Live Feature Tracking in Ultrasound Liver Sequences with Sparse Demons

Oudom Somphone, Stéphane Allaire, Benoit Mory, and Cécile Dufour

Medisys Lab, Philips Research

Abstract. We describe methods for feature tracking in temporal image sequences, based on a motion estimation framework called S*parse Demons*. It relies on a Gaussian-convolution model of the deformation field; this model is embedded in a variational formulation with a cost function defined on a finite number of points of interest. The resulting algorithm is fast and suitable for real-time, live feature tracking. Our methods are evaluated on the CLUST'14 database, consisting in 2D and 3D ultrasound liver sequences with landmarks or areas to be tracked.

1 Introduction

In this paper, we present methods for automatic tracking of anatomical features in the liver, in 2D and 3D ultrasound sequences. We apply our methods to the database of the MICCAI CLUST'14 challenge[1]. The features to track are landmarks and regions placed at locations of interest such as portal or hepatic veins bifurcations and tumors. The applicative scenario of the CLUST challenge is intervention and therapy in the liver under real-time ultrasound image guidance. More specifically, we address the issue of real-time compensation of the respiratory motion in the liver. To this end, we propose fast methods that do not rely on access to "images from the future". Moreover, we assume that:

- The ultrasound probe, be it 2D or 3D, does not drastically move during the acquisition.
- The acquisition frame rate of the ultrasound system is high, so that the motion between two consecutive frames is limited to a few millimeters.

For both the 2D and 3D datasets, we use a common motion estimation framework called *Sparse Demons*, described in section 2. Different strategies are adopted according to the objects to track – landmarks or regions, and the nature and quality of the datasets – 2D or 3D, with or without gain control. In 2D (sections 3 and 4), out-of-plane motion is expected, so that the method should be robust to appearing and disappearing features. In 3D (section 5), we designed anti-drift strategies based on the assumption that the respiratory motion is periodic. The results of our methods on the CLUST datasets were evaluated by the organization committee, based on a ground truth made of manual annotations.

[1] http://clust14.ethz.ch/

2 Sparse Demons

Feature tracking along a temporal sequence is regarded as a succession of reference-to-template motion estimation problems; at each incoming template frame the new positions of the tracked features are obtained in a causal manner by propagating the reference positions according to the estimated displacement. *Sparse Demons* is a variational approach to solve each reference-to-template problem. The key idea of our method is to find an optimal dense, non-rigid displacement field by minimizing an energy E defined only on a finite number of points of interest $\{\mathbf{x}_i \mid i \in \mathcal{P}\}$:

$$E = \sum_{i \in \mathcal{P}} \int_{\Omega} \delta(\mathbf{x} - \mathbf{x}_i)\ \mathcal{D}\Big[R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big]\ d\mathbf{x} \tag{1}$$

where R and T are the reference and template images respectively, Ω is the image domain and δ is the Dirac function. $\mathcal{D} : \mathbb{R} \to \mathbb{R}$ is a function that penalizes the dissimilarity between the reference and the transformed template; for instance, $\mathcal{D}(x) = x^2/2$ was used in [1]. As for the displacement field, we adopt a fluid-like regularization, which can be approximated by Gaussian filtering [2]; in this model, $\mathbf{u}$ is assumed to be the result of the convolution of an auxiliary field $\mathbf{v}$ with a Gaussian kernel ω_σ of scale σ:

$$\mathbf{u}(\mathbf{x}) = [\omega_\sigma * \mathbf{v}]\,(\mathbf{x}) = \int_{\Omega} \omega_\sigma(\mathbf{x} - \mathbf{y})\mathbf{v}(\mathbf{y})\ d\mathbf{y} \tag{2}$$

where $\omega_\sigma(\mathbf{x}) = \frac{1}{2\pi\sigma^2}\ e^{\frac{\|\mathbf{x}\|}{2\sigma^2}}$.

Minimizing E w.r.t. $\mathbf{v}$ is done by gradient descent; calculus of variations results in the following evolution equation:

$$\frac{\partial \mathbf{v}}{\partial t} = -\nabla_{\mathbf{v}} E = -\omega_\sigma * \left(\sum_{i \in \mathcal{P}} \delta_i \nabla_{\mathbf{u}} E \right) \tag{3}$$

where $\delta_i(\mathbf{x}) = \delta(\mathbf{x} - \mathbf{x}_i)$ and $\nabla_{\mathbf{u}} E$ is the dense gradient of E w.r.t. $\mathbf{u}$:

$$\nabla_{\mathbf{u}} E(\mathbf{x}) = -\mathcal{D}'\Big[R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big] \nabla T(\mathbf{x} + \mathbf{u}(\mathbf{x})) \tag{4}$$

The subsequent algorithm (see below) has similarities with the *demons* algorithm [3, 4]. Its computational complexity is however lower since image forces are not computed in the whole image domain but only at points $\mathbf{x}_i$.

The following sections describe how we used this general image pair registration framework to track features in a causal manner along 2D and 3D ultrasound liver sequences from the CLUST database. One of the values of this database is to provide long sequences over many breathing cycles, which are challenging for simple $(t-1)$-to-t estimation schemes as errors accumulate and cause inevitable drifting. For every subclass of the database, we investigated several anti-drift mechanisms in a live tracking scenario. In each section, we specify the point inputs, the dissimilarity measure, the respective expression of the subsequent dense energy gradient (4), and propose ways to avoid drifting.

Algorithm 1: Sparse Demons - Gradient Descent

Set $k = 0$ and $\mathbf{v}^0 = \mathbf{0}$
repeat
  Compute $\mathbf{u}^k = \omega_\sigma * \mathbf{v}^k$
  for all $\mathbf{x}_i$ **do**
    Interpolate $T(\mathbf{x} + \mathbf{u}^k(\mathbf{x}_i))$ and $\nabla T(\mathbf{x} + \mathbf{u}^k(\mathbf{x}_i))$
    Compute $\nabla_{\mathbf{u}^k} E(\mathbf{x}_i)$ according to (4)
  end
  Smooth the result to obtain the incremental update
  $\delta\mathbf{v}^k = -\omega_\sigma * \left(\sum_{i \in \mathcal{P}} \delta_i \nabla_{\mathbf{u}^k} E \right)$
  Update $\mathbf{v}^{k+1} = \mathbf{v}^k + \delta t.\delta\mathbf{v}^k$
  $k = k + 1$
until *steady state*;

3 Landmark Tracking in 2D

3.1 Method

In these datasets ("ETH" and "MED"), the gray values are consistent along the sequences. We therefore use the squared difference as dissimilarity measure, *i.e.* $\mathcal{D}(x) = x^2/2$, which yields the following energy gradient:

$$\nabla_{\mathbf{u}} E(\mathbf{x}) = -\Big[R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big] \nabla T(\mathbf{x} + \mathbf{u}(\mathbf{x})) \tag{5}$$

At each frame, the reference points of interest $\mathbf{x}_i$ are chosen in the neighbourhood of the landmarks, based on the amplitude of the image gradient: pixels on edges are selected and those in the flat regions are discarded (see Fig. 1). The neighbourhoods are squares of size Δp, centered on the landmarks.

The tracking consists of two phases:

Initial $(t-1)$-to-t tracking From frame 1 to τ (typically 100), the template is the incoming frame t and the reference is the previous frame $(t-1)$. During this phase, a mean *reference patch* is concurrently built around each initial landmark. Each reference patch consists in a small square image of size Δp, obtained by summing the patches centered on the corresponding landmark's positions at every frame (Fig. 2(b)).

1-to-t patch registration From frame $\tau+1$ onwards, for each incoming frame, we register *template patches* around the current landmark positions (Fig. 2(c)), towards the corresponding reference patches, which yields the position correction from frame $(t-1)$ to t. The aim of this scheme is to prevent drift by error accumulation, that inevitably occurs with a $(t-1)$-to-t scheme when the sequence is long. Moreover, to prevent one drifting landmark from influencing the others, we track each landmark independently.

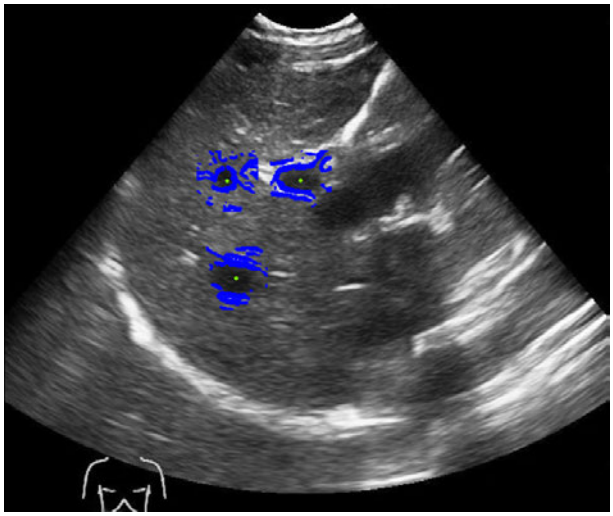
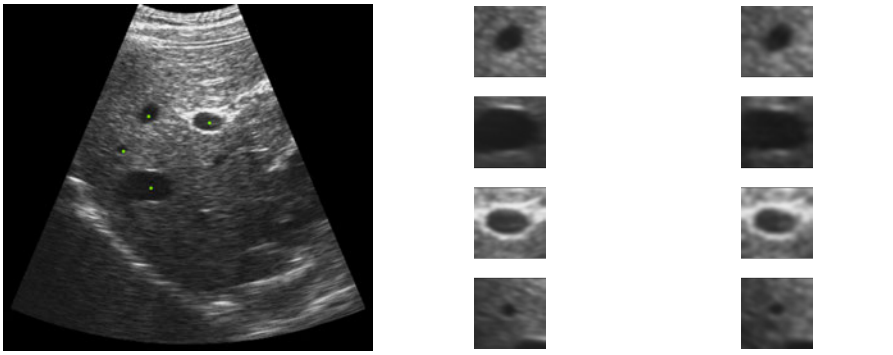


Fig. 1. Reference points (shown on a frame from the MED-13 sequence): the selected points of interest $\mathbf{x}_i$ (blue) are the pixels with larger image gradient in the neighbourhood of the current landmarks (green).



(a) Frame 19 of MED-03 (b) Reference patches (c) Template patches

Fig. 2. Reference and template patches in 2D landmark tracking.

3.2 Results

The test database for this part of the challenge contains 21 sequences of several thousands of frames (from 2424 to 14516) and 1 to 5 landmarks to be tracked. The parameters were tuned to $\sigma = 30$ mm, $\tau = 100$ frames and $\Delta p = 30$ pixels. The tracking error is defined as the Euclidean distance to the manually annotated ground truth. Table 1 below displays mean errors for all landmarks of all sequences for the "ETH" and "MED" datasets.

Dataset	MTE	SD	min	95%	max
ETH	0.98	1.14	0.00	2.45	24.16
MED	2.48	3.59	0.02	6.89	38.88
All2Dpoints	1.74	2.78	0.00	4.67	38.88

Table 1. Results for 2D landmark tracking. Mean Tracking Errors (MTE), Standard Deviations (SD), mininum errors (min), 95^{th} percentiles (95%) and maximum errors (max) are given in [mm].

4 Segmentation Tracking in 2D

4.1 Method

These sequences display some large intensity changes from one frame to the next and using the sum of squared difference as dissimilarity measure is not suitable. Instead, we minimize the entropy of the difference between the reference and the transformed template, which yields the energy:

$$E = -\int_{\mathbb{R}} p_{\mathbf{u}}(a) \log\left(p_{\mathbf{u}}(a)\right) da \tag{6}$$

$p_{\mathbf{u}}$ is the continuous Parzen estimate of the probability density function of the image difference over the the points of interest:

$$p_{\mathbf{u}}(a) = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \int_{\Omega} \delta(\mathbf{x} - \mathbf{x}_i)\ K\Big(R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x})) - a\Big) d\mathbf{x} \tag{7}$$

where K a smooth non-negative normalized Gaussian kernel. Calculus of variations results in the following dense energy gradient:

$$\nabla_{\mathbf{u}} E(\mathbf{x}) = -\left[K * \frac{p'_{\mathbf{u}}}{p_{\mathbf{u}}}\right]\Big(R(\mathbf{x}) - T(\mathbf{x} + \mathbf{u}(\mathbf{x}))\Big)\nabla T(\mathbf{x} + \mathbf{u}(\mathbf{x})) \tag{8}$$

Like in the previous section, the points of interest $\mathbf{x}_i$ are selected in the neighbourhood of the segmentation boundary, based on the amplitude of the image gradient. Since these sequences are short, the strategy to process the sequence is the $(t-1)$-to-t scheme.

4.2 Results

The test database for this part of the challenge contains 7 short sequences (from 51 to 105 frames) and 1 or 2 areas to be tracked. The Gaussian scale of the displacement was tuned to $\sigma = 30$ mm. The tracking is evaluated through the Dice coefficient between the tracked area and the manually segmented ground truth (Table 2).

Dataset	MDice	SD	min	max
OX-01_1	86.76	5.46	74.25	96.54
OX-02_1	85.66	4.99	73.25	97.74
OX-04_1	91.43	6.57	47.22	97.66
OX-05_1	79.93	6.71	61.79	95.84
OX-06_1	76.93	9.36	53.55	94.17
OX-07_1	89.71	4.39	72.41	97.74
OX-07_2	77.42	5.25	67.39	94.42
OX-08_1	88.75	2.83	79.19	98.27

Table 2. Results for 2D area tracking. Mean Dice (MDice), Standard Deviations (SD), minimum (min) and maximum (max) are given in [%].

5 Landmark Tracking in 3D

5.1 Method

In these datasets ("ICR", "SMT", and "EMC"), the gray values are consistent along most sequences. We therefore use the squared difference as dissimilarity measure (5), like in section 3. At each frame, the reference points of interest $\mathbf{x}_i$ are chosen in the neighbourhood of the landmarks on a square grid of size 80mm, regularly spaced by 10mm. Besides more sophisticated ultrasound shadow detectors, points are simply discarded in the darkest regions. The baseline is a $(t-1)$-to-t tracking, where all points of interest in all neighbourhoods are tracked together at once. To prevent drifting, an additional 1-to-t tracking is enabled if any of the two following triggers occurs:

Close histogram trigger From analyzing the differences between image histograms w.r.t. the reference difference level computed between the first two frames of the sequence, we can detect that the current incoming frame is close (not exceeding more than 20% of the reference difference level) in appearance to the initial frame where source annotations were given, which implies that a direct registration shall succeed.

Close location trigger From trajectory analysis, when landmarks positions get close (below 1.8 mm) to the positions of the source annotations given in the first frame, we also deem that a direct registration shall succeed. This trigger relies on the assumption that the probe is not moved during the sequence acquisition.

If triggered, the 1-to-t tracking overrides the $(t-1)$-to-t tracking.

5.2 Results

The database for the 3D landmark data class of the challenge contains 10 sequences of 54 to 159 frames and 1 to 4 landmarks to be tracked. The tracking error is defined as the Euclidean distance to the manually annotated ground truth. Table 3 below displays errors for all landmarks of all sequences per institution.

Dataset series	MTE	SD	min	95%	max
ICR	3.20	2.50	0.58	7.06	7.17
SMT	2.66	2.57	0.26	8.44	16.61
EMC	5.67	5.16	0.41	16.68	17.49
All3Dpoints	2.78	2.72	0.26	9.20	17.49

Table 3. Results for 3D landmark tracking. Mean Tracking Errors (MTE), Standard Deviations (SD), minimum errors (min), 95^{th} percentiles (95%) and maximum errors (max) are given in [mm].

6 Discussion and Conclusion

In terms of run-time estimation, it has to be noted that for both 2D and 3D, the tracking process is today somehow "irregular". Indeed, in 2D for instance, each given feature is first processed during a training period, then a different tracking can start. Likewise in 3D, the anti-drift strategy triggers additional motion estimations on an unsystematic basis. Also, the processing time depends on the number of features to be tracked. As the applicative scenario of the contest is not completely defined (total number of features to be tracked, adding features one by one as incoming frames flow, desired accuracy, etc.), the technical approach is not finalized. We report orders of magnitude of the computational load of these methods, taking into account that they have not been specifically optimized to that respect. On a multithreaded PC platform, the following frame rates are obtained with the described methods: around 40Hz in 2D, and around 10Hz in 3D.

The methods described in this paper are dedicated to live real time ultrasound. They are still under development, and the CLUST contest greatly helps in the design of the suitable approach and technologies. At testing and improving our methods on this challenging data, we stuck to the applicative scenario of a causal system. The progressive evolution of the tracked features been under scrutiny, and a final success criterion has been: whether the tracked features visually drift before the end of the sequence, and in that case whether the anti-drift mechanisms get them back on track w.r.t. the image content of the last frames. This is complementary to the criteria of overall average agreement or bounded disagreement highlighted by the quantitative results. Thus we have identified the drift to be the main issue of the tracking exercise. The 2D segmentation test set and the 3D test set probably do not contain a sufficient number of frames to validate that a live sequence tracking approach is reliable on the long term. In 2D however, the test sets seem long enough so as to establish a proof of concept.

References

1. Somphone, O., Craene, M.D., Ardon, R., Mory, B., Allain, P., Gao, H., D'hooge, J., Marchesseau, S., Sermesant, M., Delingette, H., Saloux, E.: Fast myocardial motion and strain estimation in 3D cardiac ultrasound with sparse demons. In: ISBI'13 Proceedings. (2013) 1182–1185
2. Mory, B., Somphone, O., Prevost, R., Ardon, R.: Real-time 3D image segmentation by user-constrained template deformation. In: MICCAI'12 Proceedings. (2012)
3. Thirion, J.P.: Image matching as a diffusion process: an analogy with Maxwell's demons. Medical Image Analysis **2**(3) (1998) 243–260
4. Mansi, T., Pennec, X., Sermesant, M., Delingette, H., Ayache, N.: iLogDemons: a demons-based registration algorithm for tracking incompressible elastic biological tissues. International Journal of Computer Vision **92**(1) (2011) 92–111