

Comparison of Methods for Picture Deformation Recovery¹

Róbert Bohdal

Abstrakt

V článku uvidíme rozličné metódy pre odstránenie deformácie obrazov. Tieto metódy sú príbuzné bežne známym metódam *registrácie obrazov* používajúcich riadiace body, ktoré určujú ako sa jednotlivé časti obrazu budú transformovať. Navrhujeme novú metódu, ktorú nazveme *hladkou trojuholníkovou metódou*. Uvedené metódy potom porovnáme z hľadiska presnosti s akou odstraňujú deformácie obrazov. Nakoniec určíme najpresnejšiu z opísaných metód. Nami navrhnutá metóda je dostatočne presná a v porovnaní s ostatnými metódami aj veľmi rýchla. Presnosť jednotlivých metód bude porovnaná pomocou novonavrhnutej metodológie.

Kľúčové slová: metódy registrácie obrazov, triangulácia, tenkostenný splajn, Shepardova metóda, Cloughova-Tocherova metóda.

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. We propose such a new method, which we call *smooth triangle method*. All methods are explored and their accuracy in the picture deformation recovery are compared. Next, we determine the most accurate method among them. Our proposed *smooth triangle method* is very fast and sufficiently accurate comparing to others. Accuracy of various methods will be compared upon the new proposed methodology.

Key words: image registration, triangulation, thin plate spline, Shepard's method, Clough-Tocher method.

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. 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.

A picture of a map is then converted into vector form (by an automatic or manual vectorization tool) and consequently coordinates of all polygons vertices are computed. These polygons are used to compute land areas, which are often written into land databases.

Our paper describes some picture transformation methods. All of them can be divided into two classes – *one-segment* or *many-segment* transformation methods. However, we can also classify these methods as *global* or *local*. The one-segment methods are global methods

¹ This work was supported by VEGA grant No. 1/3024/06.

naturally, but we can make them local. Their transformation function $f(\mathbf{x})$ is expressed by one formula on the whole domain. Among them, *radial basis functions* are the most popular. The many-segment methods (often known as *finite element methods*) are all local and their transformation function consists of more than one polynomial function. All methods in this class are based on the triangulation of the given input control points. The components of the vector function $f(\mathbf{x})$ can be considered as two explicit surfaces interpolating two sets of 3-D points. Well known methods in this section are *Clough-Tocher* and *Powell-Sabin* methods.

The discussed topics can be found (in slovak language) in more details in [3].

1. Transformation Methods

Suppose, we are given two sets of n points $\mathcal{P}, \mathcal{V} \in \mathbb{E}^2$, $\mathcal{P} = \{\mathbf{p}_i[x_i, y_i] \in \mathbb{E}^2; i = 1, \dots, n\}$ and $\mathcal{V} = \{\mathbf{v}_i[x'_i, y'_i] \in \mathbb{E}^2; i = 1, \dots, n\}$. The set \mathcal{P} consists of identification points on a deformed picture and the set \mathcal{V} comprises of points on an ideal undeformed picture². We call pairs $(\mathbf{p}_i, \mathbf{v}_i)$ as corresponding points. Then we seek a transformation function $f: \mathbb{E}^2 \rightarrow \mathbb{E}^2$ such that $f(\mathbf{p}_i) = \mathbf{v}_i$, where $i = 1, 2, \dots, n$. Using the function $f(\mathbf{x})$, we will transform all points of the input picture.

1.1. One-segment Transformation Methods

Methods which belong to this class are such methods whose transformation functions are given by one formula on the whole domain. The main 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. By some modification of these methods, we can obtain transformation functions with local character. In this section, we describe the *thin plate spline method* and *Shepard's method*.

1.1.1. 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 $f(\mathbf{x})$ is shown below [7]:

$$f(\mathbf{x}) = \sum_{i=1}^n \lambda_i R_{d,k}(\|\mathbf{x} - \mathbf{p}_i\|) + \sum_{|\alpha| < k} \mathbf{c}_\alpha \mathbf{x}^\alpha, \quad (1)$$

where $\mathbf{x}^\alpha = x_1^{\alpha_1} \dots x_d^{\alpha_d}$, $(\alpha_1, \dots, \alpha_d) \in \mathbb{N}^d$, $k \in \mathbb{N}$.

$R_{d,k}(r)$ is a class of the radial basic function, known as *polyharmonic spline*:

$$R_{d,k}(r) = \begin{cases} r^{2k-d}, & \text{if } d \text{ is odd} \\ r^{2k-d} \log(r), & \text{if } d \text{ is even} \end{cases} \quad \text{for } 2k > d.$$

In the case of thin plate spline $d = 2$, $k = 2$ and from this follow $R_{2,2}(r) = r^2 \log(r)$. If we need higher smoothness degree, then we choose $k > 2$.

If the dimension $d = 2$ and $k = 2$ we can rewrite formula (1) as:

² The position of these points can be computed from their world (geodetic) coordinates or can be determined straightforward if they are points of a rectangular grid.

$$f(x, y) = c_1 + c_2x + c_3y + \frac{1}{2} \sum_{i=1}^n \lambda_i r_i^2 \log(r_i^2), \text{ where } [x, y] \in \mathbb{E}^2, \quad (2)$$

$r_i^2 = (x - x_i)^2 + (y - y_i)^2$ and c_1, c_2, c_3, λ_i are unknown quantities. The parameters $\lambda_i, i = 1, \dots, n$ have to satisfy the boundary conditions:

$$\sum_{i=1}^n \lambda_i = \mathbf{0} \quad \text{and} \quad \sum_{i=1}^n \lambda_i p_i = \mathbf{0}. \quad (3)$$

Applying interpolation conditions $f(p_i) = v_i$, where $i = 1, 2, \dots, n$ together with boundary conditions (3) we can compute the unknown values via the next system of equations:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & \dots & 1 \\ 0 & 0 & 0 & x_1 & x_2 & \dots & x_n \\ 0 & 0 & 0 & y_1 & y_2 & \dots & y_n \\ 1 & x_1 & y_1 & 0 & r_{21}^2 \log(r_{21}^2) & \dots & r_{n1}^2 \log(r_{n1}^2) \\ 1 & x_2 & y_2 & r_{12}^2 \log(r_{12}^2) & 0 & \dots & r_{n2}^2 \log(r_{n2}^2) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & y_n & r_{1n}^2 \log(r_{1n}^2) & r_{2n}^2 \log(r_{2n}^2) & \dots & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ \lambda_1 / 2 \\ \lambda_2 / 2 \\ \vdots \\ \lambda_n / 2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}, \quad (4)$$

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

1.1.2. Shepard's Method

Perhaps the best approach of solving the *scattered data interpolation* problem is the Shepard's method, see [8].

Analogously to the previous section, we are looking for a transformation function $f(x)$, which satisfies the conditions $f(p_i) = v_i$, where $i = 1, 2, \dots, n$. Shepard defined his interpolating function $f(x)$ to be weighted mean of the coordinates v_i [6]:

$$f(x) = \sum_{i=1}^n \omega_i(x) v_i. \quad (5)$$

Weight functions $\omega_i(x)$ from formula (5) can be expressed:

$$\omega_i(x) = \frac{\sigma_i(x)}{\sum_{j=1}^n \sigma_j(x)}, \quad (6)$$

where $\sigma_i(x) = \|x - p_i\|^{-\mu_i}$, 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 $\omega_i(x)$ by *mollifying function* $\lambda_i(x) \in C(\mathbb{R}^d)$. As an example of such a mollifying function is the *Franke-Little weight* [6]:

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

Franke and Nielson proposed $R_i = 1/2D(N_w/n)^{1/2}$, where D is the maximum distance

between arbitrary two points of the set \mathcal{P} and N_w is a fixed integer (usually $N_w = 19$; heuristic obtained value).

Franke and Nielson in [4] generalized the Shepard's method by using *local interpolants*. They proposed to replace \mathbf{v}_i by local interpolating functions $L_i(\mathbf{x})$ with interpolation property $L_i(\mathbf{p}_i) = \mathbf{v}_i$. We get:

$$\mathbf{f}(\mathbf{x}) = \sum_{i=1}^n \omega_i(\mathbf{x})L_i(\mathbf{x}). \quad (7)$$

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

Modified quadratic Shepard's method is one of such methods. Using formula (7) we get:

$$\mathbf{f}(x, y) = \sum_{i=1}^n \omega_i(x, y)\mathbf{Q}_i(x, y), \quad (8)$$

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

$$\begin{aligned} \mathbf{Q}_i(x, y) = & \mathbf{c}_{i,1}(x - x_i)^2 + \mathbf{c}_{i,2}(x - x_i)(y - y_i) + \\ & + \mathbf{c}_{i,3}(y - y_i)^2 + \mathbf{c}_{i,4}(x - x_i) + \mathbf{c}_{i,5}(y - y_i) + \mathbf{v}_i. \end{aligned} \quad (9)$$

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

$$\sum_{k=1, k \neq i}^n \omega_k(x_i, y_i) [\mathbf{c}_{i,1}(x_k - x_i)^2 + \dots + \mathbf{c}_{i,5}(y_k - y_i) + \mathbf{f}_i - \mathbf{f}_k]^2 \rightarrow \min, \quad (10)$$

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

1.2. 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 whereas each region is transformed by its own transformation function. In all FEM methods, we need to ensure that all transformed points lie inside the convex hull of the set \mathcal{P} . We describe several such methods.

1.2.1. Simple Triangle Method

At first, we create a Delaunay triangulation of all input points of the set \mathcal{P} . Next, all points within each triangle of the input picture will be transformed using the corresponding *affine transformation*.

Then the transformation function can be evaluated by the following algorithm: For each picture point $X[x, y]$, we find a triangle ABC such that $X \in \Delta ABC$. If there exist more such triangles, we choose an arbitrary one. The coordinates of the new point $X'[x', y']$ are computed from equations of the affine transformation, which are given by conditions that points $A, B, C \in \mathcal{P}$ are transformed into points $A', B', C' \in \mathcal{V}$, respectively:

$$x' = a_1x + b_1y + c_1 \quad \text{and} \quad y' = a_2x + b_2y + c_2, \quad (11)$$

where

$$a_1 = \frac{1}{D} \begin{vmatrix} A_x' & B_x' & C_x' \\ A_y & B_y & C_y \\ 1 & 1 & 1 \end{vmatrix}, \quad b_1 = \frac{1}{D} \begin{vmatrix} A_x' & B_x' & C_x' \\ A_x & B_x & C_x \\ 1 & 1 & 1 \end{vmatrix}, \quad c_1 = \frac{1}{D} \begin{vmatrix} A_x' & B_x' & C_x' \\ A_x & B_x & C_x \\ A_y & B_y & C_y \end{vmatrix}$$

and

$$a_2 = \frac{1}{D} \begin{vmatrix} A_y' & B_y' & C_y' \\ A_y & B_y & C_y \\ 1 & 1 & 1 \end{vmatrix}, \quad b_2 = \frac{1}{D} \begin{vmatrix} A_y' & B_y' & C_y' \\ A_x & B_x & C_x \\ 1 & 1 & 1 \end{vmatrix}, \quad c_2 = \frac{1}{D} \begin{vmatrix} A_y' & B_y' & C_y' \\ A_x & B_x & C_x \\ A_y & B_y & C_y \end{vmatrix},$$

with
$$D = \begin{vmatrix} A_y & B_y & C_y \\ A_x & B_x & C_x \\ 1 & 1 & 1 \end{vmatrix}.$$

1.2.2. Smooth Triangle Method

We can obtain better results (from the aspect of “quality” picture deformation recovery), when we compute the coordinates of the new points as a weighted sum of the projections of the point X , where weights depend on the neighboring triangles.

Let T_i denotes triangle ABC and T_a, T_b, T_c denote its neighbors. As in the previous method, let for each triangle $T_i \subset \mathcal{P}$ be assigned a triangle $T_i' \subset v$. Let for each pair (T_i, T_i') be assigned the affine transformation f_i , such that the vertices of T_i are transformed to the vertices of T_i' . Let f_a, f_b, f_c denote the affine transformations associated to pairs $(T_a, T_a'), (T_b, T_b'), (T_c, T_c')$, respectively. Further, we denote $X_{f_i} = f_i(X)$.

The coordinates of the new point X' can be computed from formula:

$$X' = \omega_a X_{f_a} + \omega_b X_{f_b} + \omega_c X_{f_c} + \omega_i X_{f_i}, \quad (12)$$

where $\omega_a + \omega_b + \omega_c + \omega_i = 1$ and $\omega_a, \omega_b, \omega_c, \omega_i \geq 0$.

Clearly, the weights $\omega_a, \omega_b, \omega_c$ are attached to the neighbors of the triangle T_i . The weights $\omega_a = g(a)$, $\omega_b = g(b)$, $\omega_c = g(c)$ can be computed using the barycentric coordinates (a, b, c) of the point X by a function $g(x) \in C^1(\mathbb{R})$ from the conditions:

1. $g(x) \geq 0$, if $0 \leq x \leq 1$
2. $g(x)$ is decreasing on $0 \leq x \leq 1/3$
3. $g(0) = 1/2$
4. $g(x) = 0$ for $x \geq 1/3$
5. $g'(x)|_{x=1/3} = 0$.

The remaining weight ω_i can be computed from:

$$\omega_i = 1 - (\omega_a + \omega_b + \omega_c).$$

It is easy to prove that $\omega_i \geq 0$.

Any of weights ω_j belonging to the neighbor T_j of the triangle T_i is constructed so that the weight is equal to 1/2 for a point lying on their common side and is equal to 0 if the point is lying in the barycenter of the triangle (see Fig. 1).

Examples of functions, which satisfy the conditions described above are (see Fig. 2):

$$g(x) = \begin{cases} \frac{1}{2}(9x^2 - 6x + 1), & \text{if } x \in \left\langle 0, \frac{1}{3} \right\rangle \\ 0, & \text{if } x \in \left[\frac{1}{3}, 1 \right) \end{cases} \quad \text{or} \quad g(x) = \begin{cases} \frac{1}{4}(1 + \cos 3\pi x), & \text{if } x \in \left\langle 0, \frac{1}{3} \right\rangle \\ 0, & \text{if } x \in \left[\frac{1}{3}, 1 \right) \end{cases}$$

The main advantage of our triangle methods is their simple expression and also their low computational complexity. The second method provides C^1 continuous results which is commonly required.

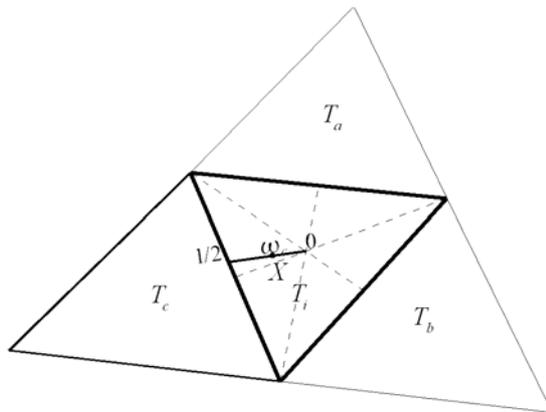


Fig. 1. The weight ω_c which corresponding to the triangle T_c is computed from the position of the point X in the triangle T_i

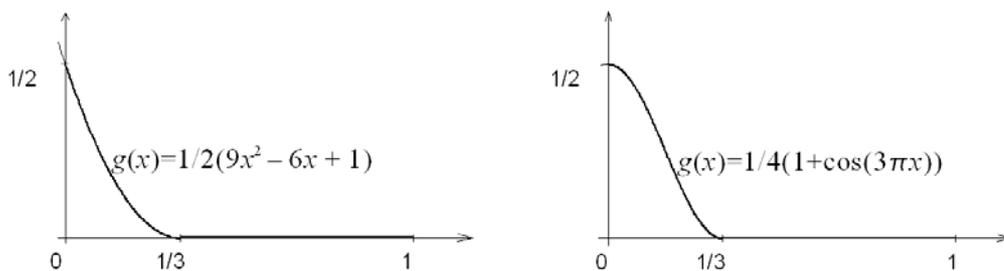


Fig. 1. Examples of the weight functions $g(x)$

1.2.3. Clough-Tocher Method

One of the main drawback of piecewise linear interpolation schemes is that they are only C^0 continuous in general. We often require C^1 continuity across the boundaries between triangles. C^1 continuity obviously requires piecewise interpolation schemes based on polynomials of higher order than 1.

In order to guarantee C^1 continuity across boundaries, we need know not only vertex coordinates $[x_i, y_i]$ and their value z_i , but also other information. This information includes

tangent plane (or gradient) at each vertex of the triangulation, as well as cross derivatives at the midpoint of each edge of the triangulation. However, the assumption of given gradients is not always realistic. In the most cases, they will have to be estimated from the given vertices. This can be done by considering the known values not only in the vertices of the actual triangle, but also in its neighborhoods. Some methods for the gradient estimation we can find in [9] and [1].

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 (usually taken as the barycenter of the triangle).

The final function $f(\mathbf{x})$ will be C^1 continuous surface which consists of *cubic Bézier triangles* over all minitriangles. Clough-Tocher method uses cubic Bézier triangles in the form:

$$\begin{aligned} X(u, v, w) = & \mathbf{b}_{300}u^3 + 3\mathbf{b}_{210}u^2v + 3\mathbf{b}_{120}uv^2 + \\ & + \mathbf{b}_{030}v^3 + 3\mathbf{b}_{021}v^2w + 3\mathbf{b}_{012}vw^2 + \\ & + \mathbf{b}_{003}w^3 + 3\mathbf{b}_{102}w^2u + 3\mathbf{b}_{201}wu^2 + 6\mathbf{b}_{111}uvw \end{aligned} \quad (13)$$

Computation of the Bézier Ordinates \mathbf{b}_{ijk}

The Bézier ordinates of the control net of three adjacent triangle patches can be evaluated by the next algorithm:

The coordinates $[x, y]$ of the ten Bézier ordinates within each minitriangle are fully determined by the control net. They are located either at the minitriangle vertices, or the 1/3 or 2/3 length of each edge, or at the barycenter of the minitriangle.

The values z of these Bézier ordinates are determined by the next steps:

1. The values z of the Bézier 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 Fig. 3):

$$\mathbf{b}_{300} = B_1 \quad \text{a} \quad \mathbf{b}_{030} = B_2.$$

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 tangent plane given by the point B_1 or B_2 and by normal at this point (see Fig. 3):

$$\mathbf{b}_{210} = \mathbf{b}_{300} + 1/3D_{\mathbf{e}_1}X(1, 0, 0)$$

$$\mathbf{b}_{120} = \mathbf{b}_{030} + 1/3D_{-\mathbf{e}_1}X(0, 1, 0),$$

where $D_{\mathbf{e}_1}X(\mathbf{u})$ denotes directional derivative³ in the direction \mathbf{e}_1 at the point \mathbf{u} (see Fig. 4).

3. The values z of vertices denoted by „●“ which lie on the lines from the centre of triangle to its vertices are determined by three already computed vertices „●“ because they all lie on the same plane:

$$\mathbf{b}_{201} = 1/3(\mathbf{b}_{300} + \mathbf{b}_{210} + \mathbf{b}'_{120}).$$

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 $B_1B_2B_3$. These vertices lie in the plane which is determined by the point S of the Bézier curve (for the parameter $t = 1/2$) given by the control points \mathbf{b}_{300} , \mathbf{b}_{210} , \mathbf{b}_{120} ,

³ The directional derivative can be estimated from the direction of the normal vector at given point.

\mathbf{b}_{030} and by the normal vector computed from the linear combination of the normals at the triangle vertices (see Fig. 3):

$$\mathbf{b}_{111} = 2D_a \mathbf{X}(1/2, 1/2, 0) - 1/2(\xi_1 + \xi_2) + 1/2(\mathbf{b}_{210} + \mathbf{b}_{120}),$$

where $\xi_1 = \mathbf{b}_{201} - 1/2(\mathbf{b}_{300} + \mathbf{b}_{210})$ and $\xi_2 = \mathbf{b}_{021} - 1/2(\mathbf{b}_{120} + \mathbf{b}_{030})$.

- 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 Fig. 3):

$$\mathbf{b}_{102} = 1/3(\mathbf{b}_{201} + \mathbf{b}_{111} + \mathbf{b}'_{111}).$$

- The last Bézier 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:

$$\mathbf{b}_{003} = 1/3(\mathbf{b}_{102} + \mathbf{b}_{012} + \mathbf{b}'_{102}).$$

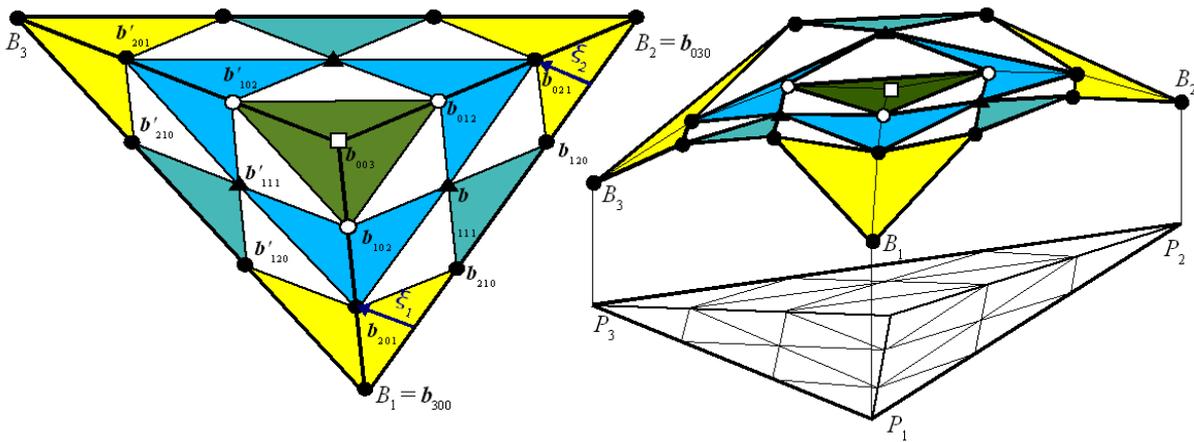


Fig. 2. Construction of the Bézier ordinates over the three minitriangles (left - top view)

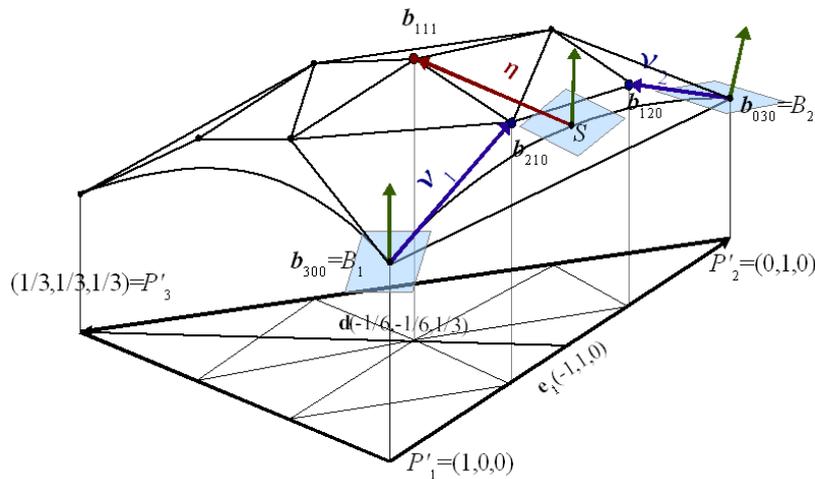


Fig. 3. Normals and crossboundary derivatives determine the position of the Bézier ordinates \mathbf{b}_{210} , \mathbf{b}_{120} , \mathbf{b}_{111}

Once we have computed all coordinates of the ordinates b_{ijk} , we can use formula (13) to evaluate points of the Bézier 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.

2. 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 Fig. 5). The first deformed picture was created by shrinking the middles of the grid boundary towards its center, the second one was deformed by transformation which warped its boundary and the last one was deformed by four local deformations (by translating and scaling).

All deformed pictures were 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 (see Fig. 6). The number of the points was chosen according to the type of the particular deformations.

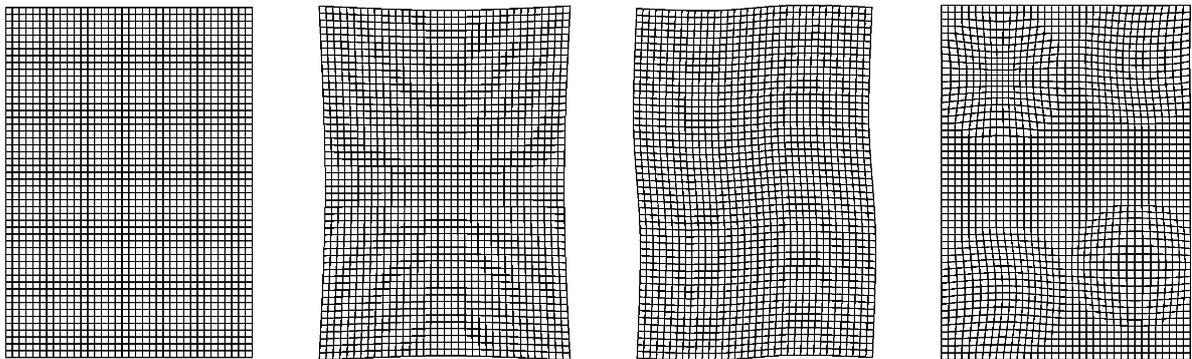


Fig. 4. Original picture and its three deformations
(from the left: original, shrunken, warped, locally deformed picture)

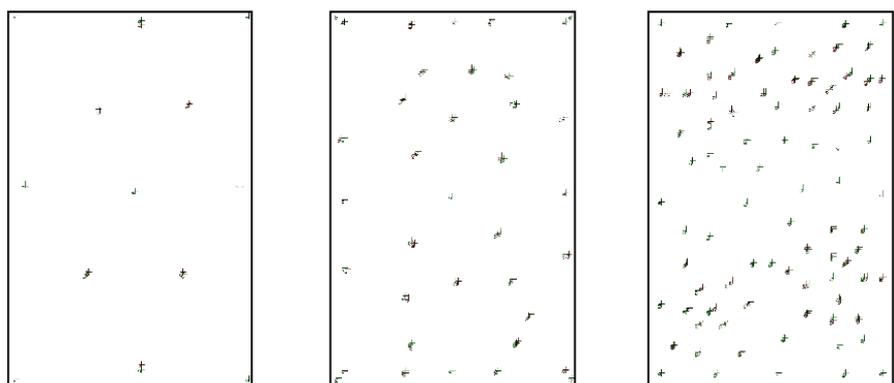


Fig. 5. Corresponding points for the image deformation recovery
of the three deformed pictures

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

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

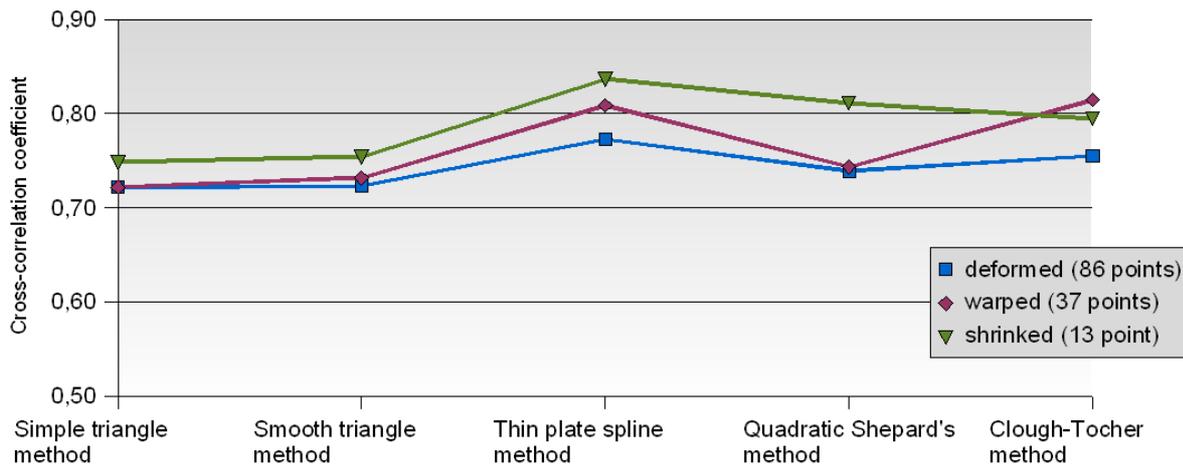
where \bar{x} , \bar{y} are averages:

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

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

	deformed (86 points)	warped (37 points)	shrunked (13 points)
Simple triangle method	0,72158	0,72185	0,74822
Smooth triangle method	0,72318	0,73162	0,75398
Thin plate spline method	0,77261	0,80886	0,83655
Quadratic Shepard's method	0,73866	0,74335	0,81108
Clough-Tocher method	0,75509	0,81454	0,79438

Table 1. Comparison of the methods for image deformation recovery of the shrunked, warped and locally deformed picture



Graph 1. Comparison of the methods for image deformation recovery of the shrunked, warped and locally deformed picture

3. Conclusion

In this paper, some transformation methods were explored and their accuracies in the image deformation recovery were compared. From the table and graph above 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.

We have also proposed new transformation method. Our simple triangle method has satisfactory accuracy and its computational simplicity can be very useful, especially in real-time computing.

Transformation functions play a major role in image registration. Examples of the transformation functions here presented methods on the sample points evaluated from the standard Franke test function $F_1(x, y)$ [5] are shown on the Fig. 6.

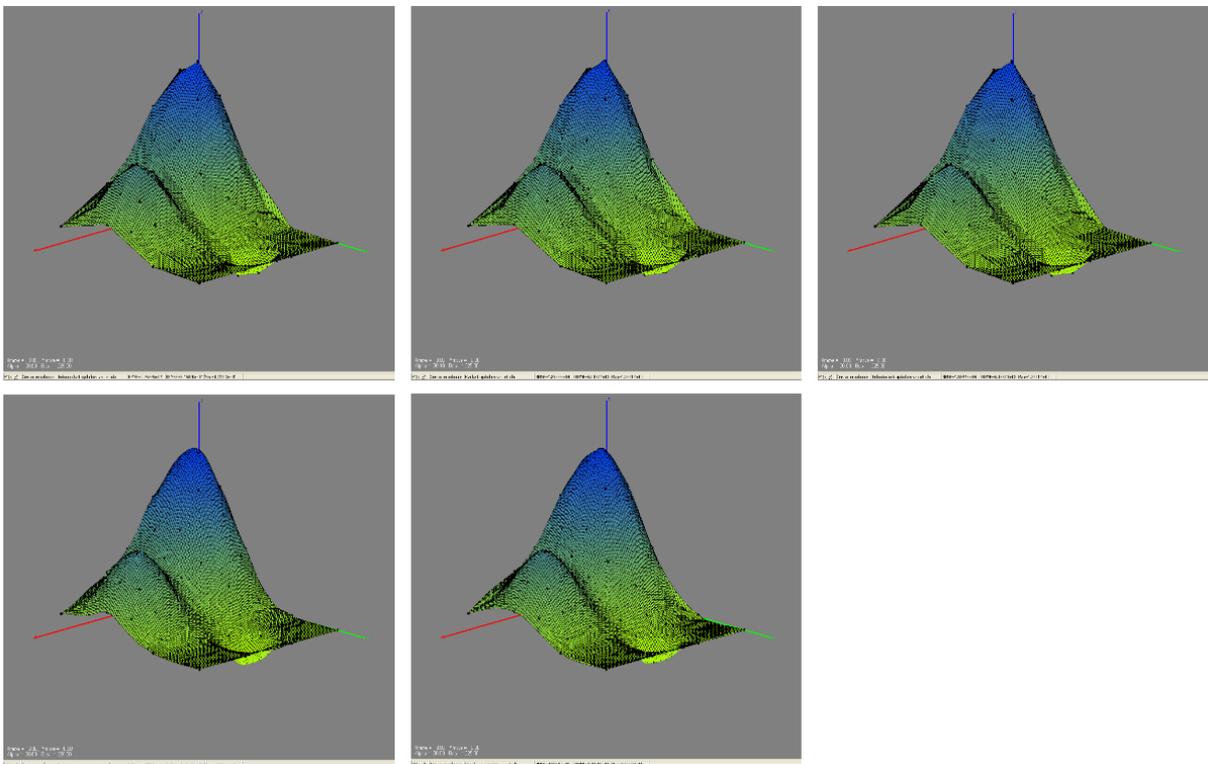


Fig. 6: Examples of the transformation functions (simple triangle method, smooth triangle method, thin plate spline method, quadratic Shepard's method, Clough-Tocher method).

References

- [1] AKIMA, H. On Estimating partial derivatives for bivariate interpolation of scattered data. In *Rocky Mountain Journal of Mathematics*, 1984, vol. 1, no. 14, p. 41–52.
- [2] AMIDROR, I. Scattered data interpolation methods for electronic imaging systems. In *Journal of Electronic Imaging*, 2002, vol. 2, no. 11, p. 157–176.
- [3] BOHDAL, R. *Odstraňovanie tvarových deformácií obrazov*. Dizertačná práca, FMFI UK, Bratislava 2006.

- [4] FRANKE, R., NIELSON, G. Smooth interpolation of large sets of scattered data. In *International Journal for Numerical Methods in Engineering*, 1980, vol. 15, no. 11, p. 1691–1704.
- [5] FRANKE, R. Scattered data interpolation: Test of some methods. In *Mathematics of Computation*, 1982, vol. 38, no. 157, p. 181–200.
- [6] HOSCHEK, J., LASSER, D. *Fundamentals of Computer Aided Geometric Design*. A K Peters, Wellesley, MA, 1993.
- [7] ISKE, A. Radial basis functions: basics, advanced topics and meshfree methods for Transport Problem. In *Seminar of Mathematics*, 2003, p. 247–274.
- [8] SHEPARD, D. A two dimensional interpolation function for irregular spaced data. In *Proceedings 23rd ACM National Conference*, 1968, p. 517–524.
- [9] STEAD, S. Estimation of gradients from scattered data. In *Rocky Mountain Journal of Mathematics*, 1984, vol. 1, no. 14, p. 265–279.

RNDr. Róbert Bohdal

Department of Algebra, Geometry
and Didactics of Mathematics
Faculty of Mathematics, Physics and Informatics
Comenius University in Bratislava
Mlynská dolina, 842 48 Bratislava, SR
e-mail: bohdal@fmph.uniba.sk