OPTICAL FLOW ESTIMATION OF THE HEART MOTION USING LINE PROCESS

Hrvoje Bogunović, Sven Lončarić
Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, HR-10000 Zagreb, Croatia
email: hrvoje.bogunovic@fer.hr, sven.loncaric@fer.hr

1 Introduction

Useful information about the cardiac function can be extracted from the motion analysis of the beating heart. Very important aspect of cardiac analysis is motion estimation. Here we try to calculate optical flow from the sequence of MR images of the heart. We shall restrict to 2-D motion estimation i.e. we shall calculate optical flow between two slices at different times. Later research will be focused on applying the algorithm to full volumes of the heart thus calculating 3-D optical flow. Very interesting result was achieved by Song and Leahy [1]. They managed to calculate full 3D optical flow of the beating heart. Same was achieved by Gorce et al. [2]. These methods, which rely on work by Horn and Schunk [3], have the same problem as the original method, which is, the smoothing of the boundaries. Since in computer vision the same problem was successfully tackled by the introduction of line process, we wanted to see what would be the effect of it in the medical imaging. Towards the calculation of full 3-D optical flow, we first wanted to explore the 2-D paradigm. Major effect of line process is that it turns optical flow field from globally smooth to piecewise smooth.
\( L \) does not depend on that constraint.

\[
P(d|f, l) = P(d|F = f) = \exp\left(\frac{1}{2\sigma^2} \sum_i \sum_j (u(i, j)d_x(i, j) + v(i, j)d_y(i, j) + d(i, j))^2\right)
\]

On the other end we have a priori probability function. Markov Random Fields (MRF) framework is utilized for displaying prior knowledge. We turn \( F \) and \( L \) from RF to MRF by defining neighbourhoods and cliques. Thus we obtained coupled MRF. A 4-point neighborhood and cliques up to second order were used. Now we can express prior probability. We can also write \( P(F = f, L = l) = P(F = f|L = l)P(L = l) \). \( P(F = f|L = l) \) should present interaction between two fields. Typical expression for this is.

\[
P(f|l) = \exp\left(-\frac{1}{\beta_f} \sum_i \sum_j [(u_x(i, j)^2 + v_x(i, j)^2)(1 - l_v(i, j))] + [(u_y(i, j)^2 + v_y(i, j)^2)(1 - l_h(i, j))]\right)
\]

Our a priori knowledge consists of the fact that motion vectors are not necessarily smooth across the edges of the image. So line process is used to help us signaling potential discontinuity of the motion field. Additionally we need to make a penalty every time line process signals discontinuity because otherwise we would end up with every site of line process having value 1. To make things easier, motion discontinuity should only appear if there is a corresponding edge discontinuity.

As a prerequisite we should first find the spatial edges. That was done using Canny edge detector. If we want the penalty for creating discontinuity outside the corresponding intensity edge to be 10 times higher than if there exists intensity edge, we would use:

\[
P(l) = \exp\left(-\frac{1}{\beta_l} \sum_i \sum_j [(1 - edge_h(i, j)) \cdot 9 \cdot l_h(i, j) + l_h(i, j) + (1 - edge_v(i, j)) \cdot 9 \cdot l_v(i, j) + l_v(i, j)]\right)
\]

It is interesting to note that the output of the edge detector should have values located at the same positions as the line process i.e. midway between pixels. Normal edge detector gives output on a regular grid which corresponds to pixel sites. That is why we had to take orientation of the edge into account. Our implementation of Canny edge detector produced edge always on the brighter side of the real edge that lays between pixels. By moving in the opposite direction from the orientation of the edge by half a pixel, we can come to the actual position of the edge.

When we multiply all the factors, and if we want MAP estimate, we have to minimize the following energy function.

\[
U(f, l|d) = \sum_i \sum_j \frac{1}{2\sigma^2} (u(i, j)d_x(i, j) + v(i, j)d_y(i, j) + d(i, j))^2 + 2\frac{1}{\beta_f} [(u_x(i, j)^2 + v_x(i, j)^2)(1 - l_v(i, j))] + [(u_y(i, j)^2 + v_y(i, j)^2)(1 - l_h(i, j))] + \frac{1}{\beta_l} [l_h(i, j) + l_v(i, j)]
\]

We have three parameters in this equation. if we take

\[
\frac{1}{2\sigma^2} = 1; \lambda = \frac{1}{\beta_f}; \gamma = \frac{1}{\beta_l}
\]
then we have only two parameters. We have not found an
efficient way to estimate these parameters so they were cho-
sen ad hoc.

We can point out though, what is the threshold needed
for the discontinuity to appear. If discontinuity appears en-
ergy will rise by $\gamma$ otherwise it will rise by $\lambda(\nabla(\text{vel})^2)$. So the threshold required for the discontinuity to appear
will be $\sqrt{\frac{1}{\lambda}}$. If $\nabla(\text{vel}) \geq \sqrt{\frac{1}{\lambda}}$ discontinuity will appear
otherwise it will not.

3 Energy minimization

Minimization was performed with HCF (Highest Con-
fidence First) algorithm, first presented by Chou and
Brown [8]. The good thing about that algorithm is that it
is does not require accurate initialization of random fields.
The energy defined by equation 6 which we have to mini-
mize is non-convex. That means it has multiple minima and
any kind of steepest descent algorithm will get trapped into
local minimum too easily. HCF algorithm uses advanced
system for site visiting which enables to avoid most of the
local minimums.

At the beginning, sites of both fields ($F$ and $L$) pos-
sess label $l_0$, meaning uncommitted. Also sites from both
fields are treated equally. Every site is visited in order of
its stability. Stability presents measure of validity of their
current label. At the beginning when all sites are uncom-
mitted the stability shows whether the observation is strong
enough at that point for site to know its label. least sta-
ble sites will be visited early since their current label obvi-
ously is not valid. Also sites which do not possess a strong
observation will be visited the least, when labels of their
neighbours will be committed and decision can depend on
context.

In such a way HCF is able to evade some local min-
ima. Since it is deterministic results are obtained quite fast.
Compared to Simulated Annealing algorithm it is couple
of orders of magnitude faster with the results being quite
similar.
4 Results

The heart is presented as 16 slices with resolution 100x100. The whole sequence consists of 16 volumes in time. Results will be presented on slice 10 since it depicts the middle slice where left ventricle and myocardium are most visible. We needed three time slices to calculate temporal gradient since we used three-point central difference.

All images were smoothed with Gaussian filter before calculating spatio-temporal derivatives due to strong noise. Typical slices of heart are given in figure 1. The resulting optical flow for slice no. 10 is given in figure 2. Result given by edge detector for slice no. 10 is given in figure 3. Line process field is depicted on figure 4.

5 Conclusion

From the results it seems that line process can be valuable in cardiac analysis. Resulting optical flow field is indeed piecewise smooth instead of globally smooth. Additional advantage of line process is that it performs simultaneously with optical flow field. In this way both fields help each other to acquire correct values.

Experiments have shown encouraging results but still require further clinical validation.

The next logical step would be to fully implement 3-D optical flow estimator. The heart is 3-D object undergoing 3-D motion and results obtained in two dimensions cannot present accurate estimate needed for modeling of real heart motion.

6 Acknowledgements

The authors wish to thank Prof. James Duncan for providing us with the ECG-gated MR images of the heart.

References


