PICTURE DEFORMATION RECOVERY OF CADASTRAL MAPS[1]

Róbert Bohdal[2]

Abstract

In this paper we will discuss various methods of *picture deformation recovery*. These methods work like commonly known *image registration methods*, which use control points to describe how parts of an image will be transformed. All methods are explored and their accuracy in the picture deformation recovery are compared.

Keywords: image registration, thin plate spline, Shepard's method, Clough-Tocher method.
2000 Mathematics Subject Classification: 65D07, 68U05, 68U10.

1 Introduction

Motivation for this paper was a need for an accurate geometric correction of scanned *cadastral maps* (maps of land lots and owners). Many of these maps are very old and they are distorted or deformed by climatic influences. The identification points (Geodetic Control Points), that represent part of distinguish objects such as old trees, corners of big buildings, etc. are often used in the process of a map creation. If we know their accurate position, they can help us remove map deformation. Cadastral maps have also another type of identification points, which make regularly spaced rectangular grid.

Our paper describes some picture transformation methods. All of them can be divided into two classes – *one-segment* or *many-segment* transformation methods.

The paper is structured as follows. Section 2 introduces image processing of maps. Section 3 describes one-segment and many-segment transformation methods. Section 4 contains comparison of here presented methods. Finally, section 5 concludes the work.

2 Image Processing

When we obtain a picture in a digital form, it is often necessary to modify it for further processing. At first, we choose proper picture cutout in order to eliminate unwanted parts of the picture. Then we adjust the *brightness* and *contrast*, eventually we apply the *gamma correction* to it. Next, the filtration process removes some unwanted artifacts like scanned hairs or dust. The cadastral maps use only black & white information therefore the picture is converted to grayscale and then it is binarized[3]. Finally, applying morphology operations we can remove isolated points, eventually we emphasize thin lines.

[1]This paper was supported by VEGA Grant No. 1/3024/06.

[2]Department of Algebra, Geometry and Didactics of Mathematics, Faculty of Mathematics, Physics and Informatics, Comenius University, Mlynská dolina, 842 48 Bratislava, Slovakia, email: bohdal@fmph.uniba.sk

[3]This conversion also takes away yellowish tint of old maps.

Filtration

Bilateral filtration or filtration based on the *anisotropic diffusion* is very convenient for filtration of map pictures because it preserves edges of objects.

Anisotropic diffusion filter. Perona and Malik formulate the anisotropic diffusion filter as a diffusion process that encourages intraregion smoothing while inhibiting interregion smoothing. Mathematically, the process is defined as follows [5]:

$$\partial_t \boldsymbol{u}(\boldsymbol{x},t) = \operatorname{div}\big(g(|\nabla \boldsymbol{u}(\boldsymbol{x},t)|)\cdot \nabla \boldsymbol{u}(\boldsymbol{x},t)\big), \tag{2.1}$$

where $g(s) = \dfrac{1}{1+(s/\lambda)^{1+\alpha}}$ or $g(s) = \dfrac{1}{e^{(s/\lambda)^{1+\alpha}}}$ with $\alpha > 0$.
In our case, $\boldsymbol{u}(\boldsymbol{x},t)$ represents the input image. The symbol $\boldsymbol{x}$ refers position in the image and t refers to the iteration step. The function $g(s)$ is called the diffusion function. It is a monotonically decreasing function of the image gradient magnitude. It allows for locally adaptive diffusion strengths; edges are selectively smoothed or enhanced based on the evaluation of the diffusion function.

Bilateral filtering. This filtering technique smooths images while preserving edges, by means of a nonlinear combination of nearby image values [6]. The kernel of a bilateral filter is composed of an inner product of two filters. The first of these filters is a normal low pass filter, which basically averages the neighboring pixel intensity values with decreasing weights for pixels at larger distances. In the second one, weights for the neighboring pixels are derived from the pixel intensity value differences to the center pixel intensity value instead of geometric distances. Combined filtering can be described as follows [6]:

$$\boldsymbol{h}(\boldsymbol{x}) = k^{-1}(\boldsymbol{x}) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \boldsymbol{u}(\boldsymbol{\xi}) c_d(\boldsymbol{x},\boldsymbol{\xi}) c_r(\boldsymbol{x},\boldsymbol{\xi}) d\boldsymbol{\xi}, \tag{2.2}$$

where $k(\boldsymbol{x}) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c_d(\boldsymbol{x},\boldsymbol{\xi}) c_r(\boldsymbol{x},\boldsymbol{\xi}) d\boldsymbol{\xi}$. The symbol $c_d(\boldsymbol{x},\boldsymbol{\xi}) = e^{-\frac{1}{2}\left(\frac{||\boldsymbol{x}-\boldsymbol{\xi}||}{\sigma_d}\right)^2}$ denotes the Euclidean distance between $\boldsymbol{\xi}$ and $\boldsymbol{x}$ and the $c_r(\boldsymbol{x},\boldsymbol{\xi}) = e^{-\frac{1}{2}\left(\frac{||\boldsymbol{u}(\boldsymbol{x})-\boldsymbol{u}(\boldsymbol{\xi})||}{\sigma_r}\right)^2}$ is distance between the two intensity values $\boldsymbol{u}(\boldsymbol{x})$ and $\boldsymbol{u}(\boldsymbol{\xi})$.

RGB to Grayscale Conversion

There are many color conversion functions which convert a pixel with the color components (R,G,B) to its grayscale value Y. One of the good choices is to use expression:

$$Y = 0.212656R + 0.715158G + 0.072186B.$$

Binarization

Binarization (or *tresholding*) is based upon simple concept. A parameter θ called *brightness treshold* is chosen and applied to the image $\boldsymbol{u}(\boldsymbol{x})$ as follows:

$$\boldsymbol{u}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{u}(\boldsymbol{x}) > \theta \\ 0 & \text{else} \end{cases}$$

An alternative approach is to use an adaptive tresholding. For each pixel in the image, a local threshold is calculated by statistically examining the intensity values of its neighborhoods.

Morphology operations

Morphology operations are very useful when we want emphasize objects in the picture or remove isolated dots. We can choose any undermentioned operation or their combinations:
dilatation – $D(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \oplus \boldsymbol{B}$,
erosion – $E(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \ominus \boldsymbol{B}$,
opening – $O(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \circ \boldsymbol{B} = (\boldsymbol{A} \ominus \boldsymbol{B}) \oplus \boldsymbol{B}$,
closing – $C(\boldsymbol{A}, \boldsymbol{B}) = \boldsymbol{A} \bullet \boldsymbol{B} = (\boldsymbol{A} \oplus \boldsymbol{B}) \ominus \boldsymbol{B}$,
where $\boldsymbol{A}$ is an object and $\boldsymbol{B}$ is a proper structural element. Symbols $\oplus$ and $\ominus$ denote two known *Minkowski* operations. Dilatation, in general, causes objects to grow in size, while erosion causes objects to shrink. The opening operation can separate objects that are connected and the closing operation can fill in small holes.

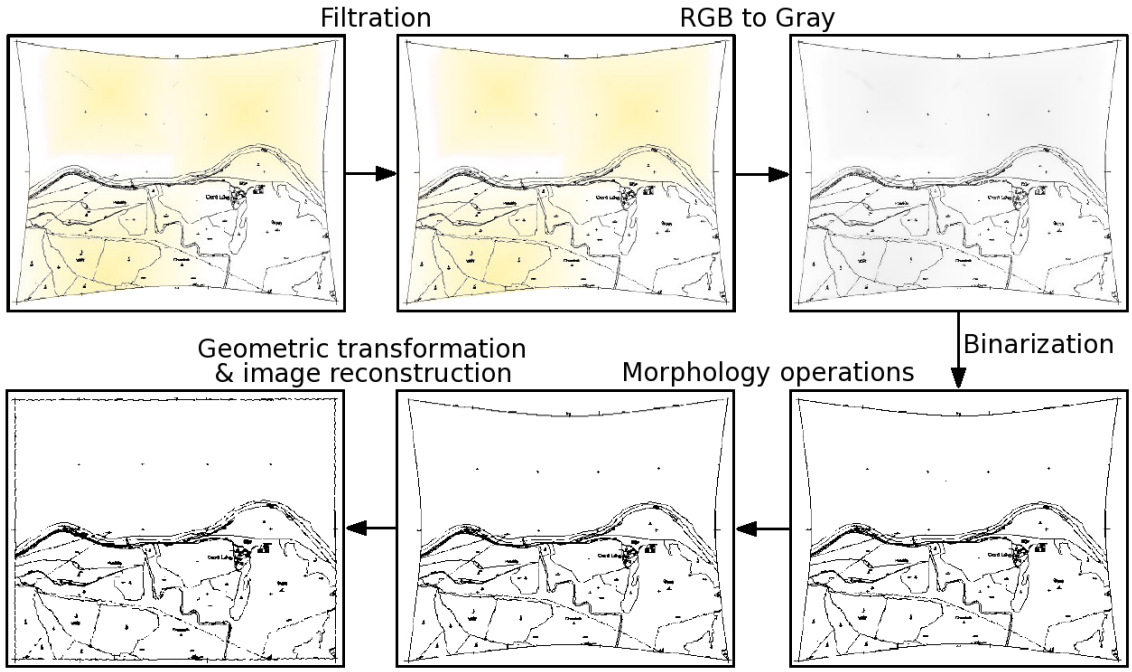

Figure 1: Stages of maps correction

3 Transformation Methods

Suppose, we are given two sets of n points $\mathcal{P}, \mathcal{V} \in \mathbb{E}^2$, $\mathcal{P} = \{\boldsymbol{p}_i[x_i, y_i] \in \mathbb{E}^2;\ i = 1, \ldots, n\}$. The set $\mathcal{P}$ consists of identification points on a deformed picture and the set $\mathcal{V}$ comprises of points on an ideal undeformed picture. We call pairs $(\boldsymbol{p}_i, \boldsymbol{v}_i)$ as corresponding points. Then we seek a transformation function $\boldsymbol{f} : \mathbb{E}^2 \mapsto \mathbb{E}^2$ such that $\boldsymbol{f}(\boldsymbol{p}_i) = \boldsymbol{v}_i$, where $i = 1, \ldots, n$. Using the function $\boldsymbol{f}(\boldsymbol{x})$, we will transform all points of the input picture.

3.1 One-segment Transformation Methods

Methods in this class are given by one formula on the whole domain. Advantage of these methods is their simple expression and possibility to compute pixels of the new picture laying outside the convex hull of the input corresponding points. Their transformation functions have almost global influence on the transformed picture.

3.1.1 Thin Plate Splines Method

Thin Plate Splines method (or *radial basis function methods*) is the most preferable method for the image warping. We can also use it in the picture deformation recovery.

The interpolating transformation function $\boldsymbol{f}(\boldsymbol{x})$ has the form [2]:

$$\boldsymbol{f}(x,y) = \boldsymbol{c}_1 + \boldsymbol{c}_2 x + \boldsymbol{c}_3 y + \frac{1}{2}\sum_{i=1}^{n} \boldsymbol{\lambda}_i r_i^2 log(r_i^2), \text{ where } [x,y] \in \mathbb{E}^2, \tag{3.1.1.1}$$

$r_i^2 = (x - x_i)^2 + (y - y_i)^2$ and $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3, \boldsymbol{\lambda}_i$ are unknown quantities. The parameters $\boldsymbol{\lambda}_i,\ i = 1,\ \ldots,\ n$ have to satisfy the boundary conditions:

$$\sum_{i=1}^{n} \boldsymbol{\lambda}_i = \boldsymbol{0} \text{ and } \sum_{i=1}^{n} \boldsymbol{\lambda}_i \boldsymbol{p}_i = \boldsymbol{0}. \tag{3.1.1.2}$$

Applying interpolation conditions $\boldsymbol{f}(\boldsymbol{p}_i) = \boldsymbol{v}_i$, where $i = 1,\ \ldots,\ n$ together with boundary conditions 3.1.1.2 we can compute the unknown values via the next system of equations:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & x_1 & x_2 & \cdots & x_n \\ 0 & 0 & 0 & y_1 & y_2 & \cdots & y_n \\ 1 & x_1 & y_1 & 0 & r_{21}^2 \log(r_{21}^2) & \cdots & r_{n1}^2 \log(r_{n1}^2) \\ 1 & x_2 & y_2 & r_{12}^2 \log(r_{12}^2) & 0 & \cdots & r_{n2}^2 \log(r_{n2}^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & y_n & r_{1n}^2 \log(r_{1n}^2) & r_{2n}^2 \log(r_{2n}^2) & \cdots & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \\ \boldsymbol{\lambda}_1/2 \\ \boldsymbol{\lambda}_2/2 \\ \vdots \\ \boldsymbol{\lambda}_n/2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{f}_n \end{pmatrix}, \tag{3.1.1.3}$$

where $r_{ij}^2 = r_{ji}^2 = (x_j - x_i)^2 + (y_j - y_i)^2$.

3.1.2 Shepard's Method

Shepard defined his interpolating function $\boldsymbol{f}(\boldsymbol{x})$ to be weighted mean of the coordinates $\boldsymbol{v}_i$ [4]:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \omega_i(\boldsymbol{x}) \boldsymbol{v}_i. \tag{3.1.2.1}$$

Weight functions $\omega_i(\boldsymbol{x})$ from formula 3.1.2.1 can be expressed:

$$\omega_i(\boldsymbol{x}) = \frac{\sigma_i(\boldsymbol{x})}{\sum_{j=1}^{n} \sigma_j(\boldsymbol{x})}, \tag{3.1.2.2}$$

where $\sigma_i(\boldsymbol{x}) = ||\boldsymbol{x} - \boldsymbol{p}_i||^{-\mu_i}$, for $\mu_i > 0$. The parameter μ_i allows to control the shape of the final surface in the neighborhood of the interpolated points.

The global character of this method can be made local by multiplying weighted function $\omega_i(\boldsymbol{x})$ by mollifying function [4]:

$$\lambda_i(\boldsymbol{x}) = \left(1 - \frac{d_i(\boldsymbol{x})}{R_i}\right)_+^{\mu}, \text{ where } R_i > 0.$$

The radius R_i we can set to $1/2D(N_w/n)^{1/2}$, where D is the maximum distance between arbitrary two points of the set $\mathcal{P}$ and $N_w = 19$.

Franke and Nielson in [3] proposed to replace $\boldsymbol{v}_i$ by local interpolating functions $\boldsymbol{L}_i(\boldsymbol{x})$ with interpolation property $\boldsymbol{L}_i(\boldsymbol{p}_i) = \boldsymbol{v}_i$. Then we get:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{i=1}^{n} \omega_i(\boldsymbol{x}) \boldsymbol{L}_i(\boldsymbol{x}). \tag{3.1.2.3}$$

If the interpolation functions are quadratic, we obtain sufficiently smooth surfaces with relatively low computational complexity.

Modified quadratic Shepard's method is expressed by formula:

$$\boldsymbol{f}(x, y) = \sum_{i=1}^{n} \omega_i(x, y) \boldsymbol{Q}_i(x, y), \tag{3.1.2.4}$$

where the local quadratic interpolant $\boldsymbol{Q}_k(x, y)$ is defined by:

$$\begin{aligned} \boldsymbol{Q}_i(x, y) =& \boldsymbol{c}_{i,1}(x - x_i)^2 + \boldsymbol{c}_{i,2}(x - x_i)(y - y_i) + \\ & \boldsymbol{c}_{i,3}(y - y_i)^2 + \boldsymbol{c}_{i,4}(x - x_i) + \boldsymbol{c}_{i,5}(y - y_i) + \boldsymbol{v}_i. \end{aligned} \tag{3.1.2.5}$$

The coefficients $\boldsymbol{c}_{i,j}$ in $\boldsymbol{Q}_j(x, y)$ can be computed by least square method using conditions:

$$\sum_{\substack{k=1 \\ k<>i}}^{n} \omega_k(x_i, y_i)[\boldsymbol{c}_{i,1}(x_k - x_i)^2 + \cdots + \boldsymbol{c}_{i,5}(y_k - y_i) + \boldsymbol{f}_i - \boldsymbol{f}_k]^2 \longrightarrow min, \tag{3.1.2.6}$$

where $\omega_k(x, y) = \left(\frac{R_q - d_k(x,y)}{R_q d_k(x,y)}\right)_+^2$ and R_q is a radius of influence around the point $\boldsymbol{p}_i[x_i, y_i]$.

3.2 Many-segment Transformation Methods

Many-segment methods (mostly known as *Finite Element Methods*) are primarily based on a triangulation of the input points of the set $\mathcal{P}$. This triangulation divides the area of the input picture into triangle regions. In these methods, we need to ensure that whole transformed picture lies inside the convex hull of the input identification points.

3.2.1 Clough-Tocher Method

To construct C^1 continuous cubic Clough-Tocher interpolant, we divide each triangle of the input triangulation into three *minitriangles* by connecting its vertices to a point lying inside the triangle (e.g. the barycenter of the triangle).

The final function $f(\boldsymbol{x})$ will be C^1 continuous surface which consists of cubic Bèzier triangles over all minitriangles.

Clough-Tocher method uses cubic Bèzier triangles in the form:

$$\begin{aligned}\boldsymbol{X}(u,v,w) =& \boldsymbol{b}_{300}u^3 + 3\boldsymbol{b}_{210}u^2v + 3\boldsymbol{b}_{120}uv^2 + \\ & \boldsymbol{b}_{030}v^3 + 3\boldsymbol{b}_{021}v^2w + 3\boldsymbol{b}_{012}vw^2 + \\ & \boldsymbol{b}_{003}w^3 + 3\boldsymbol{b}_{102}w^2u + 3\boldsymbol{b}_{201}wu^2 + 6\boldsymbol{b}_{111}uvw.\end{aligned} \tag{3.2.1.1}$$

Computation of the Bèzier Ordinates $\boldsymbol{b}_{ijk}$

The Bèzier ordinates of the control net of three adjacent triangle patches can be evaluated by the next algorithm [1]:

The coordinates $[x, y]$ of the ten Bèzier ordinates within each minitriangle are located either at the minitriangle vertices, or at the 1/3 or 2/3 length of each edge, or at the barycenter of the minitriangle.

The values z of these Bèzier ordinates are determined from the next steps:

1. The values z of the Bèzier ordinates (above P_1 and P_2) denoted by "•" are z values of the points B_1 and B_2 from the given triangulation (see figure 2).

2. The values z of the vertices denoted by "•" which lie on the boundary of the control net can be computed from condition, that these vertices lie in the tanget plane given by the point B_1 or B_2 and by normal at this point (see figure 2).

3. The values z of vertices denoted by "•" which lie on the lines from the center of triangle to its vertices are determined by three already computed vertices "•" because all they lie on the same plane.

4. The values z of three vertices denoted by "▲" can be determined from the estimated crossboundary derivative at the midpoint of each of three edges of the triangle $B_1B_2B_3$.

5. The values z of three vertices denoted by "○" can be computed from condition that they lie in the plane determined by two already calculated vertices "▲" and by one inner vertex "•" (because two adjacent microtriangles with vertices "▲, •, ○" have to be coplanar (see figure 2).

6. The last Bèzier ordinate "□", which is placed above the center of the triangle $P_1P_2P_3$, lies in the plane determined by three vertices "○" because the three "center" triangles must be coplanar.

Once we have computed all coordinates of the ordinates $\boldsymbol{b}_{ijk}$, we can use formula 3.2.1.1 to evaluate points of the Bèzier triangle over the actual minitriangle (we need three such patches to "cover" the triangle $B_1B_2B_3$). Analogously, we calculate patches of each other triangles in the given triangulation and thus we obtain C_1 continuous interpolated surface.

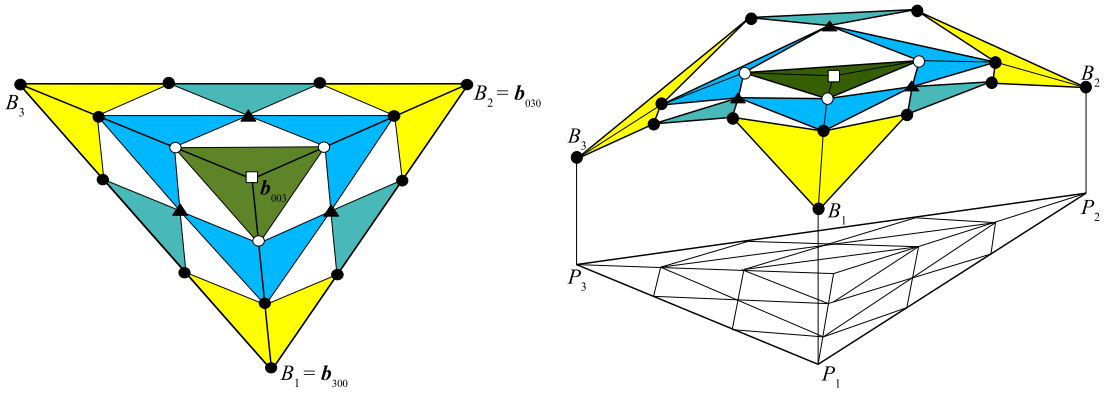

Figure 2: Construction of Bèzier ordinates over the three minitriangles

4 Comparison of the Methods

The "accuracy" of here described methods (in the term of the picture deformation recovery) was evaluated basically on the picture with black and white grid and its three deformations (see figure 4).

All deformed pictures was recovered by here presented methods. For deformation recovery in the first picture we had used 13, in the second one 37 and in the third 86 pairs of the corresponding points. The number of the points was chosen according to the type of the particular deformations.

The accuracy of the particular methods was evaluated by the *cross-correlation* coefficient:

$$CC = \left| \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}y_{ij} - mn\bar{x}\bar{y}}{\sqrt{(\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^2 - mn\bar{x}^2)(\sum_{i=1}^{m}\sum_{j=1}^{n} y_{ij}^2 - mn\bar{y}^2)}} \right|$$

where $\bar{x}, \bar{y}$ are averages:

$$\bar{x} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \text{ and } \bar{y} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} y_{ij},$$

where x_{ij} denotes pixels of the input picture and y_{ij} denotes pixels of the compared picture.

Figure 3: Comparing of the methods for image deformation recovery

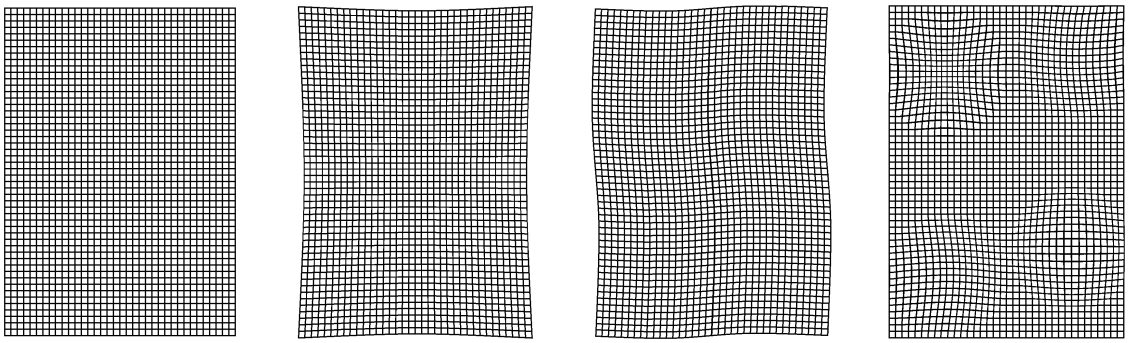

Figure 4: Original picture and its three deformations (from the left: original, shrinked, warped, locally deformed picture)

5 Conclusion

In this paper, some transformation methods was explored and their accuracies in the image deformation recovery was compared. From the graph on fig. 3 we can see that the thin plate spline method gives the best results, because its accuracy (measured by cross-correlation coefficient) has the almost highest values. We can also deduce that the thin spline method is the most suitable for the picture deformation recovery among all methods considered here.

References

[1] I. Amidror, *Scattered data interpolation methods for electronic imaging systems*, Journal of Electronic Imaging, 2 (2002), pp. 157–176.

[2] D. Fogel and L. Tinney, *Image registration using multiquadric functions, the finite element method, bivariate mapping polynomials and thin plate spline*, tech. rep., National Center for Geographic Information and Analysis, 1996.

[3] R. Franke and G. Nielson, *Smooth interpolation of large sets of scattered data*, International Journal for Numerical Methods in Engineering, 15 (1980), pp. 1691–1704.

[4] J. Hoschek and D. Lasser, *Fundamentals of Computer Aided Geometric Design*, A K Peters, Wellesley, MA, 1993, pp. 388–421.

[5] P. Mrazek, *Nonlinear Diffusion for Image Filtering and Monotonicity Enhancement*, PhD thesis, Faculty of Electrical Engineering, Czech Technical University, 2001.

[6] C. Tomasi and R. Manduchi, *Bilateral filtering for gray and color images*, in Proceedings of the 1998 IEEE International Conference on Computer Vision, 1998.

Róbert Bohdal (RNDr.) is a graduate student of geometry and topology at the Faculty of Mathematics, Physics and Informatics of the Comenius University, Bratislava. Lecturer in Bratislava at the Department of Algebra, Geometry and Didactics of Mathematics of the Faculty of Mathematics, Physics and Informatics of the Comenius University. His supervisor is doc. RNDr. Miloš Božek, CSc.