Four-dimensional imaging in radiotherapy for lung cancer patients
Wolthaus, J.W.H.

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Deformable registration

Appendix of the article “Reconstruction of a time-averaged mid-position CT scan for radiotherapy planning of lung cancer patients using deformable registration”

J.W.H. Wolthaus
J.J. Sonke
M. van Herk
E.M.F. Damen

Medical Physics 35 - 2008
1. **Introduction**

This chapter, as intermezzo, describes the motion estimation algorithm used to deform the four-dimensional (4D) CT scans to the mid-position. Conventional optical flow methods based on the gradient of the image assume intensity conservation between frames. However, this assumption can be violated in real applications (e.g., registration of cone-beam CT to conventional CT). A remedy is to compute constraints on local motion based on the image-phase (image transitions from bright to dark and vice versa). Therefore, the motion (or displacement) estimation method is divided into two parts (Figure 4-2): An image-processing operation to compute the phase of the image; and the actual motion estimation procedure based on optical flow.

**Figure 4-1.** Motion estimating scheme used in this article. First, an image-processing operation is applied to the reference \((I_R)\) and floating \((I_F)\) scan for 9 directional filters, resulting the image phase \((\theta_R \text{ and } \theta_F)\) and magnitude \((M_R \text{ and } M_F)\) of the scan. Subsequently, an optical flow approach is used to compute the local motion vectors. CM denotes the confidence measure; \(c\) is the constraint vector (equation 3).
2. Quadrature image-processing filter

The optical flow motion estimation method (Section 4-2) uses the image-phase instead of the "original" image data. The phase of an image is related to transitions from bright to dark regions in the input image and vice versa (Figure 4-2) and is computed using a quadrature filter [1]. A quadrature filter is a complex band-pass filter that simultaneously computes the local magnitude (M) and phase (θ) of the image. The phase is wrapped between 0 and 2π. The magnitude is used in the motion estimation step to compute a confidence measure, which helps to detect false motion estimates (Section 4-3). The filters are tuned in a particular direction, analogous to gradient filters. Each filter is defined as zero over one half of the Fourier domain and non-zero over the other half. The complex output is computed in the spatial domain as the convolution of the quadrature filter and the input image. The band-pass part of the filter enhances (e.g., vessels) and suppresses (noise) features of the input image. Nine quadrature filters are used in different directions in three-dimensional (3D) space (using one-dimensional –1D– filters along the 3 axes and two-dimensional –2D– filters along the 6 diagonals) [2] to split features oriented in different directions into different filter outputs to reduce interference and duplication of features, resulting in 9 different filter outputs.

3. Optical flow motion estimation

In the optical flow method, a motion constraint is derived from the image-phase and is defined as an equation that the local motion vector (v) must satisfy; in other words, a change in position is only enforced by the change in phase (in time and space), i.e., conservation of image-phase density. Using the phase θ of the image, this (linear) flow equation is:

Figure 4-2. Quadrature filtering of 3D images. (a) Coronal slice of an original 3D scan, input image for the quadrature filter. Phase (b) and magnitude (c) output after filtering image with the quadrature filter tuned in the 45 degree direction from the cranio-caudal and left-right axes. Note that phase is wrapped between 0 and 2π.
\[
\frac{\partial \theta}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \theta}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial \theta}{\partial z} \frac{\partial z}{\partial t} = 0 \Rightarrow \frac{\partial \theta}{\partial x} \Delta x + \frac{\partial \theta}{\partial y} \Delta y + \frac{\partial \theta}{\partial z} \Delta z + \Delta \theta = 0
\]  \tag{1}

where \( \Delta x, \Delta y, \Delta z \) are the translations of the features in the image from frame to frame. In Figure 4-3, a graphical representation of equation (1) in one direction is shown. If the local change of phase is a 1st order derivative (i.e., phase is locally linear), the translation between the reference image and floating image \( (\Delta x) \) can be estimated by the gradient of the phase in the image \( (c_x = \partial \theta / \partial x) \) and the difference in phase between the reference and floating image \( (\partial \theta) \). Equation 1 can be rewritten (in matrix notation) as:

\[
c^T \begin{pmatrix} v_x \\ v_y \\ v_z \\ 1 \end{pmatrix} = 0, \text{ where } c = \begin{pmatrix} c_x & c_y & c_z & c_t \end{pmatrix}^T \text{ and } v = \begin{pmatrix} v_x & v_y & v_z \end{pmatrix}^T
\]  \tag{2}

where \( c_x, c_y \) and \( c_z \) define the change in phase over space; and \( c_t \) defines the change in phase over time.

For each voxel \((x)\) and for each quadrature filter output, the parameters of the motion constraint equation are computed:

\[
c_k(x) = \begin{pmatrix} c_{k,x} \\ c_{k,y} \\ c_{k,z} \\ c_{k,t} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \frac{\partial}{\partial x} (\theta_{Floating,k}(x) + \theta_{Ref,k}(x)) \\ \frac{1}{2} \frac{\partial}{\partial y} (\theta_{Floating,k}(x) + \theta_{Ref,k}(x)) \\ \frac{1}{2} \frac{\partial}{\partial z} (\theta_{Floating,k}(x) + \theta_{Ref,k}(x)) \\ \theta_{Floating,k}(x) - \theta_{Ref,k}(x) \end{pmatrix}
\]  \tag{3}

where \( \theta_{floating} \) and \( \theta_{ref} \) are the phase of the floating image and reference image, respectively. The index \( k \) denotes the different voxels or quadrature filter outputs. Note that in the first three elements, the phase is the average of the phase of the two frames to suppress variation in phase, which improves estimation accuracy; this averaging of the phases is allowed since the conservation of the image data (phase density) after translation implies equal phase. To find a (stable) solution for this equation, different quadrature filters \( (f_k) \) as well as multiple voxels in a local region can be combined into a single constraint equation at the center of the local region (control-point).

To clarify the method, in Figure 4-4a, a single motion constraint equation for a 2D case is plotted in motion vector space \((v_x, v_y)\), \( v_y = -(c_x/c_y)v_x + (-c_t/c_y) \). It is clear that there are multiple relationships between \( v_x \) and \( v_y \) that satisfy the (single) constraint line.
A single constraint line represents the motion of a plane that can only be detected perpendicular to the plane. Therefore, multiple constraints (i.e., from different quadrature filters and/or voxels) are necessary. For a local translation, these constraints (all representing the same motion) must intersect at a common point in the motion vector space (Figure 4-4b), giving the motion of the considered voxel or ROI. In theory, only three constraints are necessary; however, a total of only three directional quadrature filters \((x,y,z)\) can be insufficient. Estimation of motion of large (larger than the quadrature filter kernel size) one-direction-oriented structures (for example vessels) suffer from the aperture problem, i.e., when determining

\[
\Delta x = \frac{\Delta \theta}{\partial \theta / \partial x}
\]

(i.e., the linear equation). A one-dimensional graphical representation of the optical flow method using the phase of the image. “Ref” and “Floating” denote the phase \(\theta\) in the reference and floating image, respectively. The translation between the reference image and floating image \(\Delta x\) can be determined by the gradient of the phase \((\partial \theta / \partial x)\) and difference in phase between the two images \(\Delta \theta\).

\[
v = \left( \frac{c_1}{c_4} v_x - \frac{c_7}{c_4} \right)
\]

The motion \(v\) is somewhere along this line

\[
\begin{align*}
t &= -v_x v_y \\
n &= \frac{1}{2} (v_x v_y)^2 + \frac{1}{2} v_y^2
\end{align*}
\]

\[
\begin{align*}
\Delta x &= \frac{\Delta \theta}{\partial \theta / \partial x} \\
\Delta x &= \frac{\Delta \theta}{\partial \theta / \partial x}
\end{align*}
\]

Figure 4-3. A one-dimensional graphical representation of the optical flow method using the phase of the image. “Ref” and “Floating” denote the phase \(\theta\) in the reference and floating image, respectively. The translation between the reference image and floating image \(\Delta x\) can be determined by the gradient of the phase \((\partial \theta / \partial x)\) and difference in phase between the two images \(\Delta \theta\).

Figure 4-4. (a) A constraint line for a 2D case is plotted in a motion vector space \((v_x, v_y)\). There are multiple relations between \(v_x\) and \(v_y\) that satisfy the constraint line. Time \(t\) is oriented perpendicular to this line. (b) Multiple motion constraints \(c\) (for example from multiple quadrature filters or pixels in the neighborhood) in an image with a pure shift motion. In this case all corresponding constraints must intersect at a single common point in the motion vector space.
motion in a kernel size (small window or aperture), only motion perpendicular to the structure can be estimated. For this reason, extra filters in the diagonal directions are used to connect the three directions. Subsequently, these local constraint-equations \( (c) \) are converted to translations. Using the least-squares method (also known as minimum-norm constraint solution method) the local motion vector \( (v) \) is estimated that matches the local constraint-equation \( (c^Tv = 0) \); i.e., finding the intersecting point of the constraints in the motion vector space. This operation has an analytical solution (involving a matrix inversion), and does not require an optimization process [3].

4. **Confidence measure**

Some motion constraints \( c_k(x) \) are unreliable, e.g., these constraints correspond to small or low-intensity (weak) features (or features that exist only in one of the two images) and noise. To distinguish these constraints, a confidence measure (CM) is used as a weight factor for the constraints: \( c'_k(x) = CM_k \cdot c_k(x) \). The CM is a combination of several factors, which will be discussed briefly in this section. A more extensive and quantitative explanation is given by Hemmendorff et al. [3,4]. First, the magnitude of the quadrature filter output is used as a measure of the strength of the features in the images. Second, the method expects that a change of phase in space must be linearly compensated by a change of phase in time (equation (1)). A first order (linear) approximation of the changes of phase in the image is, therefore, required (spatial linearity). For most voxels, the quadrature filter (which also smoothes the image data) results in locally linear phase data (depending on the size of the relevant features [3,4]). Furthermore, the tissue motion (from frame to reference frame) is assumed to be small (time linearity). However, phase may not be completely linear everywhere. The error of the linear approximation can be computed and is integrated over the neighborhood to give a confidence measure for the local linearity. The third factor is based on the similarity of the gradient of the phase in the two images (phase conservation, i.e., to check if the gradient of the phase can be averaged; equation 3). Finally, phase singularities must be avoided. These singularities can be found by comparing the direction of the phase from the quadrature filter output to the direction of the filter (which must be similar for non-singular voxels).

5. **System properties**

To estimate large motions with high accuracy, a multi-scale (coarse-to-fine) approach was used (Figure 4-1). Motion resulting from a coarse scale can be used to transform the phase and magnitude images and improve the motion results in a finer scale. Scale is defined as the resolution of the original input image divided by 2 to the power of a certain level \( (2^2, 2^1, 2^0) \) for each direction, resulting in images of \( 64^3 \),
Deformable registration

75

then 128³ and finally 256³ voxel resolution for cubic images. The entire image is divided into a number of sub-volumes, with the center of each sub-volume as control-point. Each sub-volume is chosen to be 3³ voxels large for each scale. Within each sub-volume, the constraints are averaged (Section 4-2). After each iteration, the resulting motion at the control-points is used as input for the next iteration. Between the control-points, the motion is (tri-linear) interpolated to obtain a motion vector for each voxel. In the next iteration, the phase and magnitude data of the floating image (the reference image is fixed) are first deformed using the motion output from the previous iteration, then new constraints are derived (at a finer scale) and the remaining motion is computed. This remaining motion is (vector-based) added to the estimated motion of the previous iteration. To improve accuracy, more than one iteration is done within each scale. The number of iterations depends on the distances to measure between the frames and the image quality. Generally 3 iterations and 3 scales were used.

6. Data filtering and lung-tissue border issue

To convert the motion-model to translations using the least-squares method, a matrix inversion is used (for each control-point). However, matrix inversion is sensitive to singularities of the input data (here the constraint motion-model). To prevent singularities, smoothing was applied to the constraints over the control-points before using the least-squares method (Figure 4-1). A second smoothing filter operation was performed after solving the motion-model to suppress singular results as well as to regularize the motion vectors (preventing folding of the motion vectors; Figure 4-1). In addition, this second filter was used to suppress the registration of artifacts (which is related to non-physical motion). All filters were adjusted to the scale and iteration step.

Figure 4-5. Example of filtering the motion field (in the cranio-caudal direction) using a uniform blurring filter and using the adaptive filter (section 4.5).
A main issue in deformable registration of thorax images is the fact that the motion direction of lung tissue is different from the thoracic wall and/or mediastinum (sliding tissue; Section 5-2.4 and 5-4.2). Due to this discontinuity many registration algorithms often fail (i.e., do not register correctly) in the transition regions. To overcome this problem, we used an adaptive filter besides a standard Gaussian smoothing filter. The adaptive filter smoothes only those control-points within the kernel, which have similar density value in CT (Hounsfield units), conserving the motion direction of the center control-point in the kernel.

The adaptive filter uses the minimum-projection image of the 4D CT as a weight-factor image (for each voxel, the minimum value over the frames is selected). Within the kernel of the filter, only those voxels or control-points ($p_i$) are averaged that have weight-values ($w_i$ voxel values of the minimum-projection image, $w_0$ is center voxel of the kernel) in the same range, i.e., the difference between weight-values must be small to be taken into account. The resulting output value ($p'_i$) for each voxel or control-points is computed by:

$$p'_0 = \frac{\sum_{i \in \text{kernel}} \text{WF}_i p_i}{\sum_{i \in \text{kernel}} \text{WF}_i}, \text{ with } \text{WF}_i = \exp\left(-\alpha \|v_i - w_0\|\right)$$

The coefficient $\alpha$ is a “stiffness” factor, which specifies the range of weight-value differences that is taken into account ($\alpha = 0.01$ was used). This adaptive filter preserves sharp edges (lung-thoracic wall) and smoothes small variation of the input signal (here, the motion-model or motion). In Figure 4-5, an example of the adaptive filter output is shown.

The adaptive filter with larger kernel is applied first, and finally, to remove remaining small irregularities at edge structures in the image, a Gaussian filter with very small kernel is applied.

7. Computation time

Computation time depends on the size of the image (since it is voxel based) and the number of iterations in different image scales, but less on the number of control-points. The computation time of the image-processing part and optical flow part (using 3 iterations and 3 scales) were almost equal. Total computation time is typically 10 minutes for a set of two 256x256x100 images on a Pentium 4 – 3.0 Ghz processor. However, computation time can be decreased by applying parallel computing (the method can easily be split into segments to be used in parallel computing) and avoiding computation of the background region, which is the case in our current implementation.
References