Evaluation of Advanced Lukas-Kanade Optical Flow on Thoracic 4D-CT

Christoph Bernhard Hoog Antink · Tarunraj Singh · Puneet Singla · Matthew Podgorsak

Received: March 2002 / Accepted: April 2003

Abstract Extensive use of high frequency imaging in medical applications permit the estimation of velocity fields which corresponds to motion of landmarks in the imaging field. The focus of this work is on the development of a robust local optical flow algorithm for velocity field estimation in medical applications. Local polynomial fits to the medical image intensity-maps are used to generate convolution operators to estimate the spatial gradients. A novel polynomial window function with a compact support is used to differentially weight the optical flow gradient constraints in the region of interest. Tikhonov regularization is exploited to synthesize a well posed optimization problem and to penalize large displacements. The proposed algorithm is tested and validated on benchmark datasets for deformable image registration. The ten datasets include large and small deformations, and illustrate that the proposed algorithm outperforms or is competitive with other algorithms tested on this dataset, when using mean and variance of the displacement error as performance metrics.

Keywords Optical Flow · Acute Illness · Deformable Image Registration · Radiotherapy Planning

The authors would like to thank the National Science Foundation, which funded this project under grant CMMI-#0928630. C. Hoog Antink also would like to thank Fulbright and the German National Academic Foundation for partial funding.

Christoph Bernhard Hoog Antink
Dept. of Mechanical and Aerospace Engineering
SUNY at Buffalo
Buffalo, NY 14260
USA
E-mail: choogant@buffalo.edu

1 Introduction

Optical flow for estimating velocity fields has numerous applications in the medical field. In Interstitial Laser Therapy (ILT), thermal energy is deposited via lasers to treat tumors which are accessible via needle insertion. Magnetic resonance imaging (MRI) provides the ability to locate the target, guide the optical fiber and monitor the thermal effects [17]. This especially reduces invasiveness of the procedure by making implanted thermic sensors obsolete. At the same time, spatiotemporal information of the thermal effects can be obtained and analyzed using optical flow.

In applications such as TAVI (Transcatheter Aortic Valve Implantation) where a vioprosthetic aortic valve is implanted to replace a severly stenotic aortic valve, echocardiography and fluoroscopy play an important role today. Optical flow can be used in this acute illness application for the estimation of velocity field to assist placement of the guidewire and valve positioning [9]. In addition, when thinking about future applications such as active robotic surgical devices, it is obvious that highly accurate motion information that can be computed in real-time is crucial. Optical flow provides a method to obtain these from imaging modalities that are already in use.

One of the most important factors in successful acute illness intervention such as TAVI is patient diagnosis and planning of the procedure. Four Dimensional Ultrasound provides an excellent non-invasive source of dynamic cardiac information. However, the analysis of this data can be time-consuming and is prone to human error. Optical flow can be used to assist in the derivation of qualitative and quantitative cardiac information such as Left Ventricular Long Axis over time [15,8] that could be used for patient selection, therapy planning and follow-up. With a high accuracy, real-time method, more complex parameters like stroke volume could be monitored non-invasively.

Deformable Image Registration (DIR) of Computed Tomography (CT) scans for radiotherapy planning is another area where optical flow has proven to be an invaluable tool [16,10]. Here, large deformations in the range of 4cm and artifacts in the image data pose challenges on the estimator. At the same time, high demands are put on the accuracy of the algorithm, since small errors in the estimated velocity fields can finally lead to severe over irradiating of healthy tissue when used to create faulty therapy plans. For the application of optical flow as DIR, an elaborate benchmark problem [4,2] exists, which will be used in this work. Other objective benchmark problems exist, the interested reader may find an extensive set of tests and state-of-the-art algorithms in the "EMPIRE10 Challenge" [13].

The main focus of this paper is to create an efficient optical flow algorithm that has the accuracy and robustness necessary for medical applications. For this we propose an optical flow algorithm based on the Lucas/Kanade algorithm, which is a local method [12]. Several improvements to the original algorithms are made which are described in the paper.

- *Robust gradient calculation:*
  Early optical flow algorithms used finite difference to determine the spatial and time gradients. Since smoothness assumptions are integral to optical flow algorithms, a local polynomial fit to the intensity variations about the pixel of interest is synthesized. The determination of the spatial gradient can now can carried out easily.
- *Non-Gaussian weighting function:*
  The Lucas-Kanada algorithm uses a small spatial neighborhood of the pixel of interest to determine the local flow field. A weighting function is used to give more importance to the constraint at the center of the local patch relative to the pixels at the periphery. A recently developed finite support function where the desired smoothness can be prescribed is used to weight the pixel in the local patch.
- *Tikhonov Regularization:* For least squares problem which are ill posed, preference is given to solution with specific properties such as ones with small norm of the solution vector, smooth solution. This can be achieved via Tikhonov regularization which corresponds to including an additional term to the cost function. The optical flow problem is ill-posed and to favor small velocity field, Tikhonov regularization is achieved by penalizing the norm of the velocity field.
- *Balanced Optical Flow:* There is a desire to minimize the inconsistency in the velocity field when going from image 1 to image 2 and from image 2 to image 1. To achieve this goal, a simple technique which determines the two velocity fields, inverts the second, interpolates it to match the original grid and averaged is proposed in this work.

The resulting algorithm is implemented in MATLAB / C and applied to a set of benchmark problems [4,2]. In contrast to other non-quadratic optical flow formulations, the algorithm does not rely on a complex and potentially computational expensive minimization step but on well-posed matrix inversions.

The paper is outlined as follows: Section 2 describes the mathematical and algorithmic details of the proposed algorithm. Section 3 gives a detailed description of the benchmark data used to evaluate the performance of the proposed algorithm. Section 4 presents the results, which are then discussed in Section 5. In Section 6 conclusions are drawn and future work is proposed.

2 Methods

2.1 Optical Flow

Optical flow methods calculate an approximation of the motion field from time-varying image intensities. A common assumption in optical flow methods is that voxel intensities I might translate from one frame to another but their intensities are constant [11]. If we assume this translations to be small they can be well approximated by the Taylor series expansion [12]. This leads to

the gradient constraint equation:

$$\begin{aligned} 0 &= u(x,y,z,t)I_x + v(x,y,z,t)I_y + w(x,y,z,t)I_z + I_t \\ I_x &= \frac{\partial}{\partial x} I\left(x,y,z,t\right), I_y = \frac{\partial}{\partial y} I\left(x,y,z,t\right), I_z = \frac{\partial}{\partial z} I\left(x,y,z,t\right), \\ I_t &= \frac{\partial}{\partial t} I\left(x,y,z,t\right) \end{aligned} \tag{1}$$

where $u(x,y,z,t)$, $v(x,y,z,t)$ and $w(x,y,z,t)$ refer to the normalized velocity components in the x,y,z directions at time t.

2.1.1 Gradient Calculation

For the sake of clarity, the calculation of image gradients is delineated for two-dimensional images, the three-dimensional implementation is a straightforward extension. In digital image processing applications, images are represented discretely both in space and amplitude (grey-value). Spatial gradients are obtain by convolving the image with a filter kernel h_p as given in the equation:

$$\frac{\partial}{\partial p} I\left(x,y,t\right) = \sum_{k=-\infty}^{k=\infty} \sum_{l=-\infty}^{l=\infty} I\left(k,l\right) h_p\left(x-k, y-l\right) = I\left(x,y,t\right) * h_p \tag{2}$$

To increase robustness of the gradient calculation, the intensity distribution $I\left(x,y,t\right)$ will be approximated locally ($\hat{I}_L$) by a polynomial function, in this example a cubic:

$$\begin{aligned} \hat{I}_L\left(x,y,t\right) = \quad & a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 y^2 + \\ & a_6 xy + a_7 xy^2 + a_8 yx^2 + a_9 x^3 + a_{10} y^3 \end{aligned} \tag{3}$$

Considering a patch of n by m pixels, this can be rewritten in matrix notation:

$$\hat{\underline{\mathrm{I}}}_L = M\mathbf{a} \tag{4}$$

$$M = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & y_1^2 & x_1 y_1 & x_1 y_1^2 & y_1 x_1^2 & x_1^3 & y_1^3 \\ 1 & x_2 & y_2 & x_2^2 & y_2^2 & x_2 y_2 & x_2 y_2^2 & y_2 x_2^2 & x_2^3 & y_2^3 \\ & & & & & \cdots & & & & \\ 1 & x_{nm} & y_{nm} & x_{nm}^2 & y_{nm}^2 & x_{nm} y_{nm} & x_{nm} y_{nm}^2 & y_{nm} x_{nm}^2 & x_{nm}^3 & y_{nm}^3 \end{bmatrix} \tag{5}$$

$$\mathbf{a} = \left[a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ a_{10}\right]' \tag{6}$$

Here, the number of rows in the matrix M corresponds to the number of pixels in the patch. If we select a patch $\underline{\mathrm{I}}_L$ of 5x5 pixels, the system is overdetermined and can be solved in the least-squares sense:

$$\mathbf{a} = \left(M'wM\right)^{-1} M'w \cdot \underline{\mathrm{I}}_L = W \cdot \underline{\mathrm{I}}_L \tag{7}$$

Here, w is a diagonal weighting matrix that can be used to give more weight on the central pixels. For example, a Gaussian weighting function could be used. In our approach, however, we use a novel polynomial weighting function [14].

This function gracefully ranges from zero to one over the compact support, unlike a Gaussian with infinite tails.

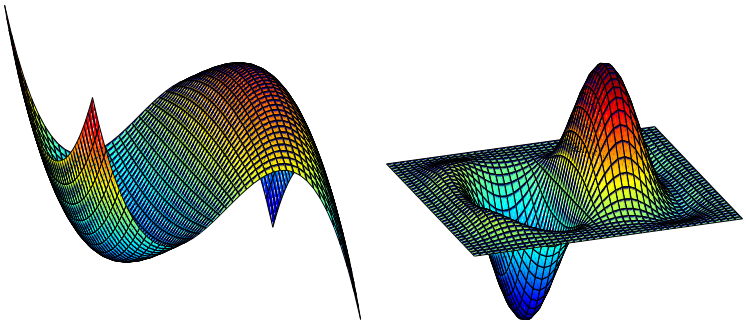
Examining Equation 7 makes obvious the fact that the matrix inversion has to be computed only once, after which the parameters can be obtained by a simple multiplication of the matrix W with the patch $\underline{\mathrm{I}}_L$. Note that this approach also makes prior image-smoothing obsolete. If we now define the central pixel of our patch to have the coordinates $x = 0, y = 0$, the gradients at this point are given by

$$\frac{\partial}{\partial x} I_L\left(0,0,t\right) = a_2, \frac{\partial}{\partial y} I_L\left(0,0,t\right) = a_3 \tag{8}$$

So, only the second and third row of the matrix W are of interest. If we rearrange them into a two-dimensional 5x5 array, the calculation of the gradient can again be calculated by a two-dimensional convolution with the filter kernel given by Equation 9 :

$$h_x = \begin{bmatrix} W(2,1) & W(2,2) & W(2,3) & W(2,4) & W(2,5) \\ W(2,6) & W(2,7) & W(2,8) & W(2,9) & W(2,10) \\ W(2,11) & W(2,12) & W(2,13) & W(2,14) & W(2,15) \\ W(2,16) & W(2,17) & W(2,18) & W(2,19) & W(2,20) \\ W(2,21) & W(2,22) & W(2,23) & W(2,24) & W(2,25) \end{bmatrix} = {h_y}' \tag{9}$$

Figure 1(a) and 1(b) illustrate the filter kernel for the uniformly weighted and non-uniformly weighted patch.



(a) No weighting (b) Polynomial weighting

Fig. 1 Filter kernel obtained by using a third order polynomial as surface approximation and (a) no weighting function w (b) a 3rd order polynomial weighting function

For the temporal derivative a simple finite difference was used. Note that the sampling time is defined to be unity, i.e. $\delta t = 1$, resulting in the gradient equation:

$$\frac{\partial}{\partial t} I\left(x,y,t\right) = I\left(x,y,t+\delta t\right) - I\left(x,y,t\right) \tag{10}$$

2.1.2 Lucas - Kanade Algorithm

Equation 1 gives one constraint equation for the three velocity components u, v and w. In order to make the problem solvable, further constraints have to be imposed. One possibility is to make the assumption that the flow is constant in a small area Ω around the point (x, y, z). Since there might be no velocity that exactly satisfies this assumption it is reasonable to find the velocity that minimizes the quadratic error function [12]:

$$E(u, v, w, t) = \sum_{\tilde{x},\tilde{y},\tilde{z}\in\Omega} g(\tilde{x}, \tilde{y}, \tilde{z}) \left[uI_x + vI_y + wI_z + I_t\right]^2 \tag{11}$$

Where $\tilde{x}, \tilde{y}, \tilde{z}$ correspond to the coordinates of the voxels at x, y, z and $g(\tilde{x}, \tilde{y}, \tilde{z})$ constitutes a weighting function around the point (x, y, z). Traditionally, a Gaussian function is used. In our approach, however, we use the novel polynomial weighting function as defined in [14] with a compact support. Note that this quadratic cost function can be minimized analytically, this involves the inversion of a 3×3 matrix.

2.1.3 Pyramidal Implementation & Iterative Optical Flow

A well established multi-scale technique to justify the assumption of linearity necessary for Equation 1 to hold true and to avoid the minimization algorithm from converging to a local minima used a coarse-to-fine resampling and warping to generate the flow field. Two techniques which were implemented in this work are the Gaussian pyramid approach and an iterative flow calculation [1]. We implemented a three-level pyramid and included a validation step in the iteration process. Before each iteration, the temporal derivative I_t of the image pair is saved. After one iteration is completed, the image is pre-warped using the calculated, temporary flow field. Now, the resulting derivative $\tilde{I}_t$ is calculated. At every voxel where the absolute value of the derivative has increased, i.e. the registration has worsened, the optical flow value is set to zero. This ensures that only flow vectors are accepted that increase similarity between images. To prevent a "choppy" flow field, a smoothing step convolving the image with the filter g_s is applied afterwards (see Section 2.2 for details).

2.1.4 Tikhonov Regularization

The assumption of small changes can be further exploited by using the Tikhonov Regularization: The general least-squares minimization tries to minimize

$$\hat{x} = \arg\min_x ||Ax - b||^2 \tag{12}$$

and the resulting solution is given by

$$\hat{x} = \left(A^T A\right)^{-1} A^T b \tag{13}$$

In order to give preference to a specific type of solution, an additional penalty term is added to the right hand side of Equation 12, leading to

$$\hat{x}_t = \arg\min_x ||Ax - b||^2 + ||\Gamma x||^2 \tag{14}$$

where Γ represents the Tikhonov matrix: This matrix can have any form in the most general case. Here, it is chosen to be a scalar multiple of the identity matrix:

$$\Gamma = \alpha I \tag{15}$$

This way, the norm of x is penalized, i.e. smaller solutions are preferred. This is reasonable since we are expecting only small changes at each iteration at each pyramid level. The solution to this minimization problem is given by

$$\hat{x}_t = \left(A^T A + \Gamma^T \Gamma\right)^{-1} A^T b = \left(A^T A + \alpha^2 I\right)^{-1} A^T b \tag{16}$$

2.1.5 Balanced Optical Flow

As described above, the calculation of spatial image gradients is carried out only in the image at time $t = 0$. In a perfect world, it would not make a difference if the flow was calculated from the image at $t = 0$ to the image at $t = \delta t$ or the other way around and then inverted. If the images contain noise, however, the results would be different. In [6] this is exploited directly by incorporating a consistency constraint in the deformation field calculation. In this work, a more simplistic approach is chosen: The optical flow is calculated by

$$[u, v, w] = X\left(I\left(t\right), I\left(t + \delta t\right)\right) \tag{17}$$

Where X denotes the optical flow calculation. Next, the reverse flow-field is calculated:

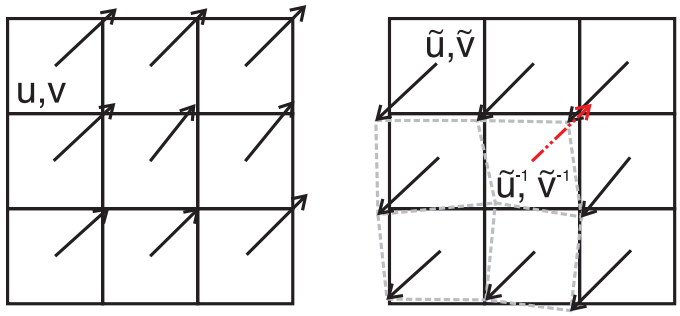
$$[\tilde{u}, \tilde{v}, \tilde{w}] = X\left(I\left(t + \delta t\right), I\left(t\right)\right) \tag{18}$$

See also Figure 2. This flow field is then inverted using the following method: Consider each point has a three dimensional coordinate, denoted x, y, z and three one-dimensional values indicating its origin, denoted ξ, ψ, ζ. If there was no deformation between the images (subscript zero), these values are the same as the respective coordinate:

$$\begin{aligned} \xi_0\left(x, y, z\right) &= x \\ \psi_0\left(x, y, z\right) &= y \\ \zeta_0\left(x, y, z\right) &= z \end{aligned} \tag{19}$$

If, however, there is a deformation between the images (subscript one), these values are different. To the coordinates, the optical flow values are added:

$$\begin{aligned} \xi_1\left(x + \tilde{u}\left(x, y, z\right)\delta t, y + \tilde{v}\left(x, y, z\right)\delta t, z + \tilde{w}\left(x, y, z\right)\delta t\right) &= x \\ \psi_1\left(x + \tilde{u}\left(x, y, z\right)\delta t, y + \tilde{v}\left(x, y, z\right)\delta t, z + \tilde{w}\left(x, y, z\right)\delta t\right) &= y \\ \zeta_1\left(x + \tilde{u}\left(x, y, z\right)\delta t, y + \tilde{v}\left(x, y, z\right)\delta t, z + \tilde{w}\left(x, y, z\right)\delta t\right) &= z \end{aligned} \tag{20}$$



(a) Forward Flow (b) Reverse Flow

Fig. 2 Two-dimensional illustration of forward optical flow field u, v, reverse optical flow field $\tilde{u}, \tilde{v}$ and inverted reverse optical flow field $\tilde{u}^{-1}, \tilde{v}^{-1}$

Note that $\delta t = 1$. To obtain an inversion of the reverse flow field, Equation 20 is interpolated on the regular grid, leading to

$$\begin{aligned} \tilde{u}^{-1}(x,y,z) &= \xi_1(x,y,z) - x \\ \tilde{v}^{-1}(x,y,z) &= \psi_1(x,y,z) - y \\ \tilde{w}^{-1}(x,y,z) &= \zeta_1(x,y,z) - z \end{aligned} \tag{21}$$

Note that since the input data to the interpolation is not placed on a regular grid, a three-dimensional implementation of the Delaunay triangulation, originally described in [7] is used. The results of Equation 21 can now be combined with the results from Equation 17 to create an averaged flow field to smoothen out noise:

$$\begin{aligned} u_{balanced} &= \frac{1}{2}\left(u + \tilde{u}^{-1}\right), v_{balanced} = \frac{1}{2}\left(v + \tilde{v}^{-1}\right), \\ w_{balanced} &= \frac{1}{2}\left(w + \tilde{w}^{-1}\right) \end{aligned} \tag{22}$$

Instead of including the inversion method at every step of the pyramidal approach, this method ensures a simple algorithm and allows parallel computation of forward- and backward-flow.

2.2 Implementation & Experimental Details

The Augmented Lucas-Kanade 3D-optical flow algorithm proposed in this paper was developed using MATLAB and C. Testing at two, three and four pyramid level resulted in the three pyramid levels with five iterations at each level, outperforming the other cases. The threshold for convergence was set to .1 pixel. The weighting function $g(\tilde{x}, \tilde{y}, \tilde{z})$ in Equation 11 was chosen as a $9 \times 9 \times 9$ polynomial window function of first order [14], the smoothing operator g_s in Section 2.1.3 was chosen as a $9 \times 9 \times 9$ polynomial window function

of third order. The polynomial derivative kernels h_p to acquire I_x, I_y, I_z, see Equation 1 and 2, was set to be of size $5 \times 5 \times 5$ (the patch size) using a third order polynomial. The weighting function w mentioned in Equation 7 was decided to be a polynomial windowing function of order 20. All convolution operations were preceded by a border padding operation mirroring the data at its borders. The regularization term α (see Equation 15) was chosen to be 100. All parameters were tuned using 300 landmark pairs per case provided for download at *http://www.dir-lab.com*. Note that the parameters of the algorithm were constant for all cases, no case-to-case tuning was performed. The calculations were performed on the Xeon X5460-server (32GB RAM, 8 x Intel(R) @ 3.16GHz with four cores each). For the first five cases, it took approximately 13 minutes per phase (i.e. $T00\% \rightarrow T10\%$) for the optical flow calculations and approximately 55 minutes for the optical flow and inversion calculations (i.e. $(T10\% \rightarrow T00\%)^{-1}$). For case six to ten, a coarse region-of-interest of 300 $\times$ 412 x Z voxels was selected, where Z corresponds to the number of voxels in $z-$ direction for the corresponding case. This bounding box was identical for all cases and data outside the ribcage that contained no useful information was cut out. This helped to reduce the data to be processed by a factor of approximately 2. After this, the corresponding computation times were approximately 40 / 73 minutes for case six to ten. The execution time of the loop processing Equation 16 could be reduced by a factor of approximately 70 by writing this function in C, this reduced the overall optical flow calculation time by a factor of 2. Note that every calculation used only one processor core at a time. To increase overall speed, up to ten optical flow calculations could be run in parallel. The bottleneck turned out to be the RAM since the inversion of the optical flow field using Delaunay Triangulation in the form of MATLAB's *TriScatteredInterp* function consumed tremendous amounts of memory. To improve this, the implementation of an approach like the one described in [5] can be used. For each case, a total of six CT images was processed, with $T00\%$ corresponding to the extreme inhale and $T50\%$ corresponding to the extreme exhale phase. The total deformation field $T00\% \rightarrow T50\%$ was obtained by sequentially synthesizing each phase-to-phase deformation field ($T00\% \rightarrow T10\%, T10\% \rightarrow T20\%, ..., T40\% \rightarrow T50\%$). The resulting displacement fields were sent to *The University of Texas M D Anderson Cancer Center* for evaluation[1].

3 Benchmark Data

Castillo et. al. published a paper [4] on the objective evaluation of DIR algorithms using a large dataset of expert determined landmarks: For their framework, they used five sets of thoracic 4DCT images that were acquired from patients undergoing treatment in their facility. The images were cropped and resampled in the xy plane (transversal plane) to contain the full rib cage at

[1] The author would like to acknowledge the help of Richard Castillo, MS who processed the result of our algorithm and provided feedback regarding its performance.

a resolution of 256 $\times$ 256 pixels. No alteration was done in the $z-$direction (Superior-Inferior (SI) direction). This leads to different voxel sizes and image dimensions. Using this data they manually identified more than 1100 landmark pairs in the extreme inhale ($T00\%$) and extreme exhale ($T50\%$) image of every set. This process was done by the primary reader, an "expert in thoracic imaging" using a software developed for this purpose. From these points, a subset of 200 points was randomly selected and re-registered by the primary reader to determine the *intra-observer error*. The same set of points was then re-registered by two secondary readers to quantify the *inter-observer error*. They then implemented two different DIR algorithms and performed a statistical analysis that showed that "large sets" of landmarks are indeed necessary to evaluate DIR accuracy with a certain accuracy. For these test cases and the optical flow algorithm they implemented, they find that 1000 pairs are necessary to evaluate the accuracy with 95% confidence.

In another paper by the same group in 2010 [2] , they extend the data by five more test cases. Here the images were not resampled and only 150 landmark pairs were registered (and re-registered), otherwise the procedure was the same. All ten test cases are publicly available at *http://www.dir-lab.com* after free registration. An overview of the data is given in table 3. Note that the displacement as well as the observer error is calculated as the euclidian distance. In the former case it is calculated using the points in the maximum inhale to the corresponding points in the maximum exhale phase. In the latter it is calculated between the data obtained by the first registration of the primary reader to the secondary registration of primary and secondary readers.

Case	Image Dimensions [pixel]	Voxel Dimensions [mm]	Landm. #	Displacement mean (sd) [mm]	Observer Error mean (sd) [mm]
1	256 x 256 x 94	0.97 x 0.97 x 2.5	1280	4.01 (2.91)	0.85 (1.24)
2	**256 x 256 x 112**	**1.16 x 1.16 x 2.5**	**1487**	**4.65 (4.09)**	**0.70 (0.99)**
3	256 x 256 x 104	1.15 x 1.15 x 2.5	1561	6.73 (4.21)	0.77 (1.01)
4	256 x 256 x 99	1.13 x 1.13 x 2.5	1166	9.42 (4.81)	1.13 (1.27)
5	256 x 256 x 106	1.10 x 1.10 x 2.5	1268	7.10 (5.14)	0.92 (1.16)
6	512 x 512 x 128	0.97 x 0.97 x 2.5	419	11.10 (6.98)	0.97 (1.38)
7	512 x 512 x 136	0.97 x 0.97 x 2.5	398	11.59 (7.87)	0.81 (1.32)
8	**512 x 512 x 128**	**0.97 x 0.97 x 2.5**	**476**	**15.16 (9.11)**	**1.03 (2.19)**
9	512 x 512 x 128	0.97 x 0.97 x 2.5	342	7.82 (3.99)	0.75 (1.09)
10	512 x 512 x 120	0.97 x 0.97 x 2.5	435	7.63 (6.54)	0.86 (1.45)

Table 1 Properties of the evaluation data as used in [2] and downloadable via *http://www.dir-lab.com*.

In the same paper, the group proposed two optical-flow based DIR algorithms: 4-Dimensional Local Trajectory Modeling (4DLTM) is an algorithm that makes use of all six images ($T00\%...T50\%$) simultaneously to calculate a polynomial trajectory for each voxel. For comparison they implement the same algorithm with the modification that it calculates a linear trajectory between two consecutive images. This technique is named "Component Phase-

to-Phase" (CPP) by the group. They show a clear advantage of 4DLTM over CPP in mean and standard deviation of 3D Euclidian error.

More algorithms were implemented by the same group in [3] and [10], the results are also available on-line and proved the superior accuracy of 4DLTM.

4 Results

Figures 3 and 4 illustrate the results for case 2 and 8 in comparison to other algorithms in terms of mean and standard deviation of the euclidian registration error as published in [4,2,3,10] and listed on *http://www.dir-lab.com* (as of 06/13/2012). The vertical bars include a marker which corresponds to the mean of the error and the length of the bar is proportional to the variance of the error. The magenta bar (leftmost bar) illustrates the error statistics of the human observer and the rest of the bars correspond to various algorithms used to estimate the motion of the landmarks. The black bar immediately to the right of the human observer statistics correspond to the results of the algorithm proposed in this paper. It is clear that for both cases (2 and 8), the algorithm presented in the paper outperform all the algorithms tested by scientists at MD. Anderson who manage the datasets and postprocess the results. Figure 5 and 6 shows the manually obtained displacement field and the error vector fields defined as pointing from the manually defined target location to the predicted location using our DIR-algorithm. These cases were chosen to

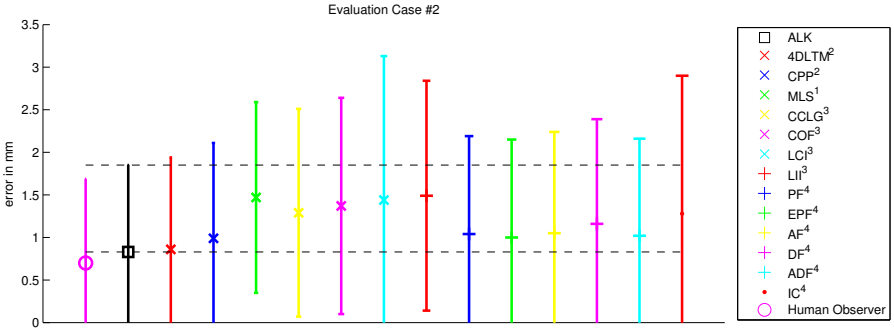


Fig. 3 Mean and standard deviation of the proposed optical flow algorithm in comparison to other algorithms for case 2: [1]described in [4], [2]described in [2], [3]described in [3], [4]described in [10]

illustrate the performance of ALK for small (case 2) as well as large deformations (case 8). The numerical results suggest that ALK has the capability of accurately registering 4DCT data, see Table 2.

It outperforms the CPP and 4DLTM approach presented in [2], although the differences to the latter are small: Evaluating all landmark pairs of all ten cases, the mean error of registration was determined to be 1.23 mm (with a

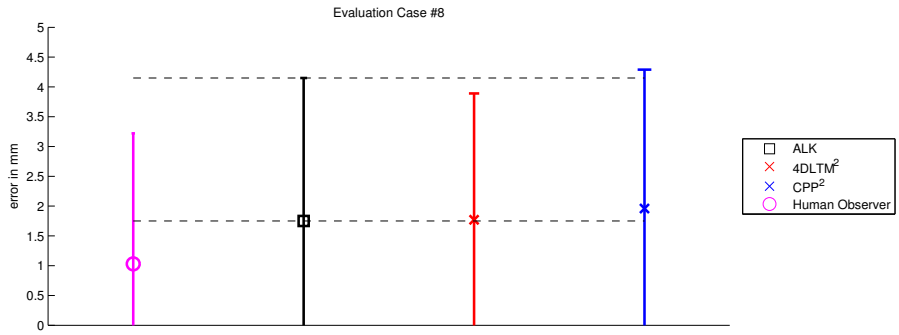


Fig. 4 Mean and standard deviation of the proposed optical flow algorithm in comparison to other algorithms for case 8: [2]described in [2]

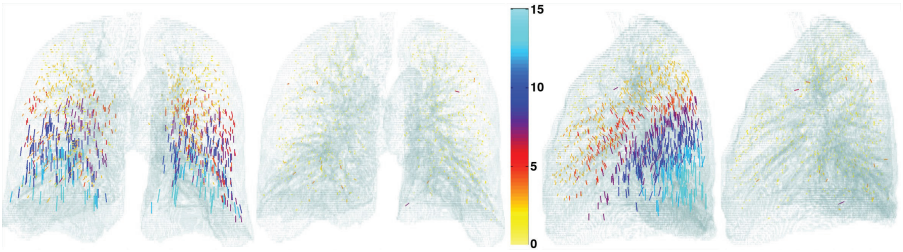


Fig. 5 Manually determined displacement vectors (left) and resulting registration error (right) for case 2 in the coronal and sagittal view of the surface rendering[2] of the lung in mm.

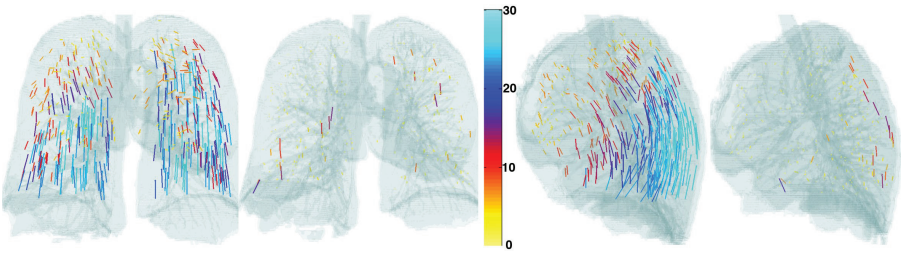


Fig. 6 Manually determined displacement vectors (left) and resulting registration error (right) for case 8 in the coronal and sagittal view of the surface rendering[2] of the lung in mm.

standard deviation of 1.42 mm). These values were 1.25 mm (1.43 mm) and 1.44 mm (1.54 mm) for the mean (standard deviation) registration error for 4DLTM and CPP, respectively. Inspecting each case individually, the algorithm proposed here has the lowest mean registration error in six and the lowest standard deviation in eight of the ten cases.

[2] Figure created using renderings provided by Richard Castillo, MS, The University of Texas M D Anderson Cancer Center

Case	ALK mean (mm)	4DLTM mean (mm)	CPP mean (mm)	ALK std (mm)	4DLTM std (mm)	CPP std (mm)
1	0.98	**0.97**	1.07	**1.00**	1.02	1.10
2	**0.83**	0.86	0.99	**1.02**	1.08	1.12
3	1.08	**1.01**	1.23	**1.15**	1.17	1.32
4	1.45	**1.40**	1.51	**1.53**	1.57	1.58
5	**1.55**	1.67	1.95	**1.73**	1.79	2.02
6	**1.52**	1.58	1.94	**1.28**	1.65	1.72
7	**1.29**	1.46	1.79	**1.22**	1.29	1.46
8	**1.75**	1.77	1.96	2.40	**2.12**	2.33
9	1.22	**1.19**	1.33	**1.07**	1.12	1.17
10	**1.47**	1.59	1.84	**1.68**	1.87	1.90

Table 2 Comparing registration error of ALK, 4DLTM and CPP in terms of mean and standard deviation error of Euclidian registration error, data for 4DLTM and CPP available at *http://www.dir-lab.com* (as of 06/13/2012).

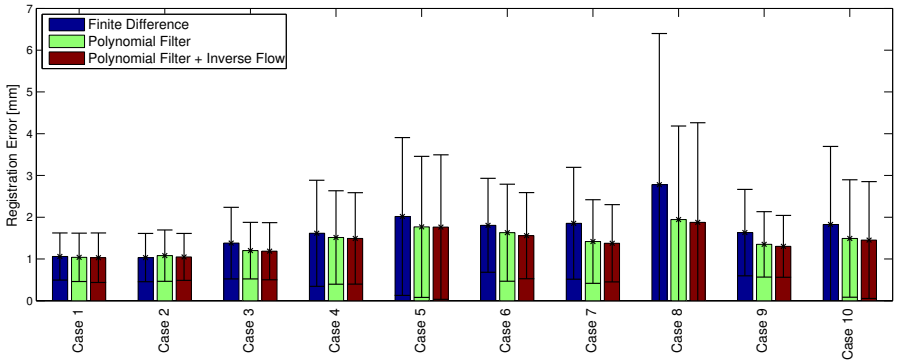


Fig. 7 Separate analysis of the contribution of polynomial gradient estimation and balanced optical flow.[3]

To analyze the contribution of each improvement separately, the 300 landmarks per case that are publicly available were used. A separate evaluation of the polynomial gradient calculation and the balanced Optical Flow can be found in figure 7. Since the problem is inherently ill-posed, Tikhonov regularization was used in all scenarios with $\alpha = const$. It can be seen that the robust gradient estimation contributes most to the increase in accuracy, especially in cases with large displacements like case 8. The balanced optical flow approach, however, only leads to smaller improvements which are most pronounced in cases with smaller displacements like case 2.

5 Conclusion & Future Work

[3] Note that this analysis was carried out using the publicly available dataset of 300 landmarks per case. Thus, the numerical values of "Polynomial Filter + Inverse Flow" differ from the values of "ALK" in Table 2 and Figures 3 & 4. Those were obtained in the final, external analysis using the (non-public) comprehensive dataset.

We have presented a novel optical flow algorithm and demonstrated its superior accuracy. Several advancements to the original Lucas/Kanade formulation have been made that allow the algorithm to outperform state-of-the-art algorithms while utilizing only two images at a time and keeping a simple, computationally efficient structure. The proposed algorithm can now be used to assist in the radiation therapy planning process. Due to its superior accuracy and robustness, an application of the algorithm in acute illness applications such as TAVI or the derivation of cardiac parameters for diagnosis seems reasonable and should be evaluated.

To further increase accuracy, we are currently exploring the possibility of using a L_1 cost function (as opposed to a quadratic, see Equation 11), which reduces the weight of the outlier errors and emphasizes minimizing the smaller errors. For a potential future real-time application it is conceivable implementing the whole algorithm in C / using a parallel architecture like GPU hardware seems reasonable.

Acknowledgements The authors would like to acknowledge the help of Richard Castillo, MS who processed the result of our algorithm and assessed its performance.

Financial Disclosure:

The authors have no financial relationship with the National Science Foundation which funded the research, the results of which are presented in this paper.

Conflict of Interest:

The authors declare that they have no conflict of interest.

References

1. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm. Intel Corporation (2001)
2. Castillo, E., Castillo, R., Martinez, J., Shenoy, M., Guerrero, T.: Four-dimensional deformable image registration using trajectory modeling. Physics in Medicine and Biology **55**(1), 305 (2010)
3. Castillo, E., Castillo, R., Zhang, Y., Guerrero, T.: Compressible image registration for thoracic computed tomography images. Journal of Medical and Biological Engineering **29**(5), 222–233 (2009)
4. Castillo, R., Castillo, E., Guerra, R., Johnson, V.E., McPhail, T., Garg, A.K., Guerrero, T.: A framework for evaluation of deformable image registration spatial accuracy using large landmark point sets. Physics in Medicine and Biology **54**, 1849 (2009)
5. Chen, M., Lu, W., Chen, Q., Ruchala, K.J., Olivera, G.H.: A simple fixed-point approach to invert a deformation field. Medical Physics **35**(1), 81 (2008)
6. Christensen, G., Johnson, H.: Consistent image registration. IEEE Transactions on Medical Imaging **20**(7), 568–582 (2001)
7. Delaunay, B.N.: Sur la sphère vide. Bulletin of Academy of Sciences of the USSR (6), 793–800 (1934)
8. Duan, Q., Herz, S., Ingrassia, C., Costa, K., Holmes, J., Laine, A., Angelini, E., Gerard, O., Homma, S.: Dynamic cardiac information from optical flow using four dimensional ultrasound. In: 27th Annual International Conference of the Engineering in Medicine and Biology Society, IEEE-EMBS 2005, vol. 4, pp. 4465–4468 (2005)

9. Gessat, M., Frauenfelder, T., Altwegg, L., Grünenfelder, J., Falk, V.: Transcatheter aortic valve implantation. role of imaging. Aswan Heart Centre Science & Practice Series **2011**(1), 3 (2011)
10. Gu, X., Pan, H., Liang, Y., Castillo, R., Yang, D., Choi, D., Castillo, E., Majumdar, A., Guerrero, T., Jiang, S.B.: Implementation and evaluation of various demons deformable image registration algorithms on a gpu. Physics in Medicine and Biology **55**(1), 207–219 (2010)
11. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17**(1-3), 185–203 (1981)
12. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence, vol. 3, pp. 674–679 (1981)
13. Murphy, K., van Ginneken, B., Reinhardt, J.M., Kabus, S., Ding, K., Deng, X., Cao, K., Du, K., Christensen, G.E., Garcia, V., Vercauteren, T., Ayache, N., Commowick, O., Malandain, G., Glocker, B., Paragios, N., Navab, N., Gorbunova, V., Sporring, J., de Bruijne, M., Han, X., Heinrich, M.P., Schnabel, J.a., Jenkinson, M., Lorenz, C., Modat, M., McClelland, J.R., Ourselin, S., Muenzing, S.E.a., Viergever, M.a., De Nigris, D., Collins, D.L., Arbel, T., Peroni, M., Li, R., Sharp, G.C., Schmidt-Richberg, A., Ehrhardt, J., Werner, R., Smeets, D., Loeckx, D., Song, G., Tustison, N., Avants, B., Gee, J.C., Staring, M., Klein, S., Stoel, B.C., Urschler, M., Werlberger, M., Vandemeulebroucke, J., Rit, S., Sarrut, D., Pluim, J.P.W.: Evaluation of registration methods on thoracic ct: the empire10 challenge. IEEE transactions on Medical Imaging **30**(11), 1901–1920 (2011)
14. Singla, P., Singh, T.: Desired order continuous polynomial time window functions for harmonic analysis. IEEE Transactions on Instrumentation and Measurement **59**(9), 2475–2481 (2010)
15. Veronesi, F., Corsi, C., Caiani, E.G., Sarti, A., Lamberti, C.: Tracking of left ventricular long axis from real-time three-dimensional echocardiography using optical flow techniques. IEEE Transactions on Information Technology in Biomedicine **10**(1), 174–181 (2006)
16. Yang, D., Lu, W., Low, D.A., Deasy, J.O., Hope, A.J., El Naqa, I.: 4d-ct motion estimation using deformable image registration and 5d respiratory motion modeling. Medical Physics **35**(10), 4577–4590 (2008)
17. Zientara, G.P., Saiviroonporn, P., Morrison, P.R., Fried, M.P., Hushek, S.G., Kikinis, R., Jolesz, F.a.: Mri monitoring of laser ablation using optical flow. Journal of Magnetic Resonance Imaging **8**(6), 1306–18 (1998)