KYBERNETIKA — VOLUME 41 (2005), NUMBER 1, PAGES 97 – 112

FAST EVALUATION OF THIN–PLATE SPLINES ON FINE SQUARE GRIDS

Petr Luner and Jan Flusser

The paper deals with effective calculation of Thin-Plate Splines (TPS). We present a new modification of hierarchical approximation scheme. Unlike 2-D schemes published earlier, we propose an 1-D approximation. The new method yields lower computing complexity while it preserves the approximation accuracy.

Keywords: Thin-Plate Spline, fast evaluation, subtabulation

AMS Subject Classification: 65D07, 65D18

1. INTRODUCTION

Interpolation and/or approximation of 2-D data measured on an irregular grid is one of traditional problems of numerical mathematics and statistics that has been investigated for many years. It arises in many application areas such as in mechanics, in surface approximation and reconstruction, in experimental data smoothing, and in image processing, to name a few.

Formally, the interpolation problem is formulated as follows. Let a set of arbitrary points (x_i, y_i), $i = 1, \dots, N$ be given in a coordinate plane. Let z_i be a data value assigned to point (x_i, y_i). The task is to find a function s such that $s(x_i, y_i) = z_i$ for $i = 1, \dots, N$.

This is an ill-posed problem having infinite number of solutions. To make the formulation meaningful, a regularization constraint must be loaded on function s. The choice of the regularization constraint also specifies the class of functions among which the resulting interpolant should be selected. There are many different possibilities how to define the regularization constraint. An intuitive approach was motivated by a physical model of elastic deformations. Let us imagine an ideal 2-D thin metal plate of infinite extent, which is fixed in 3-D space at the points (x_i, y_i, z_i). Such a plate deforms into the form which minimizes its potential energy, that means which minimizes so-called quadratic variation functional

$$V(s) = \int\int_{R^2} \left[\left(\frac{\partial^2 s}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 s}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 s}{\partial y^2} \right)^2 \right] \mathrm{d}x\, \mathrm{d}y \tag{1}$$

on Sobolev space W_2^2 of all eligible functions. Minimizing functional (1) means in fact looking for the "smoothest" interpolant. Provided that the set of given coordinate points includes at least three non-collinear points, this variational problem has a unique solution [11], which is called *Thin-Plate Spline* (TPS) and can be expressed in the explicit form

$$s(x,y) = ax + by + c + \sum_{i=1}^{N} \lambda_i \phi(x - x_i, y - y_i), \tag{2}$$

where $\phi(x,y)$ denotes the radial function

$$\phi(x,y) = (x^2 + y^2) \log \sqrt{x^2 + y^2}. \tag{3}$$

Parameters $a, b, c, \lambda_1, \dots, \lambda_N$ are determined by solving the system of $(N+3)$ linear equations

$$\begin{aligned} s(x_i, y_i) &= z_i, \quad i = 1, \dots, N, \\ \sum_{j=1}^{N} \lambda_j &= 0, \\ \sum_{j=1}^{N} \lambda_j x_j &= 0, \\ \sum_{j=1}^{N} \lambda_j y_j &= 0. \end{aligned} \tag{4}$$

The last three equations ensure consistent behavior of s in infinity.

The motivation given by the metal plate deformation can be used as well in the approximation problem. The only difference is that we relax the interpolation constraints – the imaginary plate is no longer fixed at the given data points but it is attached by elastic spirals. The functional to be minimized then consists of two terms – elastic term (1) and an error term which equals weighted mean square error. Minimization of this new functional leads to so-called *Smoothing TPS* (STPS), the form of which is exactly the same as in (2). The only difference is in coefficient calculation.

Both the TPS and the STPS have been proven to be one of the best tools for solving 2-D interpolation/approximation problems in statistical data analysis and engineering calculations. They were originally introduced by Harder [12] as a tool for wing surface interpolation in aeroelastic calculations. Basic theoretical studies on the TPS were published by Duchon [7] and Wahba [17].

The TPS have found numerous applications also in image analysis, computer graphics, and computer vision. They were used for shape recovery from stereo-images, particularly for interpolation/approximation of depth maps [11], for interpolating incomplete surfaces [6], for image sharpening and scaling [2], and for noise suppression via image smoothing [4, 13]. Probably the largest group of TPS-related image processing papers has been devoted to spatial transformations of images. Spatial transformations are, among others, the key stage of image-to-image alignment, spatial image normalization, image warping, and morphing. The TPS are used as mapping functions between two different coordinate systems. This idea was theoretically proposed by Bookstein [5] and then further developed by other authors. Successful application to face images [1], to satellite [9] and aerial images [8], and to medical images [16] have been reported. Wolberg [18] used the TPS as a general image warping model which can be employed in computer animation.

One drawback of the TPS/STPS is that they are computationally expensive to evaluate. Usually, the spline is required to be calculated at all mesh points of a fine square grid. While the number of the interpolation nodes N may be from ten to several hundreds, the number of grid points is typically from hundred thousand to million. The complexity of the TPS coefficients calculation is $O(N^3)$ which is negligible comparing to the evaluation part. The evaluation of the spline at a single grid point requires N calculations of logarithm and $O(N)$ additions/multiplications. If the number of grid points is large, the direct spline evaluation using formula (2) becomes extremely expensive.

In the last decade, considerable attention has been paid to the methods decreasing the complexity of the TPS evaluation while preserving reasonable accuracy. Flusser [8] proposed an adaptive approximation of the TPS on square or triangular regions by simpler functions. Beatson and Newsam [3] adapted an earlier method by Greengard [10] for multipole expansion of radial functions. They proposed to calculate the TPS exactly in a certain neighbourhood of the given points (x_i, y_i). Outside this neighborhood, they expanded each term of the TPS into power series, collected the contribution from all poles into one new power series and evaluated only several terms of this expansion. Powell [15, 14] used the idea of TPS subtabulation. Although the Powell's method is considered to be the fastest algorithm ever published, it still requires considerable computing time when the number of interpolation points N is large. In this paper, we present a new method which is motivated by the Powell's algorithm but performs much faster without loss of accuracy.

The rest of the paper is organized as follows. In Section 2 we recall the original Powell's method. Section 3 describes the new algorithm and discusses its accuracy. In Section 4, experimental comparisons with the Powell's method as well as with the direct evaluation are presented.

2. RECALLING POWELL

To describe the original method for the TPS tabulation developed by Powell [15, 14], let us assume that spline $s(x, y)$ given by formula (2) is required to be calculated on a square grid consisting of points (k, ℓ), $k, \ell = 0, \dots, M$.

The algorithm can be outlined as follows: First, exact spline values are calculated on an initial coarse grid consisting of points $(kh, \ell h)$; $k, \ell = 0, \dots, M/h$ where $h = 2^t$ for some suitable integer t. Next, on the two times finer grid $(kh/2, \ell h/2)$, $k, \ell = 0, \dots, 2M/h$, the missing spline values are approximated by applying carefully chosen subtabulation schemes. The linear combination of the 4×4 neighboring coarse

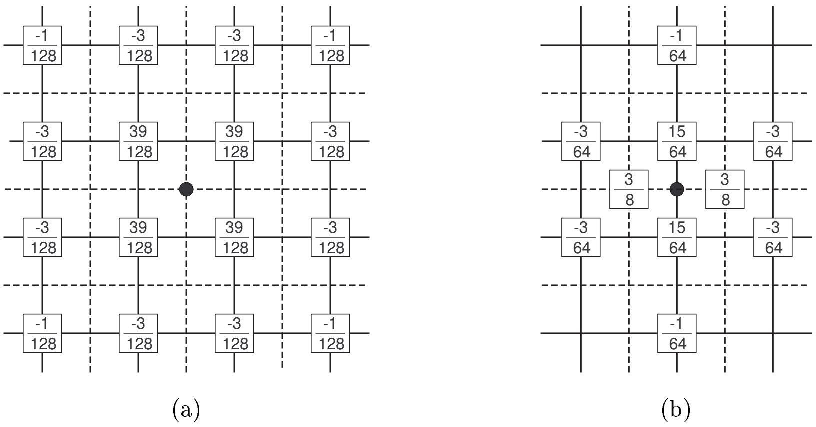

Fig. 1. Subtabulation schemes used in the Powell's algorithm.

grid values shown in Figure 1a is used to approximate $s(kh/2, \ell h/2)$ when k and ℓ are odd integers. The scheme shown in Figure 1b and its rotation by $\pi/2$ are then applied to provide estimates of $s(kh/2, \ell h/2)$ when $k + \ell$ is odd. Accuracy of these approximations depends on the distance between the point $(kh/2, \ell h/2)$ and the interpolation points (x_i, y_i), $i = 1, \ldots, N$. More specifically, Powell has shown that the subtabulation schemes provide sufficient accuracy whenever the point $(kh/2, \ell h/2)$ satisfies the conditions

$$\max[|kh/2 - x_i|, |\ell h/2 - y_i|] \geq \varrho h/2, \quad i = 1, \ldots, N, \tag{5}$$

where ϱ is a constant that depends on the required precision. Otherwise some modifications to the calculation of $s(kh/2, \ell h/2)$ are necessary to provide sufficient accuracy. The subtabulation process is then iterated to calculate the spline on ever finer grids until the final grid is reached.

To state the algorithm more precisely, we need to introduce some necessary definitions. Let us assume that the constant ϱ from conditions (5) is given. We let $N_i(h)$ be the neighbourhood

$$N_i(h) = \{(x, y) : \max[|x - x_i|, |y - y_i|] < \varrho h\} \subset \mathbb{R}^2 \tag{6}$$

of (x_i, y_i), we let $J_h(x, y)$ be the set

$$J_h(x, y) = \{i : (x, y) \notin N_i(h)\}, \tag{7}$$

and we let $s_h(x, y)$ be the function

$$s_h(x, y) = ax + by + c + \sum_{i \in J_h(x,y)} \lambda_i \phi(x - x_i, y - y_i). \tag{8}$$

In other words, $s_h(x, y)$ is the same spline as (2) except the terms expressing the contributions to $s(x, y)$ from the interpolation points whose maximum-norm distance to (x, y) is less than ϱh. Using this notation, the algorithm can be written as follows:

Step 1: Set the initial value of h to 2^t for an integer t chosen in such a way that the number of points of the initial grid $(kh, \ell h)$, $k, \ell = 0, \ldots, M/h$ is between about 20×20 and 40×40. Calculate the function $s_{h/2}$ on the initial grid using formula (8).

Step 2: Repeat t times the following steps:

(a) Set $h := h/2$.

(b) Approximate the missing values of s_h on the grid $(kh, \ell h)$, $k, \ell = 0, \ldots, M/h$ by applying the subtabulation schemes in Figure 1.

(c) Near the boundaries of the neighborhoods $N_i(h)$, $i = 1, \ldots, N$ where the usage of the subtabulation schemes is prevented by the discontinuities of the function s_h, calculate the values of s_h directly from formula (8).

(d) Replace the obtained values of s_h by the values $s_{h/2}(kh, \ell h)$, $k, \ell = 0, \ldots, M/h$. Do this by a loop through the interpolation points (x_i, y_i), $i = 1, \ldots, N$ that adds the term $\lambda_i \phi(kh - x_i, \ell h - y_i)$ to $s_h(kh, \ell h)$ for every grid point $(kh, \ell h) \in N_i(h) \setminus N_i(h/2)$ (where "$\setminus$" denotes the set difference).

Step 3: Complete the calculation by replacing the obtained values of $s_{1/2}$ on the final grid by the values of s. For this purpose, employ a loop through the neighborhoods $N_i(1/2)$, $i = 1, \ldots, N$ that adds the term $\lambda_i \phi(k - x_i, \ell - y_i)$ to $s_{1/2}(k, \ell)$ for every grid point (k, ℓ) that is in $N_i(1/2)$.

The resultant accuracy of the computed spline values depends on selection of the constant ϱ. By taking advantage of the smoothness of the TPS, Powell showed that when the spline is required to be calculated to a relative accuracy ε, it is suitable to set ϱ to the least integer such that $\varrho \geq (6/\varepsilon)^{1/6}$. For example, in the cases $\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-9}$, the recommended choices are $\varrho = 14$ and $\varrho = 43$, respectively.

The time complexities of Steps 1, 2c, 2d and 3 are $O(N)$, $O(\varrho N^2)$, $O(\varrho^2 N)$ and $O(\varrho^2 N)$, respectively. Further, since the subtabulation schemes are applied at each grid point at most once, the total work of all t iterations of Step 2b is of magnitude $O(M^2)$. It follows that the resultant time complexity of the Powell's algorithm is $O(M^2 + t\varrho N^2 + t\varrho^2 N)$. Especially when N is small in comparison to M, this yields a significant speedup over the direct use of formula (2), because the number of operations of the direct approach is of magnitude $O(M^2 N)$.

3. THE NEW METHOD

It was proven that the Powell's algorithm can provide very substantial speed-up over the direct spline evaluation [14, 15]. It is also evident, however, that the improvement rate gradually decreases as the algorithm is applied to splines with growing number of interpolation nodes N. The main cause of this slowdown is Step 2c. At every grid resolution level, $O(\varrho N^2)$ explicit evaluations of TPS terms are required because of the discontinuities of function s_h. When N is large, this part dominates the entire computation and prevents the practical usage of the algorithm. In this Section, we present a modification of the Powell's algorithm that we have developed to remove this "bottleneck".

The proposed algorithm preserves the basic structure of the original method – similarly to the Powell's algorithm, the pyramidal subtabulation of function s_h is used. The new algorithm however differs both in the subtabulation schemes used and in the way in which the discontinuities of the function s_h are treated. The major difference in comparison to the original method is that we enable the usage of the subtabulation schemes also at the points at which the approximation is calculated "across" the boundary of neighborhood $N_i(h)$. To make it possible, we propose a new 1-D approximation scheme instead of the Powell's 2-D schemes. As a result, we completely eliminate the need for any expensive explicit corrections mentioned above.

The new method is, similarly to the original Powell's method, based on hierarchical calculation from coarse to fine levels. The algorithm starts on a reasonably sparse grid on which the spline values are computed exactly from the definition. Having the spline values on a coarse grid, the values on a twice-finer grid are calculated by means of 1-D interpolation formulae. These formulae are‘ basically up-sampling convolution filters with properly chosen coefficients, the length of which (i. e. the order of the filter) is a user-defined parameter. The filters are successively applied both in x and y directions.

However, the spline values on those coarser grid points which are located near any initial interpolation node must be properly modified when they propagate to the finer level. This is necessary to keep the approximation accurate.

In the rest of this section, we first derive the approximation formulae, then we show how they are used to step from a coarse to a fine level and, finally, we formally describe the algorithm as a whole.

3.1. Subtabulation schemes

We start the detailed description of the proposed algorithm by introducing the new subtabulation schemes. In contrast to the two-dimensional subtabulation schemes used by Powell, our algorithm works with the following two one-dimensional schemes:

$$f(kh,\ell h) \approx \sum_{j=1-K}^{K} \alpha_j f(kh-h+2jh,\ell h), \tag{9}$$

$$f(kh,\ell h) \approx \sum_{j=1-K}^{K} \alpha_j f(kh,\ell h-h+2jh). \tag{10}$$

The user-defined parameter K determines how many coarse-grid points are required for the approximations at one finer-grid point. Jointly with the constant ϱ introduced in the previous section, it is used to control the accuracy of the calculated spline values. The only constraints for the selection of these parameters are $K \geq 2$ and $\varrho \geq 2K$.

Depending on the value of K, the subtabulation coefficients α_j, $j = 1-K, \ldots, K$

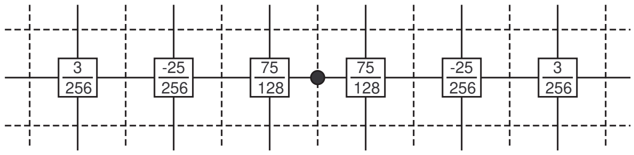

Fig. 2. Subtabulation coefficients for $K = 3$.

are defined by the following system of $2K$ linear equations:

$$\begin{aligned} &\alpha_{1-j} = \alpha_j, \quad j = 1, \dots, K, \\ &\sum_{j=1}^{K} \alpha_j = \frac{1}{2}, \\ &\sum_{j=1}^{K} \alpha_j (2j-1)^{2n} = 0, \quad n = 1, \dots, K-1. \end{aligned} \tag{11}$$

These equations ensure that formulae (9) and (10) are exact when f is a polynomial up to degree $2K-1$. For $K = 3$, the subtabulation coefficients are shown in Figure 2.

3.2. Subtabulation procedure

Having introduced the subtabulation schemes, we can proceed to the description of the subtabulation procedure which performs a single grid refinement. We consider the following model situation:

- The function s_h is required to be calculated on the grid of mesh size h consisting of points $(kh, \ell h)$, $k, \ell = 0, \dots, 2m$.
- The values of s_h are already available on the coarser grid of mesh size $2h$ consisting of points $(2kh, 2\ell h)$, $k, \ell = 1-K, \dots, m+K-1$.

The subtabulation process consists of two stages. First, scheme (9) is used to refine all grid lines that are parallel to the x-axis. This stage produces the values $s_h(kh, 2\ell h)$, $k = 0, \dots, 2m$, $\ell = 1-K, \dots, m+K-1$. Then, the grid refinement is completed by applying scheme (10) in the y-direction.

Let G be the set of all points on the coarser grid, i.e. $G = \{(2kh, 2\ell h) : k, \ell = 1-K, \dots, m+K-1\}$. For every interpolation point (x_i, y_i), $i = 1, \dots, N$, we denote A_i, B_i, C_i and D_i the following point sets:

$$\begin{aligned} A_i &= (x_i - \varrho h - 2Kh + h, x_i - \varrho h] \times (y_i - \varrho h, y_i + \varrho h) \cap G, \\ B_i &= (x_i - \varrho h, x_i - \varrho h + 2Kh - h] \times (y_i - \varrho h, y_i + \varrho h) \cap G, \\ C_i &= [x_i + \varrho h - 2Kh + h, x_i + \varrho h) \times (y_i - \varrho h, y_i + \varrho h) \cap G, \\ D_i &= [x_i + \varrho h, x_i + \varrho h + 2Kh - h) \times (y_i - \varrho h, y_i + \varrho h) \cap G. \end{aligned}$$

Schematically, these sets are shown in Figure 3 for the choice $K = 3$. Further, we use $AddTerm(i, P)$ (resp. $SubtractTerm(i, P)$) to denote the subroutine that adds

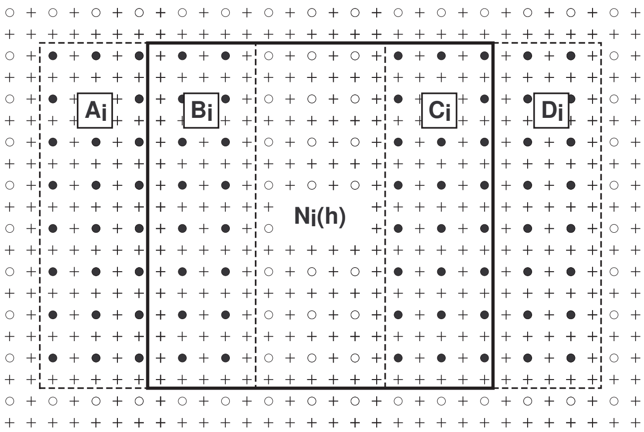

Fig. 3. Point sets A_i, B_i, C_i and D_i for $K = 3$.

(resp. subtracts) the term $\lambda_i \phi(x - x_i, y - y_i)$ to (resp. from) the current values at grid points specified by a set P.

Here are the steps used to refine the coarse grid in the x-direction:

Step 1: For each $i = 1, \ldots, N$ do
If $h \le x_i - \varrho h$ then $AddTerm(i, B_i)$
If $x_i - \varrho h < h < x_i + \varrho h$ then $SubtractTerm(i, A_i \cup D_i)$
If $h \ge x_i + \varrho h$ then $AddTerm(i, C_i)$

Step 2: Create the following lists:

$$\begin{aligned} L_k &= \{i : 2kh + h \le x_i - \varrho h < 2kh + 3h\}, \quad k = 0, \ldots, m-1 \\ R_k &= \{i : 2kh + h < x_i + \varrho h \le 2kh + 3h\}, \quad k = 0, \ldots, m-1 \end{aligned}$$

Step 3: For each $k = 0, \ldots, m-1$ do

For each $\ell = 1 - K, \ldots, m + K - 1$ do
Apply scheme (9) at point $(2kh + h, 2\ell h)$

For each i in L_k do $SubtractTerm(i, A_i \cup B_i \cup D_i)$
For each i in R_k do $AddTerm(i, A_i \cup C_i \cup D_i)$

end

Step 4: For each $i = 1, \ldots, N$ do
If $2m + h \le x_i - \varrho h$ then $SubtractTerm(i, B_i)$
If $x_i - \varrho h < 2m + h < x_i + \varrho h$ then $AddTerm(i, A_i \cup D_i)$
If $2m + h \ge x_i + \varrho h$ then $SubtractTerm(i, C_i)$

The operation of the above procedure can be logically divided into two parts: (i) subtabulation scheme application and (ii) dynamic modification of the coarse grid values near the boundaries of the neighborhoods $N_i(h)$, $i = 1, \dots, N$. It is not difficult to verify that, as a part of the main loop in Step 3, the procedure incrementally alters the coarse grid values in such a way that when scheme (9) is applied at point $(2kh + h, 2\ell h)$, the current values at points $(2kh - h + 2jh, 2\ell h)$, $j = 1 - K, \dots, K$ represent the values of the function

$$s_{h,k,\ell}(x, y) = ax + by + c + \sum_{i \in J_h(kh,\ell h)} \lambda_i \phi(x - x_i, y - y_i).$$

This ensures that Step 3 produces valid estimates of $s_h(2kh + h, 2lh)$ for all $k = 0, \dots, m - 1$ and $\ell = 1 - K, \dots, m + K - 1$. Step 4 is then necessary to restore the original values on the coarse grid (i.e. the values $s_h(2kh, 2\ell h)$, $k, \ell = 1 - K, \dots, m + K - 1$).

The time complexities of Steps 1, 2, 3 and 4 are $O(KN)$, $O(m + N)$, $O(Km^2 + K\varrho N)$ and $O(KN)$, respectively. It follows that the time complexity of the first stage of the subtabulation procedure is $O(Km^2 + K\varrho N)$. Since the second stage can be realized in a quite analogical way, this is also the resultant time complexity of the whole subtabulation procedure.

3.3. The algorithm

We are now ready to state the whole algorithm. As in Section 2, let us assume that our task is to calculate the spline $s(x, y)$ on the unit grid consisting of points (k, ℓ), $k, \ell = 0, \dots, M$. Further, let us denote M_h the least integer that satisfies the inequality $M_h h \geq M + 2h(K - 1)$. The algorithm consists of the following steps:

Step 1: Set the initial value of h to 2^t for integer t chosen in such a way that the number of points of the initial grid $(kh, \ell h)$, $k, \ell = 2 - 2K, \dots, M_h$ is between about 20×20 and 40×40. Calculate the function $s_{h/2}$ on the initial grid using formula (8).

Step 2: Repeat t times the following steps:

(a) Set $h := h/2$.

(b) Given the values $s_h(2kh, 2\ell h)$, $k, \ell = 2 - 2K, \dots, M_{2h}$, apply the subtabulation procedure described in Section 3.2 to generate the values $s_h(kh, \ell h)$, $k, \ell = 2 - 2K, \dots, M_h$.

(c) Replace the obtained values with the values $s_{h/2}(kh, \ell h)$, $k, \ell = 2 - 2K, \dots, Mh$.

Step 3: Replace the values of $s_{1/2}$ on the final grid with the values of s.

We already showed that Step 2b requires $O(KM_h^2 + K\varrho N)$ time. The time complexities of Steps 1, 2c and 3 are $O(N)$, $O(\varrho^2 N)$ and $O(\varrho^2 N)$, respectively. It follows that the total time complexity of the proposed algorithm is $O(KM^2 + tK\varrho N + t\varrho^2 N)$ which, in view of the constraint $\varrho \geq 2K$, can be simplified to $O(KM^2 + t\varrho^2 N)$.

3.4. Accuracy

We complete the description of the proposed algorithm by discussing its accuracy and the choice of the parameters K and ϱ. Basically, we use a similar approach to that of Powell [14, 15]. A key assertion is the following lemma:

Lemma 1. Let $\delta(x, y)$ be the truncation error of the estimate of the function ϕ at a point (x, y) obtained by means of subtabulation formula (9), i. e.

$$\delta(x, y) \equiv \left[\sum_{j=1-K}^{K} \alpha_j \phi(x - h + 2jh, y) \right] - \phi(x, y). \tag{12}$$

Provided that $K \geq 2$ and $\sqrt{x^2 + y^2} \geq \varrho h$ for some constant $\varrho \geq 2K$, it holds $|\delta(x, y)| \leq h^2 \Delta(K, \varrho)$, where $\Delta(K, \varrho)$ denotes the expression

$$\Delta(K, \varrho) \equiv \sum_{n=K}^{\infty} \frac{1}{n(n-1)\varrho^{2n-2}} \left| \sum_{j=1}^{K} \alpha_j (2j-1)^{2n} \right|.$$

For the proof of Lemma 1 see the Appendix.

Lemma 1 shows that after applying (9) to approximate the value of $s_h(kh, \ell h)$, the absolute value of the contribution from the ith term of s_h to the approximation error is less or equal than

$$|\lambda_i| h^2 \Delta(K, \varrho), \tag{13}$$

and the same is true for subtabulation scheme (10) due to symmetry. Thus, schemes (9) and (10) provide the relative accuracy

$$\Delta(K, \varrho)/\varrho^2 |\log \sqrt{(kh - x_i)^2 + (\ell h - y_i)^2}| \tag{14}$$

in the ith term of s_h. Now, since the algorithm uses only values $h \geq 1$, we have

$$\sqrt{(kh - x_i)^2 + (\ell h - y_i)^2} \geq \varrho h \geq 2K \geq 4.$$

Thus, the logarithm in expression (14) can be ignored. We can therefore conclude that if we wish to work to a relative accuracy ε, it is suitable to use parameters K and ϱ satisfying the condition

$$\Delta(K, \varrho)/\varrho^2 \leq \varepsilon. \tag{15}$$

Some recommended combinations of K and ϱ for various choices of ε are listed in Table 1. Besides satisfying condition (15), the given parameter values also have the property of minimizing the expression $3\varrho^2 + 12\varrho(2K - 1)$ which gives approximately the number of explicit evaluations of the TPS terms performed by the proposed algorithm in one iteration of the subtabulation procedure per interpolation point. Thus, besides providing required accuracy, the listed pairs of K and ϱ are expected to minimize the computational time as well.

Table 1. Recommended combinations of parameters K and ϱ for various choices of relative accuracy ε.

ε	K	ϱ
10^{-6}	4	13
10^{-7}	5	15
10^{-8}	5	18
10^{-9}	6	20
10^{-10}	7	22
10^{-11}	8	24

4. NUMERICAL EXPERIMENTS

To measure the performance of the proposed algorithm for the TPS evaluation and to compare it with the original Powell's algorithm, the following three experiments were carried out.

4.1. Experiment 1

In the first experiment, the new and the Powell's algorithms were compared on an artificially formed test spline.

The test spline consisted of $N = 100$ interpolation points and its shape is illustrated in Figure 4. It was obtained as a solution to the interpolation problem

$$s(x_i, y_i) = \frac{1}{2}\cos\left(\sqrt{x_i^2 + y_i^2}\right) + \frac{1}{2} \in [0,1], \quad i = 1, \dots, N$$

where the points (x_i, y_i), $i = 1, \dots, N$ were selected at random in the square $[-3\pi, 3\pi] \times [-3\pi, 3\pi]$.

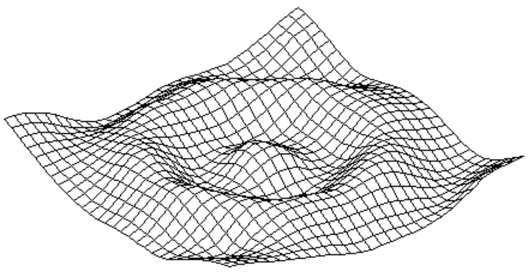

Fig. 4. The test spline used in Experiment 1.

The comparison of the algorithms was made on a grid of the size 1000×1000. Both the new and the original algorithms were applied to evaluate the spline several times, using a variety of settings of their respective input parameters. In the case of the new algorithm, the combinations of K and ϱ from Table 1 were used. In the case of the original algorithm, the value of ϱ was varied from 10 to 60.

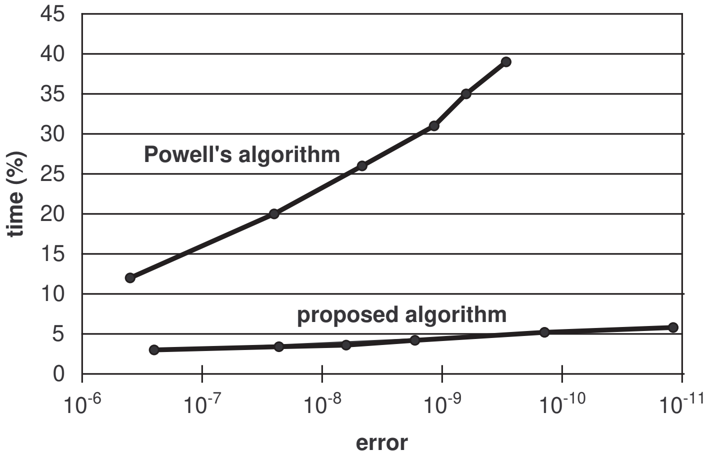

Fig. 5. Dependence of relative runtime on the approximation error.

For each run, we monitored the time of computation and the maximum approximation error between the computed and exact spline values. For the new and the original algorithms, the results of the measurements are summarized in Table 2. Graphically, comparison of the results is shown in Figure 5. The computing times are given in percents relatively to the direct spline evaluation using formula (2) which stands for 100 %.

Figure 5 clearly shows that the proposed algorithm provides substantial improvements over the original Powell's method. We can see that the computation was about five times faster in the case of the approximation error 10^{-7} and with increasing accuracy, the performance gap between the two algorithms rapidly widens. While more than 30 percent of time taken by the direct spline evaluation was required by the original algorithm to approximate the spline with the error lower than 10^{-9}, less than 6 percent was needed by the new algorithm even for approximation errors of the order of 10^{-11}.

Table 2. Results of Experiment 1. The proposed algorithm (left) and the original Powell's algorithm (right).

K	ϱ	time (%)	error
4	13	3, 0	$2,6.10^{-7}$
5	15	3, 4	$2,3.10^{-8}$
5	18	3, 6	$6,1.10^{-9}$
6	20	4, 3	$1,7.10^{-9}$
7	22	5, 1	$1,4.10^{-10}$
8	24	5, 8	$1,2.10^{-11}$

ϱ	time (%)	error
10	12	$3,8.10^{-7}$
20	20	$2,7.10^{-8}$
30	26	$4,4.10^{-9}$
40	31	$1,2.10^{-9}$
50	35	$5,9.10^{-10}$
60	39	$2,8.10^{-10}$

4.2. Experiment 2

The second experiment was aimed to study the dependence of computing time on the number of spline interpolation points N.

The experiment was made on five test splines which consisted of 100, 200, 300, 400 and 500 interpolation points and were obtained using the same procedure as in Experiment 1. As in Experiment 1, the algorithms were applied to evaluate the test splines on a grid of size 1000×1000. In the case of the new algorithm, the used parameter values were $K = 4$ and $\varrho = 13$. In the case of the original algorithm, we set $\varrho = 10$. Thus, we ensured that the achieved approximation error was always about 10^{-7}. The results of the experiment are shown in Table 3 and in Figure 6. The way of measurement of the "time" and "error" was the same as in Experiment 1. We can see that the relative runtime of the original Powell's algorithm increases approximately linearly with N. This is in accordance with the fact that while the time complexity of the direct spline evaluation is $O(M^2N)$, the number of operations required by the Powell's algorithm depends on N quadratically. On the other hand, the relative runtime of the modified algorithm almost does not depend on N, as expected. This is the main result of this paper. It shows that, in contrast to the original Powell's algorithm, the presented algorithm can provide very substantial gains over the direct evaluation even when the number of spline interpolation points is large.

4.3. Experiment 3

The aim of the last experiment was to test the applicability of the proposed algorithm in real task of computer vision – 3-D shape recovery from a depth map.

A depth map of a human face was obtained by disparity measurement on stereo images. This map consisted of 2206 irregularly distributed nodes whose z-coordinates equal the distances from the camera. These points were interpolated by a TPS to

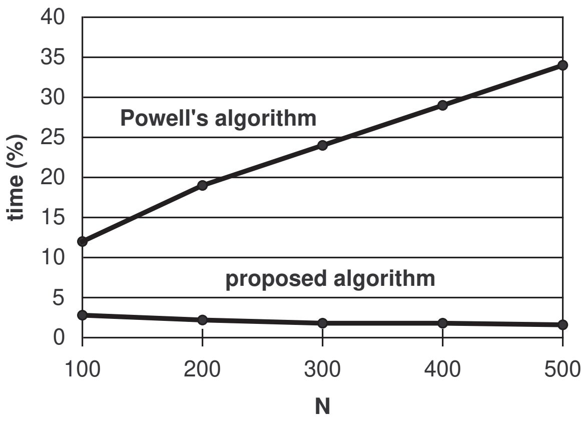

Fig. 6. Dependence of relative runtime on the number of interpolation points N.

Table 3. Results of Experiment 2: The proposed algorithm (left) and the original Powell's algorithm (right).

N	time (%)	error
100	2, 8	$2,6.10^{-7}$
200	2, 2	$1,2.10^{-7}$
300	1, 9	$1,5.10^{-7}$
400	1, 8	$2,6.10^{-7}$
500	1, 7	$1,5.10^{-7}$

N	time (%)	error
100	12	$3,8.10^{-7}$
200	19	$2,1.10^{-7}$
300	24	$1,1.10^{-7}$
400	29	$7,6.10^{-8}$
500	34	$6,7.10^{-8}$

recover the original shape of the face (see Figure 7). In accordance with the preceding experiments, the obtained spline was rescaled in the z-direction so that its range of values was approximately $[0, 1]$.

The spline was then evaluated on a regular 1200×2000 grid by means of the proposed algorithm. As in Experiment 1, all combinations of the input parameters K and ϱ from Table 1 were examined. Using the same format as in Table 2, the results of the experiment are given in Table 4.

Fig. 7. The face recovered from its depth map by the TPS interpolation.

Table 4. Results of Experiment 3.

K	ϱ	time (%)	error
4	13	1, 0	$9,6.10^{-7}$
5	15	1, 2	$8,4.10^{-8}$
5	18	1, 3	$1,3.10^{-8}$
6	20	1, 5	$9,7.10^{-10}$
7	22	1, 7	$8,4.10^{-11}$
8	24	2, 0	$7,3.10^{-11}$

5. CONCLUSION

We presented a new modification of the Powell's algorithm for Thin-Plate Splines fast calculation. The proposed method differs from the original algorithm by the dimension of the used approximation schemes. While Powell employed 2-D approximation formulae, we used only 1-D schemes. Thanks to this, our algorithm reduces the computational complexity from $O(M^2 + t\varrho^2 N + t\varrho N^2)$ to $O(M^2 + t\varrho^2 N)$. Another advantage of the presented algorithm is that it has been formulated for a general size of the subtabulation schemes. Thus, schemes of different approximation quality can be chosen for the computation, which greatly extends the range of accuracy for which the algorithm is well suited. Experimental comparison confirmed that the new algorithm performs much faster, especially when the number of interpolation nodes N is large and/or high accuracy of the calculated spline values is required.

APPENDIX

Proof of Lemma 1. The proof is based on the following two properties of the function ϕ:

$$\phi(x+t,y) = \sum_{n=0}^{\infty} \frac{t^n}{n!} \frac{\partial^n \phi(x,y)}{\partial x^n}, \quad |t| < \sqrt{x^2+y^2}, \tag{16}$$

$$\left| \frac{\partial^n \phi(x,y)}{\partial x^n} \right| \leq \frac{2(n-1)(n-3)!}{(x^2+y^2)^{\frac{n-2}{2}}}, \quad n \geq 3. \tag{17}$$

Both these properties are special cases of more general statements derived by Powell [15]. Since we assume that $\sqrt{x^2+y^2} \geq 2Kh$, it follows from (16) that the Taylor expansion

$$\phi(x-h+2jh,y) = \sum_{n=0}^{\infty} \frac{(2j-1)^n h^n}{n!} \frac{\partial^n \phi(x,y)}{\partial x^n}$$

is valid for every $j = 1-K, \ldots, K$. Substituting these expansions into (12) and invoking identities (11), we can write $\delta(x,y)$ in the form

$$\delta(x,y) = \sum_{n=K}^{\infty} \frac{2h^{2n}}{(2n)!} \frac{\partial^{2n} \phi(x,y)}{\partial x^{2n}} \left| \sum_{j=1}^{K} \alpha_j (2j-1)^{2n} \right|.$$

Provided that $K \geq 2$, inequality (17) implies the bound

$$|\delta(x,y)| \leq \sum_{n=K}^{\infty} \frac{h^{2n}}{n(n-1)(x^2+y^2)^{n-1}} \left| \sum_{j=1}^{K} \alpha_j (2j-1)^{2n} \right|.$$

Finally, in view of the inequality $\sqrt{x^2+y^2} \geq \varrho h$, we can write

$$|\delta(x,y)| \leq h^2 \sum_{n=K}^{\infty} \frac{1}{n(n-1)\varrho^{2n-2}} \left| \sum_{j=1}^{K} \alpha_j (2j-1)^{2n} \right|. \qquad \square$$

ACKNOWLEDGEMENT

This work was supported by the Grant Agency of the Czech Republic under the project No. 102/04/0155. The authors would like to thank Dr. Radim Šára, Czech Technical University, for kind providing the data used in Experiment 3.

(Received January 23, 2004.)

REFERENCES

[1] N. Arad, N. Dyn, D. Reisfeld, and Y. Yeshurun: Image warping by radial basis functions: Application to facial expressions. CVGIP: Graphical Models and Image Processing *56* (1994), 161–172.
[2] N. Arad and C. Gotsman: Enhancement by image-dependent warping. IEEE Trans. Image Processing *8* (1999), 1063–1074.
[3] R.K. Beatson and G.N. Newsam: Fast evaluation of radial basis functions. Comput. Math. Appl. *24* (1992), 7–19.
[4] M. Berman: Automated smoothing of image and other regularly spaced data. IEEE Trans. Pattern Analysis and Machine Intelligence *16* (1994), 460–468.
[5] F.L. Bookstein: Principal warps: Thin-plate splines and the decomposition of deformations. IEEE Trans. Pattern Analysis and Machine Intelligence *11* (1989), 567–585.
[6] J.C. Carr, W.R. Fright, and R. Beatson: Surface interpolation with radial basis functions for medical imaging. IEEE Trans. Medical Imaging *16* (1997), 96–107.
[7] J. Duchon: Interpolation des fonctions de deux variables suivant le principle de la flexion des plaques minces. RAIRO Anal. Num. *10* (1976), 5–12.
[8] J. Flusser: An adaptive method for image registration. Pattern Recognition *25* (1992), 45–54.
[9] A. Goshtasby: Registration of images with geometric distortions. IEEE Trans. Geoscience and Remote Sensing *26* (1988), 60–64.
[10] L. Greengard and V. Rokhlin: A fast algorithm for particle simulations. J. Comput. Phys. *73* (1987), 325–348.
[11] W.E.L. Grimson: A computational theory of visual surface interpolation. Philos. Trans. Roy. Soc. London Ser. B *298* (1982), 395–427.
[12] R.L. Harder and R.N. Desmarais: Interpolation using surface splines. J. Aircraft *9* (1972), 189–191.
[13] R. Kašpar and B. Zitová: Weighted thin-plate spline image denoising. Pattern Recognition *36* (2003), 3027–3030.
[14] M.J.D. Powell: Tabulation of thin plate splines on a very fine two-dimensional grid. In: Numerical Methods of Approximation Theory, Volume 9 (D. Braess and L.L. Schumacher, eds.), Birkhäuser Verlag, Basel, 1992, pp. 221–244.
[15] M.J.D. Powell: Tabulation of Thin Plate Splines on a Very Fine Two-Dimensional Grid. Numerical Analysis Report of University of Cambridge, DAMTP/1992/NA2, Cambridge 1992.
[16] K. Rohr, H.S. Stiehl, T.M. Buzug, J. Weese, and M.H. Kuhn: Landmark-based elastic registration using approximating thin-plate splines. IEEE Trans. Medical Imaging *20* (2001), 526–534.
[17] G. Wahba: Spline Models for Observational Data. SIAM, Philadelphia 1990.
[18] G. Wolberg: Digital Image Warping. IEEE Computer Society Press, 1990.

Petr Luner and Jan Flusser, Institute of Information Theory and Automation – Academy of Sciences of the Czech Republic, Pod Vodárenskou věží 4, 182 08 Praha 8. Czech Republic.
e-mails: luner,flusser@utia.cas.cz