MOTION ESTIMATION IN MEDICAL IMAGE SEQUENCES USING INVERSE POLYNOMIAL INTERPOLATION

Saleh Al-Takrouri
School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW, Australia and National ICT Australia Ltd, Sydney, NSW, Australia

Andrey V. Savkin
School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, NSW, Australia

Keywords: Motion estimation, polynomial interpolation, inverse interpolation, medical image sequences.

Abstract: We propose a new method for motion estimation between two successive frames in medical image sequences and videos. The method is based on inverse polynomial interpolation.

1 INTRODUCTION

The applications of motion estimation have been increasingly gaining interest in the field of medical imaging. (Hemmendorff, 2001) proposed a framework for motion estimation of 2D X-ray angiography images and 3D MRI mammograms. Deformable models were used by (Kurabayashi et al., 2005) to estimate the motion in time-series chest MR images. (Auvray et al., 2006) applied motion estimation to transparent X-ray image sequences.

Motion estimation is a key step in video coding and compression, which is an important tool to achieve bandwidth reduction when transmitting medical image sequences and videos. In addition, remote and robot-assisted surgeries and medical diagnostic tools can benefit from motion estimation in analyzing and interpreting the motions of body parts.

2 PROBLEM STATEMENT

Consider the pair of images $I_1(r,c)$ and $I_2(r,c)$, both of size $R \times C$, where the spacial arguments r and c refer to the pixel at the rth row and cth column. Here we assume that the two images are successive frames in a medical video or image sequence with spacial differences between the two images but no change in intensity. The pixels $I_2(r,c)$ of the destination image can be generated by shifting corresponding pixels in the source image in the 2D space. Let $\tau_1(r,c)$ and $\tau_2(r,c)$ be the horizontal and vertical shifts respectively, then we can write

$$I_2(r,c) = I_1(r + \tau_2(r,c), c + \tau_1(r,c)) \tag{1}$$

Sub-pixel shifts are approximated by 2D polynomial interpolations within square neighborhoods of the source image I_1. The advantage of this choice is the separability and simplicity of implementation that allows an approximation of (1) to be written in an easily manipulated form.

Now assume that $|\tau_1(r,c)| \leq p$ and $|\tau_2(r,c)| \leq p$. Then the neighborhood in consideration would be of size $(2p+1) \times (2p+1)$ and the interpolation polynomial is of order $2p$. Define a vector function

$$\mathbf{u}(\tau_i(r,c)) = \begin{bmatrix} \tau_i^{2p}(r,c) \\ \tau_i^{2p-1}(r,c) \\ \vdots \\ 1 \end{bmatrix} \tag{2}$$

The polynomial approximation of (1) can be written in the form

$$I_2(r,c) = \mathbf{u}^T(\tau_1(r,c))\mathbf{A}(r,c)\mathbf{u}(\tau_2(r,c)) \tag{3}$$

where $\mathbf{A}(r,c)$ is a $(2p+1) \times (2p+1)$ matrix . For simplicity, the spacial arguments (r,c) are dropped from this point and assumed implicitly

$$I_2 = \mathbf{u}^T(\tau_1)\mathbf{A}\mathbf{u}(\tau_2) \tag{4}$$

With τ_1 and τ_2 are the unknowns in equation (4), our goal is to solve the inverse polynomial interpolation problem represented by (4), which would also solve the motion estimation problem described above.

Al-Takrouri S. and V. Savkin A. (2008).
MOTION ESTIMATION IN MEDICAL IMAGE SEQUENCES USING INVERSE POLYNOMIAL INTERPOLATION.
In *Proceedings of the First International Conference on Bio-inspired Systems and Signal Processing*, pages 212-215
DOI: 10.5220/0001057902120215
Copyright © SciTePress

Many motion estimation methods use multiscale or hierarchial levels in order to process large motions, the proposed method can handle the size of motions that typically exist between two successive frames and therefor we are not using any multiscale pyramids.

3 POLYNOMIAL INTERPOLATION

For a pixel that is assumed to move a maximum of p pixels to the right or the left in a 1D source signal, the neighborhood considered $\mathbf{Y}$ is of length $2p+1$ and centered at the element $y(0)$. Using polynomial interpolation, $y(x)$ representing a shift from the center by a value x where $|x| \leq p$ can be approximated by using a polynomial of order $2p$

$$y(x) = c_{2p}x^{2p} + c_{2p-1}x^{2p-1} + \cdots + c_2x^2 + c_1x + c_0 \tag{5}$$

The coefficients $c_{2p} \cdots c_0$ are found by solving a system of $2p+1$ linear equations of the form

$$\mathbf{XC} = \mathbf{Y}^T \tag{6}$$

where

$$\mathbf{X} = \begin{bmatrix} (-p)^{2p} & (-p)^{2p-1} & \cdots & -p & 1 \\ (-p+1)^{2p} & (-p+1)^{2p-1} & \cdots & -p+1 & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ (p-1)^{2p} & (p-1)^{2p-1} & \cdots & p-1 & 1 \\ p^{2p} & p^{2p-1} & \cdots & p & 1 \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} c_{2p} \\ c_{2p-1} \\ \vdots \\ c_1 \\ c_0 \end{bmatrix}, \qquad \mathbf{Y}^T = \begin{bmatrix} y(-p) \\ \vdots \\ y(0) \\ \vdots \\ y(p) \end{bmatrix} \tag{7}$$

and the solution to the linear system is given by

$$\mathbf{C} = \mathbf{QY}^T, \qquad \mathbf{Q} = \mathbf{X}^{-1} \tag{8}$$

The matrix $\mathbf{X}$ in (7) is a special form of the Vandermonde matrix. Its inverse can be found using an explicit LU factorization discussed in the paper by (Olver, 2006).

Denote the ith row of the matrix $\mathbf{Q}$ in (8) as $\mathbf{q}_i$. The process of 1D polynomial interpolation can be expressed as

$$y(x) = \mathbf{Y} \sum_{i=1}^{2p+1} \mathbf{q}_i^T x^i \tag{9}$$

The 1D polynomial interpolation in (9) can be extended to the 2D case. When a pixel in a 2D neighborhood is assumed to move a maximum of p pixels along any dimension, the neighborhood in consideration is of size $(2p+1) \times (2p+1)$ and centered at the pixel $n(0,0)$.

Recall that $\mathbf{q}_i$ is the ith row of the matrix $\mathbf{Q}$ in (8). We use the fact that the 2D polynomial interpolation is separable to build the matrix $\mathbf{A}$, with each element on the ith row and jth column is given by

$$a_{(i,j)} = \sum_{m=-p}^{p} \sum_{n=-p}^{p} \mathbf{N}(m,n)\mathbf{q}_j(m)\mathbf{q}_i(n) \tag{10}$$

The process of 2D polynomial interpolation can be expressed now as

$$I_2 = \sum_{i=1}^{2p+1} \sum_{j=1}^{2p+1} a_{(i,j)} \tau_1^{2p+1-i} \tau_2^{2p+1-j} \tag{11}$$

with the matrix form of (11) is as given by (4).

4 SOLUTION OF INVERSE INTERPOLATION

4.1 The Linear Approximation

First we start by finding a linear approximation of (4) around some values $\overline{\tau}_1, \overline{\tau}_2$ (to be defined later). The first order approximation using Taylor series is easily computed since the differentiation of (4) with respect to ether τ_1 or τ_2 is trivial.

$$\begin{aligned} I_2 \approx \; & \mathbf{u}^T(\overline{\tau}_1)\mathbf{A}\mathbf{u}(\overline{\tau}_2) + \dot{\mathbf{u}}^T(\overline{\tau}_1)\mathbf{A}\mathbf{u}(\overline{\tau}_2)\left[\tau_1 - \overline{\tau}_1\right] \\ & + \mathbf{u}^T(\overline{\tau}_1)\mathbf{A}\dot{\mathbf{u}}(\overline{\tau}_2)\left[\tau_2 - \overline{\tau}_2\right] \end{aligned} \tag{12}$$

Equation (12) is written in a form of a linear equation

$$\overline{I}(\overline{\boldsymbol{\tau}}) = \mathbf{H}(\overline{\boldsymbol{\tau}})\boldsymbol{\tau} \tag{13}$$

where

$$\begin{aligned} \overline{I}(\overline{\boldsymbol{\tau}}) &= I_2 - \mathbf{u}^T(\overline{\tau}_1)\mathbf{A}\mathbf{u}(\overline{\tau}_2) + \mathbf{H}(\overline{\boldsymbol{\tau}})\overline{\boldsymbol{\tau}} \\ \mathbf{H}(\overline{\boldsymbol{\tau}}) &= \begin{bmatrix} \dot{\mathbf{u}}^T(\overline{\tau}_1)\mathbf{A}\mathbf{u}(\overline{\tau}_2) & \mathbf{u}^T(\overline{\tau}_1)\mathbf{A}\dot{\mathbf{u}}(\overline{\tau}_2) \end{bmatrix} \\ \overline{\boldsymbol{\tau}} &= \begin{bmatrix} \overline{\tau}_1 \\ \overline{\tau}_2 \end{bmatrix}, \quad \boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \end{aligned} \tag{14}$$

An approximate solution to equation (13) can be found as

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{G}(\overline{\boldsymbol{\tau}})\overline{I}(\overline{\boldsymbol{\tau}}) \\ \mathbf{G}(\overline{\boldsymbol{\tau}}) &= \left(\mathbf{H}^T(\overline{\boldsymbol{\tau}})\mathbf{H}(\overline{\boldsymbol{\tau}}) + \mathbb{I}_2\right)^{-1}\mathbf{H}^T(\overline{\boldsymbol{\tau}}) \end{aligned} \tag{15}$$

where $\mathbb{I}_2$ is the 2×2 identity matrix.

4.2 The Iterative Solution

Define $\bar{\boldsymbol{\tau}}(k)$ to be the accumulated shifts from initial step until the kth step

$$\bar{\boldsymbol{\tau}}(k) = \sum_{i=0}^{k} \boldsymbol{\tau}(i) \tag{16}$$

Starting with an initial value $\bar{\boldsymbol{\tau}}(0) = 0$, the linear equation (13) and its solution (15) can be used in an iterative manner as follows

$$\begin{aligned}
\bar{\boldsymbol{\tau}}(0) &= 0 \\
e(0) &= \bar{I}(\bar{\boldsymbol{\tau}}(0)) \\
\boldsymbol{\tau}(1) &= \mathbf{G}(\bar{\boldsymbol{\tau}}(0))e(0) \\
&= \mathbf{G}(\bar{\boldsymbol{\tau}}(0))\bar{I}(\bar{\boldsymbol{\tau}}(0)) \\
e(1) &= \bar{I}(\bar{\boldsymbol{\tau}}(1)) - \mathbf{H}(\bar{\boldsymbol{\tau}}(1))\,[\boldsymbol{\tau}(0)+\boldsymbol{\tau}(1)] \\
&= \bar{I}(\bar{\boldsymbol{\tau}}(1)) - \mathbf{H}(\bar{\boldsymbol{\tau}}(1))\bar{\boldsymbol{\tau}}(1) \\
\boldsymbol{\tau}(2) &= \mathbf{G}(\bar{\boldsymbol{\tau}}(1))e(1) \\
&= \mathbf{G}(\bar{\boldsymbol{\tau}}(1))\,[\bar{I}(\bar{\boldsymbol{\tau}}(1)) - \mathbf{H}(\bar{\boldsymbol{\tau}}(1))\bar{\boldsymbol{\tau}}(1)] \\
&\;\;\vdots
\end{aligned} \tag{17}$$

In general

$$\boldsymbol{\tau}(k+1) = \mathbf{G}(\bar{\boldsymbol{\tau}}(k))\,[\bar{I}(\bar{\boldsymbol{\tau}}(k)) - \mathbf{H}(\bar{\boldsymbol{\tau}}(k))\bar{\boldsymbol{\tau}}(k)] \tag{18}$$

Substituting $\bar{I}(\bar{\boldsymbol{\tau}}(k))$ from equation (14) into equation (18) yields the formula for the iterative numerical solution as

$$\boldsymbol{\tau}(k+1) = \mathbf{G}(\bar{\boldsymbol{\tau}}(k))\left[I_2 - \mathbf{u}^T(\bar{\tau}_1(k))\mathbf{A}\mathbf{u}(\bar{\tau}_2(k))\right] \tag{19}$$

When $\boldsymbol{\tau}(k+1)$ in the iterative equation (21) converges to zero (or a *small enough* number near zero) we get $\bar{\boldsymbol{\tau}}(k+1)$ such that

$$I_2 \simeq \mathbf{u}^T(\bar{\tau}_1(k+1))\mathbf{A}\mathbf{u}(\bar{\tau}_2(k+1)) \tag{20}$$

which is the solution to both problems of motion estimation and inverse polynomial interpolation. Finally, algebraic manipulation of (19) and using (16) simplify the solution into the iterative formula given by

$$\bar{\boldsymbol{\tau}}(k+1) = \bar{\boldsymbol{\tau}}(k) + \frac{I_2 - \mathbf{u}^T(\bar{\tau}_1(k))\mathbf{A}\mathbf{u}(\bar{\tau}_2(k))}{\mathbf{H}(\bar{\boldsymbol{\tau}}(k))\mathbf{H}(\bar{\boldsymbol{\tau}}(k))^T}\mathbf{H}(\bar{\boldsymbol{\tau}}(k))^T \tag{21}$$

Our solution in (21) is closely related to the algorithm proposed by (Biemond et al., 1987). The major difference is that in (Biemond et al., 1987) a bilinear interpolation was used to calculate the displacement frame difference, and the spatial gradients were obtained by rounding off the displacement estimates; whereas in we use polynomial interpolation which provides better interpolation and simplifies calculating the gradients. Also, (Biemond et al., 1987) used observations from a block of pixels.

Motion estimation results can be improved significantly by testing multiple initial values. Figure 1 shows the chosen initial positions for the 5×5 neighborhood (i.e. $p = 2$). For the $(2p+1)\times(2p+1)$ neighborhood, the number of initial values $\bar{\boldsymbol{\tau}}(0)$ is p^2. The different initial values are sorted and tried according to their distance from the mean shift obtained for the previously processed adjacent pixels in I_2, starting with the closest. This also establishes dependency between the motions of the image pixels. For an initial value, if the iterative equation (21) converges to a solution before reaching a specified maximum number of iterations, the result is recorded and there would be no need to try the other initial values. Otherwise, the next initial value is tried.

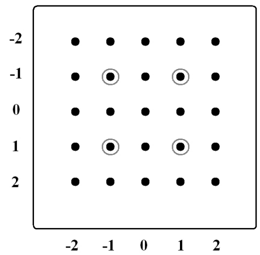

Figure 1: The circled pixel positions are the chosen initial positions for the 5×5 neighborhood.

5 RESULTS

We tested our method using gray-scaled images. For comparison, motion in the same frames was estimated by the elastic image registration method by (Periaswamy and Farid, 2003) and the widely-used optical flow method by (Lucas and Kanade, 1981). The Matlab code for Periaswamy and Farid's method is available on the internet (Web, 1). Examples show that our method provides better performance.

In the first example (Figure 2) two images are extracted from an echocardiography video (Web, 2). The images are of size 430×550 pixels. The second example (Figure 3) shows two images extracted from a video recorded during a robotic-assisted repair of a pulmonary artery (Web, 3). The images are of size 240×352 pixels. For both examples we chose $p = 7$, a convergence threshold of 0.001 and the maximum number of iterations to be 20.

For each example, we computed the peak signal-to-noise ratio (PSNR) for the displaced frame difference. The PSNR equation is defined by (22) and the results are listed in Table 1.

$$PSRN = 10\log_{10}\frac{255^2 RC}{\sum\limits_{r=1}^{R}\sum\limits_{c=1}^{C}(I_2(r,c) - I_s(r,c))^2} \tag{22}$$

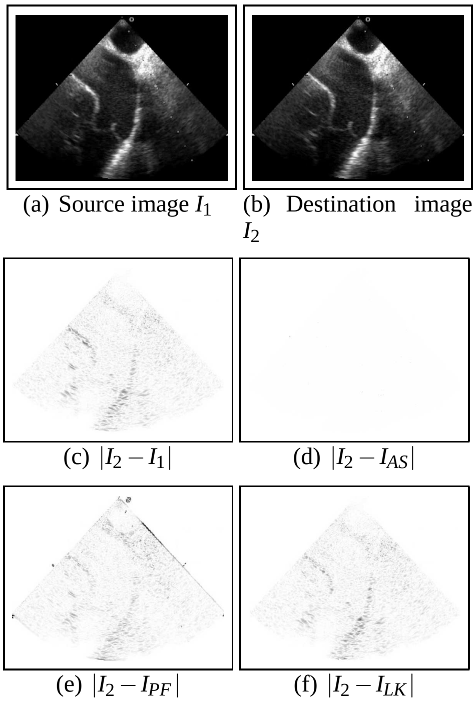

Figure 2: Motion estimation between two successive frames from echocardiography video.

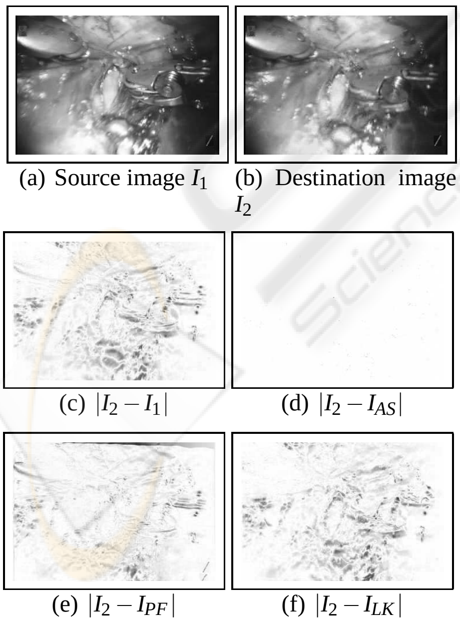

Figure 3: Motion estimation between two successive frames from robot-assisted artery surgery video.

Table 1: PSNR of displaced frame difference.

	Echocardiography	Artery Surgery
Our method	50.27 dB	42.34 dB
Periaswamy-Farid	25.66 dB	19.85 dB
Lucas-Kanade	27.10 dB	19.12 dB

In (Periaswamy and Farid, 2003) the motion within a small neighborhood was modeled locally by an affine transform. In video sequences the considered neighborhood may contain one or more different motions in addition to the stationary background. An attempt to model these motions and the static background using one affine transform will produce estimation errors. Our method does not suffer from this shortcoming because it works on a pixel level.

ACKNOWLEDGEMENTS

This work was supported by National ICT Australia and the Australian Research Council.

REFERENCES

Auvray, V., Bouthemy, P., and Lienard, J. (2006). Motion estimation in x-ray image sequences with bi-distributed transparency. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1057–1060.

Biemond, J., Looijenga, L., Boekee, D., and Plompen, R. (1987). A pel-recursive wiener-based displacement estimation algorithm. *Signal Processing*, pages 399–412.

Hemmendorff, M. (2001). *Motion estimation and compensation in medical imaging*. PhD Thesis, Linkopings University, Sweden.

Kurabayashi, Y., Kagei, S., Gotoh, T., and Iwasawa, T. (2005). Motion estimation for sequential medical images using a deformable model. *Systems and Computers in Japan*, 36:27–36.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the DARPA Image Understanding Workshop*, pages 121–130.

Olver, P. (2006). On multivariant interpolation. *Studies in Applied Mathematics*, 116:201–240.

Periaswamy, S. and Farid, H. (2003). Elastic registration in the presence of intensity variations. *IEEE Transactions on Medical Imaging*, 22:865–874.

Web (1). http://www.cs.dartmouth.edu/farid/research/.

Web (2). http://www.cts.usc.edu/videos.html.

Web (3). http://www.manbit.com/ers/ersvideo.asp.