



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

PRECISE IMAGE REGISTRATION USED FOR CORRECTION OF GEOMETRICAL IMAGE DISTORTION

PŘESNÉ LÍCOVÁNÍ OBRAZU KALIBRAČNÍCH VZORKŮ PRO KOREKCI GEOMETRICKÉ DISTRORZE

MASTER'S THESIS

DIPLOMOVÁ PRÁCE

AUTHOR

AUTOR PRÁCE

Bc. Petra Zemčíková

SUPERVISOR

VEDOUCÍ PRÁCE

Ing. Martin Mézl

BRNO 2017

Master's Thesis

Master's study field **Biomedical Engineering and Bioinformatics**

Department of Biomedical Engineering

Student: Bc. Petra Zemčíková

ID: 146207

**Year of
study:** 2

Academic year: 2016/17

TITLE OF THESIS:

Precise Image Registration Used for Correction of Geometrical Image Distortion

INSTRUCTION:

1) Make a theoretical review about methods of image registration with focus on methods for precise flexible image registration. 2) Design a process for individual steps of optimal image registration. 3) Design a methodology for evaluation of results. 4) In cooperation with FEI Czech Republic acquire suitable images for evaluation of methods. 5) Test designed algorithms on simulated and real images and compare your implementation with open-source software tools (e.g., Elastix, Matlab). 6) Discuss results and evaluate efficiency and applicability of your methods in comparison with other methods. This diploma thesis is produced in cooperation with FEI Czech Republic.

RECOMMENDED LITERATURE:

[1] JAN, J. Medical Image Processing, Reconstruction and Restoration - Concepts and Methods. Signal Processing and Comm. Signal Processing and Comm. Boca Raton, FL, USA: CRC Press, Taylor and Francis Group, 2006. 760 s. ISBN: 0-8247-5849-8.

[2] KLEIN, S., STARING, M., MURPHY, K., VIERGEVER, M.A., PLUIM, J.P.W. "elastix: a toolbox for intensity based medical image registration," IEEE Transactions on Medical Imaging, vol. 29, no. 1, pp. 196 - 205, January 2010.

**Date of project
specification:** 6.2.2017

Deadline for submission: 19.5.2017

Leader: Ing. Martin Mézl

Consultant: Ing. Miloš Malínský, Ph.D.

prof. Ing. Ivo Provazník, Ph.D.
Subject Council chairman

WARNING:

The author of the Master's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

ABSTRAKT

Cílem předkládané diplomové práce je pomocí lícování obrazů přesně popsat distorzní pole pro následné odstranění geometrické distorze. Snímky zkreslené geometrickou distorzí pochází z prozařovacího elektronového mikroskopu. První část práce se zabývá zejména teorií spojenou s elektronovou mikroskopií, vznikem geometrické distorze a samotnou obrazovou registrací s důrazem na intenzitní flexibilní metody lícování. Ve druhé části je pak představena vytvořená metoda pro modelování geometrické distorze a lícování obrazů postižených slabou geometrickou distorzí. Vyvinutá metoda je následně otestována na testovacích i reálných datech a srovnána s existujícími popsány metodami pro obrazovou registraci (například open-source softwarem Elastix).

KLÍČOVÁ SLOVA

Elektronová mikroskopie, geometrická distorze, modelování geometrické distorze, lícování obrazů, intenzitní metody, flexibilní lícování, Elastix.

ABSTRACT

The main objective of the thesis is using image registration for a precise description of a deformation field for a following removal of a geometrical distortion. The geometrically distorted images were captured by a transmission electron microscope. The first part of the thesis is dedicated to the theory associated with the electron microscopy, the formation of the geometrical distortion, and the image registration with the main emphasis on the intensity based non-rigid methods. The next part describes a developed method for modelling the geometrical distortion and for the image registration. The method is then tested on both test and real data and compared to the existing methods for image registration (such as Elastix open-source software).

KEYWORDS

Electron microscopy, geometrical distortion, geometrical distortion modelling, image registration, intensity methods of image registration, non-rigid image registration, Elastix.

ZEMČÍKOVÁ, P. *Precise Image Registration Used for Correction of Geometrical Image Distortion*. Brno: Brno University of Technology, Faculty of Electrical Engineering and Communication. Department of Biomedical Engineering, 2017. 99 p., 12 p. supplements. Master's thesis. Master's thesis supervisor: Ing. Martin Mézl, master's thesis consultant: Ing. Miloš Malínský, Ph.D.

DECLARATION

I declare that I have elaborated my master's thesis on the theme of "Precise Image Registration Used for Correction of Geometrical Image Distortion" independently, under the supervision of the master's thesis supervisor and with the use of technical literature and other sources of information which are all quoted in the thesis and detailed in the list of literature at the end of the thesis.

As the author of the master's thesis I furthermore declare that, concerning the creation of this master's thesis, I have not infringed any copyright. In particular, I have not unlawfully encroached on anyone's personal copyright and I am fully aware of the consequences in the case of breaking Regulation S 11 and the following of the Copyright Act No 121/2000 Vol., including the possible consequences of criminal law resulted from Regulation S 152 of Criminal Act No 140/1961 Vol.

Brno May 16th 2017

.....

(author's signature)

ACKNOWLEDGEMENT

I would like to thank my supervisor Ing. Martin Mézl and my consultant Ing. Miloš Malínský, Ph.D. for the expert advice, inspiration and kind and valuable help and support during the elaboration of this master's thesis.

Brno January 16th 2017

.....

(author's signature)

CONTENT

Content	iv
List of Figures	vii
List of Tables	xi
Introduction	12
1 State of the Art	14
2 Electron Microscopy	16
2.1 Transmission Electron Microscope	16
2.2 Geometrical Image Distortion	18
2.2.1 Mathematical Description of Geometrical Distortion	20
2.2.2 Application of Low Accelerating Voltage.....	20
3 Image Registration	22
3.1 Image Data.....	22
3.2 Image Registration in General	22
3.3 Image Registration Procedure.....	23
3.4 Geometrical Image Transformations	24
3.4.1 Rigid Transformations	24
3.4.2 Flexible Transformations.....	26
3.4.3 Nonlinear Transformations	27
3.4.4 Piece-Wise Transformations.....	28
3.4.5 Elastic Transformations	28
3.5 Interpolation.....	29
3.5.1 Nearest Neighbor Interpolation	30
3.5.2 Bilinear Interpolation.....	31
3.5.3 Spline Interpolation.....	32

3.6	Similarity Criteria	33
3.6.1	Intensity-Based Global Criteria	33
3.6.2	Information-Based Global Criteria	34
3.6.3	Feature-Based Methods	36
3.7	Optimization	36
3.7.1	Deterministic Algorithms	36
3.7.2	Stochastic Algorithms	37
3.7.3	Genetic Algorithms	38
4	Image Acquisition	40
5	The Proposed and Implemented Method of Image Registration	43
5.1	Model of the Distortion.....	43
5.1.1	Implementation of the Distortion Model to MATLAB	44
5.2	GA Function	47
5.2.1	Origin	47
5.2.2	Registration Engine.....	50
5.2.3	Transformation Function	53
5.2.4	Saving Results and Visualization	56
5.3	Setting Options	57
6	Existing Methods and Algorithms	60
6.1	fminsearch	60
6.1.1	fminsearch Implementation	60
6.2	elastix.....	61
6.2.1	Access from MATLAB	63
6.2.2	elastix and transformix Implementation	65
6.3	The Demon Algorithm.....	66
6.3.1	imregdemons Implementation.....	67

7	Testing and Optimization	68
7.1	The Result Visualization and Numerical Evaluation.....	68
7.2	Testing Data.....	69
7.3	Testing	71
7.4	The Evaluation of the Registration of the Real Images	80
7.4.1	The Registration of the Images with a High Magnification and a Low SNR	80
7.4.2	The Registration of the Images with a Regular Structure.....	83
8	An Application of the Registration to Suppress the Geometric Distortion	88
9	Discussion	90
9.1	GA function	90
9.2	fminsearch	91
9.3	elastix.....	92
9.4	imregdemons	93
10	Conclusion	94
	References	95
	List of Abbreviations	99
	SUPPLEMENTARY MATERIALS	100
A	Parameters' Effect to an Image	100
B	How to Control the Program	105
B.1	Input Images	105
B.2	Method Choice and Setting.....	106
C	The Parameters for the Set of the <code>elastix</code> Image Registration	109
C.1	AtW_parameters_1.txt.....	109
C.2	AtW_parameters_2.txt.....	110

LIST OF FIGURES

Fig. 2.1 Schematic representation of TEM arrangement. [26]	16
Fig. 2.2 Basic schema of the image formation process in TEM. [39]	18
Fig. 2.3 Geometrical image distortion; a) pincushion distortion, b) barrel distortion, c) spiral, azimuthal distortion. [8].....	19
Fig. 3.1 Rigid transform of a simple image: a) original and b) shifted and rotated image. [13].....	25
Fig. 3.2 Flexible transform of the previous image: a) original and b) image scaling and shearing. [13].....	26
Fig. 3.3 Regular rectangular uniform grid shown in black, nonuniform transformed grid shown in grey. [21].....	29
Fig. 3.4 Sinc function.....	30
Fig. 3.5 Nearest neighbour interpolation method: a) schematic representation of the nearest neighbor interpolation of values outside a regular grid, [38] b) the “staircase” resulting image function. [9]	31
Fig. 3.6 Bilinear interpolation method: a) schematic representation of the bilinear interpolation of values outside a regular grid, [33] b) result image function. [9]	32
Fig. 3.7 Bicubic interpolation method: a) schematic representation of the bicubic interpolation of values outside a regular grid, [33] b) result image function. [9]	32
Fig. 3.8 Joint histogram of two images schematically (only a single histogram value out of all qr values is depicted). [13].....	35
Fig. 3.9 The course of determining the new point using the simplex method. a) reflection and expansion, b) contraction, c) reduction. [25]	38
Fig. 4.1 Schematic representation of the origination of a pair of images. One of the images is created in a way in blue, the other in red. The scheme is without magnification.	41

Fig. 4.2 An example of the first described calibration sample (a carbon film with a golden shadowing of waffle pattern gratings made on a copper grid) and an example of a shift made.....	42
Fig. 4.3 An example of the second described calibration sample (a carbon film shadowed with gold with graphitized carbon particles) and an example of a shift made. .	42
Fig. 5.1 An example of a deformed field. a) An original image in red. A created auxiliary field before a deformation in white. b) Barrel distortion; an auxiliary deformed field in white. An example of two of many variants of the differently distorted original image in red and yellow. c) Pin cushion distortion; the same example for pin-cushion distortion in red and yellow.	46
Fig. 5.2 Schema of the main function.	47
Fig. 5.3 Schema of the Registration Engine Function.	50
Fig. 5.4 Schema of the Transformation Function.	54
Fig. 6.1 A block scheme of the basic registration components of image registration used in <code>elastix</code> . [18].....	62
Fig. 6.2 Add the folder with all the function and <code>elastix</code> code to MATLAB path. ..	65
Fig. 7.1 The original undistorted selected image placed in the black frame (zeros).	69
Fig. 7.2 a) The distorted image in the frame. b) The same distorted image cropped out of the frame.	70
Fig. 7.3 a) The previous image Fig. 7.2 b). b) The same distortion of the image, but a different position relative to the optical axis. c) Both images overlaid one over another.	70
Fig. 7.4 a) 7 px were cropped off the bottom and 7 px off the right side of the first distorted image. b) 7 px were cropped off the top and 7 px off the left side of the second distorted image. c) Both images overlap with performed shift.	71
Fig. 7.5 An example of the coefficients DF_x , DF_y development during the registration.	73
Fig. 7.6 A zoom of the previous Fig. 7.5.	73
Fig. 7.7 The results of the GA registration. a) Less iterations and individuals, 70 s, quality 0,9899. b) More iterations and individuals, 292 s, quality 0,9934. See the details and the exact numbers of iterations and individuals in the Tab. 7.2.	74

Fig. 7.8 One more example of the results of the GA registration. The set of the <code>dsVec</code> , <code>nIter</code> and <code>nInd</code> the same as in the previous example. a) Less iterations and individuals, 119 s, quality 0,9932. b) More iterations and individuals, 205 s, quality 0,9938.	74
Fig. 7.9 The results of the <code>fminsearch</code> registration. a) 200 iterations, 117 s, quality 0,9899. b) 500 iterations. 299 s, quality 0,9915. See the details in the Tab. 7.2.	75
Fig. 7.10 One more example of <code>fminsearch</code> result. a) 200 iterations, 109 s, quality 0,9936). b) 500 iterations, 262 s, quality 0,9895.	75
Fig. 7.11 a) The result of <code>elastix</code> ; 28 s, quality 0,9685.	76
Fig. 7.12 The result of the Demon algorithm. a) 100 iterations, 5 s, quality 0,9559. b) 500 iterations, 27 s, quality 0,9690. See the details in the Tab. 7.2.	76
Fig. 7.13 A visualization of the displacement field (GA function, results of the second registration, Fig. 7.7 b)) using rectangular grid.	77
Fig. 7.14 a) The resulting displacement field of <code>elastix</code> . The same displacement field is also a result of the successful GA and <code>fminsearch</code> function and successful <code>imregdemons</code> registration (500 iterations). b) The resulting displacement field of the unsuccessful Demon algorithm registration (100 iteration).	77
Fig. 7.15 A comparison of a cropped part of the displacement fields from the previous Fig. 7.14 a) and b). The displacement vectors at each pixel show the shift of the pixel between the fixed image grid and a corresponding location in the moving image.	78
Fig. 7.16 The quality development during the image registration. The order of the curves is the same as the order of the methods in the Tab. 7.2.	80
Fig. 7.17 The input images and their comparison. The sample is a carbon film shadowed with gold with graphitized carbon particles. The magnification was 510 000x and the dose rate 7 200 e/Å ² s ⁻¹ . Low image quality due to low SNR is apparent. The results of the registration are in the Tab. 7.4.	80
Fig. 7.18 The results of the GA registration. a) Less iterations and individuals, 401 s, quality 0,9800. b) More iterations and individuals, 542 s, quality 0,9800. See the details and the exact numbers of iterations and individuals in the Tab. 7.4.	81

Fig. 7.19 The results of the <code>fminsearch</code> registration. a) 200 iterations, 231 s, quality 0,9674. b) 200 iterations, 232 s, quality 0,9394. c) 500 iterations, 668 s, quality 0,9800. See the details in the Tab. 7.4.....	82
Fig. 7.20 a) The result of <code>elastix</code> ; 27 s, quality 0,9529. See the details in the Tab. 7.4.	82
Fig. 7.21 The result of the Demon algorithm. a) 100 iterations, 7 s, quality 0,9620. b) 500 iterations, 48 s, quality 0,9588. See the details in the Tab. 7.4.	82
Fig. 7.22 The input images and their comparison. The magnification and the dose rate are not known.	83
Fig. 7.23 The results of the GA registration. a) Less iterations and individuals, 421 s, quality 0,9944. b) More iterations, 616 s, quality 0,9944. For the exact numbers of iterations and individuals are see the Tab. 7.5.	84
Fig. 7.24 The results of <code>fminsearch</code> registration. a) 250 iterations, 376 s, quality 0,9458. b) 500 iterations, 639 s, quality 0,9463. See the details in the Tab. 7.5.	85
Fig. 7.25 Not found solution of a registration of images of a regular structure.	85
Fig. 7.26 The results of <code>elastix</code> registration, 27 s, quality 0,8981. See the details in the Tab. 7.5.	86
Fig. 7.27 The results of <code>imregdemons</code> registration. a) 100 iterations, 10 s, quality 0,9294. b) 500 iterations, 61 s, quality 0,9528. See the details in the Tab. 7.5.	86
Fig. 7.28 The result of the GA function, when only a translation coefficients were used.	87
Fig. 8.1 A scheme of the distortion correction. (The exact expression of the equation of inverse transformation can be found in [35]. px represents the corrected x coordinates and px the resulting transformed x coordinates. The same is true for y coordinates.)	89

LIST OF TABLES

Tab. 4.1 The most important specifications of Talos L120C TEM for Materials Science. [30].....	40
Tab. 5.1 Summary table of setting options of the GA function.....	58
Tab. 6.1 An example of components available in <code>elastix</code> toolbox.....	63
Tab. 7.1 An adjustment of the coefficients for distorting the test images.	71
Tab. 7.2 Results of the registration of images from Fig. 7.4.	72
Tab. 7.3 Comparison of time efficiency of the GA function and <code>fminsearch</code>	79
Tab. 7.4 The results of the registration of the images on the Fig. 7.17.	81
Tab. 7.5 The results of the registration of the images on the Fig. 7.22.	84
Tab. 10.1 Parameters' effect table.	100

INTRODUCTION

This thesis deals with the image registration of images affected by the geometrical image distortion caused by the internal properties of a transmission electron microscope system. For basic applications of a microscope including quantitative measurements of the structures in the image, automatic image processing or image correlation, an occurrence of such a distortion is not admissible. For accurate correction of this geometrical distortion, it is essential to precisely map the deformation field. The image registration was chosen as a powerful and efficient tool for the unambiguous description of the distortion.

To find new application options of transmission electron microscope, it is necessary to experiment with settings of its individual components and processes. Currently a use of lower voltage (60 or 80 kV) than is usual (120, 200 kV or even 300 kV) on the same system and its influence on the acquired image is a popular research topic. One of the consequences of the low energy imaging is an increased occurrence of a geometric image distortion.

As is well known, the force of the geometric distortion increases with increasing distance from an optical axis. Therefore, there is aperture embedded into a microscope that reduce the most distorted areas of a resulting image. In some applications, however, the large field of view is required, and it is appropriate to reduce or eliminate such a restriction. For this purpose, the present geometric distortion must be accurately corrected.

In the presented master's thesis, a particular emphasis was put on familiarization of the topic itself, the state of the art and options of solution of the problem. After creating the model of the distortion, several methods for obtaining the desired output were applied with greater or lesser success and the method that seemed as the most appropriate was developed, implemented, tested, and further improved. Finally, it was compared with available open-source methods.

The thesis is divided into eight main parts. The first chapter briefly reports the state of the art in the field of image registration of geometrically distorted images. The first half of the next chapter contains a simplified description of the transmission electron microscope and its function, whereas the second half summarizes the aberrations caused by a microscope and especially it describes the geometrical image distortions. Then, the third chapter deals with the problems of image registration with emphasis on intensity-based techniques and non-rigid methods. In the fourth chapter, there is briefly described how the samples were acquired, while the fifth chapter is devoted

to modelling of the distortion and the developed method for the image registration. The following chapter of the master's thesis sums up properties and possibilities of application of the existing methods whether already implemented in MATLAB or available open-source, such as software `elastix`. Finally, the last part of the thesis deals with testing and optimization. Eventually, the possibilities of application, the discussion of the results obtained and the conclusion of the whole work follow.

1 STATE OF THE ART

The need of the image registration in the context of the electron microscopy is coming out of several facts. Recently an interest in imaging of large samples increased, but these samples might be too large to be captured in a single image. Then several images of a sample regions are captured and stitched up together. Here the geometrical image distortions, described in chapter 2.2, comes to play. Another cause of the distortion emergence is the above-mentioned usage of low voltage.

The image processing is an easiest and cheapest way how to correct geometrical distortions. Because of their nonlinearity (distortion grows with increasing distance to the image centre), use of linear transformations and simple image registration is impossible. Also important is a usual requirement of subpixel accuracy (which is particularly important in the electron microscopy, where this requirement follows from the nature of the great magnification used and the precise accuracy required). [22]

One of the methods used for distortion correction is capturing and subsequent comparing of calibration samples (straight lined objects, checkerboards), which clearly reproduce the lenses caused distortion. Unfortunately, this can be a problem in electron microscopy. Due to the process of a replacement of the calibration samples (crystalline structures) for the real samples and refocusing after the exchange leads to a possible change of the distortion, which is difficult to control and evaluate. [22] Additional information beyond the scope of this thesis can be found in [5] or [40]. A thesis dealing with the definition of a geometric distortion using crystalline structure is for example [32]. Here a monocrystal model was aligned to a real monocrystal image at the magnification greater than 200 000 times (this magnification is necessary for a visibility of the crystalline structures).

Another method, applicable in electron microscopy and even for magnifications up to 10 000 times used in biology applications, is utilization of a model of the radial symmetric distortion. In the case of this variant, the algorithms can be categorized from two points of view. The first is the mathematical model of the distortion itself, the second is an estimate of the model coefficients. The model of the barrel or pincushion distortion (Fig. 2.3) is the simplest and the most popular way of modelling radial distortions and many algorithms are based on it with some modifications. The reason is its usability and functionality for almost all kind of regular lenses (the only problem is wide angle lenses (fisheye), for which different models were developed). The disadvantage of this model is the assumption of the specific shape; in practice, this shape might be affected and misrepresented by non-ideal electromagnetic fields. Thus, realistically, it may be

preferable not to restrict the exact shape of the distortion model. [22], [35]

The next method is parametrizing the lens distortions by Legendre polynomials where intensity variance is used as a similarity measure. [22]

2 ELECTRON MICROSCOPY

2.1 Transmission Electron Microscope

Images analysed later in the thesis were captured using TEM, transmission electron microscope, schematically in Fig. 2.1.

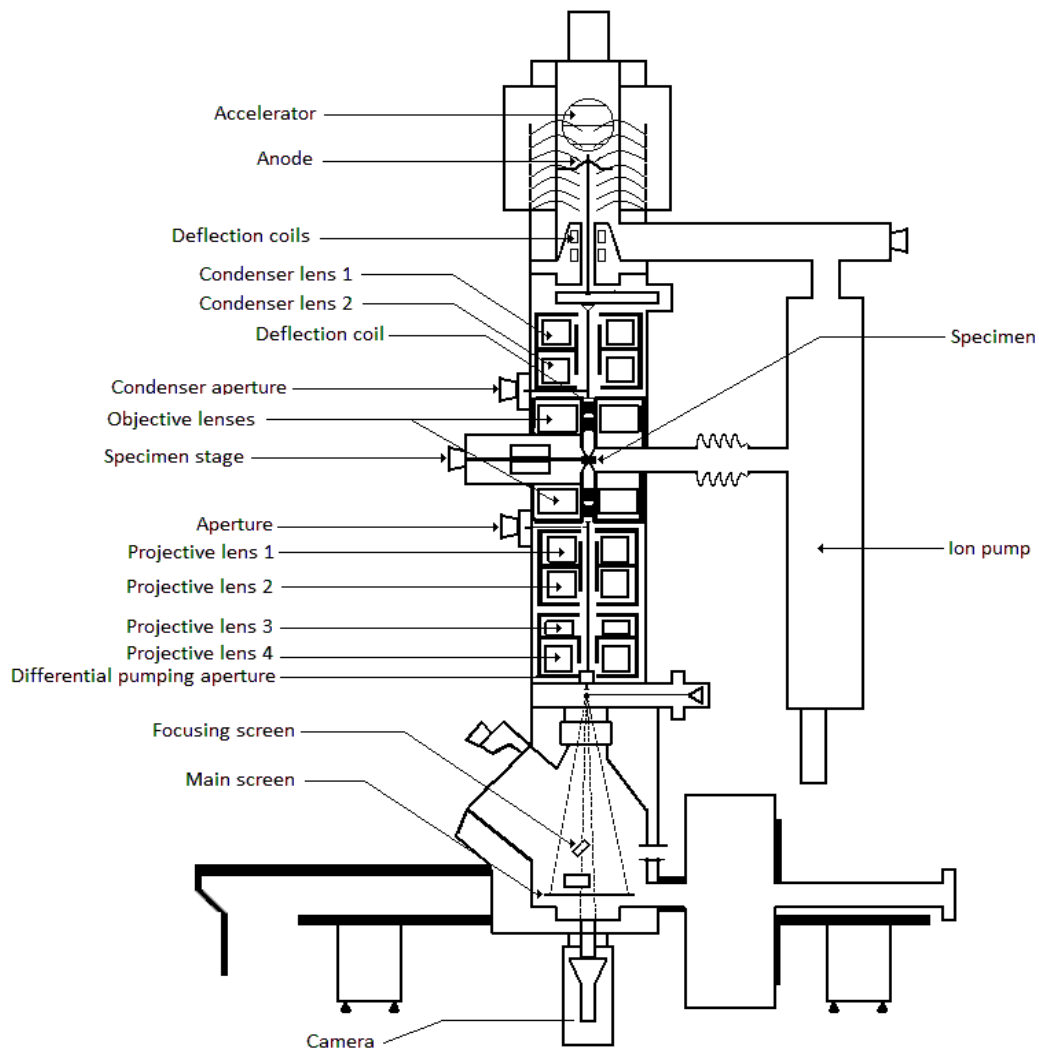


Fig. 2.1 Schematic representation of TEM arrangement. [26]

Any microscope can only distinguish between two points at a minimum distance of a half of wavelength λ of a source (light source, electron source). Wavelength of accelerated electron (60 kV) is about 0,005 nm (a hundred thousand times shorter than of visible light). Practical resolution of TEM can be nowadays greater than 0,1 nm and magnification in the range 10^3 to 10^6 . [7], [19]

The electron beam is produced by an electron gun consisting of electron source, which is a cathode with highly negative potential, and an anode plate accelerating electrons by its relative positivity to the cathode so a potential difference is created. (Only around 1% of electrons pass through the hole in the anode plate farther). The electron gun creates the *illumination system* together with following at least two magnetic condenser lenses, which determines a very important illumination diameter (and its flexibility over a desired range) of the primary electron beam at the specimen.

Specimen stage is a part of TEM where the specimen is inserted and whose stability is crucial for image resolution (every vibration is magnified in the output image). The specimen must be ultrathin (maximum 100 nm) for the electrons to pass through the material and form the image. (After the landing to the specimen, most of the primary electrons are transmitted, in case of TEM just a minimum is absorbed, and some electrons are scattered on the specimen structures, that creates the image contrast. Except the scattering phenomena, the primary beam undergoes other interactions leading to additional reactions such as an emergence of luminescence, X-radiation, generation of secondary and Auger electrons etc. [27]). As the vacuum in the TEM column must not be violated, the specimen is inserted through an airlock. [7]

Last part of the TEM is an *imaging system* consisting of several lenses producing the final magnified image (or electron diffraction pattern). Closest to the specimen, there is an objective lens; strong lens with a small focal length. Necessary for the lens is a water cooling because of its high excitation current. The excitation current must be accurately stabilized, but on the other hand, possible to change in order of maintaining the same focal length at different electron-accelerating voltages and for precise focusing of the image. Intermediate lenses serve to control the magnification of TEM (as above, $10^3 - 10^6$) by changing their focal length, and furthermore, these lenses serve to display the diffraction pattern instead of magnified image of the specimen, if needed. Varying lens excitation does this. For the final image formation and enlargement on the TEM screen, a projector lens is necessary. Because the TEM screen has several centimetres in the diameter and some electrons land far from the optical axis, distortions occur. Strong projector lens should minimize this effect. The final magnification of the image is a result of all the magnification factors of each of the imaging lenses. [7], [27]

Final part of the imaging system is a fluorescent screen (converts electron energy to visible light, which is possible to observe by auxiliary eyepiece and serves specially to focusing) and an electronic camera system (usually CCD sensors, formerly photographic film), used for projecting the image on a monitor. [27]

The two stages of the image formation are shown in the Fig. 2.2. Stage A is

the scattering of an incident electron beam by a specimen. This scattered radiation passes through an objective lens, which focuses it to form the primary image. Then, stage B magnifies the primary image obtained in stage A using additional lenses, and form a highly magnified final image. [39]

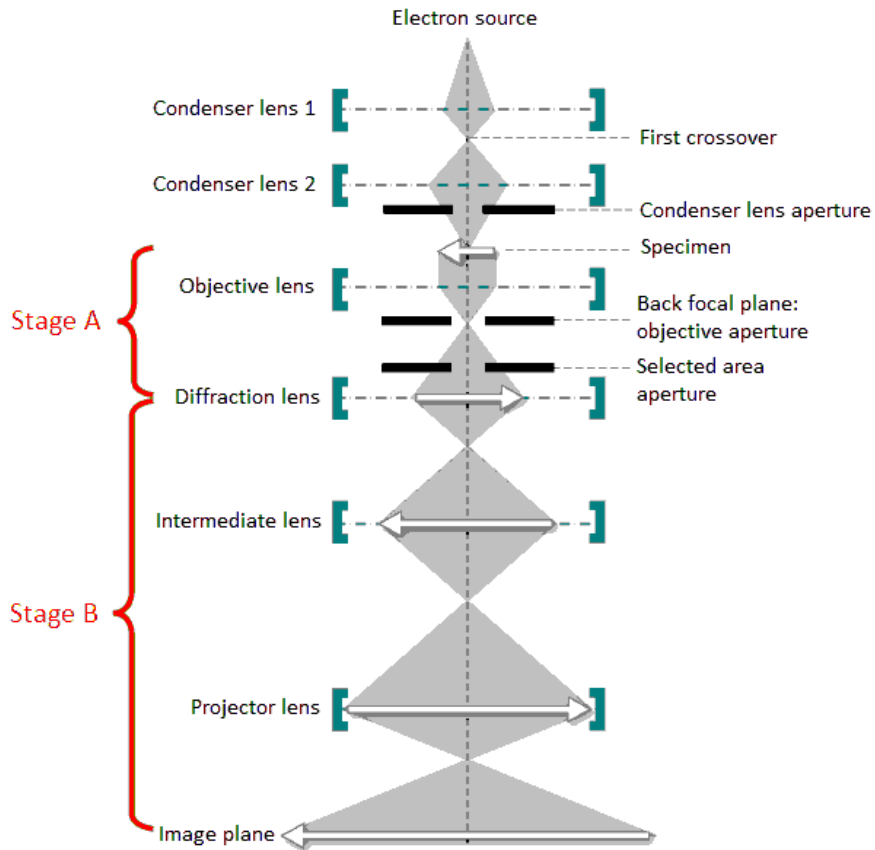


Fig. 2.2 Basic schema of the image formation process in TEM. [39]

What are the other components of TEM, such as deflectors, stigmators, apertures etc., and details of the parts described above can be found for example in [7], [19], [26] or [27].

2.2 Geometrical Image Distortion

Because of lens imperfections, various types of optical aberrations are produced, including defocus, spherical aberration, chromatic aberration, astigmatism etc. Some of them can be easily corrected (depending on the type of microscope), for some it is more complicated or impossible to correct them completely and there is always at least one aberration which limits the optical resolution of a microscope. [8]

Optical aberrations can be divided into axial and off-axial aberrations or chromatic and geometrical aberrations which include spherical aberration, coma, field astigmatism and field curvature and finally geometrical distortion. As the aim of the thesis is image registration for correction of geometrical image distortion, this type of distortion is more examined below. An elaboration of all the aberrations would be beyond the scope of the thesis and it can be found e.g. in [8].

Unlike other symmetry-permitted aberrations of a round lens mentioned above, geometrical image distortion does not affect the image information and punctuality, thus, every object point has its stigmatic image point regardless the resolution. The single image point is not astigmatic, but the shape of the whole object is distorted. The distortions affect all the points except those on the optical axis and the actual effect of distortion increases with increasing radial distance from the optical axes. The most common types of geometrical distortion are the pincushion distortion in Fig. 2.3 a) and the barrel distortion in Fig. 2.3 b). They can occur in the presence of round light optical or electrostatic electron lenses. Spiral or azimuthal distortion in Fig. 2.3 c) is in contrary caused by helical trajectories of electrons determined by the force of magnetic lenses. These distortions are nonlinear and often occur in combination. The originator is mostly the projector lens system and nowadays large CCD cameras aggravate the problem. In case of the electron microscopy, a hardware controlled distortion correction is much more complicated, unlike e.g. in the light microscopy, because the projector lenses are not available to adjustment by operator and then, image processing is the easiest way of correction. [8], [22]

The geometrical image distortion certainly is not the only aberration modifying the image geometry, but it is the most significant one. [35]

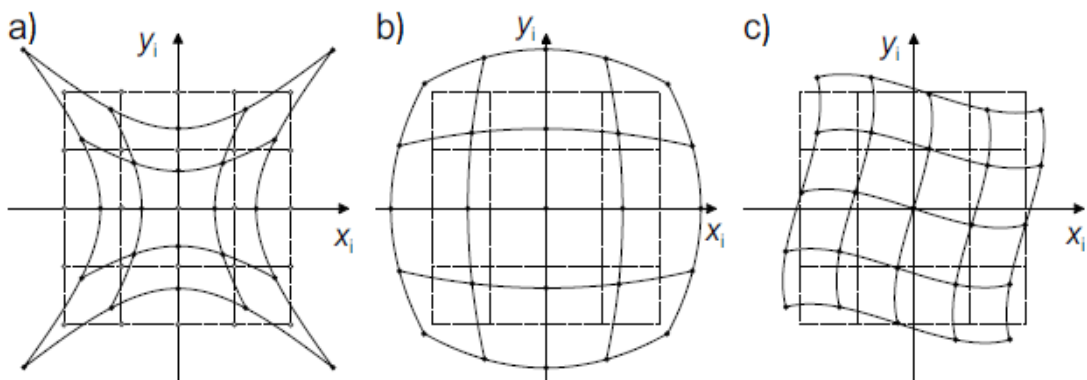


Fig. 2.3 Geometrical image distortion; a) pincushion distortion, b) barrel distortion, c) spiral, azimuthal distortion. [8]

2.2.1 Mathematical Description of Geometrical Distortion

As mentioned above, the geometrical distortion is of radial type, so the easiest method to model it, is a radial shift to the pixel coordinates which modifies the distances of the pixels from the image centre. When

$$\hat{r} = f(r), [35], \quad 2.1$$

where r is the pixel coordinate distance in the undistorted image and \hat{r} in the distorted one, then f is a distortion function. The distortion function f can be expressed by its Taylor expansion – the higher polynomial degree is, the better approximation but greater computational and time requirements are. In most cases, the usage of just one coefficient controlling the distortion is precise enough. For a treatment of more significant distortion case, the second coefficient can be added as in equation 2.2. This order of the approximation is precise enough and at the same time stable enough:

$$\hat{r} = r + \kappa_1 r^3 + \kappa_2 r^5 = r(1 + \kappa_1 r^2 + \kappa_2 r^4), [35]. \quad 2.2$$

Thus, κ_i are radial distortion coefficients, where κ_1 is coefficient controlling a general behaviour of the distortion and κ_2 is added for better approximation and easier control of the distortion. The equation 2.2 written in the context of two-dimensional Cartesian coordinate system x, y expresses the equation 2.3:

$$\begin{pmatrix} \widehat{p}_x \\ \widehat{p}_y \end{pmatrix} = \begin{pmatrix} p_x(1 + \kappa_1 r + \kappa_2 r^2) \\ p_y(1 + \kappa_1 r + \kappa_2 r^2) \end{pmatrix}, [35]. \quad 2.3$$

2.2.2 Application of Low Accelerating Voltage

The electrons are usually accelerated by applied voltage high at least 100 kV and higher, even to 300 kV. The reasons are several, the expense of high-voltage microscopy is avoided and the possibility of atomic resolution is provided, even without lens-aberration correction. [6]

For special applications (especially in cell biology because of minor damage to the sample), low voltage transmission electron microscopy (LVTEM) with accelerating voltages around 30 kV (even 5 kV) is employed. It can still provide atomic resolution in low-voltage phase-contrast images and generates highly contrasted images when the sections are thinner (10 - 40 nm) than usual and it allows eliminate otherwise

required staining of biological structures. On the other hand, so thin sample is unusable in the material science (because of a too low phase contrast). The disadvantages of LVTEM is an extremely complicated sample preparation and a requirement of a correction of created aberrations. The impact of the aberrations is greater with the lower voltage applied. [1], [6]

3 IMAGE REGISTRATION

There are thousands of papers on image registration published. Classic works published before 1992 introduced key ideas, which are still in use. Advancement of acquisition devices which has caused increase in both amount and diversity of obtained images has led to vast research and development in automatic image registration. [41]

3.1 Image Data

The most common example of multidimensional signal is an image. An image can be represented in many ways, while the most frequent is static image, two-dimensional signal defined as a function of two continuous variables,

$$f(x, y), [13], \tag{3.1}$$

and since image size is always limited, $x \in \langle -x_{max}, x_{max} \rangle$ and $y \in \langle -y_{max}, y_{max} \rangle$. The before forementioned equation 3.1 describes images of scalar values of elements, i.e. grayscale and binary (black and white) images, whilst a colour image can be represented by a vector function,

$$\mathbf{f}(x, y) = [f_R(x, y), f_G(x, y), f_B(x, y)]^T, [13], \tag{3.2}$$

whose components describes the brightness of the individual colour components (red, green, and blue). [12], [13]

3.2 Image Registration in General

Image processing includes several standard tasks, such as preprocessing, reconstruction, fusion, segmentation and description, classification, restoration and finally image registration. [13], [22]

Image registration is one of the key tasks speaking of biomedical images and applications. The image registration is a crucial step in all image analysis tasks in which the final information is gained from the combination of various data sources like in movement analysis, multi-subject analysis, image sequence analysis, image fusion, change detection, it is essential for elastography or 3D reconstruction (MRI, CT). [41] One of the image registration application is also analysis and

compensation of deformations and distortions, which is the main objective of this thesis. Per the specification of the application, the goal of the registration is to compare or fuse information, or just obtain quantitative or qualitative information of deformation. [23]

The input images can be images of the same object in different times, by different modalities, under different conditions or it can be images of the same region in different objects. The input can consist of two or more images (2D, or also 3D volumes or sequences of 2D or 3D). [23] Simply put, the goal is to obtain the corresponding structures in images on the same spatial coordinates, in other words, to find a **geometrical transformation** relating coordinates of corresponding locations in input images. [11], [38] Although more images can be put in, image registration geometrically aligns two images – the reference and the moving image, which is transformed during the process to correspond to the reference image. [38]

Not every time it is essential to process all image pixels. There is a way to register images of linear, graph-like structures (blood-vessels, nerves) by first finding correspondences between these structures or register an image out of its segmentations. [23]

Very important aspect when choosing an image registration method is a required accuracy of the image registration. For some applications, the pixel-level accuracy is enough, but for others, such as electron microscopy is, the subpixel accuracy is crucial.

3.3 Image Registration Procedure

Two input images, moving A and reference image B , will be considered. During the process image A will be transformed to A' .

Equation 3.3 can formally describe image registration process,

$$\alpha_0 = \arg \max_{\alpha} c \{B(\mathbf{x}_B), A'(T_{\alpha}(\mathbf{x}_A))\}, \quad 3.3$$

$$\mathbf{x}_B, T_{\alpha}(\mathbf{x}_A) \in \Omega_{\alpha}, [13]$$

which expresses finding the optimal vector of parameters α_0 of geometrical transformation T in terms of optimizing image similarity c between reference image B and transformed image $A'(T_{\alpha}(\mathbf{x}_A))$. Vector of parameters α_0 transforms image A into transformed image A' . With respect to the chosen criterion of image similarity $c(B(\mathbf{x}_B), A'(T_{\alpha}(\mathbf{x}_A)))$ it is possible to minimize or maximize the image similarity c . Spatial

range $X_{A'}$ differs from X_B , so the image content comparison is only possible on the common area, which is represented by its intersection $\Omega_{\alpha} = X_{A'} \cap X_B$.

3.4 Geometrical Image Transformations

Geometrical transformation is a process, during which the spatial coordinates of pixels of an image are transformed, equation 3.4, while their brightness values remain unchanged. Thus, it is a mapping of T between moving image A and transformed moving image A' as it is expressed in the following equation 3.5. In this case, the transformation of pixels is expressed, equations can be easily converted into voxels coordinates transformation by adding third dimension z .

$$r = (x, y) \rightarrow r' = (x', y'), [13] \quad 3.4$$

$$r' = T(r), [13] \quad 3.5$$

From different angles, it is possible to divide geometrical transformations into groups. The most general division is into *linear* and *nonlinear geometrical transformations*. While for linear transformations the parallelism of the lines is maintained, in nonlinear transformations, the parallelism is not maintained and the distances between pixels are changed.

According to the scope of the transformation into global and local geometrical transformation. If it is possible to describe the distortion of whole image in single formula, same for all the pixels, then it is a *global geometrical transformation*. Otherwise, it is a *local (piece-wise) transformation* when it is necessary to use different functions (and therefore different formulas) for different areas of the image (guaranteeing continuity and perhaps smoothness at the borders of the areas) [13].

And according to the nature of the distortion, there are *rigid* and *flexible geometrical transformations*.

Individual types will be discussed in following sections.

3.4.1 Rigid Transformations

Regarding rigid transformations, the shape of the displayed scene is not changed, it is not geometrically distorted, just the position is different and the distances between individual pixels are not affected. The image can be *shifted* (values Δx , Δy are added, 3.6)

$$\Delta r = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, [13], \quad 3.6,$$

or *rotated* (within angle θ) according to the transform matrix B expressed in matrix 3.7,

$$B = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. [13] \quad 3.7$$

Then the *generic rigid transformation* 3.8, as seen in Fig. 3.1, consists of shift and rotation. And it is a linear transformation. [13]

$$r' = Br + \Delta r. [13] \quad 3.8$$

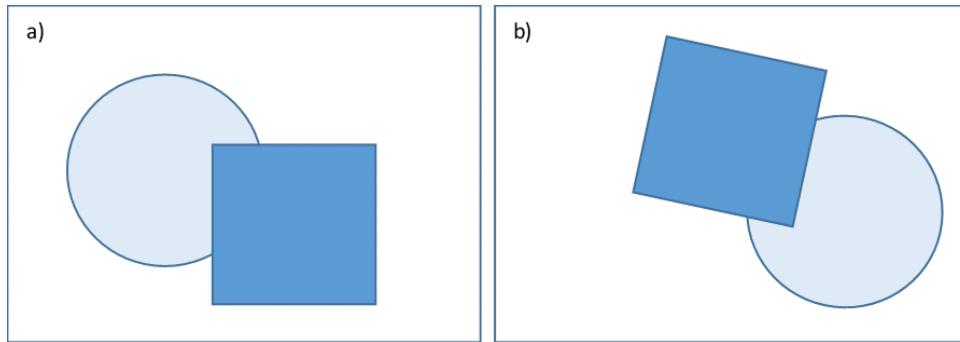


Fig. 3.1 Rigid transform of a simple image: a) original and b) shifted and rotated image. [13]

Mentioned above matrices 3.6 and 3.7 can be easily converted into just one rigid transformation matrix R , in 3.9, by extension of the spatial coordinate vector on the homogenous coordinate vector $r = (cx, cy, c)$, as expressed in 3.10, where c is an arbitrary constant, usually selected $c = 1$. [38]

$$r' = Rr \quad 3.9$$

$$r' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & \Delta x \\ \sin \theta & \cos \theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} [13] \quad 3.10$$

3.4.2 Flexible Transformations

On the contrary, flexible transformations distort the shape of the displayed scene and therefore, distances between pixels are different in the original and transformed image. Thus, shift and rotation are not sufficient enough to describe the distortion. One of the simplest flexible transformations is *plain scaling* expressed by 3.11 matrix S (for 2D),

$$S = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, [13] \quad 3.11$$

where s_x and s_y define scaling in x and y coordinates. The second simplest is *shearing* (gradually and uniformly shifting the rows or columns of the image matrix) defined in 2D as G_x and G_y matrices as in 3.12:

$$G_x = \begin{bmatrix} 1 & g_{xy} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 0 & 0 \\ g_{yx} & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. [13] \quad 3.12$$

Variables g_{xy} for G_x and g_{yx} for G_y express shearing in x and y coordinates.

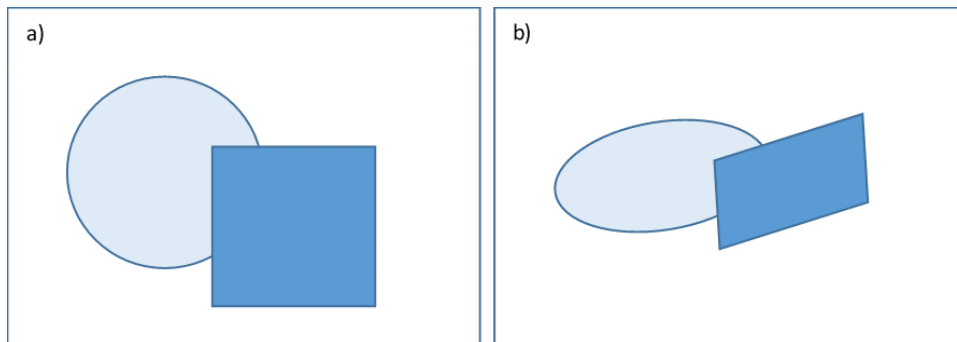


Fig. 3.2 Flexible transform of the previous image: a) original and b) image scaling and shearing. [13]

The most generic linear image transformation, *affine transformation*, 3.14, is obtained by combining all the previous transformation matrices (shift, rotation, scaling and shearing) into one. Resulting matrix A is then defined as:

$$A = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix}, [38] \quad 3.13$$

$$r' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = G_x G_y S R r = A r = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. [38] \quad 3.14$$

The affine transformation has for 2D 6 independent parameters and is maximally flexible linear transformation, which keeps the straightness and parallelism of lines and planarity of surfaces. [10]

3.4.3 Nonlinear Transformations

All the other flexible transformations are nonlinear; they are more generic, but also for the description much more parameters are needed.

This group of the transformations includes *projective transformation*, which preserves straightness of lines and planarity of surfaces, but not its parallelism anymore. [10]

All the other transformations moreover curve lines and surfaces.

The *polynomial transformation* uses the polynomial coefficients to calculate the new position vector. Utilization of this transformations relates to correction of distortions such as barrel or pincushion distortion, therefore it is also used in this thesis. The disadvantage consists in the risk of origin of false oscillations, which can cause undesirable local image warping, however, this is related particularly with higher-order (higher than third) polynomials, which are therefore appropriate to avoid. The definition in 2D for input vector $r = (x,y)$ is:

$$r' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \sum_i \sum_j \begin{bmatrix} i_j c_x \\ i_j c_y \end{bmatrix} x^i y^j, [13] \quad 3.15$$

where c are the coefficients of polynomials defining the transformation and $r' = (x',y')$ is the output vector.

Then the second-order polynomial transformation in 2D is:

$$r' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ y^2 \\ x \cdot y \end{bmatrix}. \quad [13] \quad 3.16$$

Specific application for the barrel and pincushion distortion is recorded in chapters 2.2.1 and 0. [10], [13], [37]

Eventually, the *radial basis functions* using control points (CPs) \tilde{r} can be presented. There exists known correspondence between \hat{r}' and \tilde{r} and the set of the correspondences serves for construction of the transform function (thin-plate splines are the most frequently used functions). [13]

3.4.4 Piece-Wise Transformations

If it is impossible or too complicated to describe the distortion of whole image in single formula, the image can be divided into non-overlapping areas to be transformed individually. The image split can be given by a selection of CPs defining individual triangular areas, or it can be split according to regular mesh or the image content after previous segmentation. After separate application of transforms, discontinuities can arise at the borders. To achieve smoothness, higher-order transforms must be used and the most common is a usage of separable B-spline. [13]

3.4.5 Elastic Transformations

There is also a completely different approach to modelling distortions (especially local one, too complicated for description using parameters) and thus is elastic registration. It should be noted that these methods are not reliable for a strong multipixel distortions. [13]

In case of *elastic registration*, the image is perceived as a rubber surface and to be aligned with the reference image, it is bended and stretched, but as little as possible. External forces stretch the image and internal forces represents stiffness and smoothness constraints. When image deformations are very localized, elastic registration is not so successful, here fluid registration can be used. [41]

Fluid registration methods look at the picture as at a thick fluid flows to match

the reference image and to control the image transformation, a viscous fluid model is used. The disadvantage is blurring emerging during the registration process. [41]

One of the others non-rigid approaches is *optical flow based approach* estimating relative motion between images. [41]

3.5 Interpolation

Even if an image is treated as continuous in space, the whole process of digital image processing, obviously, takes place in discrete environment - a discrete grid of pixels. Once, a geometrical transformation is performed, a new set of spatial coordinates of pixels is obtained, but in many cases, these spatial coordinates are not integer values and therefore it falls outside the current regular rectangular grid of the original image. [13], [38] Thus a new, transformed and nonuniform grid arose, which leads to a necessity to find the as accurate as possible intensity values in the nodes of the original uniform grid, as in Fig. 3.3. To do so, the interpolation is needed. [21]

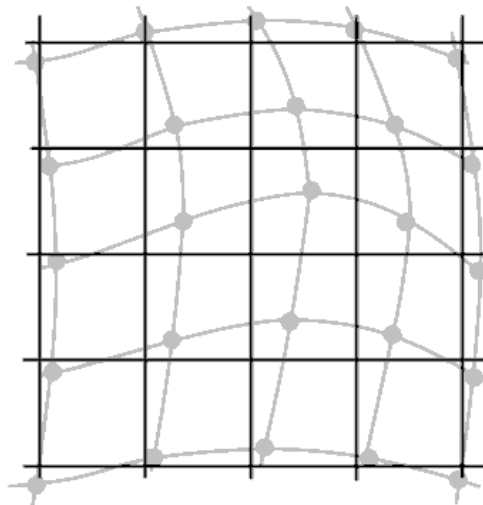


Fig. 3.3 Regular rectangular uniform grid shown in black, nonuniform transformed grid shown in grey. [21]

The most accurate interpolation method, which is theoretically able to completely restore an analog signal after sampling, is usage of (in case of images two-dimensional) *sinc function*. The function is the result of the Fourier transform of a rectangular pulse and is defined as:

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}. [2] \quad 3.17$$

The disadvantage (in fact practical inapplicability) of this function is its complexity and infinity (to count one output value theoretically infinite amount of input pixel values is needed), as shown in Fig. 3.4. [38]

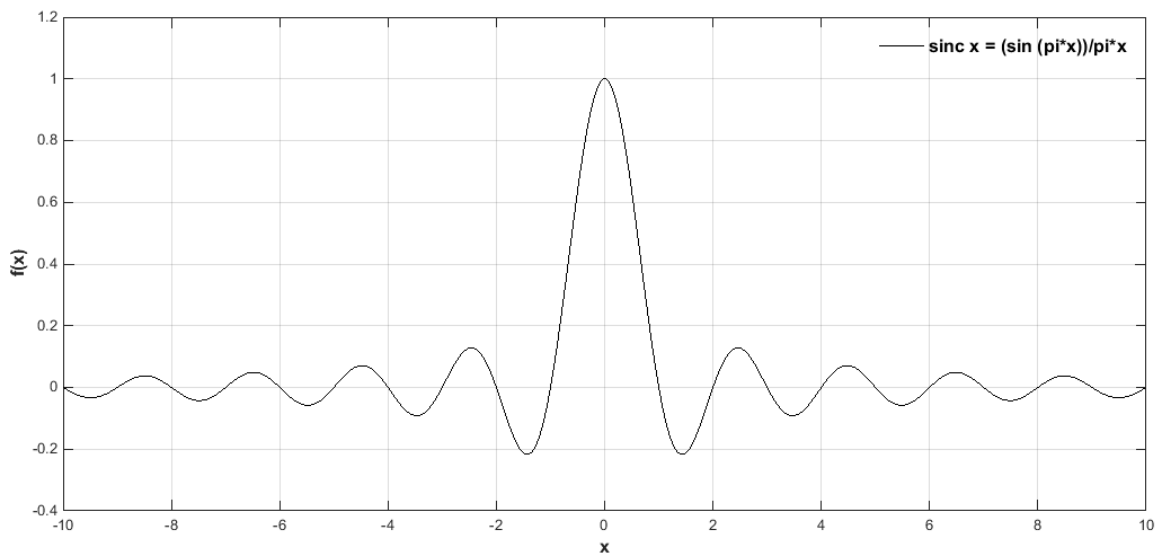


Fig. 3.4 Sinc function.

Thus, two main requirements when selecting the function are: minimization of amount of input values needed for calculation (to decrease computational and durational demands), but on the other hand the best possible approximation of sinc function (for the best possible accuracy). The most widely used compromise methods are nearest neighbor interpolation, bilinear and bicubic interpolation discussed below.

3.5.1 Nearest Neighbor Interpolation

The nearest neighbor interpolation method is the simplest, fastest, but not punctual method, therefore should not be generally used because it leads to artefacts formation. It can be used just for tentative image display or for oversampled images (much higher sampling rate than sampling theorem). [41]

The seeking value arises by simple assignment of the nearest known value, therefore very small difference between found subpixel positions ($A'/B'/C'/D'$) can lead to quite a big difference in output value simply selected out of four surrounding values ($A/B/C/D$). Thus, no new value is calculated, but the ultimate impact has only a single value, as can be seen in Fig. 3.5 a), which leads to “staircase”, not continuous, output function with deeply modified frequent content. Some pixels get lost and others are duplicated, sharpness is heavily reduced. [9]

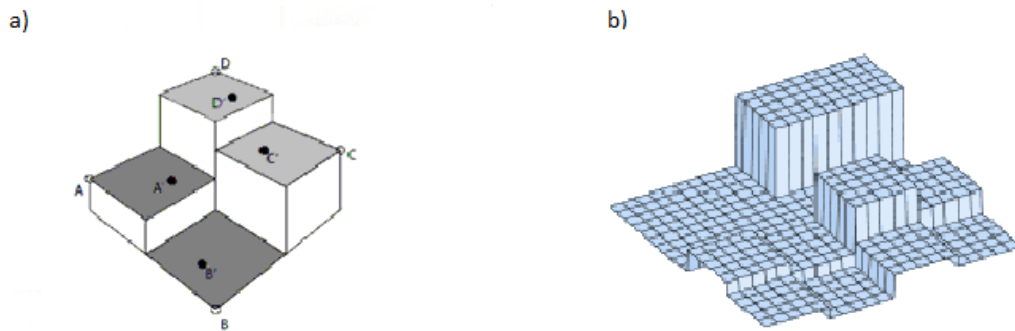


Fig. 3.5 Nearest neighbour interpolation method: a) schematic representation of the nearest neighbor interpolation of values outside a regular grid, [38] b) the “staircase” resulting image function. [9]

3.5.2 Bilinear Interpolation

It is possible to say, that the best compromise between accuracy and computational complexity is the bilinear interpolation. This method is extremely efficient and still the most commonly used one. [41]

The method is more demanding because of the need for a larger number of input values. In simplistic terms, there is created a curved surface out of four values of four ambient pixels (Fig. 3.6) and the required value at the corresponding position is assigned. The advantage is obvious, not just one value is reflected, but all four surrounding values affects the resulting number and the “staircase” character is smoothed. Mathematically, the function is continuous in the original space, but the derivatives are discontinuous at the borders. To achieve their continuity and to increase the sharpness of an image, it is necessary to use more sophisticated methods. [28], [10]

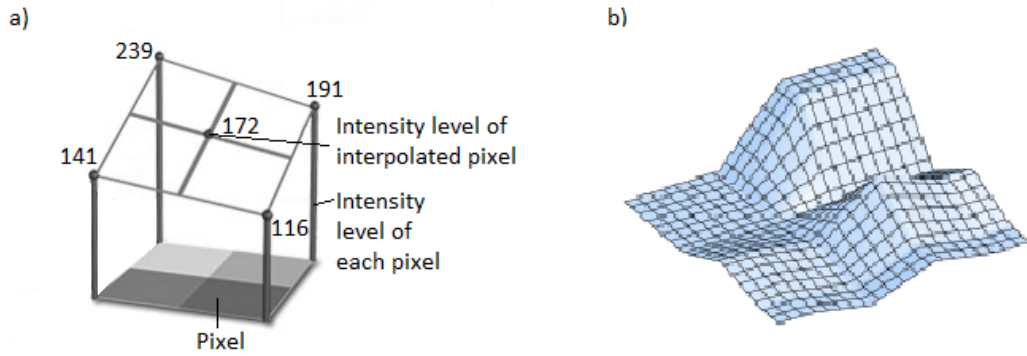


Fig. 3.6 Bilinear interpolation method: a) schematic representation of the bilinear interpolation of values outside a regular grid, [33] b) result image function. [9]

3.5.3 Spline Interpolation

A spline is a polynomial between each pair of tabulated points, guaranteeing global smoothness in the interpolated function up to some order of derivative. Splines tend to be more stable than fitting a polynomial, with less possibility of wild oscillations between the tabulated points. There exist several various splines such as linear, cubic, B-splines. [24]

Spline interpolations includes, for example, *bicubic spline* interpolation, which, as shown in the Fig. 3.7, considers the impact of sixteen surrounding pixels. Therefore, this interpolation increases the image sharpness (an interpolated function is continuous in both, the first and the second derivative), however, at the cost of increased computational complexity and especially duration. [24], [28]

In case that the geometric transformation leads to distinct enlargement of the image or a greater sharpness is required, the cubic interpolation is successfully applicable and recommended, but there still exists a risk of emergence of negative values, which is in case of intensity-based methods improper and undesirable. [41]

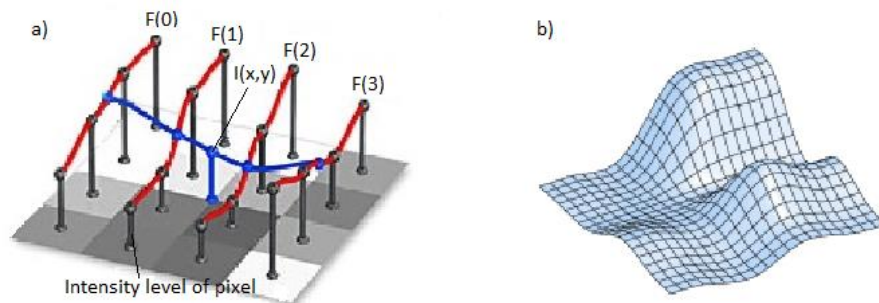


Fig. 3.7 Bicubic interpolation method: a) schematic representation of the bicubic interpolation of values outside a regular grid, [33] b) result image function. [9]

3.6 Similarity Criteria

To be able to sophisticatedly evaluate whether the parameters of the transformation in transformation vector α_0 were set correctly, it is necessary to determine the similarity between two images and that is what the similarity criterion (being punctual, global similarity criteria, the local ones are used mostly while working with CPs or in disparity map calculation etc.) serves to. Thus, global similarity is a function, which is lately optimized for obtaining the best possible results.

There are many different options and approaches to compare the similarities or specifying dissimilarities and the choice of the particular (dis)similarity criterion must respect the analysed problem to not being misleading and its suitability should be verified. [13]

Two basic types of the criterion are intensity-based and information-based global similarity criteria, the next group is formed of feature-based criteria. Having regard to the facts in the previous paragraph, first mentioned are used mostly for monomodal (images from just one imaging technique) image registration and the second for multimodal image registration. [37]

3.6.1 Intensity-Based Global Criteria

Intensity-based methods have greater computing demands; however, it is compensated by avoiding sensitive part of the feature-based techniques, which is a feature extraction stage. Other techniques are phase-correlation, Fourier-based or wavelet-based techniques. [14], [15]

Both input images, usually same sized, are transformed from matrices A and B to vectors a and b and compared by chosen similarity metrics as e.g. Euclidean distance or cosine criterion are.

Euclidean distance is the simplest criterion defined as the length of the difference vector between vector a and b , according to

$$C_E(\mathbf{a}, \mathbf{b}) = |\mathbf{a} - \mathbf{b}| = \sqrt{\sum_{i=1}^N (a_i - b_i)^2}, [13] \quad 3.18$$

where a_i is (i)-th element of a and b_i is (i)-th element of b . Obviously, if both input images are the same, their Euclidean distance is equal to zero and more different they are,

the greater the distance is. [37] If needed, the dissimilarities could be emphasized by also widely used criterion, the square of Euclidean distance, the *sum of squared differences* (SSD). In case of different size of compared images, both metrics can be normalized by N (number of pixels).

Another frequently used similarity metric is *cosine criterion* (or generally angle criterion) define as angle δ between the vectors a and b ,

$$C_A(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}\mathbf{b}}{|\mathbf{a}||\mathbf{b}|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2} \sqrt{\sum_i b_i^2}}. \quad [13] \quad 3.19$$

Although cosine takes values in range $\langle -1, 1 \rangle$, in case of cosine criterion, taken values are between $\langle 0, 1 \rangle$ because of nonnegativity of the pixel intensities. The advantage of this criterion is its robustness towards linear changes of contrast, unlike previously described Euclidean distance. While Euclidean distance marks two identical images, one of which has only linearly (even nonlinearly but monotonously) transformed contrast, as different, the cosine criterion is able to evaluate them as identical (because the angle between the vectors is not affected). The cosine criterion can be readily converted to another criterion as well. By subtracting the mean value from all its elements, the *norm-cosine criterion* is obtained,

$$C_{CC}(\mathbf{a}, \mathbf{b}) = \frac{\sum_i (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_i (a_i - \bar{a})^2} \sqrt{\sum_i (b_i - \bar{b})^2}}, \quad [13] \quad 3.20$$

where $\bar{a} = \frac{1}{N} \sum_i a_i$ and $\bar{b} = \frac{1}{N} \sum_i b_i$. In this case again, the differences between images are this way enhanced and the range is extended between $\langle -1, 1 \rangle$. [13]

One of the frequently used criterions is also *cross-correlation*, which is dot product $\mathbf{a}\mathbf{b}$. This criterion is simple but unreliable. More can be found in [13].

3.6.2 Information-Based Global Criteria

In many cases, as a different brightness range of registered images or multimodal imaging, it is not possible to use intensity-based global criteria and there it comes to information-based criteria.

The most common criterion is *joint histogram*. Simplistically, joint histogram (Fig. 3.8) is two-dimensional histogram expressing amount of the same couples of pixel intensities on matching positions in both input (same sized, grayscale) images. In Fig. 3.8, $q(r)$ is number of shades of grey in image A (B) and the count of couples increases

every time when in the same spatial position in A and B is a specific couple of grey shade i (in A) and k (in B). Thus, while the unimportant positional information is lost, the relation between pixel intensities in compared images is obtained (for identical images, all the values occur on the connector of position $(0,0)$ and (q,r) in the joint histogram). [13]

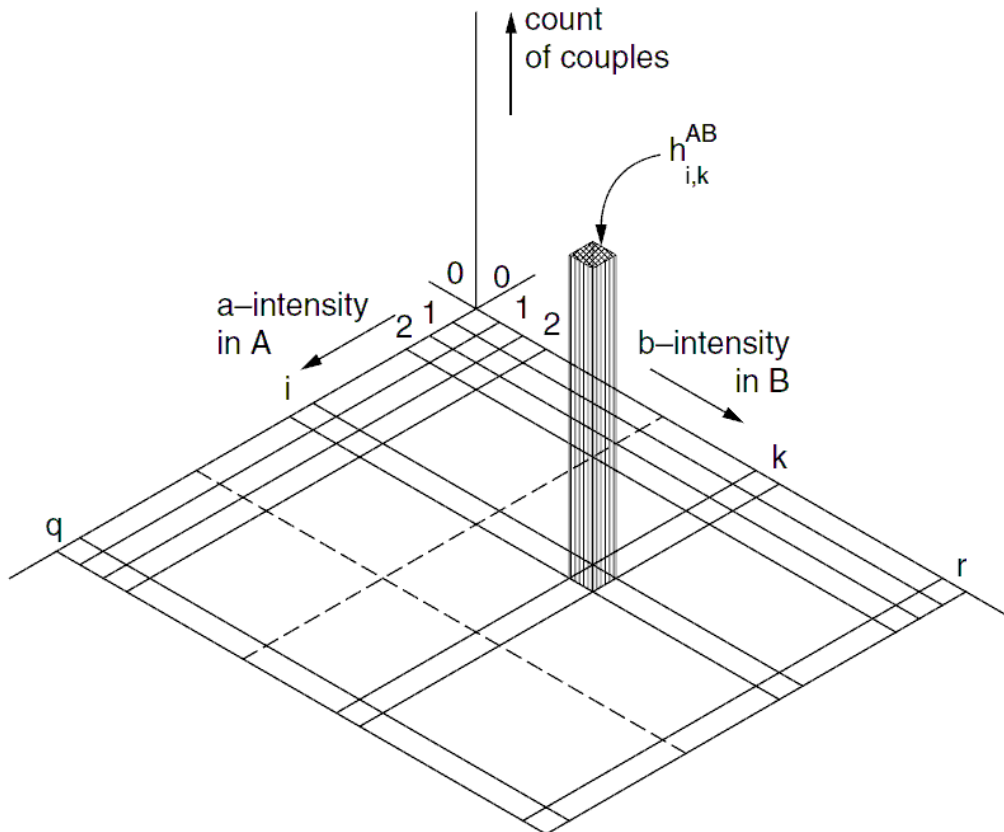


Fig. 3.8 Joint histogram of two images schematically (only a single histogram value out of all qr values is depicted). [13]

Another representative is **mutual information**. The information measure is entropy; entropy of images individually, H_A and H_B , and joint entropy H_{AB} . Mutual information itself, I_{AB} , is a difference between the sum of information individually and the joint information;

$$I_{AB} = H_A + H_B - H_{AB}. [13] \quad 3.21$$

Interpretation of mutual information is as follows, it expresses a degree of predictability of moving image A while knowing the reference image B . [37] An interdependence plays an important role; the more dependent the images are, the bigger is the predictability

of one from the second and the smaller is the union information. [13]

3.6.3 Feature-Based Methods

Feature-based methods are built on existence of characteristic, distinctive objects such as edges, line intersections, corners, clearly closed regions, etc. These objects are manually or automatically detected in both, reference and moving image, and their central points or other unambiguous points are set as control points. The correspondence between those corresponding CPs in both images is established and based on this correspondence, parameters of the mapping function are computed. That allows the transformation of the moving image. [41]

The biggest disadvantage of the method is either to detect the CPs manually, or to automatized their detection, which is considerably complicated with risk of errors.

3.7 Optimization

The actual image registration process itself involves searching for the most appropriate geometrical transformation – the best possible, optimal, transformation vector α_0 . This vector is iteratively changed to transform the moving image A to optimize (minimalize or maximize) the cost function based on the similarity criterion between images A and B . The dimension N of space where optimization takes place depends on number of parameters of geometrical transformation (in case of only rotation it is 1D, translation 2D etc.), so that with bigger number of parameters, the number of local extremes increases and the optimization approach must be adapted to the specific task character. [38]

Individual optimization algorithms are described in the following chapters from resources [25] and [34].

3.7.1 Deterministic Algorithms

Gradient descent is a numerical optimization method, where it is iteratively proceeded in the direction of the largest gradient of the function. This method is suitable for simple continuous functions, where the calculation is fast. For more complex functions tends to get stuck in local extremes.

Gradient descent is a basis for *Newton's method*, which interspaces a points neighbourhood with a quadratic function, which's minimum (maximum) forms a new estimate for the next iteration. The calculations in each iteration are quite extensive, so it is suitable for simple functions and if the first estimate is not close to the optimum,

convergence is not guaranteed. In MATLAB, this method is implemented in `fminunc` function.

Exhaustive search calculates all possible solutions (all the combinations of the transformation coefficients) and selects the optimum. If the task is not limited, it is not applicable.

However, the disadvantage of deterministic algorithms is a necessity of existence of the first and second derivative (and there exist more complications for more complex functions). [34]

3.7.2 Stochastic Algorithms

There is no general deterministic algorithm for global optimization and thus, **stochastic algorithms** are usually used. These algorithms heuristically searched the stated space and cannot guarantee the exact solution finding, but can find a practically good enough solution in acceptable time. Completely random search does not lead to results in a real-time, so there are systematic methods searching the space in various ways without calculating derivatives. [25]

Comparative methods such as Box-Wilson or simplex compare functional values of several points and in each iteration removes the worst solution and choose a new point with supposed better functional value. Thus, they are pointing towards the optimum, computationally are not demanding and are more successful for local optimization. The **Box-Wilson** compares functional values in $2^N + 1$ points. In 2D there are 5 points, the first is randomly chosen and 4 next form a square around. In the next iteration, the point with the best functional value is new central point. When the central point is the point with the best value, it is the optimum searched. The **simplex** method creates a formation with $N+1$ vertices in N -dimensional space (a triangle in 2D). The principle is similar to the previous method. When the first point estimation is made, the function values are calculated at all points and points are marked as B = Best, G = Good, W = Worst. W point is eliminated and replaced with another point of better functional value. To set a new triangle, several operations have to be made. First is reflection, R for Reflection replaces W, and if R's functional value is better than W's, expansion into point E is made (Fig. 3.9 a)). New simplex triangle is BGR or BGE. If R is not better than W, contraction into C_1 and C_2 occurs as in Fig. 3.9 b). Better of these two points is chosen and stated as C. If the functional value in C is better than in W, BGC is the new simplex triangle, otherwise, reduction is made, so W is replaced by S and G by M, see Fig. 3.9 c). Whole process is iteratively repeated and the optimum in each iteration is point B or the centre of mass of BGW triangle. This method is implemented in MATLAB to

fminsearch function, which is currently used for the optimization in the proposed method.

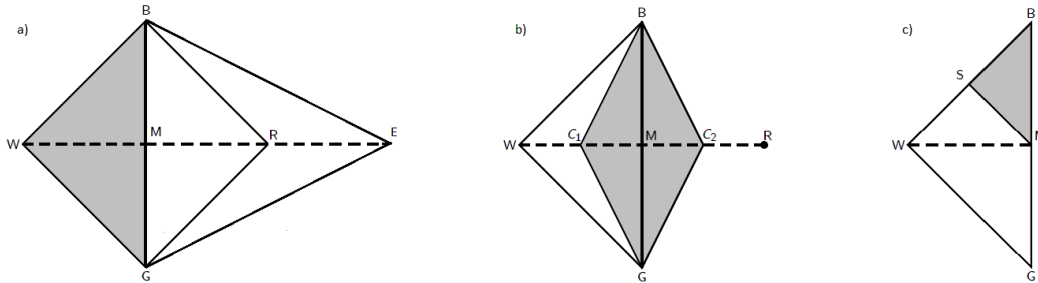


Fig. 3.9 The course of determining the new point using the simplex method. a) reflection and expansion, b) contraction, c) reduction. [25]

Controlled Random Search reminds the simplex method, but it works with population of X individuals in N -dimensional space, where $X \gg N$ and simplex is chosen from those X points.

Simulated Annealing algorithm is based on a real annealing process and with a certain probability in short-term accepts even worse solution, which provides the ability to escape from a local minimum.

3.7.3 Genetic Algorithms

The basic idea of genetic algorithms is an analogy with evolutionary processes in biological systems, namely, that only the best individuals of the population survive and proceed to the next step – the new generation. There exist many genetic algorithms based on real biological systems such as modelling of the fireflies' behaviour, pheromone tracks of ants, behaviour of bees, grey wolves or cuckoos.

Initial population is randomly created and a measure of the **quality** of every individual is determined (in case of maximization, it is directly the functional value). Also, a **termination condition** is defined; this can be a maximum number of iterations, results stagnation etc. Then, a selection, mutation, crossover, elitism begin.

The **selection** is performed for example as two individuals duel, where the better one proceeds into a new generation. During **crossovers**, parts of the individuals are randomly exchanged. In a **mutation**, there is set a probability, that a random change of the individual occurs (e.g. randomly changed one of the transformation coefficients). Mainly due to mutations, it is possible to move the solution out of a local optimum. **Elitism** is a simple passage of the best individuals to a new generation.

Till the termination condition is not accomplished, it is iteratively repeated. After the termination, the best individual of the population is chosen and this is the global optimum.

4 IMAGE ACQUISITION

The image acquisition was performed on a transmission electron microscope Talos L120C for Materials Science applications. The high-contrast, high-resolution, 120 kV, multi-discipline (2D, 3D imaging of beam-sensitive materials such as cells, polymers, and soft materials at ambient and cryogenic temperatures, in situ dynamic observations, electron tomography etc.) S/TEM with the constant-power objective lenses and low-hysteresis is equipped with the fast $4k \times 4k$ Ceta 16M camera, which provides large field-of-view. [30]

The key S/TEM specifications are mentioned in the Tab. 4.1. The Talos L120C has many other features and benefits but they are not key to this thesis' measurements.

Tab. 4.1 The most important specifications of Talos L120C TEM for Materials Science. [30]

	Parameter	Values
TEM	High tension (min steps)	20 – 120 kV
	Magnification range	25x – 650kx
Camera	Ceta CMOS	16 Mpx
	Sensor	4,096 x 4,096 1px = 14 μm

All the images were captured at a screen current of approximately 1,5 nA, the screen current in which the camera is not endangered. The magnification was approximately between 36kx and 510kx. The dose rate was then between $50 \text{ e}/\text{\AA}^2\text{s}^{-1}$ at the smallest measured magnification and $7200 \text{ e}/\text{\AA}^2\text{s}^{-1}$ at the largest measured magnification ($\text{\AA} = 0,01 \mu\text{m}$). The magnification and the connected dose rate are mentioned in each image's filename. A sampling was 1, every value; therefore, the images consist of 4096×4096 pixels, their size is 16 MB and are saved in a TIFF format.

To avoid sample damage when measuring biological samples, it is crucial not to exceed the dose rate of $80 \text{ e}/\text{\AA}^2\text{s}^{-1}$. To ensure that this value is not exceeded, it is necessary to keep an approximately constant screen current value 1,5 nA even for large magnifications. However, this leads to a significant decrease of the signal-to-noise ratio (SNR) and results in a low image quality. To sum it up, good SNR is in contradiction with preservation of biological samples. Although the biological samples were not measured during the master's thesis, it is necessary to know the results of the image registration also for the conditions described above (low SNR).

There were two measured calibration samples. The first was a carbon film with a golden shadowing of waffle pattern gratings made on a copper grid. The second was a carbon film shadowed with gold with graphitized carbon particles. The particles viewed over the holes in the film may be used for assessment of factors limiting the microscope performance. [31] The examples of the samples' images follow in the Fig. 4.2 and Fig. 4.3.

The geometric distortion mapping requires an acquisition of the image pairs using the same microscope setup. First, the magnification was set and the desired pattern or element in the sample was found, focused and acquired. When the reference image was captured, a small beam shift was performed in x, y or both axes and the second image was acquired under the same conditions. Consequently, these two images only differ by passing through different parts of the lens, which results in a changed distortion. Another consequence is that the second image is shifted to the first one due to a selective aperture which selects another part of the beam. A schema describing this procedure is in the Fig. 4.1; the scheme is without magnification.

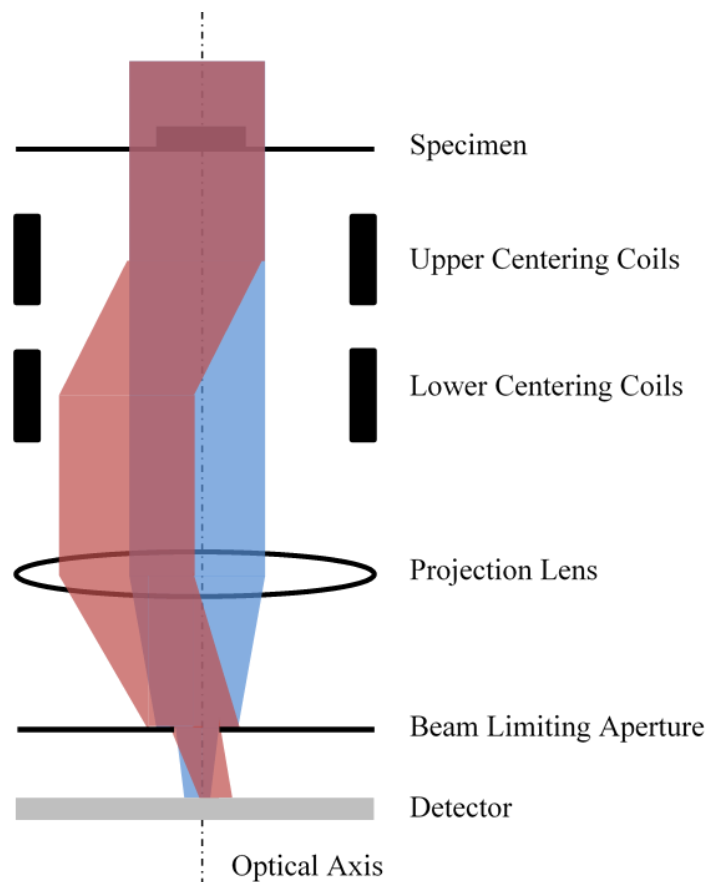


Fig. 4.1 Schematic representation of the origination of a pair of images. One of the images is created in a way in blue, the other in red. The scheme is without magnification.

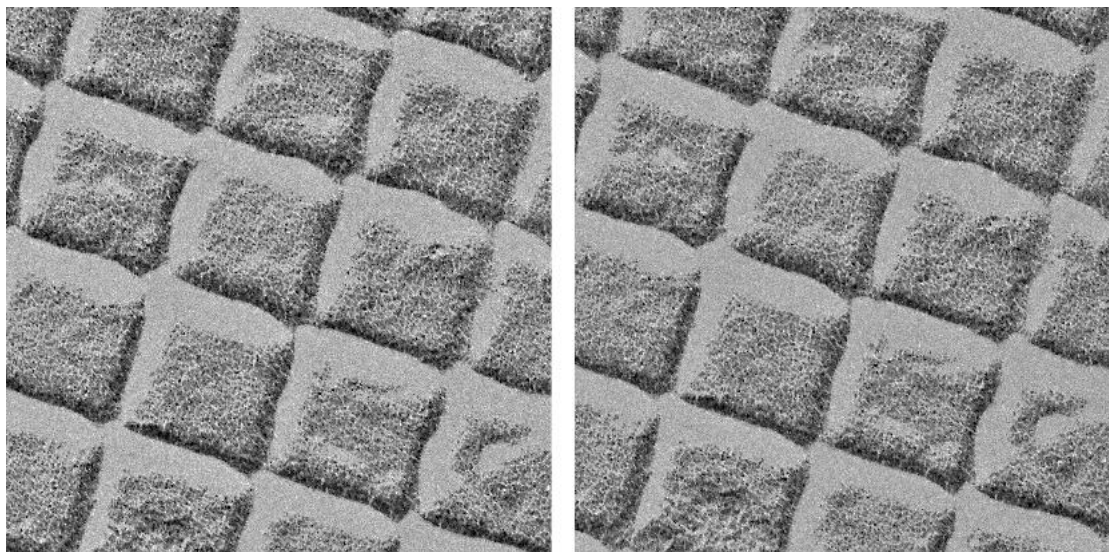


Fig. 4.2 An example of the first described calibration sample (a carbon film with a golden shadowing of waffle pattern gratings made on a copper grid) and an example of a shift made.

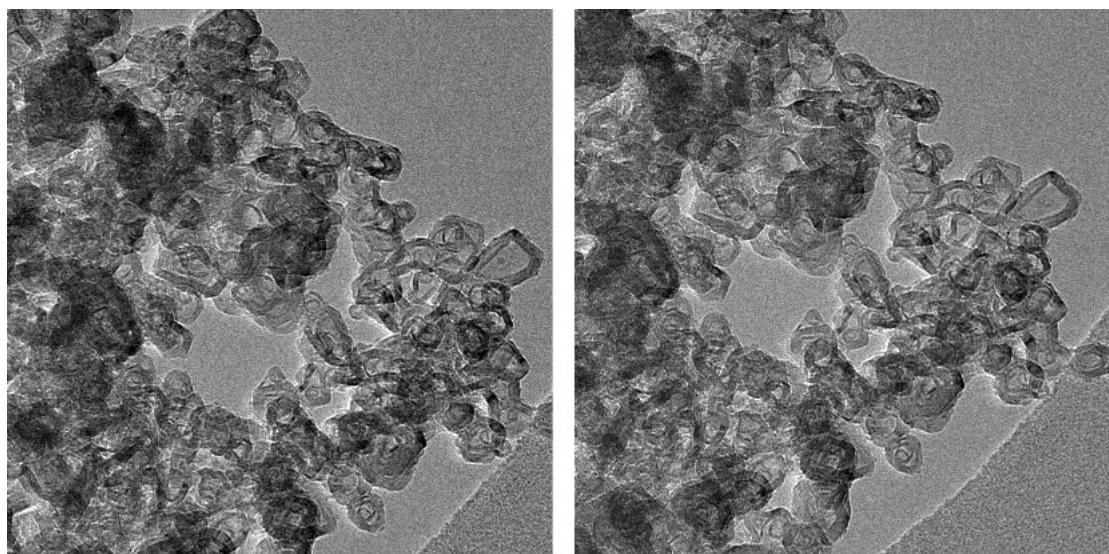


Fig. 4.3 An example of the second described calibration sample (a carbon film shadowed with gold with graphitized carbon particles) and an example of a shift made.

5 THE PROPOSED AND IMPLEMENTED METHOD OF IMAGE REGISTRATION

Images were captured as described in the previous chapter 4 Image Acquisition. Subsequently, the image registration itself comes in a row. The whole process leads to description of the distortion field. In this chapter, the proposed and in MATLAB implemented method is introduced and described.

5.1 Model of the Distortion

Based on the literature research and test results, the selected method to achieve the goals of the thesis is a usage of the model of the radial symmetric distortion. Two main aspects of the distortion correction exist. The first is a mathematical model of the geometrical distortion, and the second are the model parameters used. [35]

The general description of a geometrical distortion is described in chapter 2.2.1 Mathematical Description of Geometrical Distortion. To model the geometrical distortion the expanded equation 2.3 was used to count an impact of the real microscope system which can lead to not perfectly radial distortion and shrinking effect. The squeeze term s of possible shrinking effect was added in equation 5.1 and a fact that the distortion can be non-radial were considered in factors λ_x and λ_y in equation 5.2:

$$\begin{pmatrix} \widehat{p}_x \\ \widehat{p}_y \end{pmatrix} = \begin{pmatrix} p_x \cdot (1 + \kappa_1 r + \kappa_2 r^2) \\ p_y \cdot (1 + \frac{\kappa_1}{s} r + \frac{\kappa_2}{s} r^2) \end{pmatrix}, [35]. \quad 5.1$$

$$\begin{aligned} \widehat{p}_x &= f_x(p_x) = p_x(1 + \lambda_x p_x^2) \\ \widehat{p}_y &= f_y(p_y) = p_y(1 + \lambda_y p_y^2), [35]. \end{aligned} \quad 5.2$$

Eventually, by combination of previous equations 2.3, 5.1 and 5.2, the final model is formed:

$$\begin{pmatrix} \widehat{p}_x \\ \widehat{p}_y \end{pmatrix} = \begin{pmatrix} p_x \cdot (1 + \kappa_1 p_x^2 + \kappa_1 (1 + \lambda_x) p_y^2 + \kappa_2 (p_x^2 + p_y^2)^2) \\ p_y \cdot (1 + \frac{\kappa_1}{s} p_x^2 + \frac{\kappa_1}{s} (1 + \lambda_y) p_y^2 + \frac{\kappa_2}{s} (p_x^2 + p_y^2)^2) \end{pmatrix}, [35]. \quad 5.3$$

p_x represents the x coordinates of the original image and \widehat{p}_x the transformed x coordinates. The same is true for y coordinates.

5.1.1 Implementation of the Distortion Model to MATLAB

When the previously stated equations are rewritten into matrix syntax, a product of two matrices is obtained for both, x and y coordinates:

$$\begin{pmatrix} \widehat{p}_x \\ \widehat{p}_y \end{pmatrix} = \begin{bmatrix} 1 & \kappa_1 & \kappa_1 & \kappa_1 \cdot \lambda_x & \kappa_2 & 2 \cdot \kappa_2 & \kappa_2 \end{bmatrix} * \begin{bmatrix} p_x \\ p_x^3 \\ p_x \cdot p_y^2 \\ p_x \cdot p_y^2 \\ p_x^5 \\ p_x^3 \cdot p_y^2 \\ p_x \cdot p_y^4 \end{bmatrix} \quad 5.4$$

$$\begin{pmatrix} \widehat{p}_y \end{pmatrix} = \begin{bmatrix} 1 & \frac{\kappa_1}{s} & \frac{\kappa_1}{s} & \frac{\kappa_1}{s} \cdot \lambda_y & \frac{\kappa_2}{s} & 2 \cdot \frac{\kappa_2}{s} & \frac{\kappa_2}{s} \end{bmatrix} * \begin{bmatrix} p_y \\ p_y \cdot p_x^2 \\ p_y^3 \\ p_y^3 \\ p_y \cdot p_x^4 \\ p_y^3 \cdot p_x^2 \\ p_y^5 \end{bmatrix}$$

which expressed in terms of MATLAB looks as follows:

```

coeffs = [1, kappa1, kappa1, kappa1*lambdax, kappa2,
          2*kappa2, kappa2;
          1, kappa1/s, kappa1/s, (kappa1/s)*lambday,
          kappa2/s, 2*(kappa2/s), kappa2/s];

```

```

nx = [X(1, :); ...
      X(1, :).^3; ...
      X(1, :).* (X(2, :).^2); ...
      X(1, :).* (X(2, :).^2); ...
      X(1, :).^5; ...
      X(1, :).^3.* (X(2, :).^2)
      X(1, :).* (X(2, :).^4)];
ny = [X(2, :); ...
      X(2, :).* (X(1, :).^2); ...
      X(2, :).^3;
      X(2, :).^3;
      X(2, :).* (X(1, :).^4); ...
      X(2, :).^3.* (X(1, :).^2); ...
      X(2, :).^5];

```

In the matrix $X(1, :)$ must be saved a vector of all the x pixel coordinates and in the $X(2, :)$ all the y pixel coordinates.

The coefficients κ_1 and κ_2 corresponds to the coefficients κ_1 and κ_2 from the chapter 2.2.1 as well as coefficients λ_x , λ_y and s to the λ_x , λ_y and s from the previous chapter 5.1.

Not only for registering TEM images, but anywhere a position of the optical axis is unknown or where its position to the image centre may change, it is necessary to add another parameter that takes this into account. The position of the optical axis is at the same time the centre of the distortion. This parameter represents a shift of the moving image in a large auxiliary deformed field, which is deformed in the form of a barrel distortion or a pin cushion distortion, according to the distortion coefficients described above. DF_x is a shift in the deformed field in x axis and DF_y in y axis. The resulting matrices then have the following form:

```

nx = [X(1, :) + DFx; ...
      X(1, :).^3; ...
      X(1, :).* (X(2, :).^2); ...
      X(1, :).* (X(2, :).^2); ...
      X(1, :).^5; ...
      X(1, :).^3.* (X(2, :).^2)
      X(1, :).* (X(2, :).^4)];

```

```

ny = [X(2, :) + DFy; ...
      X(2, :) .* (X(1, :).^2); ...
      X(2, :).^3;
      X(2, :).^3;
      X(2, :) .* (X(1, :).^4); ...
      X(2, :).^3 .* (X(1, :).^2); ...
      X(2, :).^5]; .

```

The Fig. 5.1 illustrates an example of how an auxiliary deformed field may look and how may look the resulting image shifted therein. The main purpose is to show how much the position relative to the optical axis affects the image.

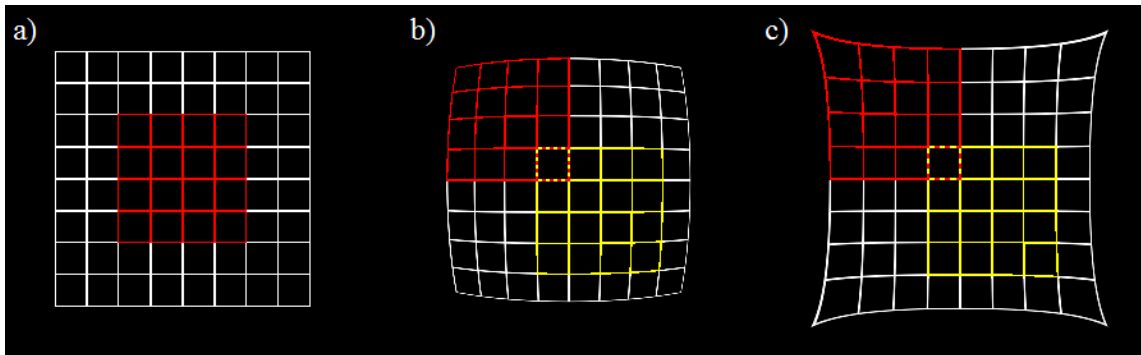


Fig. 5.1 An example of a deformed field. a) An original image in red. A created auxiliary field before a deformation in white. b) Barrel distortion; an auxiliary deformed field in white. An example of two of many variants of the differently distorted original image in red and yellow. c) Pin cushion distortion; the same example for pin-cushion distortion in red and yellow.

When both the coefficients and the coordinate products are ready, the matrices can be multiplied:

```

xp = coeffs(1, :) * nx;
yp = coeffs(2, :) * ny; .

```

There are two more parameters which were not mentioned before but could influence the model later; parameters T_x , T_y . As described in the chapter 4 Image Acquisition, when capturing images to illustrate a geometric distortion, there was performed a shift between a pair of images. This shift is simulated by the T_x and T_y translation parameters. They are embedded into the algorithm later, as described in the chapter 5.2.3 Transformation Function. Altogether, there are 9 parameters for modelling and correcting the distortion. These parameters form a 9D space where a solution is searched for. What is the effect of each parameter on an image is illustrated in the supplement A Parameters' Effect to an Image.

5.2 GA Function

The following paragraphs describe a principle and a process of the proposed method. The method is based on a continuous genetic algorithm (which is briefly described in the chapter 3.7.3 Genetic Algorithms), therefore, the function will be further referred to as a GA function.

5.2.1 Origin

The whole algorithm is initialized from the function called `OriginFunction.m`, whose simplified diagram illustrates the following Fig. 5.2.

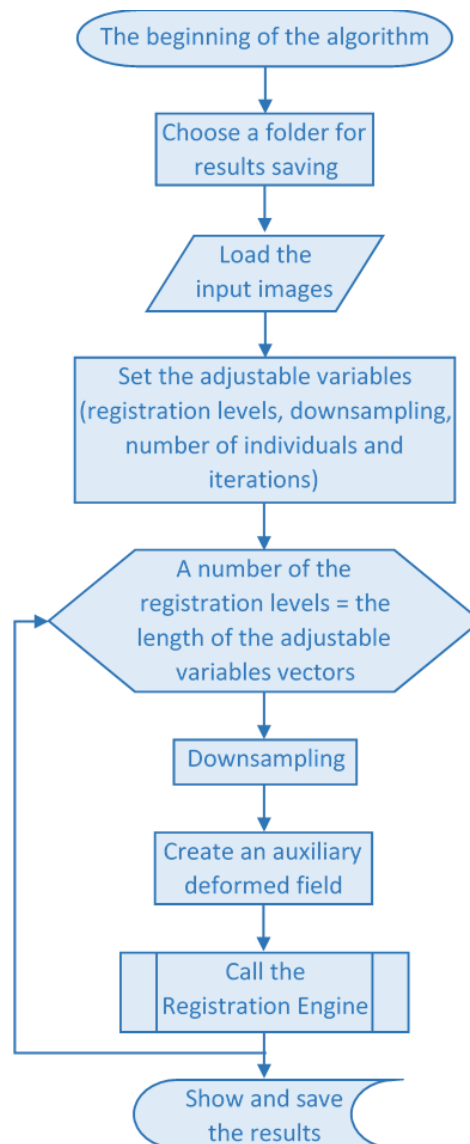


Fig. 5.2 Schema of the main function.

First of all, a folder where the registration results will be saved is selected by a user in an interactive window. Then a performance (time efficiency) measurement is launched by a `tic` command. When the pair of images intended for the registration is loaded and adjusted to the required double precision data format, the registration process can start. The details of images' loading and function's setting can be found in supplement B How to Control the Program.

As already mentioned, the original images from a microscope consist of a large number of samples (4096 x 4096 pixels). Therefore, to avoid a stuck of the function in local minima and to speed the registration up, it is appropriate to downsample these images for the first, basic registration. That is why a scale space approach was used and the registration passes of several levels. A number of the levels and a downsampling force is optional – a user enters the required number of the registration levels. This is done into a variable `dsVec` (downsampling vector), by specifying how many samples will be dropped in which registration level; e.g.:

```
dsVec = [32, 16, 8, 1];
```

Such a vector means that there will occur four registration levels and in an original image of 4096 x 4096 pixels, in the first registration level, every 32nd sample will be preserved – the image will be downsampled by 32 to 128 x 128 samples. In the next level, it will be upsampled to 256 x 256 samples, then to 512 x 512 samples and finally, when the registration is already well, 4096 x 4096 samples.

As said above, the method is based on a genetic algorithm. Thus, the right solution of the image registration, the global optimum, is found using a population of individuals. Using these individuals, a 9D space of parameters is searched through. In each level of the registration, there can be used a different number of the individuals of the population. Likewise, a different number of iterations may occur in each registration level. These two variables, the number of the individuals `nInd` and the number of the iterations `nIter`, are adjustable by a user. Obviously, because of an increase of a computational complexity (a decrease of a time and space efficiency) with the increasing number of the image samples, it is appropriate to choose higher numbers of individuals and iterations in the first levels of the registration and decrease it in the subsequent levels. An example follows:

```
nIter = [12, 10, 8, 4];
```

```
nInd = [20, 16, 10, 8];
```

in the first registration level, for 128 x 128 samples, 12 iterations will take place using 20 individuals etc.

After the image downsampling, the auxiliary variables are created. One of them is mentioned auxiliary deformed field simulating different positions of the distortion centre. At the beginning, it is necessary to normalize the image coordinates to have the [0,0] coordinate in the exact middle of this deformed field. (It is considered as a pixel where optical axis passes through and the distortion is zero). The x and y pixel coordinates are saved as vectors into a matrix X. This will be needed for later calculation of interpolation.

5.2.2 Registration Engine

After the initial preparation, it is possible to start the registration itself. The genetic algorithm is implemented in a function named `RegistrationEngineGA.m`, whose flowchart is in the Fig. 5.3.

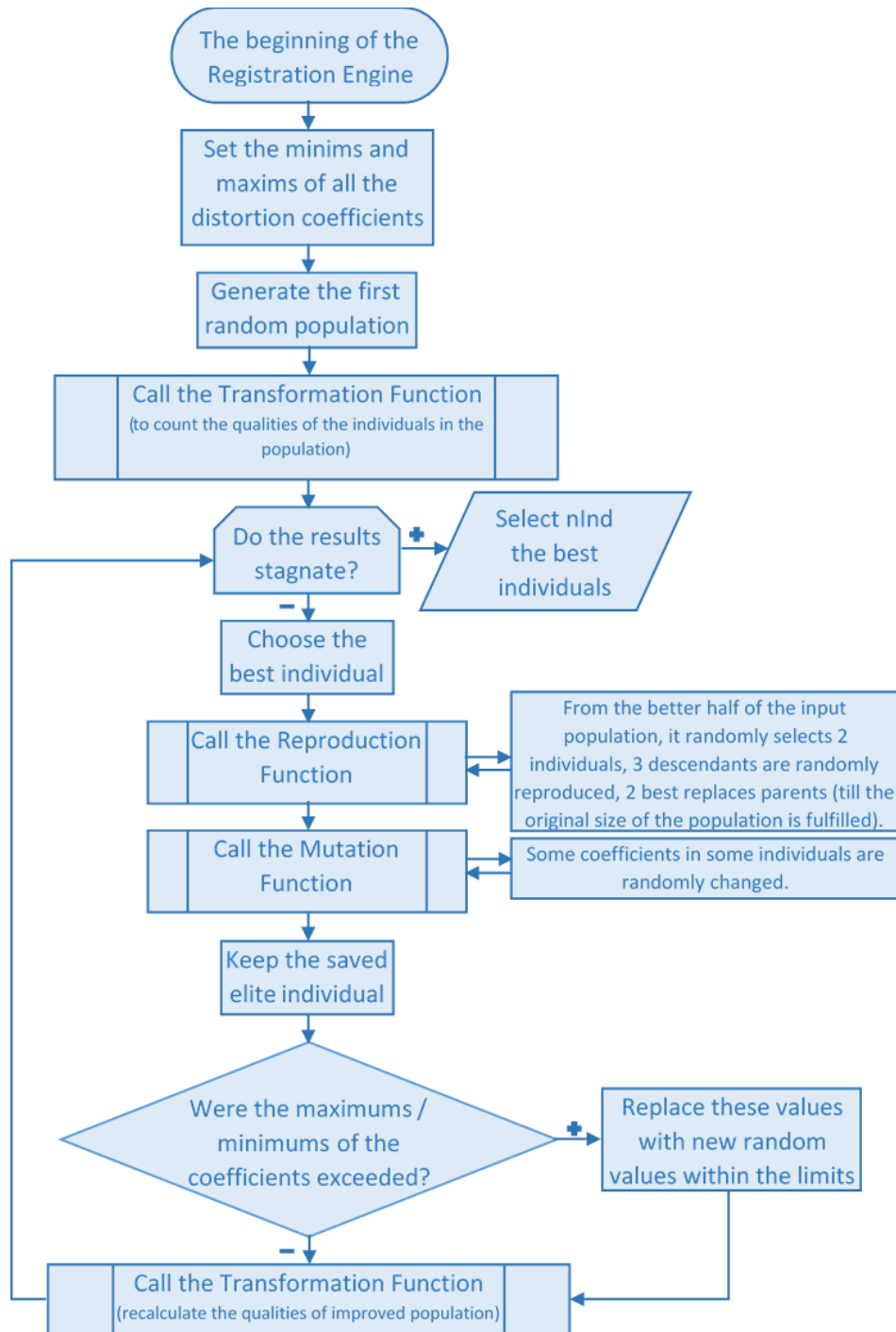


Fig. 5.3 Schema of the Registration Engine Function.

When the registration starts, the genes (bits) of the individuals of the first generation are generated randomly. Each individual of the generation is performed as a vector, where each number represents one of the transformation parameters. An example of one random individual from a random population is:

```
ind =
    12.5889 % DFx
     6.2296 % DFy
    -0.0368 % kappa1
     0.1508 % kappa2
     0.3517 % lambdax
     0.1622 % lambday
     0.1067 % s
     6.2133 % Tx
     4.9324 % Ty.
```

A population is a matrix where one individual is one row in such a population matrix.

Before the random generation of individuals, it is appropriate to set limits of all the transformation coefficients. The individuals' bits are then generated from the specified range. This determination can greatly help and speed up the global optimum search. However, it may also avoid the function from finding a global optimum if set incorrectly. Therefore, it is necessary to set these limits carefully; according to input knowledge or after testing the impact of individual coefficients on a particular image. During the development of the function, they were experimentally determined (more in the chapter 7 Testing and Optimization), but a user can modify them easily.

Once the first random generation is created, it is necessary to determine a quality of each individual representing a possible solution. For this purpose, a transformation function `TransformationFunctionGA.m` is called. The function's input is the generated population of the individuals and it returns the qualities of all the individuals in the population. The quality of each individual is measured using a cosine criterion between the reference and the moving image. For details, see the next chapter 5.2.3 Transformation Function.

At this point, it is possible to start the evolution. The process is terminated, when the number of iterations set earlier in the `nIter` variable is reached or when the results start to stagnate.

To perform elitism, a variable `elite` must be set to 1 (for any other number elitism do not occur). The best individual from the population, according to the calculated

quality, is found and saved into a variable `elit`.

A reproduction is done by calling the `Reproduction.m` function. Inside this function, the individuals are first sorted according to their quality from the best to the worst. A selection is then simulated; only a better half of the individuals get the opportunity to participate in the reproduction. Out of the better half of the population, two individuals are selected and these individuals reproduce together. For a greater diversity of the descendants, repeated reproduction of the same pair of two individuals is avoided by a simple condition. Of course, one individual cannot be selected twice in one iteration and reproduce himself. After selecting the parents, three descendants are created and their quality is calculated (again using the transformation function `TransformationFunctionGA.m`). These descendants are then sorted by quality, and two better descendants advance to the next generation. After the corresponding number of reproductions, when the new generation is as big as the original one, the reproduction ends and the new generation becomes a new population. This new population is subject to further modifications back at the `RegistrationEngineGA.m`.

A mutation follows the reproduction. The mutation is a main modification that may help the algorithm get out of a fall to a local minimum. Mutations occur in a mutation function `Mutation.m`. The probability of the mutation is determined in an adjustable variable `mutProb` (in a range 0 – 1). In the function, a random number is generated for each bit of all the individuals in the population. If this number is smaller than the set mutation probability, a new random number is generated for the particular individual's bit (a new transformation parameter is tried). Such a modified population returns to the `RegistrationEngineGA.m` and an elitism is applied.

During the elitism, the first individual in the population is replaced by the previously found and saved elite individual `elit`, to avoid losing this possible good solution.

In such a modified population, the values exceeding the predetermined `DFx`, `DFy` and `Tx`, `Ty` ranges are replaced by new randomly generated numbers from the given ranges.

The qualities of the individuals of the modified population must be recalculated (`TransformationFunctionGA.m`) and the best individual's quality is saved into a variable `fbest`.

Before proceeding to the next iteration, it is necessary to assess whether the results stagnate or not. At the beginning, a variable `stagnate` is set to zero, and a maximum

number of iterations with repeated result's stagnation is set to a variable `stagnateCeil`. The difference between the best stored quality from previous iteration `fbest` and the best individual's quality in a current iteration is calculated. If this difference is lower than the before set stagnation threshold `stagnateDiff`, the results are labelled `stagnant` and the `stagnate` variable is incremented by one. Once the `stagnate` variable reaches the set `stagnateCeil` limit, this registration level ends without further iterations.

At the end of the iteration, an iteration counter (`ite`) is also incremented by one and a proceeding to the next iteration follows.

After all the iterations, the individuals who advance to the next registration level must be selected. The number of these individuals is given by the specific value in the vector `nInd` (defined at the beginning of the algorithm). Therefore, the improved population is sorted according to the quality and `nInd` best individuals advance to the next registration level. Then the algorithm returns to the original function `OriginFunction.m`.

It is important to note that after obtaining the results of the current registration level, it is necessary to recalculate the coefficients related to a certain number of the pixels. And these are the T_x and T_y coefficients. These coefficients must be increased accordingly to have the same meaning and effect after upsampling.

5.2.3 Transformation Function

As has been mentioned several times, the transformation function is called for the calculation of the individuals' qualities. Its flowchart follows in the Fig. 5.4. The input of the function is the population. The quality of each individual, the quality of possible solution, is measured as a cosine criterion between the reference and the moving image.

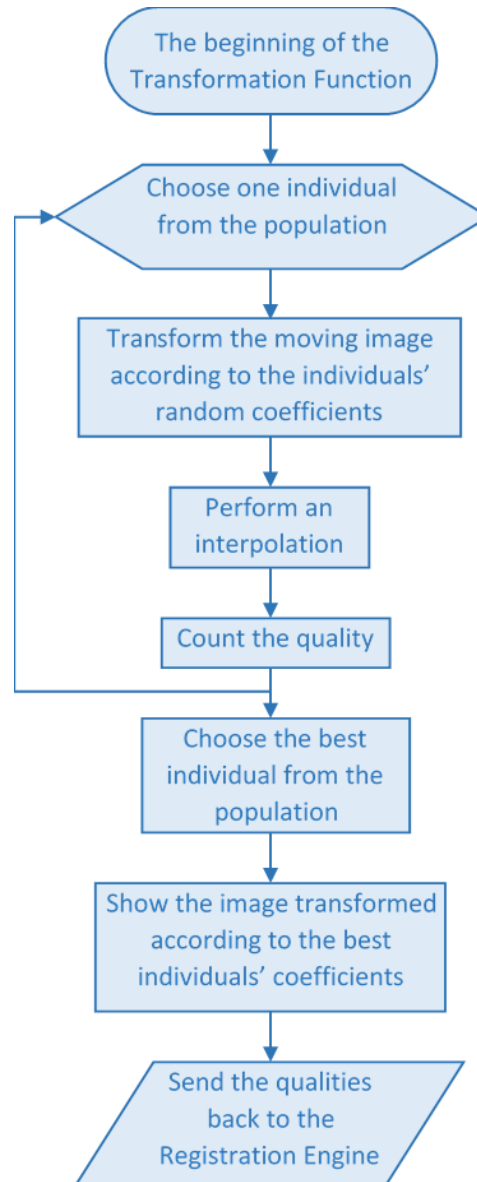


Fig. 5.4 Schema of the Transformation Function.

Above all, in this function it is necessary to avoid the possibility that images will not fit into the auxiliary deformed field after a translation. Therefore, the translation values that could cause this are replaced by the limit values possible for the movement in the deformed field.

The function then picks one individual after another from the input population. One individual (one row) in the form of all the transformation coefficients represents one possible solution of the image registration. The principle of the function is that it transforms regular coordinate grid using these parameters and create the deformation field.

The transformation function proceeds according to the respective equations:

$$\begin{aligned}
 nx &= [X(1, :) + DFx; \dots \\
 &\quad X(1, :).^3; \dots \\
 &\quad X(1, :).*(X(2, :).^2); \dots \\
 &\quad X(1, :).*(X(2, :).^2); \dots \\
 &\quad X(1, :).^5; \dots \\
 &\quad X(1, :).^3.*(X(2, :).^2); \dots \\
 &\quad X(1, :).*(X(2, :).^4)]; \\
 \\
 ny &= [X(2, :) + DFy; \dots \\
 &\quad X(2, :).*(X(1, :).^2); \dots \\
 &\quad X(2, :).^3; \\
 &\quad X(2, :).^3; \\
 &\quad X(2, :).*(X(1, :).^4); \dots \\
 &\quad X(2, :).^3.*(X(1, :).^2); \dots \\
 &\quad X(2, :).^5];,
 \end{aligned}$$

already and more broadly explained in the chapter 5.1.1 Implementation of the Distortion Model to MATLAB.

The coordinate transformation is followed by selecting a position in the deformed field. This field is several times larger than the image size, and the selection of the particular position (the distortion shape), in this field depends on the DFx , DFy parameters.

After this process, the interpolation is performed and the solution quality criterion, a similarity criterion, is calculated. As an interpolation method, the linear interpolation has been selected for its simplicity, sufficiency, and no emergence of negative values. In this case, cosine criterion was selected as the quality criterion. A microscope software includes image histogram optimizing function for automatic brightness and contrast adjustment. However, if this feature is disabled or the images are of another modality, registration works. But to count and compare the qualities of image registrations, a cosine criterion was selected because of its robustness towards linear changes of contrast (unlike SSD criterion).

In order to monitor the progress of the registration and an improvement of the results in each iteration of each registration level, the best solution found for each iteration is displayed and saved at the end of the transformation function. It is important to note, that the projected result does not necessarily have to be the best result ever found. Only the best result of the given iteration. On the other hand, the population continually

improves, thus, usually it is the best result overall, but it may not always be the case.

The vector of the population's qualities is returned to the `RegistrationEngine.m`.

5.2.4 Saving Results and Visualization

At the end of the program, visualization takes place. Of course, the result of the registration is shown, but also the resulting x and y transformation matrices and the displacement field is displayed.

There are used two ways of plotting the displacement field. The first is a usage of the quiver function according to the resulting transformation matrices `xpReshRES`, `ypReshRES` (counted as a difference between the original rectangular grid of the image and the resulting counted deformed grid):

```
quiver(xpReshRES, ypReshRES, 0.5);
```

the last parameter of the function is a density of rendered arrows. The arrows represent the displacement vectors at each pixel showing the shift of the pixel between the fixed image grid and a corresponding location in the resulting registered moving image.

The second used way is a deformation of a pre-arranged regular grid, which is deformed according to the resulting set of coefficients.

Examples of described visualizations of deformation fields can be found in the chapter 7 Testing and Optimization.

For a possibility of a comparison of the registrations' results, it is necessary to store and save the results of each registration and each of its iterations on a continuous basis. For this purpose, the `STATS` variable structure is created.

First, the basic registration information is saved into an `info` field of the structure. Specifically, the reference and the moving image names; the name of the folder where the results are saved; the name of the registration method used and the used vectors `dsVec`, `nIter`, `nInd`.

Into the next field, `regEngineInfo`, the information about the registration engine is stored. Namely, `elite`, `mutProb`, `stagnateCeil` and `stagnateDiff` variables are saved.

The next field, `time`, stores the time calculated using a tic toc function.

The qualities are stored into a `quality` field. All the qualities of the best individuals of every iteration are stored.

Coefficients connected with these qualities are stored in `coeffs` field.

The resulting registered image is saved into `result` field and respective transformation matrices `x` and `y` are saved in `xpReshRES` and `ypReshRES` fields.

This way, all the information obtained during the registration are available anytime. After a few registrations, the results can be easily visualized and compared.

To simplify the visualization of the development of the qualities and the coefficients during the registrations, a `Visual.m` function was created. Once it runs, an interactive window opens. There a user enters a number specifying how many results' files will be compared. Then, in the next interactive window, the folder browser is opened and it is possible to chooses the files to be compared. All stored qualities and individual coefficients are then automatically plotted into charts.

5.3 Setting Options

In the chapter 5.2 GA Function, it was stated that many parameters and function coefficients are adjustable. In this chapter, it will be described, what values were tested as the most appropriate, leading to the best results, while registering both, the test data and real TEM images. The following Tab. 5.1 summarizes all the adjustable variables that affect the developed registration method based on a genetic algorithm.

Tab. 5.1 Summary table of setting options of the GA function.

	Relevant variable	Range	Limitations and Consequences
Downsampling of the input images	dsVec	From one to a reasonable number according to the image size. (For the test data, the best range is 4 – 1; for the real sample images 32 – 8.*). Numbers in vector must decrease from biggest to smallest.	In case of a large image and downsampling 1, the program will have large time and computational demands. The smaller is the number of samples used for registration, the sooner an approximate registration will be reached.
Number of iterations	nIter	Reasonable numbers depending on the force of the distortion of registered images and their size; depending also on the requirements for the results accuracy and time efficiency. (Recommended range 30 – 8.*)	Too big number of iterations for the same downsampling level is useless (see the ‘Maximum of the iterations for stagnant results’ and ‘Stagnation threshold’ in this table). The best way is to decrease the number of iterations with increasing number of samples used.
Number of individuals	nInd	2 – unlimited (with respect to time and space efficiency and in a balance with the selected number of iterations). (Recommended range 30 – 10.*). Numbers in vector must decrease from biggest to smallest.	Because of reproduction, only even numbers can be used. With the growing number of individuals, the performance of the programme decreases significantly. On the other hand, an increase of the number of individuals has a better effect on the resulting quality than the increase of the number of iterations. The best way is to decrease the number of individuals with increasing number of samples used.

	Relevant variable	Range	Limitations and Consequences
Elitism	<code>elite</code>	1	To perform elitism, the variable must be set to 1. (For any other number elitism do not occur.) When the elitism is turned off, the function also finds the solution, but it is unnecessarily slowed as the already found good solution is not saved.
Probability of mutation	<code>mutProb</code>	0 – 1	The higher the number is, the higher the probability of a mutation occurrence in each of the individual's bit is. The number choice depends mainly on the tendency of the function to get stuck at a local minimum. Choosing the probability of mutation is not critical to the success of the algorithm.
Maximum of the iterations for stagnant results	<code>stagnateCeil</code>	The number is set to one third of the total number of iterations set for the particular registration level (<code>nIter</code>).	There is a different number of iterations in each registration level; therefore, it is pointless to set this number specifically (then in some cases it could be even higher than the number of iterations to be done). One third proved to be appropriate during testing.
Stagnation threshold	<code>stagnateDiff</code>	A reasonable number depending on the method of the quality calculation and the order of the quality values. Currently set to $1 \cdot 10^{-4}$.	The number that defines when the iteration results are stagnant. If a user determines that the results of individual iterations do not vary for a long time, it is possible to increase this threshold or decrease the value of the previous variable <code>stagnateCeil</code> .

(*Specific values and their combinations can be found in all the presented solution examples in the chapter 7 Testing and Optimization.)

6 EXISTING METHODS AND ALGORITHMS

In the previous chapter, a proposed and implemented function based on a genetic algorithm was presented. It is stated, that the algorithm starts from the `OriginFunction.m`. This function is also used as a start point for all the other methods which are implemented, tested, and compared to the developed function. This way a user can easily choose what function to use to evaluate his data, while also test what function is the best for evaluation of these particular data.

The method is selected using a `Method` variable. A choice of `Method = 0` represents the use of the developed GA function (5.2 GA Function). `Method = 1` choose a usage of the MATLAB function `fminsearch` (6.1 `fminsearch`). If `Method = 2`, an open-source software `elastix` is used (6.2 `elastix`). In `Method = 3` a MATLAB function `imregdemons` using a Demon algorithm is performed (6.3 The Demon Algorithm).

6.1 `fminsearch`

`fminsearch` is a simple feature for searching the minimum of a specified problem and is implemented in MATLAB. The function is easy to use for image registration. It is based on a simplex algorithm for finding functional maxima or minima described in the chapter 3.7.2 Stochastic Algorithms. Unfortunately, simplex is a simple optimization algorithm with a large tendency to get stuck in a local minimum. Therefore, `fminsearch` contains a variety of relatively significant shortcomings and is very sensitive to the initial estimation of the coefficients. However, with sufficient input limitations and a number of iterations large enough, the function could be considered successful in some cases.

6.1.1 `fminsearch` Implementation

As already mentioned, the algorithm starts in the `OriginFunction.m`. In case of `fminsearch`, it is the same, and the schema in the Fig. 5.2 is identical to the only difference; the function `RegistrationEngineFMIN.m` is called instead of `RegistrationEngineGA.m`. `fminsearch` is selected by setting `Method = 1`.

No scale space approach is used, and only one number is entered in the `dsVec` variable. This number is equal to one if it is not required to downsample the to be

registered images. However, if it is appropriate to downsample images because of their size, an appropriate number is specified. When initializing `fminsearch`, one additional variable is set, namely `MaxFminEval`, which specifies how many iterations a user wants to perform.

Then, as in the genetic algorithm, the `RegistrationEngineFMIN.m` is called. In this case, the limits are set only for generating the first generation of coefficients. These generated coefficients are one of the `fminsearch` inputs. Then only the termination condition of the `fminsearch` is left to be determined into a variable `opts`. This is the mentioned maximum number of iterations – how many times the function is evaluated:

```
opts = optimset('MaxFunEvals',MaxFminEval);
```

The whole `fminsearch` input looks like:

```
coeffs1 = fminsearch (@TransformationFunctionFMIN, coeffs0,  
                    opts, refIm, movIm, X, imHandle);
```

where `coeffs0` is the initial, randomly generated, coefficients estimate. `refIm` is the reference image and `movIm` is the moving, to be registered, image. Variable `opts` expresses the termination condition. `X` is the matrix filled with x and y pixel coordinates and `imHandle` is only a handle of current MATLAB figure for the visualisation of the result of every iteration step. The most important in the `fminsearch` function is the iteratively called function `TransformationFunctionFMIN.m`. This function is based on the `TransformationFunctionGA.m` (5.2.3 Transformation Function). The transformation equations are the same, only in each iteration just one set of coefficients enters, is evaluated, and displayed. The coefficients are iteratively changed by `fminsearch` function until a specified number of iterations is reached and the solution found.

The resulting image, transformation matrices and displacement field are visualized and saved as in the previously described GA function.

6.2 `elastix`

`elastix` is an open source software by Stefan Klein and Marius Staring, developed at the Image Science Institute in the University Medical Center Utrecht under supervision of Josien P. W. Plum. It is a toolbox for both rigid and nonrigid image registration, based on Insight Segmentation and Registration Toolkit (again an open-source, cross-platform

system providing software tools based on leading-edge algorithms for registering and segmenting multidimensional data, more in [42]). The whole source code, a description, a manual, details, and other useful information might be found at [16]. As the authors focus suggests, `elastix` was developed mainly but not only for registration of medical images in both, 2D and 3D.

The general disadvantage of the use of one particular image registration method is its narrowness on the application for which it was developed, because none of the methods can be used for all the applications. This is a problem that `elastix` tries to overcome. The whole toolkit contains a range of algorithms of commonly used registration methods in the medical imaging field, so a user would have an option of configuring, testing, and comparing many various possible solutions of their specific tasks and in the same time use the same software with different settings for an entirely different registration task. [18]

`elastix` has several basic components (see Fig. 6.1) and then the user can choose from a variety of transformation models, interpolators, optimization methods, cost functions and multiresolution schemes. There is a huge amount available. Between the most popular might be included only a few examples in Tab. 6.1, and full list can be found at the webpage in [16] (section Documentation → Modules). For some of the components, additional setting is available, the details at the same webpage (Documentation → Related Pages → Parameters). `elastix` has a command-line interface and a parameter text file exists, where the user can configure a registration algorithm by specifying the names of the desired components as well as any additional settings for the components. [18]

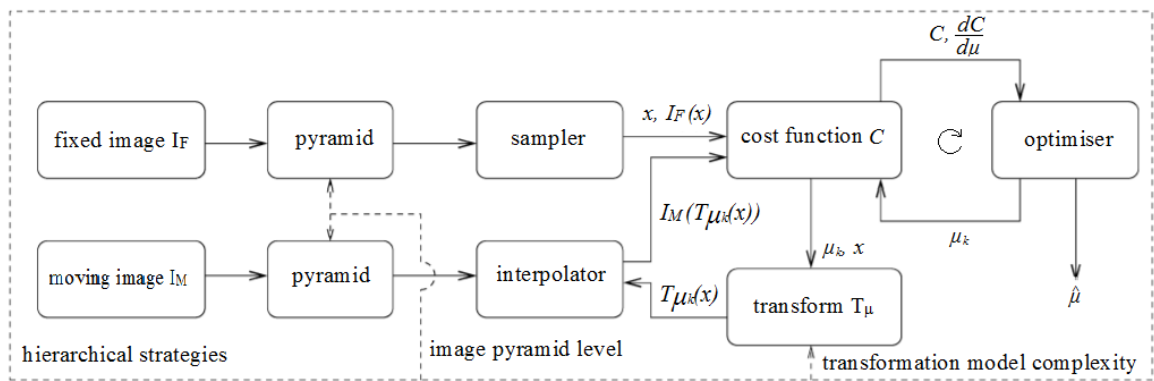


Fig. 6.1 A block scheme of the basic registration components of image registration used in `elastix`. [18]

Tab. 6.1 An example of components available in `elastix` toolbox.

Transform model	Rigid (2D, 3D, Translation)
	Affine
	Nonrigid (B Spline, Thin Plate Spline)
Similarity measure	Mutual information
	Euclidean distance
Optimisation method	Full search
	Gradient descent
	Simplex
Interpolation method	Nearest neighbour
	Linear
	Cubic
	B Spline

The input reference and moving images are supplied as command-line arguments together with the parameter text file. The output of the registration, the deformed moving image, and any progress information such as transformation description in text file, is saved.

It is also important to mention accompanying program `transformix` that can be used to evaluate the transformation at user-defined points, or to see the deformation field and gain the exact transformation matrices, which is used in the thesis. [18]

6.2.1 Access from MATLAB

First of all, it must be mentioned that for the purpose of this thesis, all the original and modified codes needed to run `elastix` and `transformix`, are part of the attachment.

Nevertheless, the open-source software `elastix` can be downloaded from [16]; section ‘Download’. (The downloaded zip folder contains `elastix`, `transformix` and licences). Because `elastix` is cross-platform without any graphical user interface, it can be called from the system command-line, but this is not always convenient.

There exist several ways how to call `elastix` from MATLAB and one of them is

MelastiX. It allows `elastix` to be called from within MATLAB as though it is native MATLAB command. MelastiX can be downloaded from [4]. Downloaded folder than contains three subfolders and a README text file with basic info. One of the subfolders consist of the main needed codes (`elastix`, `transformix`, codes for `.mhd`, `.mha` files reading and many others). The second subfolder contains a few examples of parameters' files which define `elastix` setting and the last subfolder includes several examples of `elastix` usage.

`elastix` and `transformix` are originally created in C++ language and are compiled as `.exe` files. To be able to run it, all the needed files must be added to the operation system default file path and MATLAB path. To add the operation system default file path, a user needs to open Windows Control Panel \ System and Security \ System \ Advanced system setting \ Advanced \ Environment Variables; there is a section 'System Variables' and an item 'Path'. Into the 'Path', there must be added the path to the folder, where all the codes (`elastix`, `transformix` and the other needed files) are saved. (Some computer systems require a user to restart the computer after adding a new path to the system path. In case of persistent inconvenience, `elastix` can be installed using the `elastix.exe` file, but it should not be necessary to run it from MATLAB.)

These codes, their folder, must be together with the function and tested images in the same folder. This folder and its subfolders must be added also to the MATLAB path (right click to the folder \ Add to Path \ Select Folders and Subfolders, Fig. 6.2).

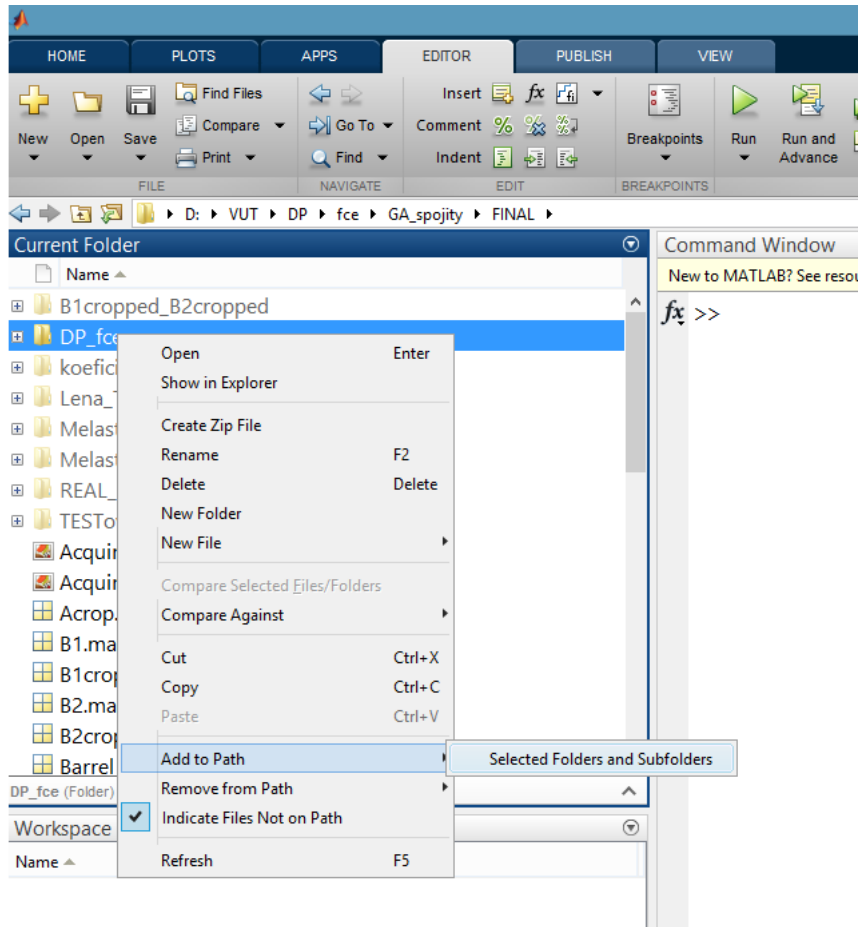


Fig. 6.2 Add the folder with all the function and `elastix` code to MATLAB path.

If something is changed in one of the `elastix` codes or something is added into the folder, the folder must be added to the MATLAB path again. This is also true for each new MATLAB launch.

6.2.2 `elastix` and `transformix` Implementation

As in the previous cases, the `elastix` call is initialized from the main function `OriginFunction.m` by setting the `Method = 2`. When evaluating large images, it is better to set input downsampling (e.g. `dsVec = [8]`); it is not necessary but significantly reduces the computational complexity.

`elastix` and then `transformix` create their folders into a folder that is initially specified by a user. They store there the necessary interim files. It must be noted, that when the folder to store the results is chosen at the beginning of the `OriginFunction.m`, this folder and its whole path must not contain any white spaces in the name. `Elastix` cannot work with such a path with white spaces.

The function `DoElastix.m` performing `elastix`, is called (instead of any `RegistrationEngine` function). This function is based on the function "`example_2D_affineThenWarping_withParams`" which is a part of the `Melastix` (MATLAB Elastix wrapper, downloaded from [4]). The original function can be found in `matlab_elastix-master \ MelastiX_examples \ elastix`. The command line to call `elastix` itself is following:

```
[~,out] = elastix (img2, img, [], {'AtW_parameters_1.txt',  
                                'AtW_parameters_2.txt'});.
```

`img2` is the moving image, `img` is the reference image. In the files '`AtW_parameters_1.txt`' and '`AtW_parameters_2.txt`', there is a set of the input parameters. It is available in the supplement C.1 for the affine transformation and C.2 for the non-rigid transformation. It means, that the registration run in two phases, first is affine, where the translation is performed and then follows non-rigid transformation. Parameters set in these two files are key to the registration. As can be seen in the files, the basic registration setting is: B-spline interpolation, standard gradient descent optimization method, pyramidal approach, the registration run in 1000 iterations and the result is given in a double data format and saved as a `.mhd` file.

After the `elastix` run, the registered image is displayed and stored. One of the `elastix` outputs contain info out of which the `transformix` can get the corresponding transformation matrices and the displacement field. It creates the `deformationField.mhd` file, which needs to be read by another modified function `mha_read_volume_PZ`. Original is '`mhd_read_volume.m`', a part of the `Melastix` (MATLAB Elastix wrapper, downloaded from [4]). The original function can be found in `matlab_elastix-master \ code`.

```
[regImage, log] = transformix([],out,1);  
[V1, V2] = mha_read_volume_PZ('deformationField.mhd');
```

Finally, the resulting transformation matrices and displacement field can be plotted and stored and the resulting quality of the registration calculated.

6.3 The Demon Algorithm

The Demon Algorithm is an efficient non-parametric diffeomorphic image registration algorithm which can be adapted and then applied to many sorts of problems for various applications. [36] The method is based on pixel velocities caused by edge based forces. The result is a transformation field of pixel velocities, which filtered by a Gaussian kernel,

perform a global registration. [20]

The demon registration uses the optical flow equation for finding small deformations in image sequences / image pairs. The estimated displacement (velocity) u (2D: $u = (u_x, u_y)$) between a point p in a static image F and the corresponding point in a moving image M is following:

$$\mathbf{u} = \frac{(m-f)\nabla f}{|\nabla f|^2 + (m-f)^2}, \quad [20]. \quad 6.1$$

f is the intensity in F and m is the intensity in M and Δf is the gradient of the static image. Δf then represents internal edge based force and $(m - f)$ the external force. To make the equation more stable and use it in image registration, the term $(m - f)^2$ was added. The resulting displacement \mathbf{u} is based on a local information; therefore, it is appropriate to regularize the velocity field by Gaussian smoothing. [20]

6.3.1 imregdemons Implementation

There is implemented a demon algorithm for a displacement field estimation in MATLAB in a function `imregdemons`. This function allows easy estimating of a displacement field aligning two 2-D or 3-D images. The registration passes of three pyramidal levels of a hundred iterations in each.

When using this function (`Method = 3`). When evaluating large images, it is better to select input downsampling (e.g. `dsVec = [8]`); it is not necessary but significantly reduces the computational difficulty. If a user want to improve the results, it is possible to set variable `iterDemon` which specifies a number of iterations to be done in each registration level; default is 100 of iterations. The required command in the `OriginFunction.m` looks as follows:

```
[DF, movingReg] = imregdemons(Moving, Static);
```

where `Moving` is the to be registered image and `Static` is the reference input image. An output consists of the estimated displacement field `DF` and the registered image `MovingReg`. Thus, moving image is warped exactly according to the displacement field. In the displacement field, there are displacement vectors at each pixel showing the shift of the pixel between the fixed image grid and a corresponding location in the moving image.

From this registered image `movingReg` and the original `static` image, a quality is counted (cosine criterion) and all the needed results are visualized and saved.

7 TESTING AND OPTIMIZATION

7.1 The Result Visualization and Numerical Evaluation

The first and irreplaceable assessment of the image registration accuracy is its visualization. Whether to display images overlap or render a checkerboard. Any visualization provides a first but fairly accurate estimate of the precision of the registration and determines whether the method used has a potential for further development.

Visually judgable is also the resulting displacement field (visualized can be e.g. by a function `quiver` or by using a regular grid distorted by the resulting transformation coefficients).

The next visualizable result is a difference between the resulting transformation matrices of two same registrations. When a pair of images is registered several times (by different methods or just to compare the results of a repeatedly performed one method), the resulting x and y transformation matrices can be compared by subtracting them. (Obviously, the resulting transformation matrix x from one method is subtracted from the transformation matrix x from the second method. The same is true for y .) When registering correctly, the difference of two transformation matrices x should be zero or very close to zero.

There exist several objective criteria enabling the evaluation of the registration results, see chapter 3.6. The criterion chosen in the function as a measure of a solution's quality is a cosine criterion. This criterion has been selected and used based on the nature of the processed images, which should in principle have an equivalent range of values and contrast, but in practice it was not always the case.

The methods used can be also evaluated in terms of time and space efficiency of the algorithms. How many iterations is needed to find a solution and how long one iteration takes.

Another important factor would be the effectiveness of the methods to various distortions. This can be modelled and evaluated systematically by changing a strength of the distortion.

7.2 Testing Data

The algorithms in general should be tested on test data to make the results of the methods verifiable. For this purpose, a small database of test pairs of images with known parameters were created. For a better imitation of the real samples' images, a weak distortion of the whole deformed field was created. On the other hand, images deformed by stronger distortions were created too. On such images, it is easier to see the differences and to assess the ability of the registration methods for an. One example of a test image creation and a testing procedure follows.

First, the original undistorted selected test image was placed in a black frame (zeros; Fig. 7.1) and in this state, it was distorted.



Fig. 7.1 The original undistorted selected image placed in the black frame (zeros).

Such a distorted image was then cropped out of the frame for testing itself, Fig. 7.2. In this figure, there is the image distorted by the coefficients as shown in the Tab. 7.1. In Fig. 7.2 a), it is possible to see the resulting distortion, but in Fig. 7.2 b), which is cropped, the weak distortion is not apparent.



Fig. 7.2 a) The distorted image in the frame. b) The same distorted image cropped out of the frame.

A presence of a different distortion can be proven by overlaying a distorted image over the original one or by overlaying two distorted images one over another; Fig. 7.3 c).



Fig. 7.3 a) The previous image Fig. 7.2 b). b) The same distortion of the image, but a different position relative to the optical axis. c) Both images overlaid one over another.

In case of presented images, when the image is cropped out of the frame, the influence of the shift can be suppressed. Thus, the shift of the images to each other was then performed manually as shown in the following Fig. 7.4.

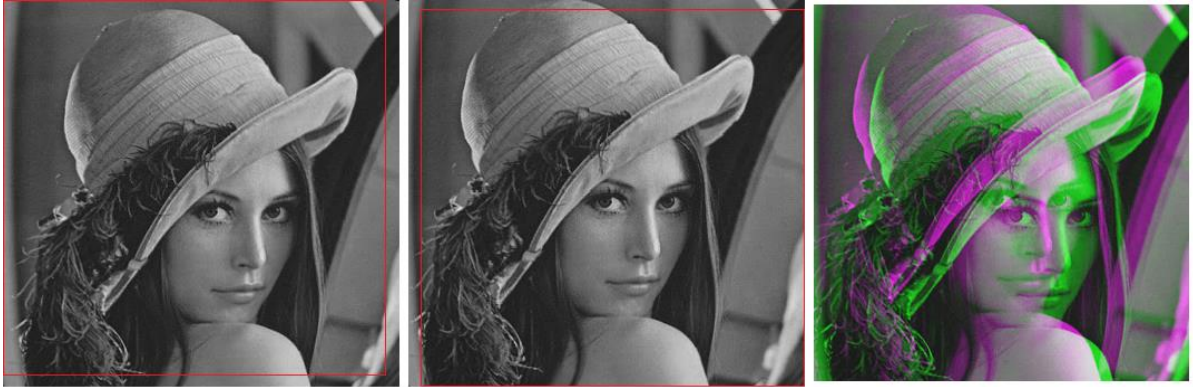


Fig. 7.4 a) 7 px were cropped off the bottom and 7 px off the right side of the first distorted image.
 b) 7 px were cropped off the top and 7 px off the left side of the second distorted image.
 c) Both images overlap with performed shift.

Tab. 7.1 An adjustment of the coefficients for distorting the test images.

	Fig. 7.2 b) Fig. 7.3 a) Fig. 7.4 a)	Fig. 7.3 b) Fig. 7.4 b)
DFx	-15	20
DFy	-12	8
kappa1	$2 \cdot 10^{-9}$	$2 \cdot 10^{-9}$
kappa2	$3 \cdot 10^{-12}$	$3 \cdot 10^{-12}$
lambdax	0	0
lamday	50	50
s	1	1
Tx	7*	7*
Ty	7*	7*

(*The shift means a shift of the images towards each other.)

7.3 Testing

The images described in the example in the previous chapter 7.2 Testing Data were tested using all the before described methods. The results found and generally valid are discussed in this chapter.

All the registrations were performed on Intel® Core™ i5-5200U CUP (2.2 GHz) with 8 GB RAM.

Tab. 7.2 Results of the registration of images from Fig. 7.4.

	Input data:		Output results:						
	Images distorted by coefficients:		GA*		fmin		elastix	demon	
			dsVec =	nIter =	nInd =	MaxFminEval =	-	iterDemon=	
Fig. 7.4 a) ref.	Fig. 7.4 b) mov.	= [4, 1] = [20, 10] = [14, 8] (GA 1)** a)	= [4, 1] = [30, 16] = [20, 12] (GA 2) b)	= 200 (FMIN 1) a)	= 500 (FMIN 2) b)	***	= 100	= 500	
DFx	-15	20	3,07	-3,43	-3,97	-2,20	-	-	-
DFy	-12	8	-9,39	-12,32	-2,87	4,43	-	-	-
kappa1	$2 \cdot 10^{-9}$	$2 \cdot 10^{-9}$	$-5,54 \cdot 10^{-9}$	$0,71 \cdot 10^{-9}$	$0,80 \cdot 10^{-9}$	$0,32 \cdot 10^{-9}$	-	-	-
kappa2	$3 \cdot 10^{-12}$	$3 \cdot 10^{-12}$	$3,35 \cdot 10^{-12}$	$-0,60 \cdot 10^{-12}$	$-1,54 \cdot 10^{-12}$	$0,18 \cdot 10^{-12}$	-	-	-
lambdax	0	0	-51,52	40,52	66,50	24,48	-	-	-
lamday	50	50	-0,18	7,32	4,02	-15,85	-	-	-
s	1	1	1,81	-7,69	0,73	2,46	-	-	-
Tx	7*	7*	-19,75	12,64	-11,88	-18,46	-	-	-
Ty	7*	7*	-5,05	-2,43	-12,87	-13,62	-	-	-
Quality	-		0,9899	0,9934	0,9899	0,9915	0,9685	0,9559	0,9690
Time [s]	-		70	292	117	299	28	5	27
Result	-		Fig. 7.7 a)	Fig. 7.7 b)	Fig. 7.9 a)	Fig. 7.9 b)	Fig. 7.11 a)	Fig. 7.12 a)	Fig. 7.12 b)

(*The other setting of genetic algorithm method was: Elitism turned on. The probability of mutation 0,25. 4 was the maximum of the iterations while the results stagnated. The stagnation limit was $1 \cdot 10^{-6}$.)

**A note on legends in charts in Fig. 7.5, Fig. 7.6 and Fig. 7.16.

***The setting of elastix is in supplement C.1 and C.2.)

From the Tab. 7.2, it is obvious, that not the specific values of the coefficients create the exact distortion, but their combination does. Different combinations of coefficient values result in an identical distortion. Therefore, it is not essential to observe the progress of coefficients' values. It is, however, appropriate to point out that in case of the fminsearch, the coefficients values' change only in a small range, during the registration. In the case of the GA function, the coefficients values' change very distinctively at the beginning; only after finding the best direction the changes decrease

until a complete stabilization. This is important for finding the global optimum. An illustration of described facts follows in Fig. 7.5, Fig. 7.6. There can be seen clearly, how big the differences at the beginning of the GA function are. And it must be emphasized, that not all the solutions have been saved and plotted in this case; saved and later plotted are only the best individuals from each iteration of each registration level (more about this follows in the next paragraphs).

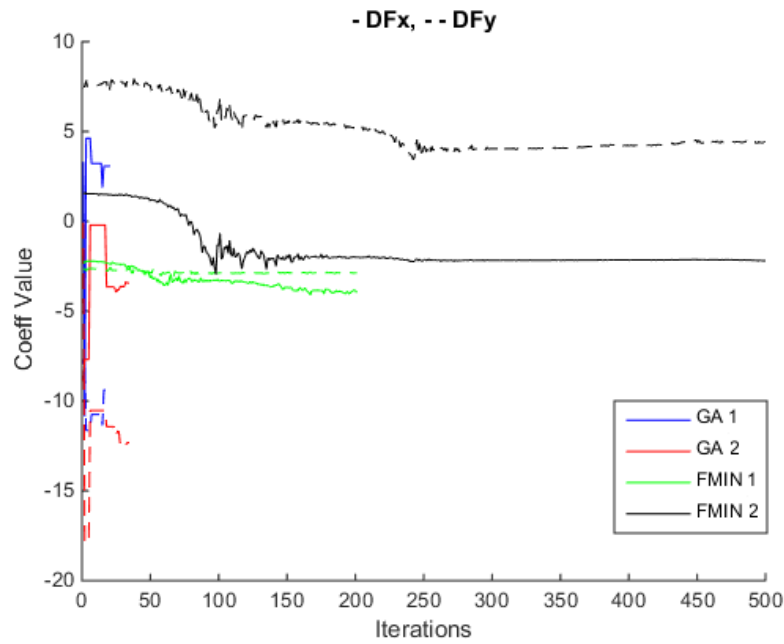


Fig. 7.5 An example of the coefficients DF_x , DF_y development during the registration.

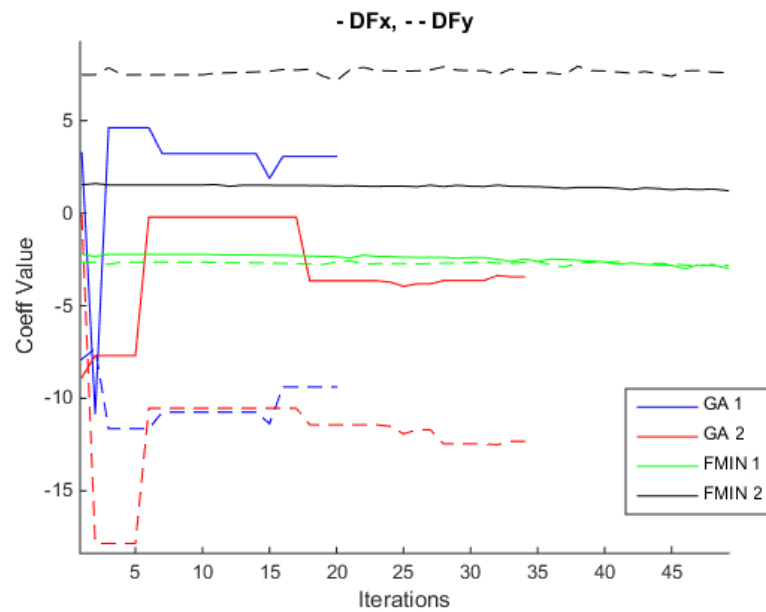


Fig. 7.6 A zoom of the previous Fig. 7.5.

The results' visualization is in following figures. The resulting images overlap the reference ones.



Fig. 7.7 The results of the GA registration. a) Less iterations and individuals, 70 s, quality 0,9899. b) More iterations and individuals, 292 s, quality 0,9934. See the details and the exact numbers of iterations and individuals in the Tab. 7.2.



Fig. 7.8 One more example of the results of the GA registration. The set of the $dsVec$, $nIter$ and $nInd$ the same as in the previous example. a) Less iterations and individuals, 119 s, quality 0,9932. b) More iterations and individuals, 205 s, quality 0,9938.

These four registration results say, that in case of GA function, the result is almost exclusively dependent on the selected vectors $dsVec$, $nIter$, $nInd$. Of course, it is still a stochastic algorithm that does not guarantee the right solution finding even after particular number of iterations. But, with an increasing number of searched options (with increasing number of iterations and individuals), the global optimum is more likely to be found. Thus, comparing the results in Fig. 7.7 a) and Fig. 7.8 a), for which the $nIter$ and $nInd$ were smaller than in Fig. 7.7 b) and Fig. 7.8 b), the results are sort of different.

On the other hand, comparing the results in Fig. 7.7 b) and Fig. 7.8 b), for which the `nIter` and `nInd` were set bigger, the results are better and very similar. The direction of registration is therefore obvious. If the registration was running longer, even more precise solution would be found.



Fig. 7.9 The results of the `fminsearch` registration. a) 200 iterations, 117 s, quality 0,9899. b) 500 iterations. 299 s, quality 0,9915. See the details in the Tab. 7.2.



Fig. 7.10 One more example of `fminsearch` result. a) 200 iterations, 109 s, quality 0,9936). b) 500 iterations, 262 s, quality 0,9895.

Unfortunately, `fminsearch` does not show up such a robustness of a direction of the solution finding. In the case of Fig. 7.9 a) and Fig. 7.10 a), after 200 iterations, the results differ; as in GA function. However, they do not get stable even after 500 iterations. Not even another increase of the number of the iterations can guarantee the stability and reliability of the results. Especially in a case of a stronger distortion.



Fig. 7.11 a) The result of `elastix`; 28 s, quality 0,9685.



Fig. 7.12 The result of the Demon algorithm. a) 100 iterations, 5 s, quality 0,9559.
b) 500 iterations, 27 s, quality 0,9690. See the details in the Tab. 7.2..

The results in the Fig. 7.11 show the success of the `elastix` software. In contrast, the method implemented in MATLAB, `imregdemons`, is not successful. However, in this case, only default setting of `imregdemons` ran (3 levels of 100 iterations). When the number of iterations is increased to 500 iterations per level, the solution significantly improves.

A visualization of deformation field follows.

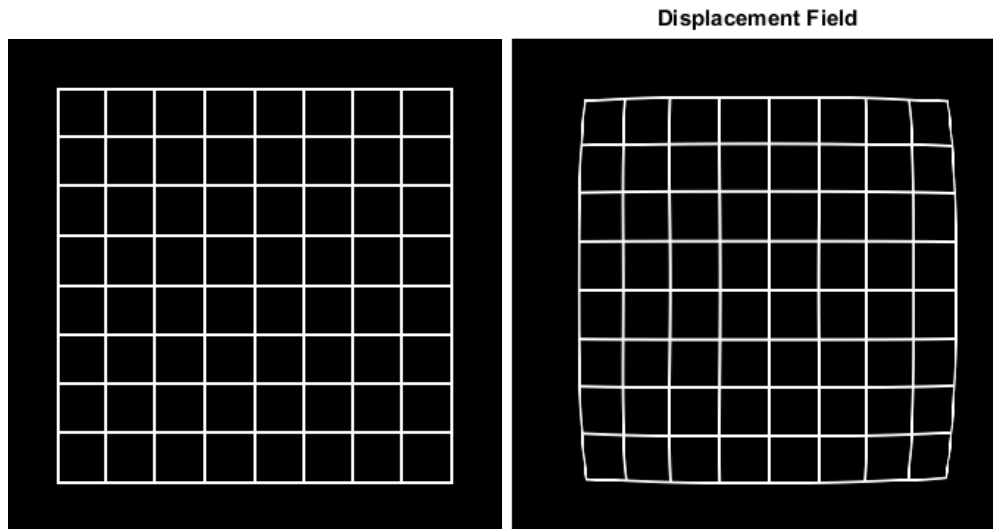


Fig. 7.13 A visualization of the displacement field (GA function, results of the second registration, Fig. 7.7 b)) using rectangular grid.

In the Fig. 7.13 a), there is a regular, rectangular grid, which is an input of a Transformation Function. The function distorts this grid according to the resulting coefficients describing the distortion. The deformation field thus look like in the Fig. 7.13 b).

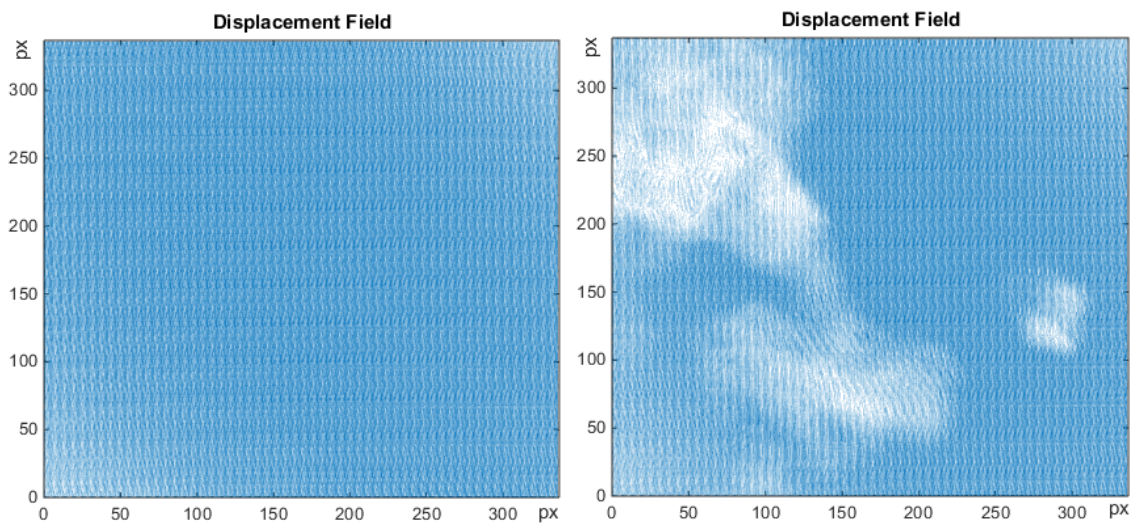


Fig. 7.14 a) The resulting displacement field of `elastix`. The same displacement field is also a result of the successful GA and `fminsearch` function and successful `imregdemons` registration (500 iterations). b) The resulting displacement field of the unsuccessful Demon algorithm registration (100 iteration).

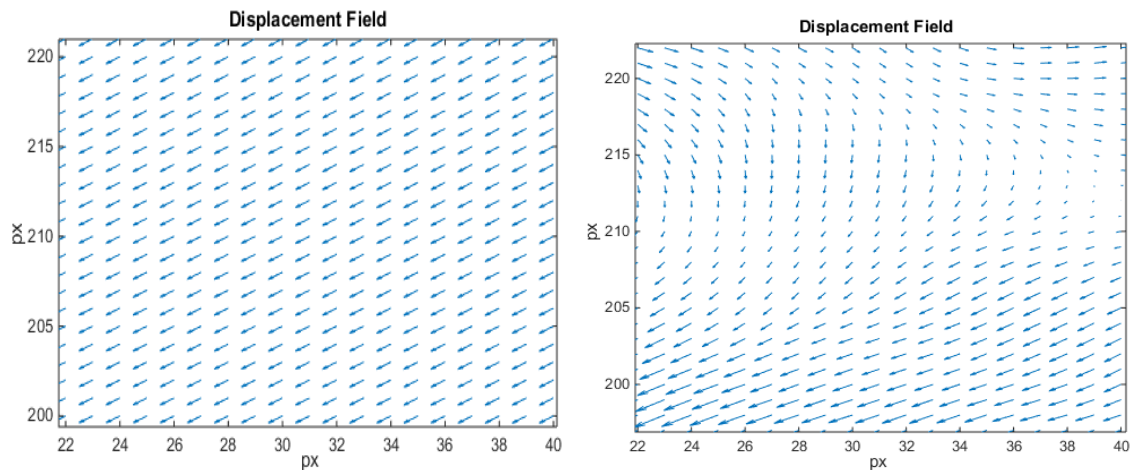


Fig. 7.15 A comparison of a cropped part of the displacement fields from the previous Fig. 7.14 a) and b). The displacement vectors at each pixel show the shift of the pixel between the fixed image grid and a corresponding location in the moving image.

For the `elastix` and `imregdemons` function, the coefficient values are not available. Therefore, only the `quiver` function does a displacement field visualization in these cases. The Fig. 7.14 clearly shows a regularity of the deformation field as assessed by the `elastix` and the irregularity obtained with the unsuccessful registration by `imregdemons`.

Comparing the time efficiency is appropriate only between the GA function and `fminsearch` function. It should be remembered that in the case of one `fminsearch` iteration, only one coordinate recalculation occurs (according to the input coefficients); one iteration means one `TransformationFunctionFMIN.m` run and one computation inside. In case of the GA function, there are as many calculations as how many individuals are in the particular registration level; one iteration of registration level with 20 individuals lead to a `TransformationFunctionGA.m` run with 20 repetitive coordinate recalculations and a continuous storage and comparison of individual runs. Thus, for the second example of GA function and second example of `fminsearch` from the Tab. 7.2 applies the Tab. 7.3. For a comparable time, the GA function performs 960 calculations, `fminsearch` only 500 calculations.

Tab. 7.3 Comparison of time efficiency of the GA function and `fminsearch`.

	GA		fminsearch
	<code>dsVec = [4, 1]</code> <code>nIter = [30, 16]</code> <code>nInd = [20, 12]</code>		<code>MaxFminEval = 500</code>
Time [s]	292		299
Calculations:	Initialization	20	500
	30 iterations with 20 individuals	30*20	
	30 reproductions, calculation for 3 descendants	30*3	
	16 iterations with 12 individuals	16*12	
	16 reproductions, calculation for 3 descendants	16*3	
Resulting number of calculations:		= 950	= 500

A quality development during the registration is interesting too, Fig. 7.16. To describe the figure, it is important to recall what is one iteration in the GA function and one iteration in `fminsearch` (see the previous paragraph). Compared to `fminsearch`, during one iteration in GA function, many more possible solutions are tested. Nevertheless, the quality of solution increases with each other selected solution in GA. `Fminsearch` does have upward trend, but it definitely does not apply to every two consecutive solutions.

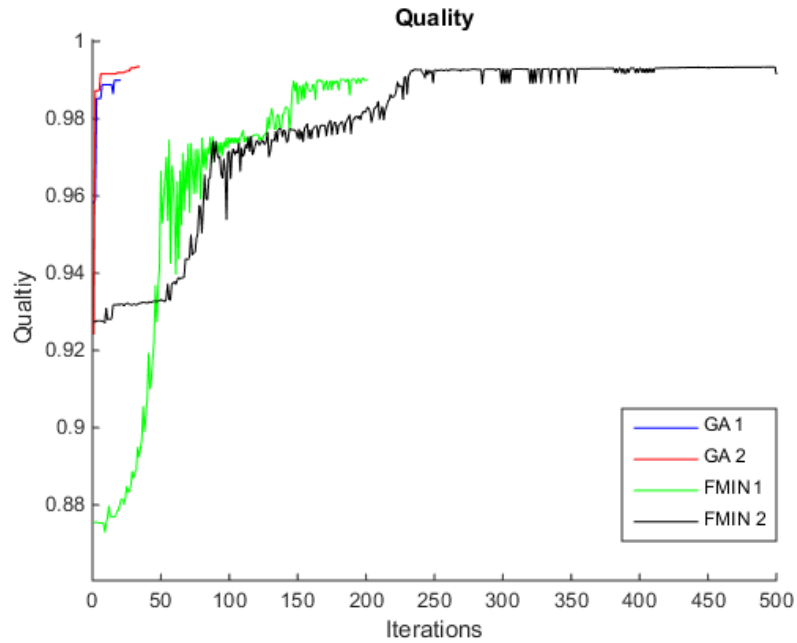


Fig. 7.16 The quality development during the image registration. The order of the curves is the same as the order of the methods in the Tab. 7.2.

7.4 The Evaluation of the Registration of the Real Images

7.4.1 The Registration of the Images with a High Magnification and a Low SNR

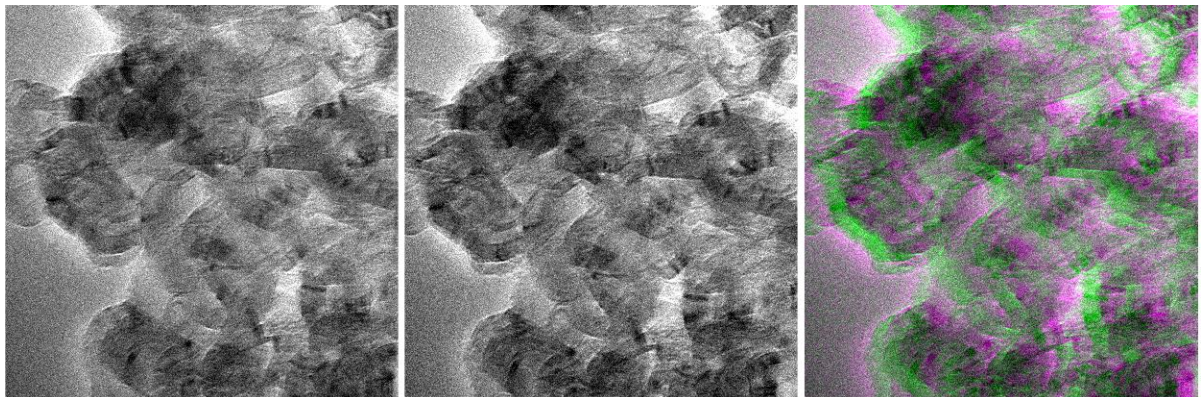


Fig. 7.17 The input images and their comparison. The sample is a carbon film shadowed with gold with graphitized carbon particles. The magnification was 510 000x and the dose rate $7\,200\text{ e}/\text{\AA}^2\text{s}^{-1}$. Low image quality due to low SNR is apparent. The results of the registration are in the Tab. 7.4.

Tab. 7.4 The results of the registration of the images on the Fig. 7.17.

	GA*		fminsearch			elastix	imregdemons	
	dsVec =		dsVec =			dsVec =	dsVec =	
	nIter =		MaxFminEval =				iterDemon =	
	= [32, 8]	= [32, 8]	= [8]	= [8]	= [8]	= [8]	= [8]	= [8]
	= [14, 10]	= [30, 12]	= 200	= 200	= 500		= 100	= 500
	= [20, 12]	= [24, 12]						
Quality	0,9800	0,9800	0,9674	0,9394	0,9800	0,9529	0,9620	0,9588
Time [s]	401	542	231	232	668	27	7	48
Result	Fig. 7.18 a)	Fig. 7.18 b)	Fig. 7.19 a)	Fig. 7.19 b)	Fig. 7.19 c)	Fig. 7.20 a)	Fig. 7.21 a)	Fig. 7.21 b)

(*The other setting of genetic algorithm method was: Elitism turned on. The probability of mutation 0,25. 4 was the maximum of the iterations while the results stagnated. The stagnation limit was $1 \cdot 10^{-6}$.)

Quality in the form of a similarity criterion is a good pointer during the run of a registration method. However, comparing the resulting qualities of different methods, it cannot be marked as very reliable, especially when there is a noise in the images. Although the visual quality of the registration (images overlap) is better in case of *elastix* registration (Fig. 7.20 a)) than *imregdemons* registration (Fig. 7.20 b)), the quality in the form of the similarity criterion says the opposite.

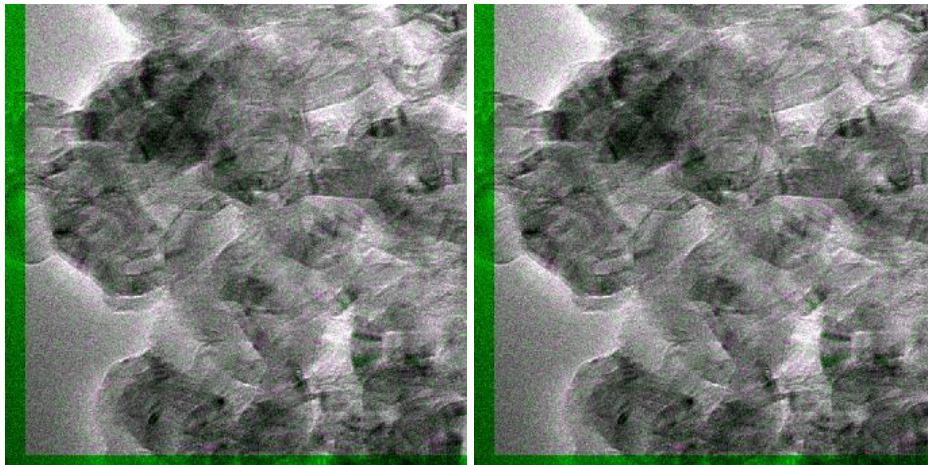


Fig. 7.18 The results of the GA registration. a) Less iterations and individuals, 401 s, quality 0,9800. b) More iterations and individuals, 542 s, quality 0,9800. See the details and the exact numbers of iterations and individuals in the Tab. 7.4.

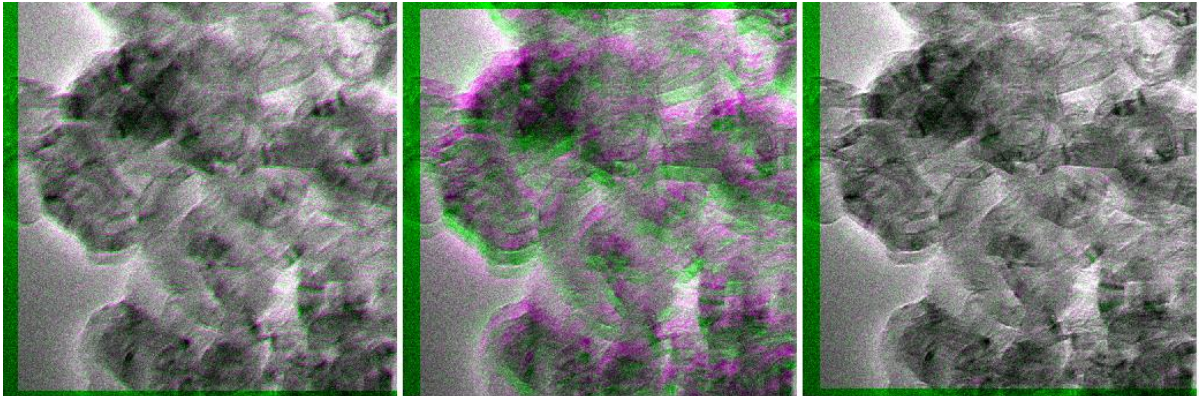


Fig. 7.19 The results of the `fminsearch` registration. a) 200 iterations, 231 s, quality 0,9674.
 b) 200 iterations, 232 s, quality 0,9394. c) 500 iterations, 668 s, quality 0,9800.
 See the details in the Tab. 7.4.

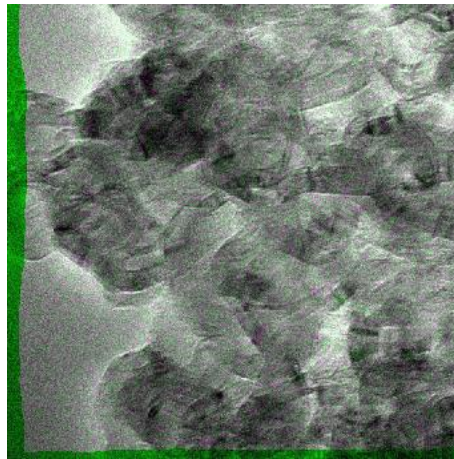


Fig. 7.20 a) The result of `elastix`; 27 s, quality 0,9529. See the details in the Tab. 7.4.

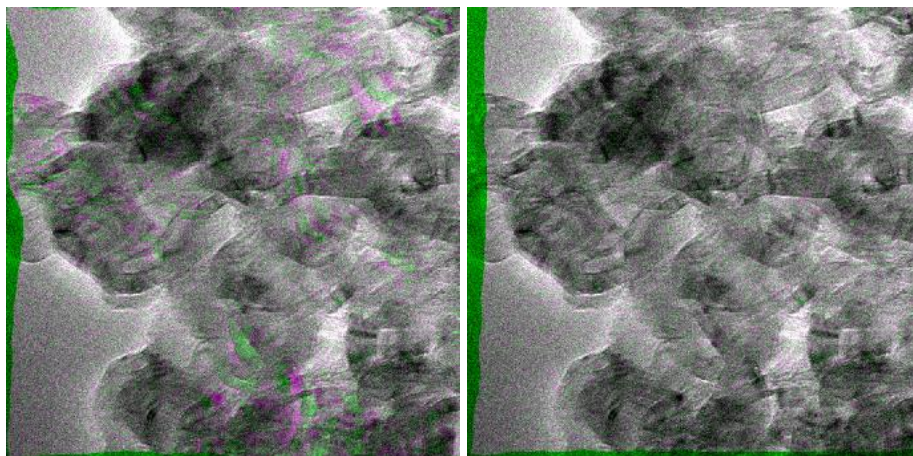


Fig. 7.21 The result of the Demon algorithm. a) 100 iterations, 7 s, quality 0,9620.
 b) 500 iterations, 48 s, quality 0,9588. See the details in the Tab. 7.4.

In the Fig. 7.20 and Fig. 7.21, there are results of the `elastix` and `imregdemons`. The `elastix` result and `imregdemons` result after 500 iterations are comparable, as well as their qualities. On both mentioned results shape is possible to see an evidence, that it is not enough to use only a translation to obtain a precise deformation field description. When comparing demon registrations visually, there is a significant improvement after 500 iterations comparing it to 100 iterations. On the contrary, the counted quality deteriorated. This can happen because the shades of gray are in a very small range and the noise is present in the images.

7.4.2 The Registration of the Images with a Regular Structure

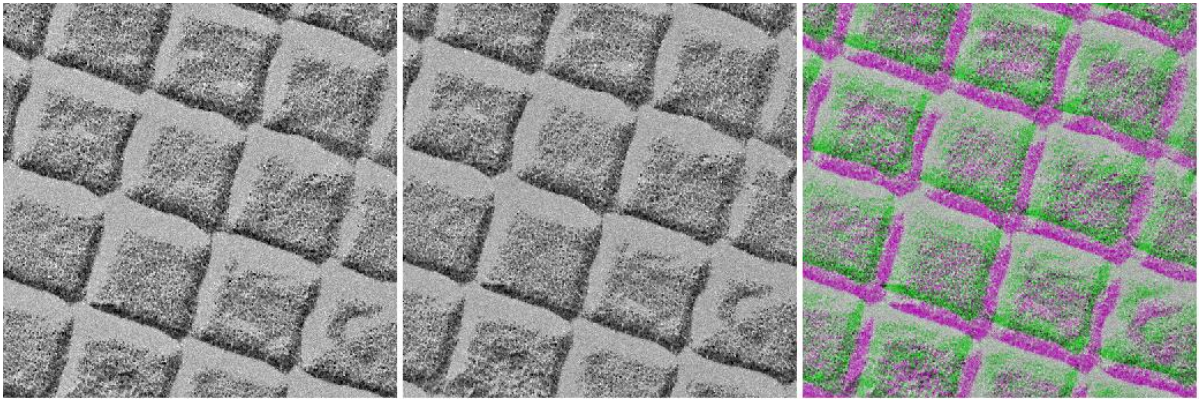


Fig. 7.22 The input images and their comparison. The magnification and the dose rate are not known.

The images in the Fig. 7.22 are the images of carbon film with a golden shadowing of waffle pattern gratings made on a copper grid. An apparent recurring pattern is captured in the images, that can lead to registration errors.

Tab. 7.5 The results of the registration of the images on the Fig. 7.22.

	GA*		fminsearch		elastix	imregdemons	
	dsVec = nIter = nInd =		dsVec = MaxFminEval =		dsVec =	dsVec = iterDemon =	
	= [32, 8] = [14, 10] = [20, 12]	= [32, 8] = [30, 12] = [24, 12]	= [8] = 250	= [8] = 500	= [8]	= [8] = 100	= [8] = 500
Quality	0,9944	0,9944	0,9458	0,9463	0,8981	0,9294	0,9528
Time [s]	421	616	376	639	27	10	61
Result	Fig. 7.23 a)	Fig. 7.23 b)	Fig. 7.24 a)	Fig. 7.24 b)	Fig. 7.26 a)	Fig. 7.27 a)	Fig. 7.27 b)

(*The other setting of genetic algorithm method was: Elitism turned on. The probability of mutation 0,25. 4 was the maximum of the iterations while the results stagnated. The stagnation limit was $1 \cdot 10^{-6}$.)

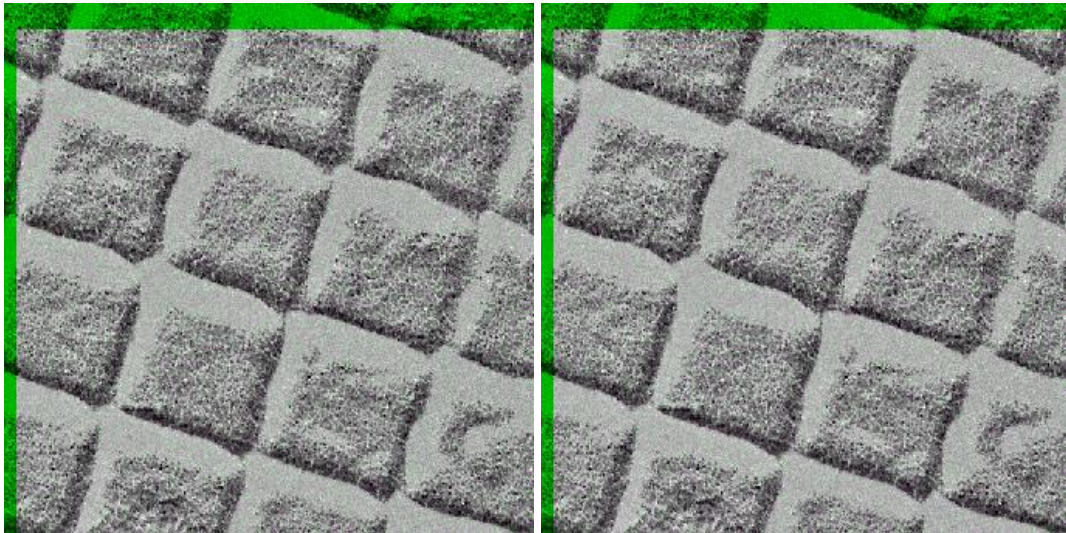


Fig. 7.23 The results of the GA registration. a) Less iterations and individuals, 421 s, quality 0,9944. b) More iterations, 616 s, quality 0,9944. For the exact numbers of iterations and individuals are see the Tab. 7.5.

From such a result, Fig. 7.23, it can be judged that an increased number of iterations and individuals do not always lead to an improved result; even after a small number of iterations and fewer individuals, the result may be good enough.

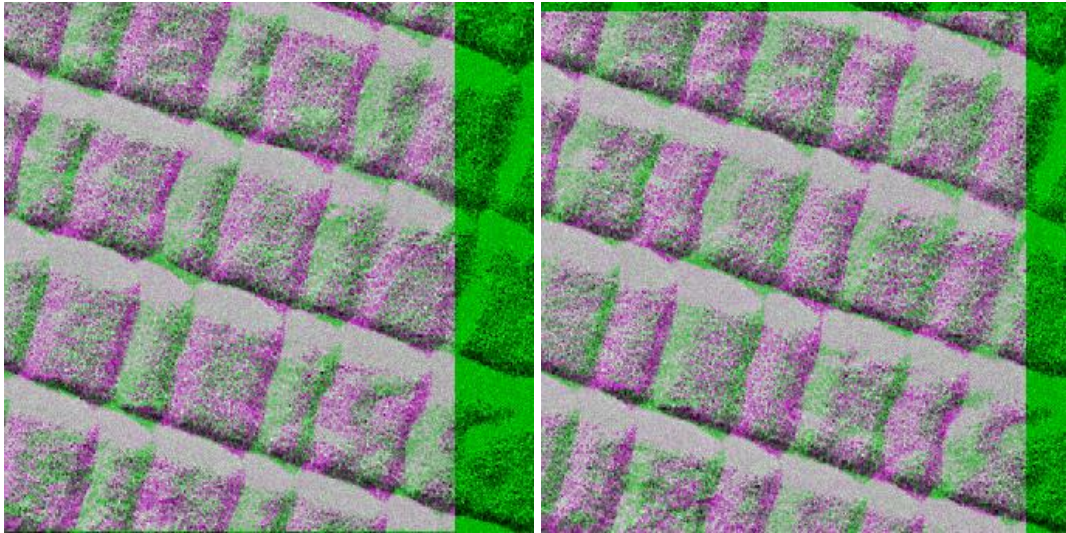


Fig. 7.24 The results of `fminsearch` registration. a) 250 iterations, 376 s, quality 0,9458.
b) 500 iterations, 639 s, quality 0,9463. See the details in the Tab. 7.5.

On the other hand, according to the results in the Fig. 7.24, in case of the `fminsearch` function used to register a regular structure, the result may not be found even after many iterations.

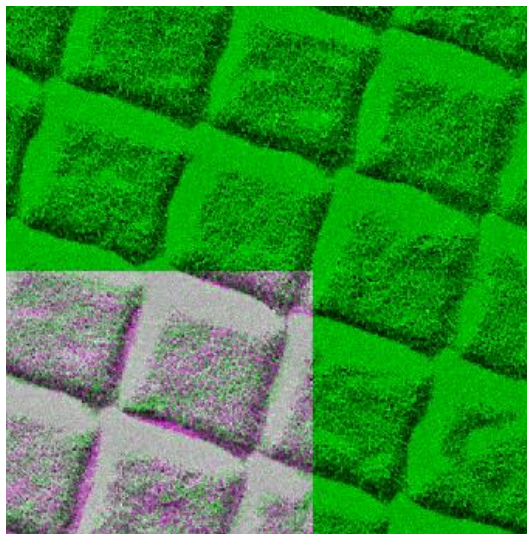


Fig. 7.25 Not found solution of a registration of images of a regular structure.

In the Fig. 7.25, there is an example of a danger of a regular structure. Such a result can arise from both, the GA and the `fminsearch` function. In case of the GA function, however, the translation coefficients can be simply reduced. In case of the limited range exceedance, these values are after reproduction and mutation performance returned to the allowed range. This is not easily doable in `fminsearch`.

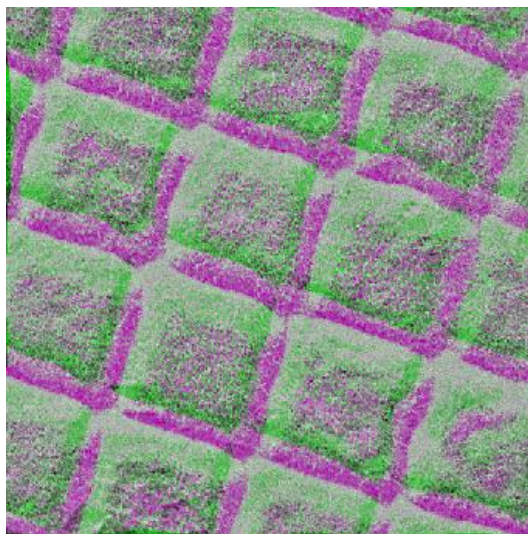


Fig. 7.26 The results of `elastix` registration, 27 s, quality 0,8981. See the details in the Tab. 7.5.

According to the Fig. 7.26, it has to be stated that `elastix` could not register a regular structure with the same settings as before.

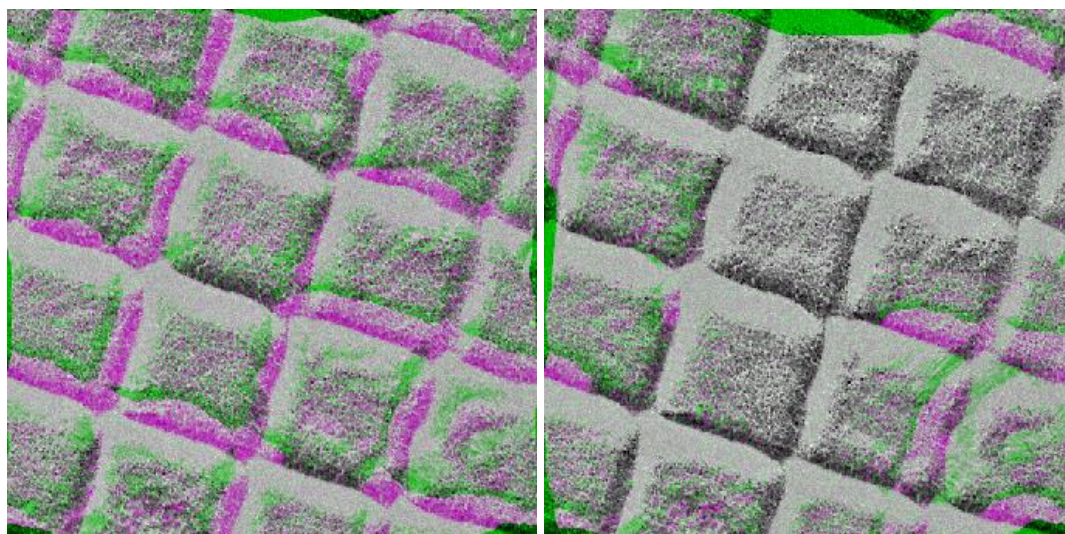


Fig. 7.27 The results of `imregdemons` registration. a) 100 iterations, 10 s, quality 0,9294. b) 500 iterations, 61 s, quality 0,9528. See the details in the Tab. 7.5.

According to the result in the Fig. 7.27, the demon registration was not successful after 100 iterations. After 500 iterations, the result was significantly better, but not perfect. It must be mentioned, that other increase of number of iterations is no longer conducive to such an improvement. Even after 1000 iterations, the images were not registered properly.

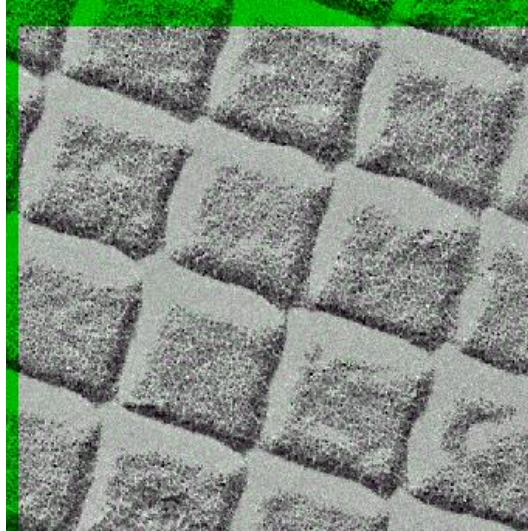


Fig. 7.28 The result of the GA function, when only a translation coefficients were used.

As was mentioned many times, the distortions of the real samples' images are very weak, therefore, the results (whereas visualized overlap or the calculated quality) are very good even when only the translation coefficients T_x and T_y are used for the registration. On the other hand, the quality counted is 0,9940 in this case. This number is lower than the best qualities counted for these images using the proper full GA function. From this, it can be judged that the use of translation is not sufficient and the images are truly slightly distorted. Another evidence of the distortion can be seen on the Fig. 7.20 and Fig. 7.21 b), the previous result of `elastix` and `imregdemons` registration.

8 AN APPLICATION OF THE REGISTRATION TO SUPPRESS THE GEOMETRIC DISTORTION

As was described in the Introduction, the force of the geometric distortion increases with increasing distance from the optical axis. Therefore, there is aperture embedded into a microscope that reduce the most distorted areas of a resulting image. In some applications, however, the large field of view is required, and it is appropriate to reduce or eliminate such a restriction. For this purpose, the present geometric distortion must be accurately corrected.

To this end, the registration was made and the exact description of the deformation field was obtained. It was the purpose of this master's thesis. Then, such a deformation field can be used to correct the geometric distortion according to the Fig. 8.1. Although the resulting deformation field describes the relationship between the two input images, it can also be used to describe the lens deformation field itself. Considering weak image distortion; see the Fig. 5.1, the image acquired at the centre of the field can be considered as undistorted. Taking account of this assumption and symmetry of the lens, only two acquired images are enough to describe the whole field. This way, the deformation can be easily counted wherever in the field.

The resulting displacement field is inversely applied to a following image acquisition. However, this is valid only in case the microscope user is working with the same system setting. Using this way of working, the aperture embedded into a microscope and reducing the most distorted areas of a resulting image, can be eliminated. The correction can be applied directly in the microscope software and a user will then see the corrected undistorted image.

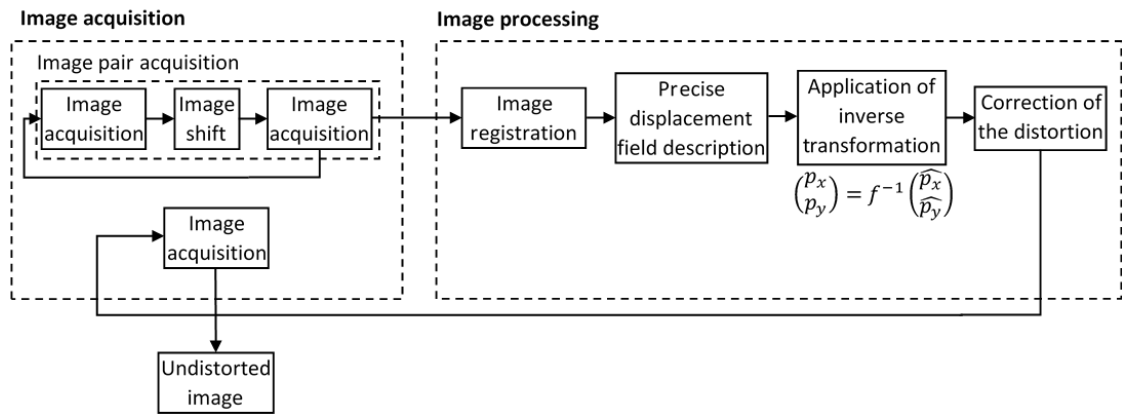


Fig. 8.1 A scheme of the distortion correction. (The exact expression of the equation of inverse transformation can be found in [35]. p_x represents the corrected x coordinates and \widehat{p}_x the resulting transformed x coordinates. The same is true for y coordinates.)

9 DISCUSSION

The registration method is based on two basic assumptions. The first is that the entire described field has a zero distortion exactly in its centre, and the second is that the deformation of the image is caused only by basic defects and the distortion. Therefore, in the register function GA and the registration engine for `fminsearch`, only the translation, shrinking and nonradiality of the distortion with not known distortion centre are considered. If there are other significant distortions in the images it may not be possible to precisely map the displacement field by the stated equation (used in both, GA function and `fminsearch`).

9.1 GA function

The designed and developed method is indicated for the particular type of images, deformed by weak geometrical distortions. Thus, the program is especially effective for previously mentioned weak geometric distortion. On the other hand, the function is considerably effective even for stronger distortions on which it was successfully tested.

The proposed method cannot describe the displacement field with a specific set of coefficients. For sufficient flexibility of the method for various and irregular distortions, there are 9 coefficients describing the distortion; as a result, different sets of coefficients lead to the same distortion. What is important is the combination, not the individual coefficients. Very important is an appropriate setting of the coefficients' limits. To limit the distortion creation enough but not to deny the variability. The most importantly, the translation and the movement in the deformation field must be limited. Otherwise, the moving image can escape out of the figure.

The proposed and developed method is very successful in the registration itself. It could evaluate all the real images which were acquired (23 image pairs of different samples, magnifications and noises). However, an accurate description of the deformation field is also important.

Since the correct solution is not verifiable by testing the exact coefficients' values, it can be tested over the transformation matrices. For each image registration, there are resulting transformation matrices x and y describing the positional difference of the same pixel in the reference and the moving image. Thus, when the resulting transformation matrix x from one registration and the same transformation matrix x from another registration are subtracted, the resulting matrix should contain zero elements or of very

low values. Such precise results, however, were not achieved by the presented method. In the case of real images evaluation, the resulting matrix of differences is formed by values in order of pixels on the edges and tenths or hundredths of pixels inside the image. And such a precision manifests itself only after considerably high number of iterations with a big number of individuals used. On the other hand, by subtracting the two resulting images after two identical registrations (of identical setting), very low numbers of differences (in hundredths of pixels) are obtained; registration itself is subpixel. Such a result is not surprising, because it is still a stochastic algorithm, which starts in different origins and can thus come to different areas during searching of solution.

To use this method in the microscope system, it must be tested and significantly refined on hundreds or thousands of images. It can be combined with any different registration method, which uses e.g. a gradient method for optimization, so the solution obtained has a greater chance of being a perfect global optimum. Such a method could follow up the proposed method and refine its solution.

It could be suggested that GA function's improvement could be caused by a prior evaluation of the translation. After a translation, e.g. on downsampled images, the function could focus only on the distortion itself. Unfortunately, due to the importance and strength of DF_x and DF_y coefficients, and its close relation to T_x , T_y coefficients, the translation and distortion are not easily separable.

The GA function is not as fast as `elastix` function or `imregdemons`. The acceleration of the GA function was one of the most difficult tasks. Downsampling and upsampling and its connection to the use of coefficients led to many blind alleyways. But the function is finally more robust for the evaluated type of images than `imregdemons` and even `elastix` (with tested setting); the GA function can handle even a regular structure. Among the acquired real samples' image pairs, there was not one pair that the GA function could not register. An average time for image pair evaluation is in units of minutes.

9.2 `fminsearch`

`fminsearch` function has several already mentioned disadvantages which have repeatedly manifested during testing.

The importance of appropriate limitation of the coefficients is even more highlighted in `fminsearch`, where the input coefficient estimation is absolutely crucial for success

of the registration. The input estimation decides whether the right solution is found or the function get stuck at a local minimum. This is `fminsearch`'s biggest weakness and a reason, why the function is not able to search for an optimum in an extensive range of values. The test kits of coefficients in individual iterations do not change sufficiently and quickly enough. In conclusion, such a function is more suitable for local optima searching. Despite this, when a high enough number of iterations and a strongly limited estimation of input values were used, the function could find correct solution. Only the individual solutions of the same problem (same image pair) were too diverse and its quality were too variable.

As proven in the chapter 7.3 Testing, in case of `fminsearch`, a good result cannot be guaranteed even after setting a high number of iterations. In addition to the large influence of input coefficient estimation, there is also considerable variability of quality in consecutive solutions; even after a big amount of iterations performed. When the `fminsearch` is stopped, the result can be significantly worse than another previously found solution. In the GA function, there is selected the best individual in every iteration and compared to the previous best individual. Therefore, the final solution cannot be worse than any previous.

Another disadvantage of `fminsearch` is an uncontrollability of the coefficient values that `fminsearch` tests; unlike the GA function, where the coefficients can be adjusted to the prescribed interval to avoid testing too low or high values and to help to find the right solution this way.

What could possibly improve the `fminsearch` results is a scale space approach. However, due to the uncontrollability of the coefficient values, it probably would not be an essential improvement.

The time requirements are comparable to GA function's. In this case, however, a lot less quality calculations occurs in the same time interval. The time efficiency of `fminsearch` is thus the smallest of all the tested functions.

Similarly, when used on real image pairs, the success of `fminsearch` was not very good and it did not rise with the increasing number of iterations. One of the factors that influenced it was the noise in the images.

9.3 elastix

After `elastix` registration, in most of the cases, the visual result is very good. The only problem of `elastix` is a registration of a regular structure. `elastix`'s great advantage

are very low time demands – high speed and especially its repeatability. Because the algorithm is based on the gradient descent optimization method, the solution is unambiguous. Therefore, the displacement field is described precisely and unambiguously. However, for the possibility of using `elastix` for correction of the geometric distortion, it is necessary to interpolate the displacement field by a function corresponding to the geometric distortion (for example, using the smallest square method).

9.4 `imregdemons`

The MATLAB function `imregdemons` must be labelled completely unsuccessful if it is running with the default iteration setting (100 iterations in each of 3 registration levels). However, if the number of iterations is increased, the function is significantly more successful and even retains its high speed. Out of the tested methods, `imregdemons` has the biggest problem with the noise present in the image, but after the number of iterations is increased, it could overcome it quite well. However, in case of real data, the success of `elastix` or GA is far from reaching for `imregdemons`. In the case of a regular structure, the result is better than in the case of `elastix`, but is far from equal to the quality of the GA function.

10 CONCLUSION

In the presented master's thesis, the main objective is to register images from transmission electron microscope. Such a registration can be used to correct the undesirable geometrical image distortion of microscopic images. For this purpose, it is necessary to precisely map the deformation field and unambiguously describe the distortion by the methods of image registration.

First, a research was conducted in the field of the image registration with a focus on the intensity based flexible image registration methods. This research was presented in the first three chapters of the thesis.

Subsequently, modelling of the geometric distortion preceded the image registration itself and can be considered successful. The image registration method based on polynomial transformation were proposed and tested with generally satisfying results.

The geometric distortion mapping requires an acquisition of the image pairs using the same microscope setup. This acquisition was performed in cooperation with FEI Czech Republic on a transmission electron microscope Talos L120C for Materials Science applications and is broadly described in the fourth chapter.

The developed method is presented and broadly described in the fifth chapter. The implemented GA function is sufficiently robust with scale space approach, adjustable parameters for different image sizes and differently strong distortions, user-friendly with its time requirements and precise enough on issues of registration itself. It displays and stores easily reproducible results and statistics.

For an objective comparison of the success of the developed method, it was compared with available existing methods and open-source programs (in MATLAB implemented `fminsearch` and `imregdemons` and open source software `elastix`). The sixth chapter of the thesis is devoted to the presentation of these methods and their usage, while their testing and comparison is in the seventh chapter together with visualized results. Although the `elastix` software was not effective for all the real images, thanks to its high speed, precision, repeatability and adaptation options, it can be described as the most appropriate method for accurate description of the deformation field. However, in this case, the resulting `elastix` transformation matrices will need to be interpolated by a function corresponding to the geometric distortion (for example, using the smallest square method).

There also exists a possibility of combination of the robustness of the GA function and repeatability and speed of the `elastix`. After the GA registration, the resulting images would then become `elastix` input. Such a result could indeed be realistically usable.

REFERENCES

- [1] BENDAYAN, M., PARANSKY, E. *Low Voltage Transmission Electron Microscopy in Cell Biology*. Progress in Histochemistry and Cytochemistry [online]. 2015 [cit. 2016-12-18]. B.m.: Elsevier GmbH, edn. 50, no. 1–2, p. 1–10. doi:10.1016/j.proghi.2015.05.001.
- [2] BOURKE, P. *Miscellaneous functions* [online]. 2001 [cit. 2016-10-25]. Available from: <http://paulbourke.net/miscellaneous/functions/>.
- [3] BROWN, L. G. *A Survey of Image Registration Techniques*. ACM Computing Surveys, Vol 24, No. 4, December 1994. Available from: <http://www.geo.uzh.ch/microsite/rsi-documents/research/SARlab/GMTILiterature/PDF/Bro92.pdf>.
- [4] CAMPBELL, R. *MATLAB Elastix; Elastix wrappers for MATLAB* [online]. 2016 [cit. 2016-11-22]. Available from: <http://www.mathworks.com/matlabcentral/fileexchange/52982-matlab-elastix> or https://github.com/raacampbell/matlab_elastix.
- [5] DEVERNAY, F., FAUGERAS, O. *Straight lines have to be straight*. Mach. Vision Appl. 13 (1), 14–24. Available from: <https://hal.archives-ouvertes.fr/inria-00267247/document>.
- [6] EGERTON, R. F. *Choice of operating voltage for a transmission electron microscope*. Ultramicroscopy [online] 2014. B.m.: Elsevier, edn. 145, p. 85–93. ISSN 18792723. doi:10.1016/j.ultramic.2013.10.019.
- [7] EGERTON, R. F. *Physical principles of electron microscopy; An introduction to TEM, SEM, and AEM*. 2005. ISBN 978-038-7260-167.
- [8] ERNI, R. *Aberration-Corrected Imaging in Transmission Electron Microscopy; An Introduction*, 2010. 2nd edition. Hackensack, NJ: Distributed by World Scientific Pub. Co., Chapter 7: Aberrations, p. 189–228. ISBN 978-1-84816-536-6. Available from: http://www.worldscientific.com/doi/suppl/10.1142/p703/suppl_file/p703_chap07.pdf.
- [9] GETREUR, P. *Linear Methods for Image Interpolation*. IPOL Journal, Image Processing On Line 2011. Edn. 1, p. 1–22. Available from: http://www.ipol.im/pub/art/2011/g_lmii/article.pdf.
- [10] CHMELÍK, J. *Affine Registration of Native and Contrast-Enhanced CT Brain Images*. Master's thesis. Czech Republic: Brno University of Technology, The Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering, 2013. Supervisor: Ing. Petr Walek 55p. ISBN 9780824758493. Available from: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=66415.
- [11] JAKUBÍČEK, R. *Movement Correction in Thoracic Dynamic Contrast CT Data*. Master's thesis. Czech Republic: Brno University of Technology, The Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering, 2015. 67 p., 2p. of attachments. Supervisor: Ing. Petr Walek. ISBN 3540401865. Available from: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=65710.

- [12] JAN, J. *Číslicová filtrace, analýza a restaurace signálu*. 2002. ISBN 80-214-1558-4.
- [13] JAN, J. *Medical Image Processing, Reconstruction and Restoration Concepts and Methods*, 2006. ISBN 9780824758493.
- [14] KARYBALI, I. G., PSARAKIS, E. Z., BERBERIDIS, K., EVANGELIDIS, G. D. *An efficient spatial domain technique for subpixel image registration*. *Signal Processing: Image Communication* [online] 2007. Edn. 23, no. 9, p. 711–724. ISSN 09235965, doi:10.1016/j.image.2008.08.003.
- [15] KARYBALI, I. G., PSARAKIS, E. Z., BERBERIDIS, K., EVANGELIDIS, G. D. *Efficient image registration with subpixel accuracy*. *European Signal Processing Conference, Eusipco* p. 1–5. 2006. ISSN 22195491. Available from: <http://www.eurasip.org/Proceedings/Eusipco/Eusipco2006/papers/1568982167.pdf>.
- [16] KLEIN, S., STARING, M. *Elastix* [online]. Webpage, 2004 [cit. 2016-11-20]. Available from: <http://elastix.isi.uu.nl/index.php>.
- [17] KLEIN, S., STARING, M. *Elastix; the manual* [online]. 2015 [cit. 2016-11-27]. Available from: http://elastix.isi.uu.nl/download/elastix_manual_v4.8.pdf.
- [18] KLEIN, S., STARING, M., MURPHY, K., VIERGEVER, M.A., PLUIM, J.P.W. *elastix: a toolbox for intensity based medical image registration*. *IEEE Transactions on Medical Imaging*, 2010. Vol. 29, no. 1, pp. 196 – 205. Available from: http://elastix.isi.uu.nl/marius/downloads/2010_j_TMI.pdf.
- [19] KORMUNDA, M. *TEM (Transmission Electron Microscopy), HRTEM (High Resolution TEM), SEM (Scanning Electron Microscopy), EDX (Energy-dispersive X-ray spectroscopy)*. A lecture on Material Science and Physics [online]. Ústí nad Labem: Jan Evangelista Purkyně University, Faculty of Science, [cit. 2016-11-19]. Available from: <http://physics.ujep.cz/~mkormund/P219/NanoMataChar-prednaska4.pdf>.
- [20] KROON, D.-J., SLUMP, C. H. *MRI Modality Transformation in Demon Registration*. Netherlands: University of Twente, Institute for Biomedical Technology & Technical Medicine.
- [21] KUHN, M. *Matematické modelování deformací tkání v MRI obrazech*. Bachelor thesis. Czech Republic: Masaryk University, Faculty of Science, 2010. Available from: http://is.muni.cz/th/269560/prif_b/BP_Kuhn_2010.pdf.
- [22] KAYNIG, V., FISCHER, B., MÜLLER, E., BUHMANN, J. M. *Fully automatic stitching and distortion correction of transmission electron microscope images*. *Journal of Structural Biology* 2010. B.m.: Elsevier Inc., edn. 171, no. 2, p. 163–173. ISSN 10478477, doi: 10.1016/j.jsb.2010.04.012.
- [23] KYBIC, J. *Image registration and its biomedical applications*. Prague: Czech Technical University, 2015. ISBN 978-80-01-05740-7.

- [24] LIU, S. *Spline interpolation*. Scotland: University of St Andrews, Numerical Analysis course, chapter 3: Interpolation. 2011. Available from: http://www.geos.ed.ac.uk/~yliu23/docs/lect_spline.pdf.
- [25] MÉZL, M., KOZUMPLÍK, J. Lectures on FEAL: *Evoluční algoritmy*. 2015. Czech Republic: Brno University of Technology, The Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering.
- [26] MALÍNSKÝ, M., SHÁNĚL, O. *Úvod do elektronové mikroskopie*. A lecture on FMZT: Mikroskopická zobrazovací technika [online]. Czech Republic: Brno University of Technology, The Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering, 2015. [cit. 2016-11-19].
- [27] NOVOTNÝ, R., VÁLOVÁ, P. *Princip a metody elektronové mikroskopie*. A lecture on Světelná a elektronová mikroskopie [online]. Czech Republic: Palacký University Olomouc, Faculty of Science, 2014 [cit. 2016-11-19].
- [28] PARKER, J. A., KENYON, R. V., TROXEL, D. E. *Comparison of Interpolating Methods for Image Resampling*. 1983. ISBN 0278-0062. Available from: <http://www2.cs.uic.edu/~kenyon/Papers/Comparison.of.Interpolating.Methods.Parke.r.Kenyon.Troxel.pdf>.
- [29] SHAMONIN, D.P., BRON, E.E., LELIEVELDT, B.P.F., SMITS, M., KLEIN, S., STARING, M. *Fast Parallel Image Registration on CPU and GPU for Diagnostic Classification of Alzheimer's Disease*. *Frontiers in Neuroinformatics*, 2014. Vol. 7, no. 50, pp. 1-15. Available from: http://elastix.isi.uu.nl/marius/downloads/2014_j_FNI.pdf.
- [30] *Talos L120C for Materials Science*. Datasheet. FEI, part of Thermo Fisher Scientific, 2016. Available from: <https://www.fei.com/talos-l120c-for-materials-science/>.
- [31] *TEM/STEM Test Specimens*. Ted Pella, Inc. Microscopy Products for Science Industry. Available from: https://www.tedpella.com/calibration_html/TEM_STEM_Test_Specimens.htm.
- [32] TEMELOVÁ, K. *Correction of image distortion of microscopic scene*. Master's thesis. Czech Republic: Brno University of Technology, Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering, 2016, p. 82. Supervisor of the thesis: Ing. Jan Odstrčilík, Ph.D. Available from: https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=124987.
- [33] THYSSEN, A. Examples of ImageMagick Usage (Version 6) [online]. 2008. [cit. 2016-11-05]. Available from: <http://www.imagemagick.org/Usage/misc/>.
- [34] TVRDÍK, J. *Evoluční algoritmy*. Ostrava: University of Ostrava, Faculty of Science, 2004.
- [35] VASS, G., PERLAKI, T. *Applying and removing lens distortion in post production*. Colorfront Ltd., Budapest. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.136.3745&rep=rep1&type=pdf>.

- [36] VERCAUTEREN, T., PENNEC, X., PERCHANT, A., AYACHE, N. *Diffeomorphic demons: Efficient non-parametric image registration*. NeuroImage, 2009. Vol. 45, no. 1, pp. S61-S72. Available from: http://ac.els-cdn.com/S1053811908011683/1-s2.0-S1053811908011683-main.pdf?_tid=24248f04-1e31-11e7-b9a9-00000aacb35d&acdnat=1491858386_ae35e69d26a583283e834685deae74ea.
- [37] WOLBERG, G. *Geometric transformation techniques for digital images: a survey*. NY: Columbia University, Department of Computer Science, 1988. Available from: <https://core.ac.uk/download/pdf/27291581.pdf>.
- [38] WALEK, P., LAMOŠ, M., JAN, J. *Analýza biomedicínských obrazů počítačová cvičení*. Czech Republic: Brno University of Technology, The Faculty of Electrical Engineering and Communication, Department of Biomedical Engineering, 2015. ISBN 9788021447929.
- [39] WILLIAMS, D., B., CARTER C. B. *Transmission Electron Microscope; Training module* [online] 1996. [cit. 2016-12-18]. MyScope, training for advanced research, no. 18, p. 729. Available from: <http://www.ammrf.org.au/myscope/pdfs/tem.pdf>, doi:10.1007/978-1-4757-2519-3_1.
- [40] ZHANG, Z. *A flexible new technique for camera calibration*. IEEE T. Pattern Anal. 22 (11), 1330–1334. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?rep=rep1&type=pdf&doi=10.1.1.220.534>.
- [41] ZITOVÁ, B., FLUSSER, J. 2003. Image registration methods: A survey. *Image and Vision Computing*. An. 21, no. 11, p. 977–1000. ISSN 02628856. doi:10.1016/S0262-8856(03)00137-9.
- [42] Collective of authors. *Insight Segmentation and Registration Toolkit* [online]. Webpage, 2002 [cit. 2016-12-30]. Available from: <https://itk.org/>.

LIST OF ABBREVIATIONS



CPs	Control Points
GA	Genetic Algorithm
LVTEM	Low Voltage Transmission Electron Microscope
SNR	Signal-to-Noise Ration
SSD	Sum of Squared Differences
TEM	Transmission Electron Microscope




SUPPLEMENTARY MATERIALS




A PARAMETERS' EFFECT TO AN IMAGE



The figures in this attachment illustrate how each coefficient affect image distortion.

Tab. 10.1 Parameters' effect table.

	Image	Parameters
1.		<pre>DFx = 0; DFy = 0; kappa1 = 0*10⁻⁹; kappa2 = 0*10⁻¹²; lambdax = 0; lambday = 0; s = 1; Tx = 0; Ty = 0;</pre> <p>This is a default setting with no influence on an image.</p>
2.		<pre>DFx = 0; DFy = 0; kappa1 = 0*10⁻⁹; kappa2 = 10*10⁻¹²; lambdax = 0; lambday = 0; s = 1; Tx = 0; Ty = 0;</pre> <p>Only one variable (kappa2, in bold) was changed to simulate a barrel distortion.</p>

3.		<p>DFx = 50; DFy = 0; kappa1 = 0*10⁻⁹; kappa2 = 10*10⁻¹²; lambdax = 0; lambday = 0; s = 1; Tx = 0; Ty = 0;</p> <p>The same setting as in previous image with a shift in a displacement field (DFx, x axis (columns)).</p>
4.		<p>DFx = 50; DFy = -20; kappa1 = 0*10⁻⁹; kappa2 = 10*10⁻¹²; lambdax = 0; lambday = 0; s = 1; Tx = 0; Ty = 0;</p> <p>A shift in a deformed field in y axis added into DFy.</p>
5.		<p>DFx = 0; DFy = 0; kappa1 = 1000*10⁻⁹; kappa2 = 10*10⁻¹²; lambdax = 0; lambday = 0; s = 1; Tx = 0; Ty = 0;</p> <p>Comparing with the second image, only kappa1 was added to make a barrel distortion stronger.</p>

6.		<pre>DFx = 0; DFy = 0; kappa1 = 1000*10⁻⁹; kappa2 = 10*10⁻¹²; lambdax = 0; lambday = 10; s = 1; Tx = 0; Ty = 0;</pre> <p>In this image, lambday was changed to simulate different distortion in x and y axis.</p>
7.		<pre>DFx = 0; DFy = 0; kappa1 = 1000*10⁻⁹; kappa2 = 10*10⁻¹²; lambdax = 10; lambday = 0; s = 1; Tx = 0; Ty = 0;</pre> <p>To highlight the previous effect, lambdax was modified too.</p>
9.		<pre>DFx = 0; DFy = 0; kappa1 = 0*10⁻⁹; kappa2 = 10*10⁻¹²; lambdax = 0; lambday = 0; s = 1; Tx = -10; Ty = 40;</pre> <p>In this case, only a translation in both axes is represented. A shift -10 pixels in x axis leads to a shift to the right side and 40 pixels in y axis leads to a shift upward.</p>

10.		<pre>DFx = 0; DFy = 0; kappa1 = 0*10⁻⁹; kappa2 = -1*10⁻¹²; lambdax = 0; lambday = 0; s = 1; Tx = -25; Ty = 40;</pre> <p>Finally, the crucial importance of the sign is presented. Only a sign before kappa2 was changed and a pin cushion distortion occurred. The order of this parameter was also decreased, because the pin cushion is stronger than the barrel distortion.</p>
12.		<pre>DFx = 0; DFy = 0; kappa1 = 0*10⁻⁹; kappa2 = -1*10⁻¹²; lambdax = 0; lambday = 0; s = 6; Tx = -25; Ty = 40;</pre> <p>Eventually, the shrinking effect of parameter s is introduced.</p>

13.



```
DFx = 0;  
DFy = 0;  
kappa1 = 10*10-9;  
kappa2 = -100*10-12;  
lambdax = 0;  
lambday = 0;  
s = 1;  
Tx = 0;  
Ty = 0;
```

A brief reminder of how important it is to set the limits and not to overcome them.

B HOW TO CONTROL THE PROGRAM

The whole program is running and controlled of the `OriginFunction.m`. When a user run the function, first, an interactive window opens. Here a user selects the folder where the results will be saved. Here it is necessary to emphasize that if a user wants to run the `elastix` program, the directory must be without white spaces. `Elastix` is not able to work with such a directory with white spaces in it.

B.1 Input Images

In a section `%% INPUT IMAGES`, a user must specify the name of the image pair to evaluate.

Now, it will be described how to enter the image names if they are stored in a matrix, `.mat` file. The names are specified into the variables `nameImgStat` for the reference image and `nameImgMov` for the moving image. These images are then loaded. Please note, that important is the inner name of the image; how the image was saved into the loaded matrix. In a following example, the name of the image saved into the matrix `No_Dist.mat` was actually `I1`. During the whole run, the program works with the images as with `img` for the reference image and `img2` for the moving one. So, the images must be saved into these variables after loading (as in the following example) or they must be already saved that way.

```
nameImgStat = 'No_Dist'; % reference image
nameImgMov = 'LenaBarrel_10'; % moving image
load (nameImgStat);
img = I1;
load (nameImgMov);
img = I2;
```

The second option is to have images saved not in matrices `.mat`, but in an image format, such as `.tif`. In this case, the loading is a little different. Example follows again. In this case, there is no need to be careful about the name, only it must be set into the variables `nameImgStat`, `nameImgMov` with the suffixes.

```
nameImgStat = 'AcquireCCD11.tif';
nameImgMov = 'AcquireCCD12.tif';
img = im2double(imread(nameImgStat));
img2 = im2double(imread(nameImgMov));
```

B.2 Method Choice and Setting

Here follows a section `%% METHOD CHOICE & SETTING CHOICE` where a user select the method of registration and the parameters of the method will be set. The method is selected in a variable `Method`:

```
% Choose the method:
% 0 ... Genetic Algorithm (GA)
% 1 ... fminsearch
% 2 ... elastix
% 3 ... demon algorithm
Method = 2;
```

It may not necessarily be the same every time, but generally, the test data is stored in the matrix format `.mat` and the real samples images are in the image format `.tif`. These two types of data are treated differently. Especially because the test data are usually about 400x400 px size and the real images 4096x4096 px size. Thus, for example `dsVec` (downsampling vector) for all the optional functions is advisable to be set differently. Due to the distortion nature in this two cases, the orders of the `kappa1` and `kappa2` coefficients in the `TransformationFunctionGA.m` (and the `TransformationFunctionFMIN.m`) are also different for test and real data. Therefore, an `ImgType` variable was introduced to simplify changing between these two data types. So, a user does not have to change the entire function for each change of data. In case of small images (test data deformed by stronger distortions), the `ImgType = 0`, for large images (real samples data deformed very slightly) `ImgType = 1`. This variable is specified in the previous section along with the loading of the images. Then an input set for `fminsearch` function, `elastix` and `demon` registration may look as follows.

```
if ImgType == 0 % example for Lena
    dsVec = [1];
    nIter = [];
    nInd = [];
elseif ImgType == 1 % example for real images
    dsVec = [8];
    nIter = [];
    nInd = [];
end
```

In case of the GA function, the setting will probably be a little more specific. There are

vectors specifying the number of samples, iterations and individuals in each level of the registration, for example:

```
if Method == 0 % example for GA function
    if ImgType == 0 % example for Lena
        dsVec = [4 1]; % downsampling
        nIter = [20 10]; % number of iterations
        nInd = [24 12]; % number of individuals
    elseif ImgType == 1 % example for real images
        dsVec = [32, 8];
        nIter = [12, 8];
        nInd = [20, 10];
    end
end
```

The coefficients order inside Transformation Functions then looks like:

```
if ImgType == 0;
    DFx = coeffs0(1);
    DFy = coeffs0(2);
    kappa1 = coeffs0(3).*10e-9;
    kappa2 = coeffs0(4).*10e-12;
    lambdax = coeffs0(5);
    lambday = coeffs0(6);
    s = coeffs0(7);
    Tx = coeffs0(8);
    Ty = coeffs0(9);
elseif ImgType == 1;
    DFx = coeffs0(1);
    DFy = coeffs0(2);
    kappa1 = coeffs0(3).*10e-20;
    kappa2 = coeffs0(4).*10e-20;
    lambdax = coeffs0(5);
    lambday = coeffs0(6);
    s = coeffs0(7);
    Tx = coeffs0(8);
    Ty = coeffs0(9);
end
```

Of course, this way can be easily introduced another type of data and its processing setting.

After appropriate setting of variables `dsVec`, `nIter` and `nInd`, all the functions can be run. The details for such a setting can be found in the function description in the chapter 5.2.1 Origin and in the Tab. 5.1 Summary table of setting options of the GA function. in the chapter 5.3 Setting Options.

If `fminsearch` function is chosen for the registration, one more variable, `MaxFminEval`, must be set. It specifies in how many iterations the function will terminate.

For a proper setting and functionality of `elastix`, `Method = 2`, see the chapter 6.2.1 Access from MATLAB.

C THE PARAMETERS FOR THE SET OF THE **elastix** IMAGE REGISTRATION

C.1 AtW_parameters_1.txt

```
(FixedImageDimension 2)
(MovingImageDimension 2)
(UseDirectionCosines "true")
(Registration "MultiResolutionRegistration")
(Interpolator "BSplineInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(Optimizer "StandardGradientDescent")
(Transform "AffineTransform")
(Metric "AdvancedMattesMutualInformation")
(AutomaticScalesEstimation "true")
(AutomaticTransformInitialization "true")
(HowToCombineTransforms "Compose")
(FinalGridSpacingInVoxels 16)
(NumberOfMovingHistogramBins 32)
(NumberOfFixedHistogramBins 32)
(ErodeMask "true")
(NumberOfResolutions 5)
(ImagePyramidSchedule 8 8 4 4 2 2 1 1)
(MaximumNumberOfIterations 400)
(NumberOfSpatialSamples 1000)
(NewSamplesEveryIteration "true")
(ImageSampler "Random")
(BSplineInterpolationOrder 1)
(FinalBSplineInterpolationOrder 3)
(DefaultPixelValue 0)
(WriteResultImage "true")
(ResultImagePixelFormat "short")
(ResultImageFormat "mhd")
(SP_alpha 0.600000)
(SP_A 50)
```

```
(WriteTransformParametersEachIteration "false")
(WriteTransformParametersEachResolution "false")
(ShowExactMetricValue "false")
(FixedInternalImagePixelType "float")
(MovingInternalImagePixelType "float")
(CompressResultImage "false")
```

C.2 AtW_parameters_2.txt

```
(FixedImageDimension 2)
(MovingImageDimension 2)
(UseDirectionCosines "true")
(Registration "MultiResolutionRegistration")
(Interpolator "BSplineInterpolator")
(ResampleInterpolator "FinalBSplineInterpolator")
(Resampler "DefaultResampler")
(Optimizer "StandardGradientDescent")
(Transform "BSplineTransform")
(Metric "AdvancedMattesMutualInformation")
(AutomaticScalesEstimation "true")
(AutomaticTransformInitialization "true")
(HowToCombineTransforms "Compose")
(FinalGridSpacingInVoxels 16)
(NumberOfMovingHistogramBins 32)
(NumberOfFixedHistogramBins 32)
(ErodeMask "true")
(NumberOfResolutions 5)
(ImagePyramidSchedule 8 8 4 4 2 2 1 1)
(MaximumNumberOfIterations 1000)
(NumberOfSpatialSamples 1000)
(NewSamplesEveryIteration "true")
(ImageSampler "Random")
(BSplineInterpolationOrder 1)
(FinalBSplineInterpolationOrder 3)
(DefaultPixelValue 0)
(WriteResultImage "true")
```



```
(ResultImagePixelFormat "double")
(ResultImageFormat "mhd")
(SP_a 4000)
(SP_alpha 0.600000)
(SP_A 50)
(WriteTransformParametersEachIteration "false")
(WriteTransformParametersEachResolution "false")
(ShowExactMetricValue "false")
(FixedInternalImagePixelFormat "float")
(MovingInternalImagePixelFormat "float")
(CompressResultImage "false")
```