Image Warping based on 3D Thin Plate Spline

Chanjira Sinthanayothin  
NECTEC, NSTDA  
112, Thailand Science Park, Phahon Yothin Road  
Klong Luang, Pathumthani, 12120, Thailand  
chanjira.sinthanayothin@nectec.or.th

Wisarat Bholsithi  
NECTEC, NSTDA  
112, Thailand Science Park, Phahon Yothin Road  
Klong Luang, Pathumthani, 12120, Thailand  
wisarat.bholsithi@nectec.or.th

Abstract—This article has shown the simulation change after moving landmarks on medical images by applying 3-dimension thin plate spline through the following 3 steps: the first step is to writing the program to open and display 3-dimension medical images using Borland C++ Builder 6 along with OpenGL as the engine to display the images. The second step is to distort 3-dimension images by defining the initial set of landmarks along with the final destination of the landmarks on the cube corners with the texture image. The last step is to apply the distortion algorithm on different corresponding points within the skeletal model in three-dimension such as the skull, limbs, and joints and display the result after distortion. The results from the experiment have shown that the distortion of three-dimension image by applying thin plate spline can simulate the results after moving the landmarks on the medical images.

Keywords: 3D Thin Plate Spline, 3D Image Warping, 3D Image Distortion, OpenGL

1 Introduction

This paper presents a technique for 3D image warping simulation based on thin plate spline. The users will be able to see 3D structure movement in the directions according to the users' decisions with this simulation. Therefore, the users who need to develop more complicated software such as animated virtual reality or further study in 3D images would find this technique more informative and useful.

3D Image warping technique uses the 3D vertex or 3D coordinates as the data source. Although, there are a number of techniques for doing image warping such as the faster 3-dimension thin plate spline algorithm as shown in the research of Roberts et al. [1] as well as the fast reconstruction of 3D object by radial basis function on 3D thin plate spline algorithm by Carr et al. [2] to improve the performance of 3-dimension simulation.

However, the faster 3-dimension thin plate spline technique of Roberts has been applied to the 3-dimension head image implemented using MATLAB, which is difficult to exported as commercial product in the future.

Furthermore, the application of radial basis function on thin plate spline algorithm of Carr would take longer to render 3D objects when the radial basis function algorithm has to manipulate the model data.

Therefore, our paper is another offer for someone who has strong interests to developed more research work in this 3D image field.

For further researches and developments, there is an application of this technique on the 3D face recognition with Vectors [3] as well as the application medical image registration in the same way as the registration of MRI images to find the exact position of breast cancer while reducing the motion artifacts [4].

2 Three-Dimension Thin Plate Spline

Thin plate spline (TPS) [5] is the technique for estimating the random data from 2 paring sets of data to construct spline map from the affine factor for linear distortion and weighting factor for nonlinear distortion for image registration. The first step of thin plate spline is to calculate the affine factor \( A \) and weighting factor \( W \) by solving eq. 1

\[
\begin{bmatrix}
K \\
\hat{P}^T \\
O(4,4)
\end{bmatrix}
\begin{bmatrix}
W \\
A
\end{bmatrix} =
\begin{bmatrix}
V \\
O(4,3)
\end{bmatrix}
\]

(1)

\( O(4,4) \) is a zero matrix of 4x4 while \( O(4,3) \) is a zero matrix of 4 x 3.

\( P \) is a matrix of the initial landmark positions before moving without the additional value defined in eq. 2 while \( \hat{P} \) is an initial landmark position set matrix before moving with the additional value 1 in every row defined in eq. 3 and \( \hat{P}^T \) used in eq. 1 is a matrix \( \hat{P} \) that has rows and columns switched (transposed \( \hat{P} \) matrix).

\[
P =
\begin{bmatrix}
x_1 & y_1 & z_1 \\
... & ... & ... \\
x_n & y_n & z_n
\end{bmatrix}
\]

(2)