Thin Plate Spline Warping: Code Organization, Demo Usage, and Test Results

Fitzgerald J. Archibald

Electrical and Computer Engineering

University of California, Santa Barbara

USA

Table of Contents

Scope

This document provides the code organization of thin-plate-spline (TPS) warping and spatial interpolation of non-uniform grids using nearest neighbor and inverse distance weighting methods. Further, this document explains the usage (internal, input and output parameters) of these modules. Also, usage details on the test and demo applications are provided in this document. Finally, the test results and performance are described in this document. For more details on code design, please refer to the Matlab source code. Also, refer to the white papers for details on thin plate splines, nearest neighbor interpolation, inverse distance weighted interpolation, and uniform interpolation using spline and bilinear interpolation.

Directory Structure

students\fitzgeraldarchibald\tpsWarp

tpswarp.m

interp2d.m

nearestInterp.m

idwMvInterp.m

students\fitzgeraldarchibald\tpsWarp\tst

interpTimeDemo.m

tpsWarpDemo.m

File Organization

tpsWarp.m

```
% Description: Thin-Plane spline warping of the input image (img) to
% output image (imgw). The warping is defined by a set of reference points
% (Zp=[Xp Yp]) on the [img] image and a set of corresponding points (Zs)
% on the (imgw) image. The warping will translate Zp in img to Zs in imgw.
%
% Input:
% img - input image
% outDim - Output canvas ([W H])
% Zp - landmark in img
% Zs - landmark in canvas
% interp.method - interpolation mode('nearest', 'invdist', 'none')
% interp.radius - Max radius for nearest neighbor interpolation or
%                 Radius of inverse weighted interpolation
% interp.power - power for inverse weighted interpolation
%
% Output:
% imgw - warped image with no holes
% imgwr - warped image with holes
% map - Map of the canvas with 0 indicating holes and 1 indicating pixel
%
% Reference:
% F.L. Bookstein, "Principal Warps: Thin-Plate splines and the
% decomposition of deformations", IEEE Transaction on Pattern Analysis and
% Machine Intelligence, Vol. 11, No. 6, June 1989
```

interp2d.m

```
% Description:
% Interpolation of the warped grid using nearest neighbor or inverse
% distance weighted interpolation. This routine is invoked by
% tpsWarp.m
%
% Inputs:
% X,Y - mesh grid of input image
% Xwr, Ywr - warped grid
% img - input image
% outH, outW - output warped image dimensions
% interp.method - interpolation mode('nearest', 'invdist', 'none')
% interp.radius - Max radius for nearest neighbor interpolation or
%                 Radius of inverse weighted interpolation
% interp.power - power for inverse weighted interpolation
%
% Output:
% imgw - interpolated image (warp+interpolate)
% imgwr - warped image with holes
% map - Map of the canvas with 0 indicating holes and 1 indicating pixel
```

nearestInterp.m

```
% Description:
% Fill holes using nearest neighbor (median filter of neighbors) interpolation.
% This module is invoked by interp2d.m. However, this module maybe useful
% as a standalone unit in other non-uniform interpolation applications.
%
% Inputs:
```

```
% imgw - input image
% map - Map of the canvas with 0 indicating holes and 1 indicating pixel
% maxhw - Max radius for nearest neighbor interpolation
%
% Output:
% out - interpolated image
```

idwMvInterp.m

```
% Description:
% Fill holes using Inverse Distance Weighting multivariate interpolation.
% This module is invoked by interp2d.m. However, this module maybe useful
% as a standalone unit in other non-uniform interpolation applications.
%
% Inputs:
% imgw - input image
% map - Map of the canvas with 0 indicating holes and 1 indicating pixel
% maxhw - Radius of inverse weighted interpolation
% p - power for inverse weighted interpolation
%
% Output:
% out - interpolated image
%
% Reference:
% http://en.wikipedia.org/wiki/Inverse_distance_weighting
```

Demo/Test Programs

tpsWarpDemo.m

```
% Demonstration of Thin-Plate-Spline Warping. In this program,
% the input image is warped from landmark points to their
% correspondence using thin plate spline interpolation.
%
% Inputs:
% imgInFilename - input image filename
% mapFilename - colormap data file for display
% lmFilename - landmark data file for display
%
% Output:
% Display of warped image
%
% Example Usage:
% tpsWarpDemo('..\data\0505_02.jpg','map.mat','tpsDemoLandmark.mat')
%
% Note:
% 1. To select landmarks interactively, change line number 30 from
% "if 0" to "if 1". After execution of the program, enter the
% number of landmark points using keyboard in the command window.
% Then, select the landmark points in subplot 211 and
% correspondence points in subplot 212. Once all the pointsare
% selected, the program will display the warped image with and
% without interpolation.
% 2. To change interpolation methods, modify line numbers 21 to 23
% with appropriate values for interp data structure.
%
% Example output:
% The subplots in Fig. 1 show input image, warped image with no
% interpolation, and warped image with inverse distance weighted
% interpolation. The blue dots correspond to landmarks and their
% correspondence.
```

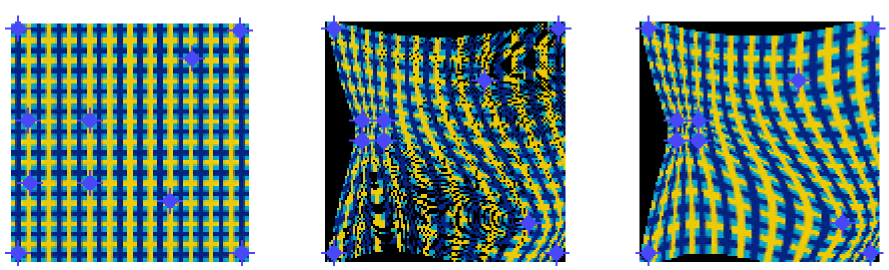

Fig.1 TPS Warping and inverse distance weighted interpolation demonstration

interpTimeDemo.m

```
% Demonstration of application of TPS warping for temporal
% interpolation of video sequences. This demo consists of
% 3 parts: temporal interpolation using TPS, temporal
% interpolation using frame averaging, and measurements.
% For temporal interpolation, alternative frames are

% discarded from the input video sequence. For example,
% consider the frames numbered as 1, 2, 3, and so on.
```

```
% In this case, all even numbered frames are dropped.
% That is, the even numbered frames are constructed
% using temporal interpolation schemes (TPS warping and
% frame averaging). The original even numbered frames in
% the video sequence are used for error measurements and
% quality comparison.
%
% For TPS warping, the landmark points provided through
% the data file input are used. It is advisable to select
% adequate number of landmarks in stationary region to
% minimize deformations caused by warping in addition to
% the landmarks capturing non-rigid object form. The
% landmark points are selected on the odd numbered frames.
% The correspondence points on the even numbered (missing)
% frames are generated using spline interpolation of the
% landmark positions of the odd numbered frames. Then, the
% previous odd numbered frame's landmark points are warped
% to the missing even numbered frame's correspondence points.
%
% In case of frame averaging, previous and next frames
% are averaged to create current missing frame.
%
% Dependencies:
% 1. mmread, mmwrite, [mmplay] available at:
% http://www.mathworks.com/matlabcentral/fileexchange/8028
% http://www.mathworks.com/matlabcentral/fileexchange/15881
% http://www.mathworks.com/matlabcentral/fileexchange/15880
%
% Inputs:
% vidInFilename - input video filename
% lmFilename - data file containing landmark points (stationary region)
% NLPs - number of landmark points (non-rigid motion)
% numFrames - number of frames to be processed (need to match with
%             lmFilename)
%
% Output:
% inp.avi (input), avg.avi (averaging), wrp.avi (warping) video files
% erra.avi (error of averaging) and errw.avi (error of warping) video files
% Display of warped video, error and interpolated positions
%
% Example:
% interpTimeDemo('..\data\MOV03798.MPG', 'MOV03798ext.mat',15, 25)
%
% Example output:
% Figure 2 shows the spline interpolated result of X-coordinate of
% a correspondence point and the original landmark point across 25
% frames of a video sequence.
%
% Figure 3 subplot 311 shows the original frame. Subplot 312 shows
% temporal interpolated frame using frame averaging. Subplot 313
% shows temporally interpolated frame using TPS warping.
%
% Figure 4 subplot 221 shows error of interpolated frame using
% frame averaging and original frame. Subplot 223 shows
% corresponding error histogram. Subplot 222 and 224 correspond
% to interpolation using TPS warping.
%
% Figure 5  shows the mean squared error of interpolation using
% frame averaging and TPS warping for 25 frames of a video
% sequence.
```

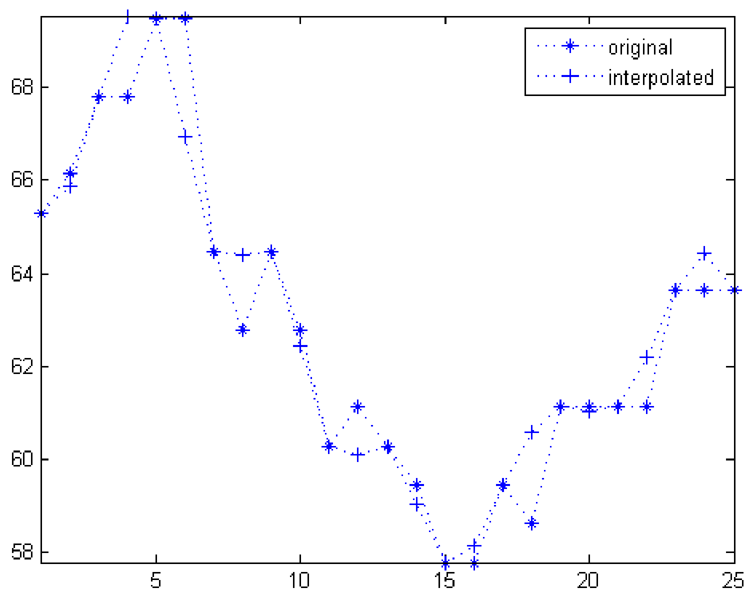

Fig. 2 X-coordinate of Interpolated vs Original landmark point for 25 frames in a video sequence

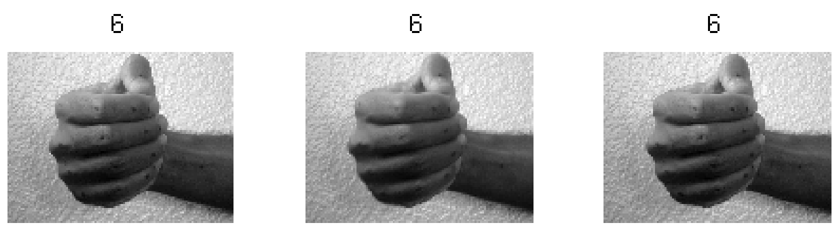

Fig. 3 Temporal interpolation using TPS warping and frame averaging

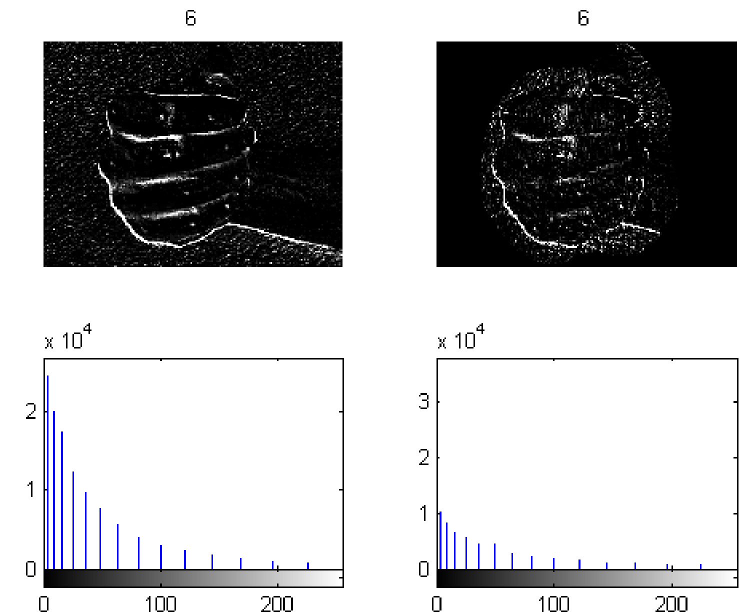

Fig. 4 Error and histogram of error of interpolated frames using frame averaging and TPS warping wrt original frame.

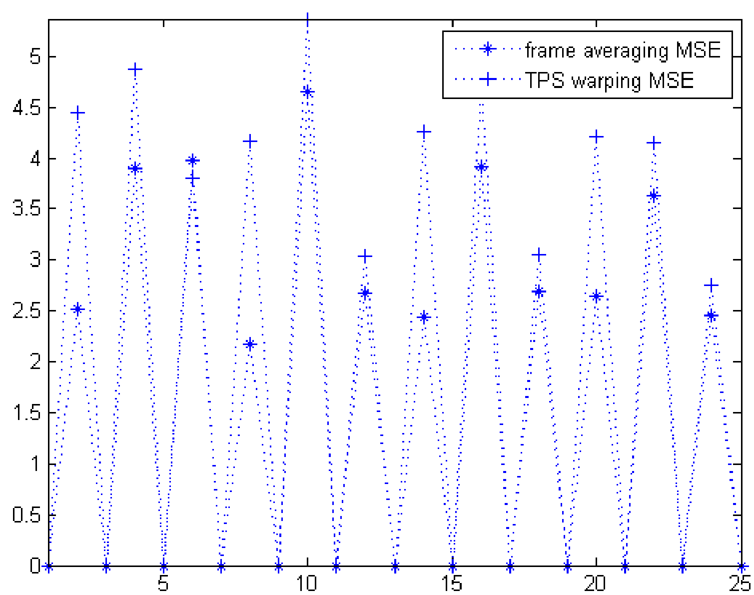

Fig. 5 MSE of interpolated frames using frame averaging and TPS warping wrt original frame for a video sequence of 25 frames.

Test Procedure and Results

Warping

1. With 3 landmark points, affine transformation is visualized.

2. With 4 or more points, various rigid transformations like scaling, rotation and scaling are visualized.

3. With 4 or more points, non-rigid transformations (both expansion and compression) are visualized.

Non-uniform Spatial Interpolation

A 128x128 random image is generated using rand function in Matlab. The image is thresholded with 0.5 as threshold to create binary image. The pixel values with 0s are taken as holes. Then, nearest neighbor and inverse distance weighted interpolation are used to fill the holes. The input image, for interpolation, of size 128x128 is generated using a combination of sine waves. Then, the mean squared error between the input image and the interpolated images (nearest neighbor and inverse distance weighted interpolation) are compared.

In general, inverse distance weighted interpolation is better by 2 times in comparison to the nearest neighbor interpolation for images with single frequency components. For multiple tone images, the mean squared error of inverse distance weighted interpolation is better than nearest neighbor only in fractional digits.

Table I provides the mean squared error comparison of few synthesized test images of size 128x128.

Table I. Non-uniform interpolation test results

Input image	**MSE of nearest neighbor**	**MSE of inverse distance weighting**
sinxmsiny_allf.bmp	4.7587	4.7230
sinxmsiny_allf1.bmp	5.3068	5.3582
sinxmsiny.bmp	0.6038	0.3584
sinxmsiny_1f.bmp	1.2040	0.6783

Temporal Interpolation

The test procedure is elaborated in interpDemo.m. Figure 5 provides the quality comparison of the frame averaging and TPS warping based interpolation methods. It can be seen generally that frame averaging has smaller mean squared error compared to TPS warping based interpolation. This is observed for a test video stream (MOV3798.MPG) generated using finger movements of a hand. The fingers are marked with a pen for landmark points. Then these landmark points are manually selected and stored in file MOV3798ext.mat. Landmarks are selected in regions of change as well as in regions of no change to constrain the warping. Total of 41 points are used of which 15 are on regions of shape change. Only the first 25 frames of the video are used for testing.

Performance

Table II. Execution time of TPS warping on a 340x340 image with 10 landmark points

Module	**Execution** ** **Time (sec)**	**Load (%)**
Grid Warping	0.51	9.99
Warped image without interpolation	0.08	1.52
Inverse Distance Weighted Interpolation*	4.47	88.48
Nearest Neighbor Interpolation*	4.88	89.34

* Only one of these two non-uniform interpolation methods are used in warping

** The execution time is for 10 landmark/correspondence points on a 340x340x3 image. The total number of holes that required filling are 37301 pixels. Total number of pixels in the image are 115600 (340x340).

Table III. Execution time of TPS warping on a 640x480 image with 41 landmark points

Module	**Execution** ** **Time (sec)**	**Load (%)**
Grid Warping	4.13	89.51
Warped image without interpolation	0.18	3.84
Inverse Distance Weighted Interpolation	0.31	6.65

** The execution time is for 41 landmark/correspondence points on a 640x480x3 image. The total number of holes that required filling are 1586 pixels. Total number of pixels in the image are 307200 (640x480).

Table III is more realistic data of TPS warping routine usage for temporal interpolation of biological images.

Notes:

1. The non-uniform interpolation needs to be calculated 3 times (once per each color) on each hole.
2. The execution time of non-uniform interpolation and warped image generation routines depend on the number of holes. Hence, the load (%) distribution also may vary depending on the number of holes.
3. The execution time of "Grid Warping" is dependent on the number of landmark points and the image dimension. However, it is independent of the number of holes (positions in warped image without backward mapping).

Future Work

1. Extend the the thin plate spline interpolation routine for 3D data. If 3D data can be treated as multiple 2D planes, then the current routine is sufficient as 3D extension can be achieved by multiple invocations of the same routine for different planes. Each plane having its own set of landmark and correspondence points. If 3D data need to be treated as a volume, then the radial functions in thin plate spline interpolation may need to be extended to account for 3 dimensions. Also, the nearest neighbor and inverse distance weighted interpolation would require 3D support of volumetric data.

2. Device a inverse mapping procedure for measuring the accuracy of warping and interpolation routines. A synthesized image like sine pattern maybe used as input.

3. Develop alternate procedure(s) to compare temporal interpolation using warping and averaging. The current method of mean squared error is not sufficient. A possible addition could be comparison of edge strengths.

4. Optimization of the non-uniform interpolation and warping routines for use in video or volumetric (3D) data.

5. Compare TPS warping quality vs frame averaging quality for biological data samples.

6. Measure and compare quality of inverse distance weighted and nearest neighbor interpolation using biological images.