Algorithm for spatio-temporal heart segmentation

Zoran Majcenic and Sven Loncaric*

Faculty of Electrical Engineering and Computing, Unska 3, Zagreb, Croatia

ABSTRACT

Heart image analysis is a challenging and important process used for a range of purposes such as image based measurement, visualization, etc. The most important step in medical image analysis is segmentation. In this work we present an algorithm for CT heart image segmentation. The segmentation is based on two basic pieces of information, pixels brightness and motion. The motion information is gathered as the optical flow information. Such information is later used for definition of an energy function for image labeling. This energy function represents a Markov random field (MRF) posterior distribution function. The MAP estimation of the segmented image has been determined using the simulated annealing (SA) algorithm.

Keywords: simulated annealing, heart, segmentation, optical flow, motion estimation

1. INTRODUCTION

Segmentation of medical images is commonly required but challenging step in image analysis. Due to the nature of images produced by various medical imaging modalities (CT, MRI, ...) we have a difficult task of determining key regions and their properties. In addition, there is often a need to compute motion information from these images for various purposes.

There are a number of techniques that deal with a problem of segmentation such as clustering algorithms([1,2,3]), neural networks ([4,5,6]), morphological methods([7,8]) and knowledge-based methods([9]). A number of stochastic approaches to the image analysis is presented in literature([10,11]). There are also a number of methods dealing with motion estimation([12,13])

In this work, we present an algorithm for medical image segmentation reinforced by motion estimation information. The algorithm is applied to the problem of segmentation of heart CT image sets. The algorithm is divided in two parts. The first part performs computation of optical flow which provides an estimate of motion field. The detailed information about this step can be found in Section 2. The optical flow information is used in the second step to reinforce the quality of segmentation. The segmentation step is based on minimization of energy function that maps labeling information into a set of real numbers. The minimization procedure is based on a simulated annealing (SA) algorithm. SA is a stochastic method for function optimization. Details of SA algorithm can be found in Section 3. Results and discussion are presented in Section 4. Conclusion is placed under Section 5.

2. OPTICAL FLOW ESTIMATION

Optical flow algorithms provide estimations of the motion fields. The motion field is represented by the field of vectors that show the displacement of points in the optical field relative to the observer, i.e. the optical flow shows a velocity field of pixels in the image. There are a number of ways to compute the optical flow. The basic classification of methods is into gradient-based ([12]), correlation-based and energy-based methods([14]).

The method used in this work is based on energy function minimization. The algorithm is shown in Figure 1. The algorithm described above attempts to find a minimum energy state of the velocity field. The minimal energy state correlates to the optimal optical flow estimation. The minimization is based on steepest descent algorithm. The energy function E used in this algorithm consists of three terms.

$$E = E_1 + E_2 + E_3$$

The first term, E_1, takes into consideration a brightness difference between the first and the second picture used to extract the optical flow information.

* Send correspondence to Sven Loncaric: E-mail: sven.loncaric@fer.hr; www: http://ipg.zesoi.fer.hr

```
Choose an inital region size S, brightness and velocity deviation for region building and maximum movement value.
for i=1, ..., N_iter do
  Choose a random site in the image P = (x, y)
  Build a region r around point P with maximum size S
  Compute a r region's average velocity av
  Compute a r region's energy E_1 with current velocity av
  Let best_vel = av and lowest_E2 = MAX_FLOAT
  for i=-MAX_MOVEMENT, ..., MAX_MOVEMENT do
    for j=-MAX_MOVEMENT, ..., MAX_MOVEMENT do
      Let vel = (i, j)
      Compute regions r energy E_2 with velocity vel
      if E_2 < E_1 then
        best_vel = vel and lowest_E2 = E_2
      end if
    end for
  end for
  if lowest_E2 < E_1 then
    set velocity of all pixels in region r to best_vel
  end if
end for
Decrease the region size S
if S > MIN_REGION_SIZE then
  goto 2
end if
```

Figure 1. The algorithm for computation of optical flow.

$$E_1 = K_1 * average\_brightness(picture1, r, zero\_velocity) - average\_brightness(picture2, r, best\_vel)$$

The r region contains the information about region position and size. Combined with velocity parameter we can extract the information about difference in brightness of region in the first image and proposed position in the second image.

The second term, E_2, takes into consideration a standard deviation of region brightness as the parameter for region comparison. The more the region standard deviation differs from the standard deviation of the region in the second image the energy will be higher.

$$E_2 = K_2 * std\_dev(picture1, r, zero\_velocity) - std\_dev(picture2, r, best\_vel)$$

The third term, E_3, calculates the neighborhood influence. Instead of imposing a global smoothness constraint we try to achieve a smooth flow vector using the third term. The neighborhood influence raises the energy of the region if its velocity differs from its neighbors normalized by brightness difference.

The procedure for computing the neighborhood term E_3 is shown in Figure 2. In this algorithm information about average brightness and velocity is used. When the region is created it is also given its unique number and average brightness and velocity of its pixels. This information is stored in a separate field.

The algorithm allows larger regions to move more easily than small regions in early stages or computation of optical flow. This is to create some good neighborhood information for small regions. The algorithm or optical flow estimation works with dinamically created regions. Their size is limited by the parameter S. As seen in the algorithm the random point is selected and the surrounding region is built. Every such region gets its own unique identifier, its average brightness, and velocity. This data is written to a special field that is later used for the purpose of computing an energy of the neighborhood. Instead of working with single pixels we have decided to work with

Let $num = 0$ and $sum = 0$
for i=1, $\ldots$, $r.GetNumBorderPoints()$ **do**
  Let $p = r.BorderPoint[i]$
  Let $label$ =r's unique number
  Let $q = (p.x + 1, p.y)$
  if label of q != label **then**
    Let $num = num + 1$
    Let
$$sum = sum + \frac{|XVel(q) - XVel(p)| + |YVel(q) - YVel(p)|}{|Brightness(q) - Brightness(p)|}$$
  end if
  Let $q = (p.x - 1, p.y)$
  if label of q != label **then**
    Let $num = num + 1$
    Let
$$sum = sum + \frac{|XVel(q) - XVel(p)| + |YVel(q) - YVel(p)|}{|Brightness(q) - Brightness(p)|}$$
  end if
  Let $q = (p.x, p.y + 1)$
  if label of q != label **then**
    Let $num = num + 1$
    Let
$$sum = sum + \frac{|XVel(q) - XVel(p)| + |YVel(q) - YVel(p)|}{|Brightness(q) - Brightness(p)|}$$
  end if
  Let $q = (p.x, p.y - 1)$
  if label of q != label **then**
    Let $num = num + 1$
    Let
$$sum = sum + \frac{|XVel(q) - XVel(p)| + |YVel(q) - YVel(p)|}{|Brightness(q) - Brightness(p)|}$$
  end if
end for
if $num > 0$ **then**
  return $(K_3 * sum/(num * r.GetNumPoints())$
else
  return $(K_4/r.GetNumPoints())$
end if

Figure 2. The algorithm for computing the neighborhood term.

regions in order to achieve a smoothness in the optical flow field. Single pixels tend to cause a very diverse velocity field. By changing the maximum region size in the initial steps we allow the building of large regions. Because of the lack of neighborhood information in the initial steps of the algorithm, by working with small regions we would risk creating incorrect motion information. For large regions it is highly unlikely that their motion will be incorrectly computed. When we compute the movement for large regions we get a good neighborhood information. Also we must lower the maximum region size to be able to detect smaller changes and changes such as rotation. A rotation of a large region is detected as translations of multiple smaller regions. The regions are created dinamically using the algorithm shown in Figure 3.

The pixel inclusion condition is defined by pixel brightness and velocity values as well as the brightness and velocity of the initial pixel. If their difference is larger than some threshold the pixel is not included in the region. The performance of the optical flow algorithm based on the artificial example of moved image region is shown in Figure 4.

3. SEGMENTATION

The segmentation process can be defined as an image labeling problem.

3.1. Image Labeling Problem

A labeling problem consists of accurately labeling a set of objects with a group of given labels. A usual set of objects to be labeled in the segmentation process are image's pixels, however as explained in the Section 2 the object set in this work are dinamicaly created regions. A label set is defined as $L = \{l_1, l_2 \ldots, l_G\}$,[10] and represents the pattern classes in the image. The process and decision making, of labeling an object, varies depending on the method used for segmentation.

3.2. Segmentation algorithm

The segmentation algorithm tries to estimate the true object labeling denoted by $\mathbf{x}=\{x_1, x_2, \ldots, x_M\}$. The input image, that is to be segmented, can be viewed as one realization of the random field. This random field is denoted as $\mathbf{Y}= \{Y_1, Y_2 \ldots, Y_M\}$, where Y_t is the feature vector associated with the t-th pixel. A neighborhood information is also taken into consideration for the labeling process. The only problem is to define the neighborhood. Neighborhood influence enters the labeling problem through an Markov Random Field (MRF) model of the statistical dependence among the neighboring regions. Given a set of observed feature vectors (our image), $\mathbf{Y}=\mathbf{y}$, and the contextual information as an MRF, $P(\mathbf{X}=\mathbf{x})$, the problem lies in finding the 'optimal' estimate of the true labeling $\mathbf{x}$. The MAP (Maximum A-Posteriori) method estimates $\hat{\mathbf{x}}$ that maximizes the posterior probability of $\mathbf{X}=\hat{\mathbf{x}}$, given $\mathbf{Y}=\mathbf{y}$.

$$P(\mathbf{X} = \mathbf{x}|\mathbf{Y} = \mathbf{y}) = \frac{P(\mathbf{Y} = \mathbf{y}|\mathbf{X} = \mathbf{x})P(\mathbf{X} = \mathbf{x})}{P(\mathbf{Y} = \mathbf{y})}$$

Such a problem of finding an 'optimal' pixel labeling, defined above, is highly complicated. In order to find a MAP estimate we have to minimize a complicated function with the large number of variables. The number of variables is generally unknown since we cannot estimate the number of regions in the image because they are dinamically created and can overwrite one another. *Simulated annealing* is a function optimization method capable of finding the global minimum and avoiding the entrapment of the algorithm in the local minimum. SA doesn't require the smoothness of functions that is to be optimized, but the process of optimization is computationally demanding. In this paper we describe the application of SA algorithm to the problem of segmentation of heart CT images. The algorithm itself has been altered in a way to become more faster, but still keep the good perfomance when applied to a problem at hand. The algorithm is shown in Figure 5.

In the above algorithm, $P(.|.)$ is a conditional probability density function of the gray levels in the image for a specified label. The probability density function is approximated with a Gaussian function of the form

$$P(.|.) = e^{-\frac{(y_t - \mu_l)^2}{2\pi\sigma_l}}$$

where variance and mean values are extracted from the image histogram as described below in the paper. G is the number of labels in the image, and $U(.|.)$ is the energy function that defines the energy of a region. In the original

Let (x, y) be the starting point in the image
Let $retreat = false$ and p the initial point
if not $retreat$ **then**
  insert pixel position $(posx, posy)$ on a region pixel list
  insert pixel position $(posx, posy)$ on a stack
end if
Let $q = (p.x + 1, p.y)$
if q is not in region **then**
  if adding q wouldn't exceed region size limit S **then**
    if pixel at q qualifies with its features as a region pixel **then**
      goto 3
    end if
  end if
end if
Let $q = (p.x - 1, p.y)$
if q is not in region **then**
  if adding q wouldn't exceed region size limit S **then**
    if pixel at q qualifies with its features as a region pixel **then**
      goto 3
    end if
  end if
end if
Let $q = (p.x, p.y + 1)$
if q is not in region **then**
  if adding q wouldn't exceed region size limit S **then**
    if pixel at q qualifies with its features as a region pixel **then**
      goto 3
    end if
  end if
end if
Let $q = (p.x, p.y - 1)$
if q is not in region **then**
  if adding q wouldn't exceed region size limit S **then**
    if pixel at q qualifies with its features as a region pixel **then**
      goto 3
    end if
  end if
end if
Let $retreat = true$
Get pixel position from stack and put it in *posx* and *posy*
if stack is not empty **then**
  goto 3
end if

Figure 3. The region creation algorithm.

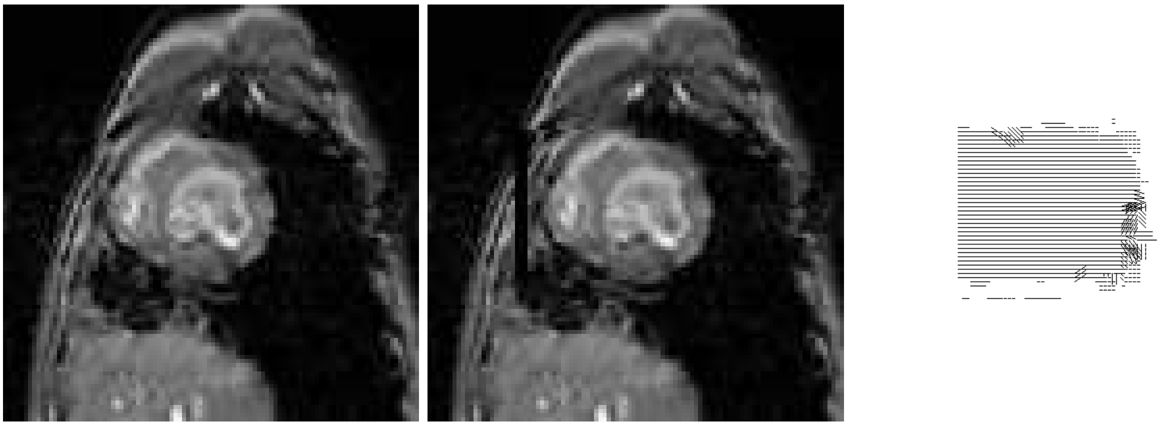

Figure 4. Optical flow estimation

1. Choose an initial temperature T.

2. Initialize $\hat{\mathbf{x}}$ by choosing x_t as the color $\hat{x}_t$ that maximizes $P(Y_t = y_t|X_t = x_t)$ for each region t.

3. Perturb x into z by randomly choosing region t and setting its label to a random value in the interval {0, 1, ..., G-1}. Let
$$\Delta = U(\hat{X}_t = \hat{x}_t|Y_t = y_t) - U(\hat{X}_t = z|Y_t = y_t)$$
If $\Delta > 0$ then replace $\hat{X}_t$ by z else replace $\hat{X}_t$ by z with probability $e^{\Delta/T}$.

4. Repeat step (3) N_{iter} times.

5. Replace T by $f(T)$ where f is the monotonically decreasing function.

6. Repeat steps (3)-(5) until frozen.

Figure 5. The algorithm for segmentation.

algorithm $U(.|.)$ would be the energy of an entire image. The energy U consists of two parts, the first part describes how well the regions label fits its gray value, and the second part describes how compatible the region is with its neighborhood in terms of its label. The first part can simply be calculated using the expression $1-P(.|.)$, where $P(.|.)$ is the probability density function. The neighborhood influence is calculated by checking the regions surrounding regions. If the neighbor region differs with its label from the currently selected region the energy U is increased. The optical flow information is taken into consideration in the way that it prevents the increase of the region's energy in case of label mismatch if the velocity fields of the two watched regions differ for a small percentage and their average brightness for a constant factor. The choice of a cooling function f, the initial temperature T, and the number of iterations is done experimentally. The temperature must be lowered slowly to prevent entrapment of the algorithm in the local minima. There is a number of ways to lower the temperature. A good choice would be

$$T_{k+1} = \frac{ln(1+k)}{ln(2+k}T_k$$

, where T_k is the kth execution of the outer loop of the algorithm. In this work the following function is used:

$$f(T) = \alpha * T$$

The distribution of gray values in the image is supposed to be normal. Mean and variance values for different region labels can be set manually or extracted from the image's histogram. One way to achieve extraction from the histogram is as follows.

1. Cross-corelating the histogram with an exponential function. By doing this the histogram becomes smooth allowing us to easily extract peaks and set means for all pixel labels.

2. Calculating cross-corelation factor between histogram at site μ_l and an exponential function with incresing variance. By assigning the variance that produces the largest value of the factor we acquire variance for all region labels.

4. RESULTS AND DISCUSSION

The input ECG-gated magnetic resonance images were 100 by 100 pixels grayscale mode. The image set consists of sixteen volumes per heart cycle. The initial stage of the algorithm performs the optical flow computation that is later used for the segmentation process by simulated annealing procedure. As can be seen in the optical flow estimation example the algorithm has difficulties on the edges of the moved region. Testing has shown that the noise present is MR images causes the errors in optical flow estimation. The segmentation step takes into consideration the motion information and boosts the segmentation quality compared to the regular SA algorithm based on the brightness alone. The drawback of this algorithm compared to other methods is its complexity. The segmentation result can be seen in Figure 6.

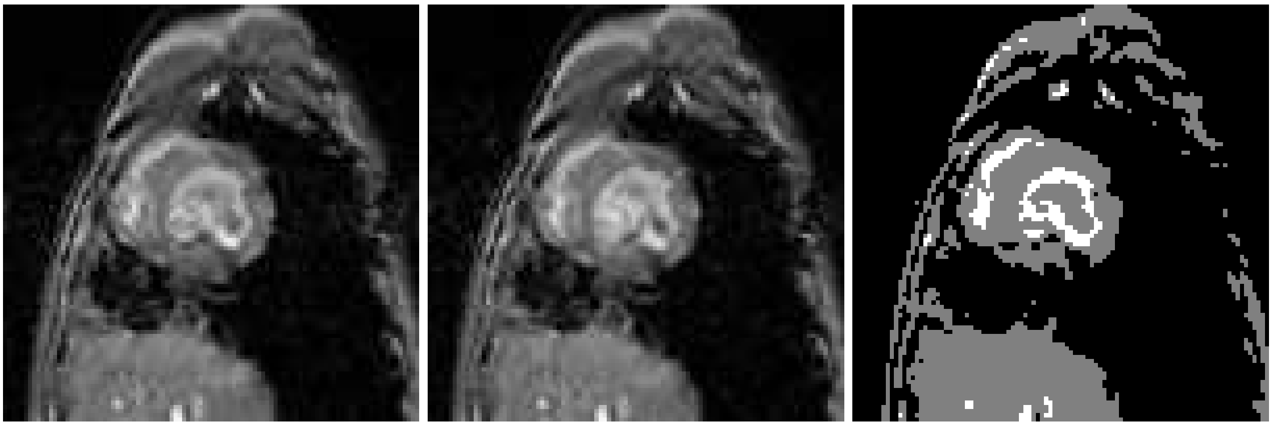

Figure 6. Input images and segmented output image.

5. CONCLUSION

The algorithm performs well and produces satisfactory segmentation results. The future work consists of expanding the segmentation to 3D images and using the motion information for computing the heart tissue stress.

ACKNOWLEDGMENT

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

REFERENCES

1. J. C. Bezdek, L. O. Hall, and L. P. Clarke, "Review of MR image segmentation techniques using pattern recognition," *Medical Physics* **20**, pp. 1033–1048, 1993.
2. S. Loncaric, D. Cosic, and A. P. Dhawan, "Segmentation of CT head images," in *Proceedings of the 10th International Symposium Computer Assisted Radiology*, p. 1012, Elsevier, Amsterdam, 1996.
3. S. Loncaric, D. Cosic, and A. P. Dhawan, "Hierarchical segmentation of CT images," in *Proceedings of the 18th Annual Int'l Conference of the IEEE EMBS*, IEEE, 1996. Amsterdam.
4. G. I. Chiou and J.-N. Hwang, "A neural network-based stochastic active model (NNS-SNAKE) for contour finding of distinct features," *IEEE Transactions on Image Processing* **4**, pp. 1407–1416, 1995.
5. M. Ozkan, B. Dawant, and R. J. Maciunas, "Neural-network-based segmentation of multi-modal medical images: A comparative and prospective study," *IEEE Transactions on Medical Imaging* **12**, pp. 534–544, 1993.
6. S. C. Amartur, D. Piraino, and Y. Takefuji, "Optimization neural networks for the segmentation of magnetic resonance images," *IEEE Transactions on Medical Imaging* **11**, pp. 215–220, 1992.

7. S. Loncaric, A. P. Dhawan, T. Brott, and J. Broderick, "3-D image analysis of intracerebral brain hemorrhage," *Computer Methods and Programs in Biomedicine* **46**, pp. 207–216, 1995.
8. J. G. Thomas, R. A. Petters, and P. Jeanty, "Automatic segmentation of ultrasound images using morphological operators," *IEEE Transactions on Medical Imaging* **10**, pp. 180–185, 1991.
9. C. Li, D. B. Goldgof, and L. O. Hall, "Knowledge-based classification and tissue labeling of MR images of human brain," *IEEE Transactions on Medical Imaging* **12**, pp. 740–750, 1993.
10. R. Dubes and A. Jain, "Random field models in image analysis," *Journal of Applied Statistics* **16**, pp. 131–164, 1989.
11. D.-W. Kim, G.-H. Lee, and S.-Y. Kim, "Stochastic segmentation of severely degraded images using gibbs random fields," *Optical Review* **3**, pp. 184–191, 1996.
12. B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence* **17**, pp. 185–203, 1981.
13. T. S. Denney and J. L. Prince, "Optimal brightness functions for optical flow estimation of deformable motion," *IEEE Transactions on Medical Imaging* **10**, pp. 187–199, 1991.
14. B. M. Ju S. and J. A., "Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 307–314, 1996.