

# Kernel-based Tracking in Ultrasound Sequences of Liver

Tobias Benz<sup>1</sup>, Markus Kowarschik<sup>2,1</sup> and Nassir Navab<sup>1</sup>

<sup>1</sup> Computer Aided Medical Procedures, Technische Universität München, Germany

<sup>2</sup> Angiography & Interventional X-Ray Systems, Siemens AG, Healthcare Sector, Forchheim, Germany

**Abstract.** *Objective:* Object tracking in 2D ultrasound sequences of liver to infer real-time respiratory organ movement and offer motion compensation in image-guided abdominal interventions.

*Methods:* A kernel-based tracking algorithm that is adaptive to scale and orientation changes of the tracking target is applied to 54 vessel targets in 21 ultrasound sequences acquired from volunteers under free breathing. Tracking performance is evaluated based on manually annotated ground truth information.

*Results:* Tracking results show that the algorithm is able to track the assessed targets in a precise and robust manner in real-time performance. The overall mean tracking error is  $1.43 \pm 1.22$  mm.

## 1 Introduction

Object tracking in ultrasound (US) sequences of liver under respiratory motion is a challenging task with several applications in, for instance, motion compensation in abdominal interventions like needle biopsies, radio frequency ablations, and radiation therapy.

In this work, we present the application of a scale and orientation adaptive mean shift procedure to track vessel targets in long 2D US sequences acquired under free breathing.<sup>3</sup>

The mean shift procedure was first introduced by Fukunaga et al. [7] for data clustering. Cheng et al. [1] and Comaniciu et al. [3] later applied it to the task of visual object tracking. Recently, Ning et al. proposed modifications to make the mean shift tracker adaptive to orientation and scale [9]. In medical image processing, the mean shift algorithm was used for vessel segmentation in CT data [11] and for blood cell segmentation in images of blood smear [2]. In regard to tracking in US sequences, the mean shift was employed to myocardial border tracking [5]. An application to vessel tracking in US series of liver has, to our knowledge, not been presented before.

---

<sup>3</sup> The US image data was obtained from the "CLUST 2014 MICCAI Challenge on Liver Ultrasound Tracking" (<http://clust14.ethz.ch/>).

## 2 Methods

This section gives an introduction to kernel density estimation, the general mean shift algorithm for visual tracking, the adaptations to this algorithm as recently proposed by Ning et al. [9], and details on our implementation and modifications.

### 2.1 Kernel Density Estimation and the Mean Shift

Given a set of samples assumed to be drawn from some probability distribution, kernel density estimation (KDE) is a method to obtain a non-parametric estimate of the underlying probability density function. The kernel density estimator  $\hat{f}_h(x)$  at location  $x \in \mathbb{R}^D$  of a function  $f$  is

$$\hat{f}_h(x) = \frac{1}{Nh^D} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right), \quad (1)$$

where  $N$  is the number of samples  $x_i \in \mathbb{R}^D$  within the kernel  $K$  with window size  $h$ .

Using the kernel profile  $k$  of the radially symmetric kernel  $K$  which satisfies  $K(x) = c_k k(\|x\|^2)$  ( $c_k$  is a normalization factor), Equation (1) can be rewritten into

$$\hat{f}_h(x) = \frac{c_k}{Nh^D} \sum_{i=1}^N k\left(\left\|\frac{x-x_i}{h}\right\|^2\right). \quad (2)$$

The output of KDE is a function that is a smoothed representation of the given sample distribution (cf. Figure 1a) and can intuitively be understood as a generalization of weighted histograms (cf. Figure 1b).

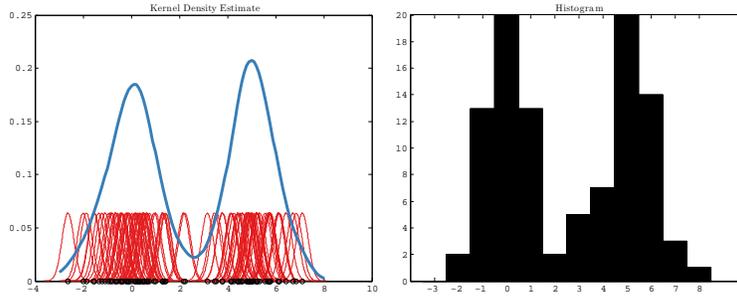
To find modes in a given KDE, mean shift procedures can be applied [3,7]. The mean shift is a gradient ascent on the gradient of the density estimate

$$\nabla \hat{f}_h(x) = 2 \frac{c_k}{Nh^{D+2}} \sum_{i=1}^N (x-x_i) k'\left(\left\|\frac{x-x_i}{h}\right\|^2\right), \quad (3)$$

$$\stackrel{g(x)=-k'(x)}{=} 2 \frac{c_k}{Nh^{D+2}} \sum_{i=1}^N x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) - 2 \frac{c_k}{Nh^{D+2}} \sum_{i=1}^N x g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad (4)$$

$$= 2 \frac{c_k}{Nh^{D+2}} \sum_{i=1}^N g\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \underbrace{\left[ \frac{\sum_{i=1}^N x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^N g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x \right]}_{=m_K(x)}. \quad (5)$$

The second term in Equation (5) is also referred to as the generalized mean shift vector  $m_K(x)$ . We can find modes in the gradient of the density estimate



(a) Gaussian KDE (blue) of 100 samples (black) drawn from two shifted normal distributions. Kernel weights for each sample point are indicated in red. (b) A histogram of the same set of samples as in (a).

Fig. 1: KDEs are a generalization of weighted histograms.

obtained by kernel  $K$  by iteratively shifting the center of  $K$  from an initial location by  $m_K(x)$ . When mode-seeking is applied to images, where pixels form a regular grid the generalized mean shift has to be extended to introduce the notion of pixel density. This can be achieved by employing a weighted mean shift, where each pixel location  $x_i$  is assigned a weight  $w_i$  derived from, for instance, the pixel's intensity.

## 2.2 Mean Shift for Tracking in Ultrasound Sequences

To apply the mean shift mode-seeking procedure to visual tracking the tracking target is selected in the first frame and represented in a suitable feature space to obtain a model of the target. For each subsequent frame a weight image is computed by assigning a weight to each pixel which depends on the probability of the pixel belonging to the target. On this weight image, which is also referred to as a target confidence map, the mean shift algorithm is initialized with the target location in the previous frame and an appropriate kernel size. After mean shift convergence the found mode is taken as the target location in the current frame.

For mean shift tracking in US sequences we used normalized weighted intensity histograms to represent the target model  $q = \{q_u\}_{u=1\dots m}$  and the target candidate model  $p(y) = \{p_u\}_{u=1\dots m}$  at location  $y$ , where  $m$  is the number of bins. The weights that determine the contributions of each pixel to a histogram bin  $u$  are based on a radially symmetric kernel  $K$ . For the target model location

$y = (0, 0)$  and kernel size  $h = 1$  is assumed by using normalized pixel locations  $x_i^*$ :

$$q_u = \frac{1}{C} \sum_{i=1}^N k(\|x_i^*\|^2) \underbrace{\delta[b(x_i^*) - u]}_{=1 \Leftrightarrow I_t(x_i) = u}, \quad (6)$$

where  $\delta$  is the Kronecker delta function,  $k$  the kernel profile of  $K$  and  $b(x)$  a function that maps the image intensity at location  $x$  to a bin number.  $C$  is the sum of the kernel weights at all locations such that the sum of all  $q_u$  is 1. For the target candidate model in the current frame the same model representation is used, but with the kernel of size  $h$  shifted to the current target location  $y$ :

$$p_u(y) = \frac{1}{C_h} \sum_{i=1}^N k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[b(x_i) - u], \quad (7)$$

where  $C_h$  is the sum of the kernel weights of all locations on the regular pixel lattice within the kernel with window size  $h$ .

Intuitively, the histograms  $q$  and  $p$  give the probability of a pixel's intensity belonging to the target and the target candidate model, respectively. The kernel assigns smaller weights to pixel locations farther away from the center. This increases robustness since pixels closer to the center are also closer to the target center and pixel locations close to the target center offer more reliable features due to, for instance, changes in the appearance of the target propagating from its boundaries towards the center.

Since the aim in the current frame is to find the target candidate model  $p(y)$  that best matches the target model  $q$ , a similarity metric is introduced next. We follow Comaniciu et al. [4] and use the discrete Bhattacharyya coefficient [8]  $\rho(y)$  to compare the target model  $q$  and the candidate model  $p(y)$  at location  $y$ :

$$\rho(y) = \rho[p(y), q] = \sum_{u=1}^m \sqrt{p_u(y)q_u}. \quad (8)$$

Intuitively, the Bhattacharyya coefficient is a measure for the amount of overlap between two sample distributions. Based on  $\rho(y)$ , a distance metric  $d$  between  $p(y)$  and  $q$  can be defined as  $d(y) = \sqrt{1 - \rho(y)}$ .

$$d(y) = \sqrt{1 - \rho(y)}. \quad (9)$$

Practically,  $d(y)$  is minimized by maximizing  $\rho(y)$ . In each frame  $t$ , the procedure to find the location  $\hat{y}$  that maximizes  $\rho(y)$  is started at location  $\hat{y}_0$ , which in the beginning is set to the position of the target in the previous frame  $\hat{y}_{t-1}$ . By linearization through a Taylor series expansion around  $y_0$ ,  $\rho(y)$  can be approximated as

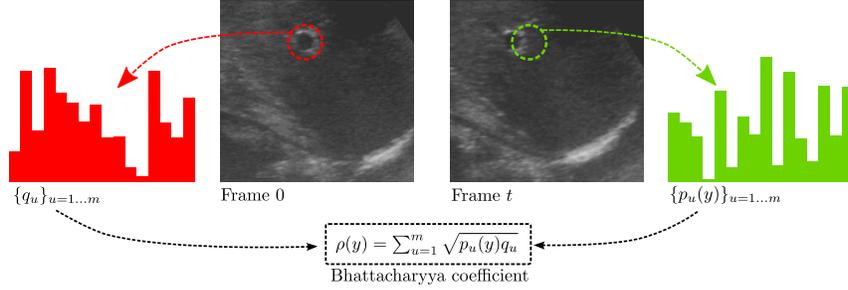


Fig. 2: Illustration of the set-up for model representation and comparison. The target model  $q$  at location 0 is computed in the first frame. In frame  $t$ , the target candidate model  $p(y)$  at location  $y$  is computed. The  $m$ -bin histograms are compared using the Bhattacharyya coefficient  $\rho$ . The mean shift procedure is used to find a  $\hat{y}_t$  that maximizes  $\rho$ .

$$\rho(y_0) \approx \frac{1}{2}\rho[p(y_0), q] + \frac{1}{2} \sum_{u=1}^m p_u(y) \sqrt{\frac{q_u}{p_u(y_0)}} \quad (10)$$

$$\stackrel{(7)}{=} \frac{1}{2}\rho[p(y_0), q] + \frac{1}{2} \sum_{u=1}^m \frac{1}{C_h} \underbrace{\sum_{i=1}^N k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \delta[b(x_i) - u]}_{p_u(y)} \sqrt{\frac{q}{p(y_0)}} \quad (11)$$

$$= \underbrace{\frac{1}{2}\rho[p(y_0), q]}_{\text{independent of } y} + \underbrace{\frac{1}{2C_h} \sum_{i=1}^N w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}_{\text{KDE obtained with kernel } K \text{ at location } y}, \quad (12)$$

where

$$w_i = \sum_{u=1}^m \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y)}}. \quad (13)$$

The first term in Eq. (12) is independent of  $y$ , whereas the second term is a KDE obtained using the kernel  $K$  at location  $y$  and weights  $w_i$ , which can be maximized using the mean shift algorithm (cf. Section 2.1). Maximizing this KDE means maximizing the Bhattacharyya coefficient, which finally leads to the minimization of the distance between  $p(y)$  and  $q$ .

The mean shift iteration step to move the kernel center position from  $\hat{y}_0$  to the new position  $\hat{y}_1$  is

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{\hat{y}_0 - x_i}{h}\right\|^2\right)}. \quad (14)$$

A favorable choice for the kernel  $K$  is the Epanechnikov kernel [6]

$$K_E(x) = \begin{cases} \frac{1}{2} \frac{D+2}{c_D} (1 - \|x\|^2) & \text{if } \|x\| < 1 \\ 0 & \text{else} \end{cases}, \quad (15)$$

where  $x \in \mathbb{R}^D$  and  $c_D$  is the volume of the  $D$ -dimensional unit sphere, e.g.  $c_d = 2$  for  $D = 1$  and  $c_d = \pi$  for  $D = 2$ . The Epanechnikov kernel minimizes the mean integrated squared error between the KDE and the true density [10] and, since the profile of  $K_E$  is half-triangular

$$k_E(x) = \begin{cases} \frac{1}{2} \frac{D+2}{c_D} (1 - x) & \text{if } x < 1 \\ 0 & \text{else} \end{cases}, \quad (16)$$

we see that  $g(x) = -k'(x) = 1$  and Equation (14) can be reduced to

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i}{\sum_{i=1}^{n_h} w_i}. \quad (17)$$

After each mean shift iteration, convergence is checked based on maximum number of iterations and minimum length of the mean shift vector. In case of convergence  $\hat{y}_t$  is set to  $\hat{y}_1$ , otherwise  $\hat{y}_0$  is set to  $\hat{y}_1$  and the mean shift procedure is repeated.

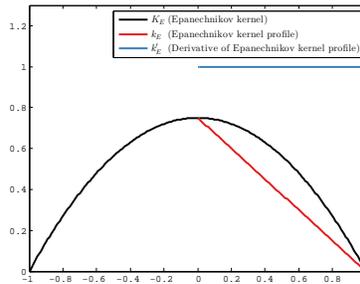


Fig. 3: The derivative of the Epanechnikov kernel is uniform.

### 2.3 Scale and Orientation Adaptive Mean Shift Tracking

In the original mean shift tracking algorithm the kernel window size and orientation remains fixed. This is unfavorable when tracking a target that changes its size and orientation over the course of the image sequence. We follow modifications proposed by Ning et al. [9] to make the procedure adaptive to scale and orientation. To this end, first the target’s scale is estimated, which is the area in the target search region occupied by the target. The estimated area is then used to adjust an ellipsoid target descriptor to match the current width, height, and orientation of the tracking target.

**Estimating the Target’s Scale** In each frame  $t$  the kernel center position is initialized with the target’s position in the previous frame  $\hat{y}_{t-1}$ . Also the kernel is slightly enlarged by a factor  $\Delta d$ , enabling the algorithm to capture a tracking target that increased in size since the last frame. Since the weight  $w_i$  for each pixel within the increased search region (cf. Equation (13)) gives the likelihood of the pixel being part of the target, the sum of all weights (i.e. the 0<sup>th</sup> order image moment of the search region in the weight image or target confidence map)

$$M_{00} = \sum_{i=1}^N w_i \quad (18)$$

is a good initial approximation of the area of the search region covered by the tracking target. However, if background features are present in the target search region, the weights of pixels within the search region that belong to the target are amplified. This is because the probability of target features in the target search area is decreased in the presence of background pixels, which, as per Equation (13) (the target candidate intensity distribution  $p(y)$  is in the denominator), increases the weights of target pixels. Therefore, the 0<sup>th</sup> order moment overestimates the size of the tracking target in case background features are present.

On the other hand, the Bhattacharyya coefficient between the target model  $q$  and the target candidate model  $p(y)$  is a measure for how many target and background features are in the current search region. Therefore Ning et al. [9] proposed to use the Bhattacharyya coefficient to adjust the 0<sup>th</sup> order moment approximation for the target scale. The estimated area is computed as

$$\hat{A} = \exp\left(\frac{\rho}{\sigma}\right) M_{00}, \quad (19)$$

where  $\sigma$  is a parameter that governs the magnitude of adjustment of the  $M_{00}$  estimate given a certain Bhattacharyya value. In the experiments described below  $\sigma$  was empirically set to 0.2.

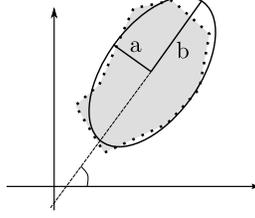


Fig. 4: Ellipsoid target descriptor.

**Estimating the Target's Orientation** For estimating the target's orientation an ellipsoid image descriptor (cf. Figure 4) is introduced which is defined by a covariance matrix based on the first and second order central image moments

$$\text{Cov} = \begin{pmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{pmatrix}, \quad (20)$$

where

$$\mu'_{pq} = \frac{\sum_{i=1}^N (x_{i,1} - \bar{x}_1)^p (x_{i,2} - \bar{x}_2)^q w_i}{\sum_{i=1}^N w_i}, \quad (21)$$

where  $(\bar{x}_1, \bar{x}_2)$  is the kernel center position. An orthogonal decomposition of Cov

$$\text{Cov} = U \times S \times U^T = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^T \quad (22)$$

yields the semi-minor and semi-major axis of the target descriptor as column vectors in  $U$  and the aspect ratio  $\frac{a}{b} = \frac{\lambda_1}{\lambda_2}$  through the singular values in  $S$ . Subsequently, a scaling factor  $k$  can be introduced such that

$$a = k\lambda_1 \quad (23)$$

$$b = k\lambda_2. \quad (24)$$

Introducing the previously estimated target area  $\hat{A}$  (cf. Section 2.3) and the general area formula for an ellipse, we can further derive

$$\hat{A} = \pi ab \quad (25)$$

$$= \pi(k\lambda_1)(k\lambda_2) \quad (26)$$

$$k = \sqrt{\frac{\hat{A}}{\pi\lambda_1\lambda_2}}, \quad (27)$$

which finally allows us to adjust the ellipsoid descriptor based on the estimated target scale:

$$\text{Cov} = U \times S \times U^T = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix} \times \begin{bmatrix} \frac{\hat{A}\lambda_1}{\pi\lambda_2} & 0 \\ 0 & \frac{\hat{A}\lambda_2}{\pi\lambda_1} \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}^T. \quad (28)$$

## 2.4 Tracking Failure Recovery

Tracking may be lost over the course of the US sequence due to, for instance, drastic change of appearance of the tracking target, too large target displacements between frames, or erroneous estimation of the target’s scale and orientation leading to a disadvantageous search area. Based on the assumed periodicity in the motion of liver vessels induced by respiration we integrated strategies to detect frames in which tracking performance is problematical and to recover from these situations. A first check is based on the analysis of the Bhattacharyya coefficient  $\rho$ . If it drops below 0.8 we discard the found target position and instead use the target position from the previous frame. Furthermore, the search region is reset to the one set by the user in the first frame. If this check triggers twice in two subsequent frames, the target position is reset to the centroid of the search area selected in the first frame. A second check is based on the analysis of the estimated target size. If the target search area in the current frame is found to be larger than three times the initial target’s size, the search area and its position is reset to the one set in the first frame. These failure recovery strategies are rather crude but were found to only be triggered in rare situation where tracking would otherwise fail completely.

## 3 Results

The scale and orientation adaptive mean shift procedure was applied to track 54 vessel targets in 21 2D US sequences of liver acquired from volunteers under free breathing. In total, the sequences comprised 91619 frames. The overall mean tracking error (MTE) was 1.43 mm with a standard deviation (SD) of 1.22 mm and 95th-percentile 3.67 mm. The minimum tracking error over all frames and tracking targets was 0.01 mm, the maximum tracking error 16.01 mm. The algorithm was developed in MATLAB Release 2013b, and the experiments were conducted on a machine equipped with an Intel i5-3320M processor at 2.6 GHz clock speed and 8 GB RAM. Tracking speed using this hardware set up was about 20 Hz. Table 1 gives an overview of the data set and the results obtained. Fig. 5 gives a visual impression of the tracking of three vessels in series MED-02.

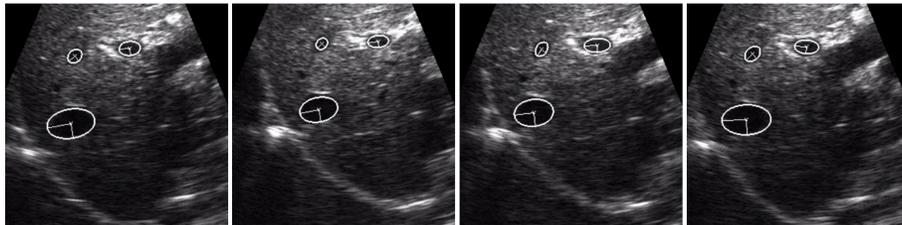


Fig. 5: Ellipsoid target descriptors of three tracking targets overlaid on four frames of the US sequence MED-02. <sup>4</sup>

Sequence information						Results				
Sequence	Targets	Frames	Resolution [mm/px]	Probe freq. [Hz]	FPS [Hz]	MTE [mm]	SD [mm]	95% [mm]	Min [mm]	Max [mm]
ETH-01	1	14 516	0.71	2.22	25	2.47	1.29	4.15	0.16	11.14
ETH-02	1	5244	0.40	2.00	16	0.60	0.38	1.26	0.04	2.64
ETH-03	3	5578	0.36	1.82	17	1.34	0.69	2.32	0.07	10.34
ETH-04	1	2620	0.40	2.22	15	1.05	0.80	2.10	0.05	7.33
ETH-06	2	5586	0.37	1.82	17	2.67	1.17	4.65	0.04	8.21
ETH-07	1	4588	0.28	2.22	14	0.85	0.54	1.95	0.01	3.21
ETH-08	2	5574	0.36	1.82	17	1.53	0.54	2.56	0.06	4.89
ETH-09	2	5247	0.40	1.82	16	0.85	0.46	1.67	0.01	5.89
ETH-10	4	4587	0.40	1.82	15	0.83	1.05	1.72	0.01	16.01
All ETH sequences						<b>1.46</b>	<b>1.31</b>	<b>3.77</b>	<b>0.01</b>	<b>16.01</b>
MED-01	3	2470	0.41	5.50	20	0.67	0.49	1.60	0.01	5.07
MED-02	3	2478	0.41	5.50	20	1.04	0.67	2.48	0.04	6.29
MED-03	4	2456	0.41	5.50	20	1.17	0.66	2.43	0.04	4.31
MED-05	3	2458	0.41	5.50	20	1.17	0.65	2.31	0.07	4.33
MED-06	3	2443	0.41	5.50	20	1.84	0.94	3.54	0.10	5.89
MED-07	3	2450	0.41	5.50	20	1.52	0.88	3.15	0.04	6.72
MED-08	2	2442	0.41	5.50	20	1.46	0.81	2.89	0.05	4.76
MED-09	5	2436	0.41	5.50	20	1.29	0.78	2.90	0.05	10.73
MED-10	4	2427	0.41	5.50	20	1.79	1.20	4.16	0.03	13.02
MED-13	3	3304	0.35	4.00	11	1.21	0.70	2.48	0.03	6.32
MED-14	3	3304	0.35	4.00	11	1.73	0.98	3.51	0.05	8.23
MED-15	1	3304	0.35	4.00	11	2.62	1.36	5.10	0.13	6.50
All MED sequences						<b>1.40</b>	<b>1.13</b>	<b>3.49</b>	<b>0.01</b>	<b>13.02</b>
All sequences						<b>1.43</b>	<b>1.22</b>	<b>3.67</b>	<b>0.01</b>	<b>16.01</b>

Table 1: Overview of the data set comprising 54 vessel targets in 21 US sequences. The table gives the sequence identity, the number of targets, the image resolution, the probe frequency, the sampling frequency (FPS), the mean tracking error (MTE), the standard deviation of the tracking error (SD), the minimum tracking error (Min), and the maximum tracking error (Max).

## 4 Conclusion

Tracking of vessel targets in 2D US series of liver under free breathing using a scale and orientation adaptive kernel-based tracking algorithm is feasible, fast, robust and precise. For future work the incorporation of a target descriptor that is adaptive to the outline of the tracking target seems worthwhile. By suggestion of one reviewer of an initial draft of this article, we will also look into making

<sup>4</sup> Link to video: <http://campar.in.tum.de/files/benz/CLUST2014/MED-02.webm>

the histogram representations adaptive to global illumination changes. The application to native 3D US is also desirable. Furthermore, other feature spaces for model representation could be investigated with a focus on, for instance, gradient-based descriptors and joint-histograms. Finally, a more sophisticated failure recovery strategy based on, for instance, a collection of keyframes or predictive motion regularization could be advantageous.

## References

1. Cheng, Y.: Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 17(8), 790–799 (1995)
2. Comaniciu, D., Meer, P.: Cell image segmentation for diagnostic pathology. In: *Advanced algorithmic approaches to medical image segmentation*, pp. 541–558. Springer (2002)
3. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24(5), 603–619 (2002)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 25(5), 564–577 (2003)
5. Comaniciu, D., Zhou, X.S., Krishnan, S.: Robust real-time myocardial border tracking for echocardiography: an information fusion approach. *Medical Imaging, IEEE Transactions on* 23(7), 849–860 (2004)
6. Epanechnikov, V.A.: Non-parametric estimation of a multivariate probability density. *Theory of Probability & Its Applications* 14(1), 153–158 (1969)
7. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. *Information Theory, IEEE Transactions on* 21(1), 32–40 (1975)
8. Kailath, T.: The divergence and Bhattacharyya distance measures in signal selection. *Communication Technology, IEEE Transactions on* 15(1), 52–60 (1967)
9. Ning, J., Zhang, L., Zhang, D., Wu, C.: Scale and orientation adaptive mean shift tracking. *Computer Vision, IET* 6(1), 52–61 (2012)
10. Scott, D.W.: *Multivariate density estimation: theory, practice, and visualization*, vol. 383. John Wiley & Sons (2009)
11. Tek, H., Comaniciu, D., Williams, J.P.: Vessel detection by mean shift based ray propagation. In: *Mathematical Methods in Biomedical Image Analysis, 2001. MM-BIA 2001. IEEE Workshop on*. pp. 228–235. IEEE (2001)