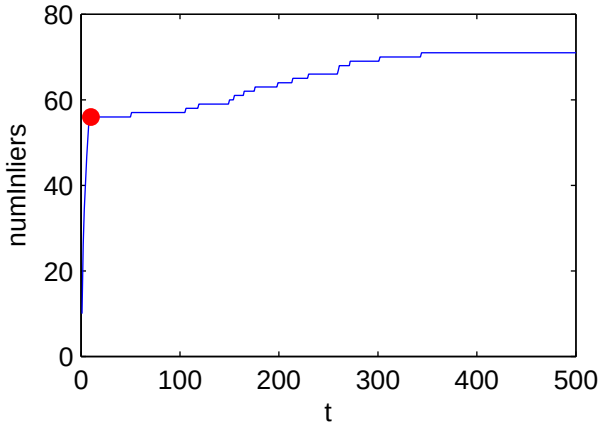

Figure 2: Determination of the threshold *t*.

the image. It is possible to map images locally with the so called B-splines.

2.3.1 Definition

A B-spline function is a sum of piecewise polynomials called B-spline basis functions. Its support is minimal with respect to a given degree. More precisely, a *p*-th degree B-spline curve is defined by:

$$\mathbf{C}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{P}_i \qquad a < u < b$$

where *u* is a parameter that depends on the position *x* of the data point; P_i are the control points, and $N_{i,p}(u)$ are the *p*-th B-spline basis functions defined on the nonperiodic (and in general nonuniform) knot vector

$$U = \underbrace{a, \ldots, a}_{p+1}, u_{p+1}, \ldots, u_{m-p}, \underbrace{b, \ldots, b}_{p+1}$$

Usually, *a*=0 and *b*=1.

2.3.2 Properties

- Local modification: moving P_i changes $\mathbf{C}(u)$ only in the interval $[u_i, u_{i+p+1})$. This follows from the fact that $N_{i,p}(u)=0$ for $u \notin [u_i, u_{i+p+1})$.
- Moving along the curve from u=0 to u=1, the $N_{i,p}(u)$ functions act like switches; as *u* moves past a knot, one $N_{i,p}(u)$ (and hence the corresponding P_i) switches off, and the next one switches on.
- The continuity and differentiability of $\mathbf{C}(u)$ follow from that of the $N_{i,p}(u)$ (since $\mathbf{C}(u)$ is just a linear combination of the $N_{i,p}(u)$). Thus, $\mathbf{C}(u)$ is infinitely differentiable in the interior of knots intervals, and it is at least *p-k* times continuously differentiable at a knot of multiplicity *k*.

2.3.3 Implementation

The paper [5] explains how to approximate scattered data with B-splines. It follows a simple process that, first, fixes the position of the control points P_i as a grid that overlaps all over the image, and then, parameterizes the data points according to their positions in the grid.

The method uses only uniform B-splines of degree 3, and the only user-defined parameter is the size of the grid. Nevertheless, it is sufficient to achieve a good trade-off between shape smoothness and accuracy of the approximation. Indeed, a coarse grid provides a rough approximation but smooth throughout the image, whereas a fine grid approximates better each data points, but is also more local. In the paper, a hierarchy of control grids is used to generate a sequence of B-spline functions whose sum results in a smooth and accurate approximation. This is called multilevel B-spline approximation. In conclusion, the user just has to choose the size of the coarsest and finest grid that will determine the locality of the transformation and its accuracy.

2.4 Polynomial model

A polynomial model has also been tested for the refinement of the displacement field.

2.4.1 Definition

A polynomial involves a sum of powers in one or more variables multiplied by coefficients. A bi-variate polynomial of degree *n* (specified by the user) is used in the program:

$$x' = \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{ij}x^i y^j = a_{00} + a_{01}y + \ldots + a_{0n}y^n + \ldots + a_{n0}x^n$$

$$y' = \sum_{i=0}^{n}\sum_{j=0}^{n-i} b_{ij}x^i y^j = b_{00} + b_{01}y + \ldots + b_{0n}y^n + \ldots + b_{n0}x^n$$

2.4.2 Properties

A polynomial transformation can cope with complex models thanks to its degree and number of unknowns. Moreover, the interesting property is the smoothness because polynomials are infinitely differentiable. But in return, polynomials higher than the first degree may result in unwanted stretching or squeezing due to oscillations between the data points. This is known as the Runge's phenomenon.

2.4.3 Implementation

Before the estimation of the polynomial's parameters, more correspondence points have been added in the non-overlapping area. Since the polynomial transformation is a global one, these points aim at constraining the transformation in the region where no information about the displacement is available. In the following, these points will be called "fixing points". The polynomial model is then estimated in the least-squares sense.

3 Results

Two sets of images have been used for testing the algorithm: *Caton* set and *Garstang* set. The evaluation of the registration's quality is achieved with the root mean square (RMS) of the alignment error between the correspondence points, and with the observation of the registered image in RGB color.space where the reference image is in red and the transformed sensed image is in green.

3.1 Affine vs perspective

This section aims at comparing affine transformation and perspective transformation in order to determine which is the most appropriate for the global mapping. The most important criterion at this stage is that the right number of inliers is reached. Indeed, the alignment error for those inliers would be reduced in the following process. Thus, the threshold *t* is determined such that the right number of inliers is obtained, and then, the quality of the transformation can be evaluated with the alignment error.

Tables 1 and 2 allow comparing both transformations for the images *Caton* and *Garstang* respectively. These two tables show that the performances of affine and perspective transformation are very similar. This observation suggests that the right global transformation model is probably close to affine. Another interesting result is the difference of the alignment error between the two sets. This is due to a difference of the deformation between these sets. Indeed, the local deformations are much stronger in *Garstang* set than in *Caton* set.

Caton set	Affine	Perspective
numInliers	56	56
t	10	10
alignError	3.87	3.32

Table 1:Caton images, comparison of the performances of affine transformation and perspective transformation.

Garstang set	Affine	Perspective	Perspective
numInliers	95	95	90
t	26	31	26
alignError	13.27	13.60	12.53

Table 2: Garstang images, comparison of the performances of Affine transformation and perspective transformation.

Figure 3 shows a part of the image Garstang registered with perspective transformation. The correspondence points emphasize the misalignment, the black points are the reference ones and the white points are the sensed which have to be transformed (in the color version of the paper, the black points are blue and the white points are yellow). Clearly, the points don't overlap. This region will have to be corrected with the deformable registration using B-splines. Although the difference between the two transformations is very small, perspective transformation is used in the algorithm because it theoretically can model more complex deformations than the affine transformation.

3.2 Perspective + polynomial vs perspective + B-splines

This part aims at correcting local misalignment that the previous transformation couldn't reduce. The method that uses polynomial transformation with "Fixing points" in the non-overlapping areas is compared with B-spline approximation. A first remark is that the method using polynomial is not easy to apply because of its big distortions in the non-overlapping areas if the degree of the polynomial is too high. Hence, for each new image, the user must determine the right polynomial degree. This distortion problem doesn't occur with B-spline approximation because the correction is only local. Table 3 gives the results. In any case the alignment error is much smaller than the previous results, but the quality of B-spline approximation appears clearly superior for *Garstang* images. Indeed, the B-splines have reduced the error by 93%, whereas the polynomial transformation has reduced the error by 1.6% only.

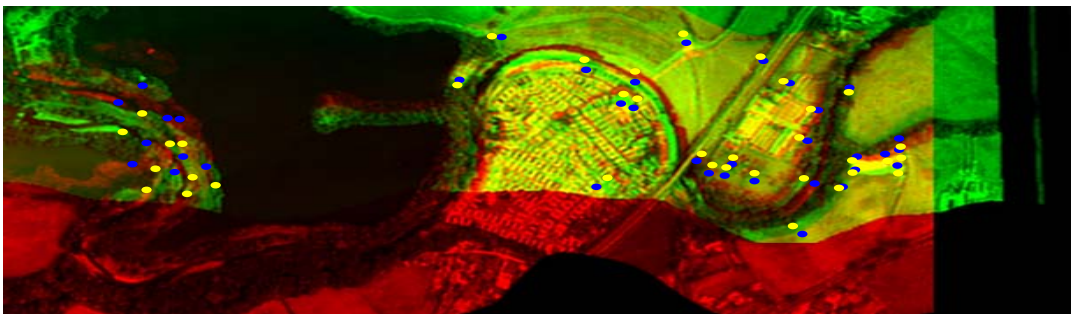

Figure 3: Perspective transformation applied on Garstang set of images.

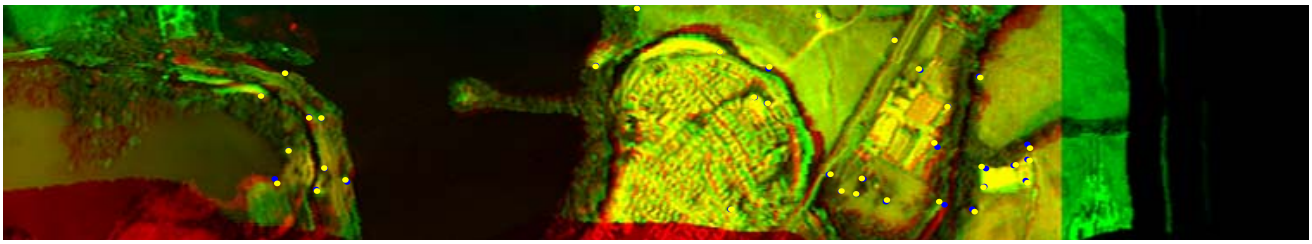

Figure 4: Deformable registration applied on *Garstang* set of images

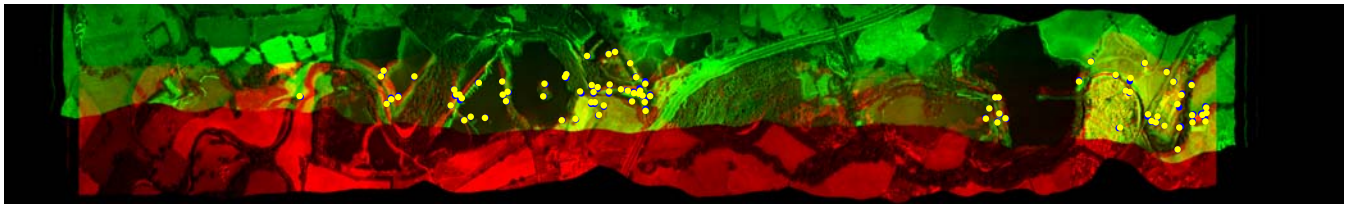

Figure 5: Deformable registration applied on the full images

	Polynomial	B-splines
Caton alignerror	1.30	0.55
Garstang alignerror	10.86	1.45

Table 3: Comparison of polynomial transformation and B-spline approximation.

Figure 4 shows the resulting image after registration with the B-splines. The correspondence points now overlap, and much less misalignment is visible which means that the local correction has been efficient in this region of the image. However, figure 5 shows that the correspondence points are not regularly spread along the images and therefore, the local correction is limited to small regions only.

5.3 Comparison with ENVI Software

ENVI (*Environment for Visualizing Images*) is an image processing system specialized for satellite and aircraft remote sensing data. A registration method is available; it warps the image with a polynomial transformation of a

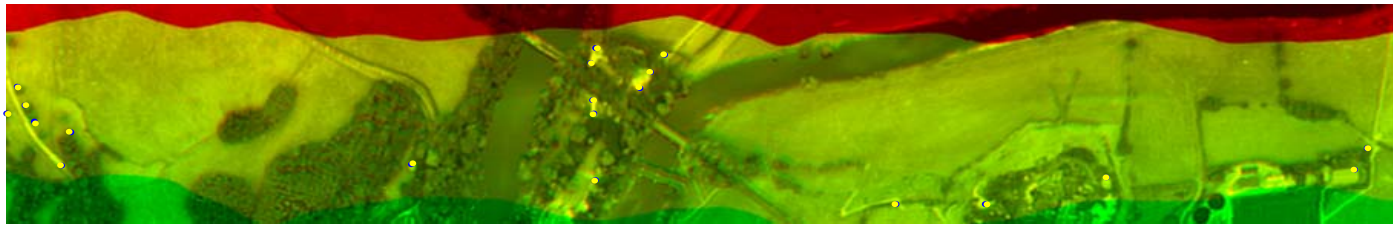

Figure 6: Deformable registration applied on *Caton* set of images

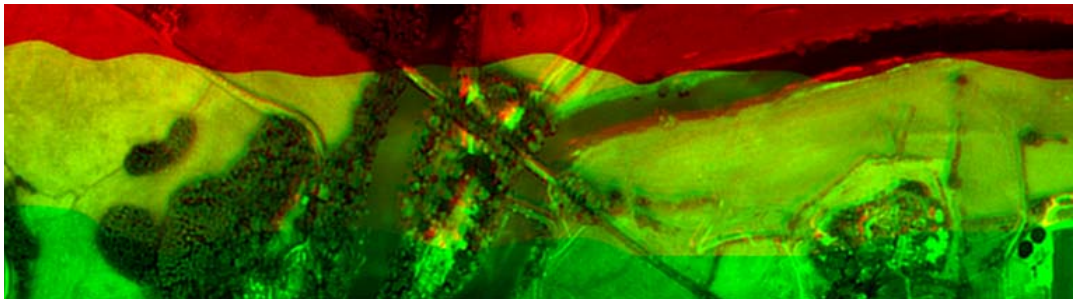

Figure 7: Affine Registration with ENVI applied on *Caton* set of images

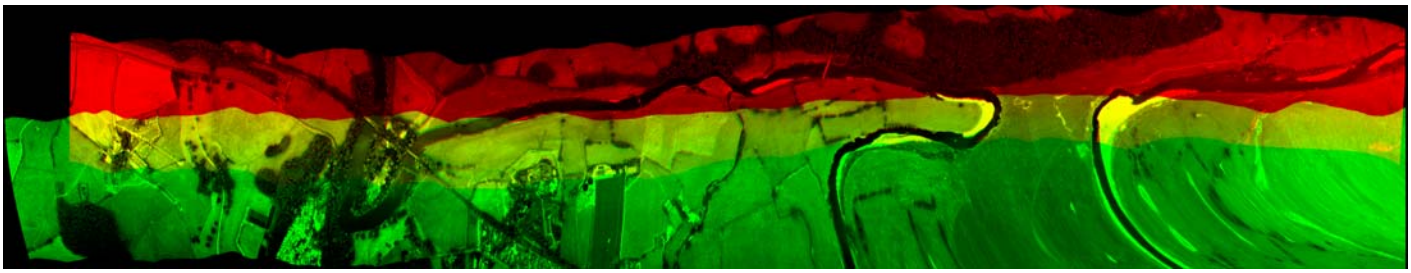

Figure 8: Polynomial registration with ENVI applied on *Caton* set of images

certain degree (chosen by the user). Although two other transformation models are available (RST and triangulation), only the polynomial one is tested since it is the most efficient in this case.

First, the software needs the correspondence points. ENVI proposes automatic feature detection, but the mismatching with the set *Caton* is too high, so the correspondence points obtained with the method presented in this paper are used. Then, the polynomial transformation of ENVI is compared visually with the algorithm presented in this paper.

Figure 6,7 and 8 show the results for *Caton* images. The deformable registration appears more efficient thanks to its local corrections. It was expected since the correction used by ENVI is affine in this case, thus giving the same result as in section 3.1 when using affine transformation. The image 8, gives the registered image with a polynomial of degree 2. Although the local misalignment is better corrected, a big deformation appears in the bottom right of the image showing that this transformation cannot be used for registering *Caton* images.

4 Conclusions and future works

A deformable registration method based on features is proposed here. First the features are detected thanks to the Harris corner detector and the initial correspondence is established through a rotation invariant cross-correlation measure. Then, RANdom SAmple Consensus method is applied in order to find the final correspondence and estimate a perspective transformation in the same time. Finally, multilevel B-spline approximation enables reducing the local misalignment. The performances of perspective transformation have been compared with the affine transformation and it was shown that both transformations give very similar results. Then, two other models were tested as local transformations: polynomial transformation with fixing points in the non-overlapping areas, and B-spline approximation. The B-spline method appeared to be more efficient in terms of alignment error and also in the preservation of the shape in the non-overlapping areas. The improvement achieved with the final algorithm is important when the real model presents the same kind of transformation (i.e. a global transformation close to homography, and some small local transformation) which is exactly the case in the example of *Caton* images. On the other hand, if the real model has a lot of local transformations that avoids finding a global one, then the first step of the mapping gives big misalignment and few inliers, involving a weak correction of the local transformations in the second step. This is the case in the example of *Garstang* images. Although the algorithm was designed to be as automatic as possible, some parameters must be initialized by the user. The most sensitive parameter is the threshold t used in RANSAC, and on which depend the number of inliers and the alignment error.

For this parameter, the algorithm displays the graphs showing the number of inliers as a function of the threshold. This graph should help the user to find the value for t by localizing where the first outliers appear.

In the future, a modification in the feature detection and feature matching parts could improve the performances of the algorithm. Indeed, if more inliers are found, a better correction can be performed. To do so, an affine invariant feature detector combined with affine invariant matching could increase the number of inliers while reducing the number of outliers. Moreover, the algorithm should be tested with more images in order to know if the method proposed here can correct those images as well as the *Caton* images, or if it gives unsufficient correction like in the *Garstang* example.

Summary of the parameters' values

	Caton	*Garstang*
threshold (affine)	10	26
threshold (homography)	10	31
nFP, *d* (polynomial)	20, 5	20, 2
m, *n*, *nlevel (B-spline)*	23, 23, 1	23, 23, 1

Table 5: Summary of the parameters

Acknowledgment

This work was supported by the PIMHAI project run under the INTERREG IIIB programme.

References

[1] Larry R. Tinney David N. Fogel. Image registration using multiquadratics functions, the finite element method, bivariate mapping polynomials and thin plate spline. Technical report from the National Center for Geographic Information and Analysis. 1996.

[2] M. Fornefett, K. Rohr, and H.S. Stiehl. Radial Basis Functions with Compact Support for Elastic Registration of Medical Images. *Image and Vision Computing*, 19(1-2): pages 87–96, 2001.

[3] C. J. Harris and M. Stephens. A combined corner and edge detector. *Proc. 4th Alvey Vision Conferences*, pages 147–151, 1988.

[4] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[5] S. Lee, G. Wolberg, and S. Y. Shin. Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3): pages 228–244, 1997.

[6] L.A. Piegl and W. Tiller. *The NURBS Book*. SpringerVerlag, ISBN: 3540615458, second edition, 1996.

[7] K. Rohr, H.S. Stiehl, M. Fornefett, S. Frantz, and A. Hagemann. Project IMAGINE: Landmark-Based Elastic Registration and Biomechanical Brain Modelling. *KI – Künstliche Intelligenz*, 00(3):37–39, 2000.

[8] Richard Szeliski and James Coughlan. Spline-Based Image Registration. Technical Report from Cambridge research laboratory. 94/1, April 1994.

[9] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.