

LV - Left Ventricle  
ST - Speckle Tracking  
PCA - Principal component analysis  
ICA - Independent component analysis  
PyrLK - Pyramid Lucas-Kanade

## 1 Introduction

## 2 Method

Contrary to the commonly-used speckle tracking methods, our approach employs several principal differences. We are using optical flow estimation for tracking the points of interest instead of correlation. However, the estimation of speckle movement is local only. The usage of generative probability shape model is used in order to preserve valid shape estimation of tracked LV.

## 3 Optical flow estimation

Our experiments shown that the Pyramid Lucas-Kanade (PyrLK) method [4] is a good choice for speckle motion estimation. The goal of the tracking is to find for given point  $\mathbf{u} = (u_x, u_y)$  in image  $I$  corresponding point  $\mathbf{v} = \mathbf{u} + \mathbf{d}$  in image  $J$ . There is the assumption that the neighborhood of the point  $\mathbf{u}$  is similar to the neighborhood of the point  $\mathbf{v}$ . The vector  $\mathbf{d}$  is referred as the image velocity or the optical flow at  $\mathbf{u}$ . It is defined as the vector that minimizes the function:

$$\epsilon(\mathbf{d}) = (d_x, d_y) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} (I(x, y) - J(x + d_x, y + d_y))^2 \quad (3.1)$$

The  $w_x$  and  $w_y$  define integration (neighborhood) window of size  $(2w_x + 1) \times (2w_y + 1)$ . Their typical values for are 2,3,4,5,6,7 and usually  $w = w_x = w_y$ .

There is a trade-off between local accuracy and robustness when choosing the integration window size. Smaller window size is able to capture even tiny motion. On the other hand, the tracking is lost when the movement exceeds the search window. Contrary, the large window size lacks accuracy. Therefore, an involvement of the image pyramid representation has been proposed in [1].

The image pyramid consists of the original image  $I^0$  of size  $n_x^0 \times n_y^0$  and  $L_m$  levels – usually 3. Each image  $I^L$  at level  $L = 1, 2, \dots, L_m$  is resized, such its size is recursively set to  $\frac{1}{2}n_x^{L-1} \times \frac{1}{2}n_y^{L-1}$ . In the image pyramid perspective, the point  $\mathbf{u}^L$  is computed from the point in base image  $\mathbf{u} = \mathbf{u}^0$ :

$$\mathbf{u}^L = \frac{\mathbf{u}}{2^L} \quad (3.2)$$

The pyramidal tracking starts at the highest level image  $L_m$ . This will produce the optical flow estimation guess  $\mathbf{g}^{L_m-1}$  that is propagated to the next pyramid level  $L_{m-1}$  where the more correct flow estimation is computed. This is repeated until the base pyramid level is reached. The general equation for computing the optical flow at level  $L$  is:

$$\epsilon(\mathbf{d}^L) = \sum_{x=u_x-w_x}^{u_x+w_x} \sum_{y=u_y-w_y}^{u_y+w_y} \left( I^L(x, y) - J^L(x + g_x^L + d_x^L, y + g_y^L + d_y^L) \right)^2 \quad (3.3)$$

The initial guess  $\mathbf{g}^{L_m} = (0, 0)$  and the propagation of the guess from level  $L$  to  $L - 1$  is accomplished by  $\mathbf{g}^{L-1} = 2(\mathbf{g}^L + \mathbf{d}^L)$ . The final optical flow  $\mathbf{d}$  is calculated from partial flow estimation at the base level and the corresponding guess:

$$\mathbf{d} = \mathbf{g}^0 + \mathbf{d}^0 \quad (3.4)$$

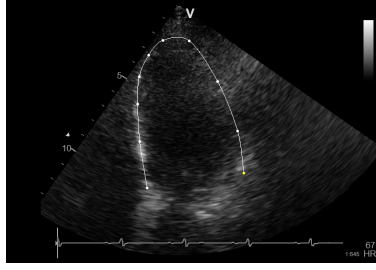


Figure 4.1: Manual annotation of mid-myocardium

## 4 Statistical shape model

The PyrLK point tracker provides robust technique for following individual points of interest. However, it is important to maintain a specific shape of the LV during cardiac cycle. We are using a statistical shape model that has been previously described in [3, 2]. The main idea of statistical shape models is employment of probability density model. Some arbitrary shape  $s = ((x_1, y_1), (x_2, y_2), \dots, (x_n, y_n))$  can be unambiguously represented as a column vector  $\mathbf{s} = [x_1, y_1, x_2, y_2, \dots, x_n, y_n]^T$ . Modeling of the shape variation is accomplished using Principal Component Analysis (PCA).

Our shape model takes into account several different origins of the shape variation. The first is the overall variability of LV shape during cardiac cycle as well shape difference between different subjects. The second origin of the shape variation is the thickness of the myocardium. The last origin of the shape variation is connected with the local strain in the myocardium.

The overall shape variation is modeled from the manually annotated data. 50 shapes from 4 LV cycles from 2 different subjects were annotated. The operator marked the points of the mid-myocardium and the spline was interpolated between points (see Figure 4.1).

New points on the spline were established, such that the distance between consecutive points was uniform. All these uniformly distributed points were triplicated, such that the inner points represent endocardium and the outer points represent epicardium. Let the  $p$  is some arbitrary mid-myocardium point. The  $p_{prev}$  is its predecessor and  $p_{next}$  is its successor. The corresponding points  $p_{inner}$  and  $p_{outer}$  on the endocardium and epicardium respectively are:

$$\begin{aligned} p_{inner\ x} &= p_x - w \frac{(p_{next\ y} - p_{prev\ y})}{2\|p_{next} - p_{prev}\|} \\ p_{inner\ y} &= p_y + w \frac{(p_{next\ x} - p_{prev\ x})}{2\|p_{next} - p_{prev}\|} \end{aligned} \quad (4.1)$$

$$\begin{aligned} p_{outer\ x} &= p_x + w \frac{(p_{next\ y} - p_{prev\ y})}{2\|p_{next} - p_{prev}\|} \\ p_{outer\ y} &= p_y - w \frac{(p_{next\ x} - p_{prev\ x})}{2\|p_{next} - p_{prev}\|} \end{aligned} \quad (4.2)$$

where  $w$  is desired width of the myocardium and  $\|\cdot\|$  represents Euclidean distance between consecutive points.

The local variation of the strain among the myocardium is accomplished with the frequency modulation added to the uniformly distributed points. The distance between consecutive points is altered by different modulation frequencies with various phase shifts. An arbitrary local strain can be thus modeled by the linear combination of basal frequencies and shifts. Let the uniformly distributed points  $P = (p_1, p_2, \dots, p_n)$  forms a carrier wave. This wave is modulated with the following signal:

$$s(i) = \sin\left(\frac{2\pi \cdot i}{n \cdot f} + 2\varphi\pi\right) \text{ for } i \in (0, 1, \dots, n-1) \quad (4.3)$$

where  $n$  is number of points,  $f$  is desired frequency, and  $\varphi$  is phase shift. The frequency modulation of points  $P = (p_1, p_2, \dots, p_n)$  with modulation values  $S = (s_1, s_2, \dots, s_n)$  is accomplished by:

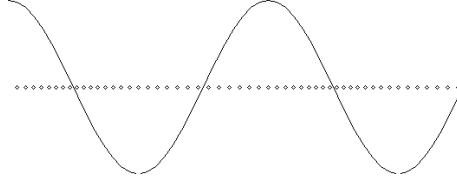


Figure 4.2: Point distribution frequency modulation

$$p_i = \begin{cases} p_i + 0.5 \cdot (p_{i+1} - p_i) \cdot s_i & \text{if } s_i \geq 0 \\ p_i + 0.5 \cdot (p_{i-1} - p_i) \cdot s_i & \text{if } s_i < 0 \end{cases} \quad (4.4)$$

The Figure 4.2 shows the example of point distribution frequency modulation. Please note that the highest point distribution frequency is shifted compared to the amplitude of the modulation wave. This is caused by the manner of the frequency modulation calculation.

The training set of shapes for the PCA calculation is formed from the input annotated shapes in the following manner:

1. For each shape  $(s_1, s_2, \dots, s_{n_s})$ , uniformly distributed points denoting the mid-myocardium are computed.
2. The uniformly distributed mid myocardium points are modulated with various frequencies  $(f_1, f_2, \dots, f_{n_f})$  and phase shifts  $(\varphi_1, \varphi_2, \dots, \varphi_{n_\varphi})$ .
3. Corresponding epicardial and endocardial points are added to the resulting modulated mid-myocardium points using different tissue widths  $(w_1, w_2, \dots, w_{n_w})$ .

The total number of training shapes is thus  $N = n_s \cdot n_f \cdot n_\varphi \cdot n_w$ . Finally, the PCA is calculated on the input samples:

1. Mean vector  $\bar{s}$  of all shapes

$$\bar{s} = \frac{1}{N} \sum_{i=1}^N s_i \quad (4.5)$$

2. Covariance matrix  $S$

$$S = \frac{1}{N-1} \sum_{i=1}^N (s_i - \bar{s})(s_i - \bar{s})^T \quad (4.6)$$

3. Compute the eigenvectors  $\phi_1, \dots, \phi_m$  and corresponding eigenvalues  $\lambda_1, \dots, \lambda_m$  of  $S$ .
4. Create projection matrix  $\Phi = (\phi_1 | \phi_2 | \dots | \phi_t)$  that contains  $t$  largest eigenvalues from previous step ( $t \leq m$ ). Usually, the  $t$  is selected to such value, that  $\sum_{i=1}^t \lambda_i = 0.95 \sum_{j=1}^m \lambda_j$ .

Now, every shape  $s$  can be represented in  $t$ -dimensional PCA space:

$$b = \Phi^T (s - \bar{s}) \quad (4.7)$$

Figure 4.3 shows mean calculated LV shape and some of its variations modeled with various vectors  $b$ .

## 5 Algorithm description

Our tracking algorithm consists of individual function blocks where each block deals with a specific task and can be easily replaced with another block that solves the same problem, e.g., the image processing block can be empty or consists of several consecutive image filters. See Figure 5.1.

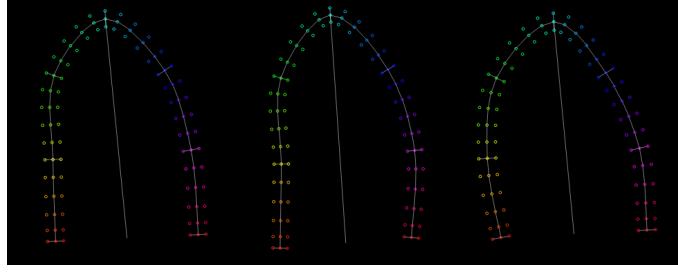


Figure 4.3: Mean LV shape (left) and some examples of it's variation

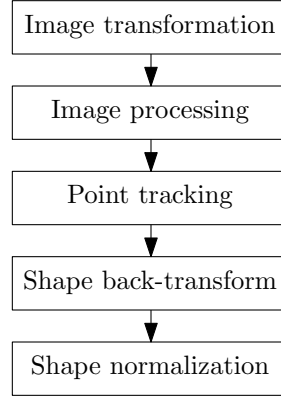


Figure 5.1: Algorithm pipeline

## 5.1 Image transformation and processing

The image transformation block transforms the input image from the input  $x$ - $y$  coordinate system to the polar (angle-distance  $\theta$ - $d$ ) coordinate system. The center of the  $\theta$ - $d$  coordinate system is placed to the center of the circular sector that contains B-mode image data (see Figure 5.2). An arbitrary point  $X = (x, y)$  is thus transformed to the point  $X' = (\theta, d)$ , such:

$$\theta = \text{atan2}(y, x) \quad (5.1)$$

$$d = \sqrt{x^2 + y^2} \quad (5.2)$$

The image processing block involves the application of image filters to the transformed frame. We have tested contrast enhancement, histogram equalization, edge detection, sharpening, and noise reduction in our experiments.

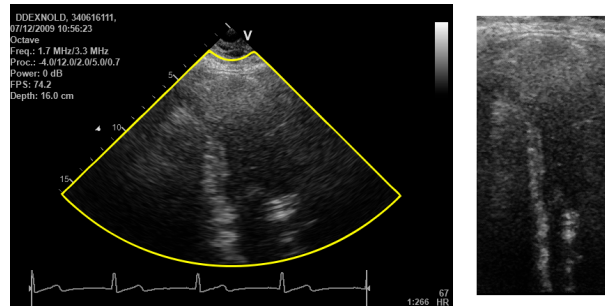


Figure 5.2: Image transformation from  $x$ - $y$  coordinate system to the angle-distance ( $\theta$ - $d$ ) coordinate system

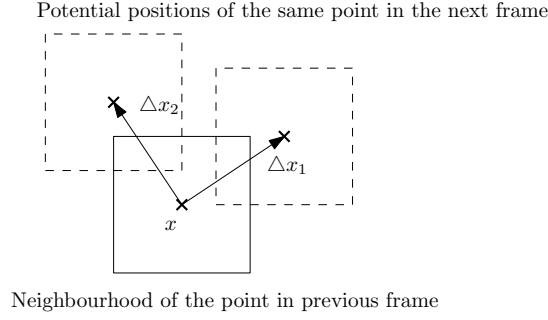


Figure 5.3: Point tracking using arbitrary metric

## 5.2 Point tracking

The point tracking block has three inputs - the previous frames, the annotated points within the previous frames, and the consecutive frame, where the point tracking algorithm should find the desired points. The consecutive frame is also processed with image transformation and filters. Additionally to the PyrLK point tracker, we have implemented speckle tracking based on the minimizing the Euclidean distance, correlation metric, cosine metric, or sum of absolute differences between the neighbor of tracked point in previous frame and potential position of the point in the next frame (see Figure 5.3). Let the pixel intensity neighborhood of the point  $x$  is denoted  $\mathbf{x}$ . The shift of the point  $x$  between the previous and the consecutive frame  $\Delta x$  is:

$$\Delta x = \arg \min_{i \in I} \|\mathbf{x} - \mathbf{x}_i\| \quad (5.3)$$

where  $\mathbf{x}_i$  is the pixel intensity neighborhood of the point  $x_i$  in the next frame and  $I$  is the set of all possible neighborhoods in the next frame.

Our algorithm also allows to have multiple weighted input frames. Suppose that we are estimating the LV shape in frame  $I_i$ . The input consist of  $j$  frames  $(I_{i-j}, I_{i-j+1}, \dots, I_{i-1})$  with corresponding weights  $\mathbf{w} = (w_j, w_{j-1}, \dots, w_1)$ . Usually, the weights are set to  $(w_1 = 1, w_2 = 0.5, w_3 = 0.25, \dots)$ . The estimation of shape  $\mathbf{s}_i$  in frame  $I_i$  is then calculated from all previous frames. The weighted mean using weights  $\mathbf{w}$  is used to form the final shape.

## 5.3 Shape back-transform and normalization

After all points in the next frame are located, the output shape is back-transformed to the  $x$ - $y$  coordinate system. The next step is the employment of the statistical shape model. The shape is projected to the PCA subspace (eq. 4.7).

The task of the shape normalization is to find optimal projection vector vector  $\mathbf{b}$  that normalizes the input shape  $\mathbf{S}$ .

1. Initialize the projection parameters  $\mathbf{b}$  to zeros.
2. generate model instance  $\mathbf{s} = \bar{\mathbf{s}} + \Phi \mathbf{b}$ , where  $\bar{\mathbf{s}}$  is the mean shape and  $\Phi$  is the eigenvector matrix calculated from 4.6.
3. Find the optimal rotation and scale  $T_{\theta, s}$  that transforms the  $\mathbf{s}$  to the input shape  $\mathbf{S}$ . See [3] for more details.
4. Apply the inverted transform  $T_{\theta, s}^{-1}$  to the input shape:  $\mathbf{S}' = T_{\theta, s}^{-1} \mathbf{S}$ .
5. Project transformed shape  $\mathbf{S}'$  to the PCA subspace:  $\mathbf{b} = \Phi^T (\mathbf{S}' - \bar{\mathbf{s}})$
6. On each component  $b_i$  of the vector  $\mathbf{b}$  apply the limit  $\pm 3\sqrt{\lambda_i}$ , where  $\lambda_i$  is the  $i$ th eigenvalue. This truncation ensures that the resulting shape will be similar to those in the original training set.
7. If the difference between previously calculated  $\mathbf{b}$  and current  $\mathbf{b}$  is greater than some threshold go to 2.

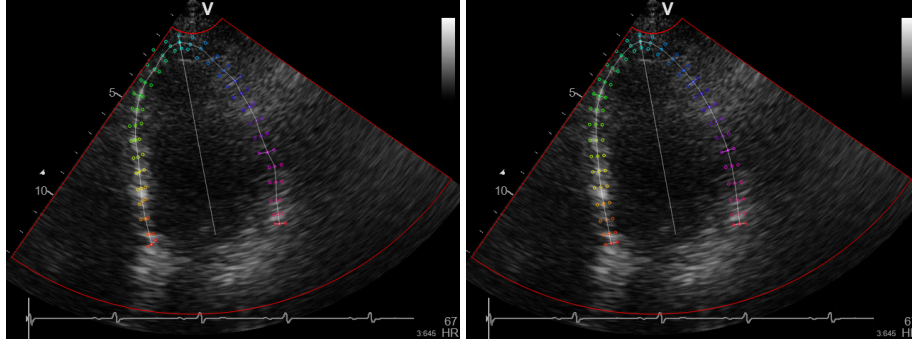


Figure 5.4: Points of the shape before normalization and after the PCA shape normalization

8. apply the last transformation  $T_{\theta,s}$  to the generated model instance  $\mathbf{s}$ :  $\mathbf{S} = T_{\theta,s}\mathbf{s}$

The result of the shape normalization is in Figure 5.4.

## 5.4 Result post-processing

The optional processing of the tracing result is the application of the simple moving average filter. The main parameter  $n$  of this filter is the floating window size that determines to which extent the data are smoothed. The arbitrary point  $p_i$  within shape  $i$ th shape  $\mathbf{s}_i$  is smoothed using average of the same points from different time indices  $j \in \langle i - \lfloor \frac{n}{2} \rfloor, i + \lfloor \frac{n}{2} \rfloor \rangle$ :

$$p_i \leftarrow \frac{\sum_{j=i-\lfloor \frac{n}{2} \rfloor}^{i+\lfloor \frac{n}{2} \rfloor} p_j}{n} \quad (5.4)$$

We have used this filter with the size parameter set to 1 (no filter), 3, 5, and 7.

# 6 Evaluation

## 6.1 Database description and performance metrics

**TODO popsat zarizeni, ktery se vse skenovalo.** We have tested our LV tracker on the dataset consisting of 10 subjects – 3 or 4 beat cycles were recorded for each subject resulting in total 36 beat cycles. The frame-rate ranges from 62FPS to 82FPS. The individual beat starts (EKG peaks) as well as the initial shape of the LV were manually annotated. For the shape annotation, just the mid-myocardium control points were annotated. The real LV shape was automatically deduced using the PCA shape model.

The longitudinal strain (main strain,  $MS$ ) represents myocardial deformation along whole LV. The strain equation  $\epsilon$  is defined as:

$$\epsilon = \frac{L - L_0}{L_0} \quad (6.1)$$

where  $L$  is the length of the LV after deformation and  $L_0$  is the initial length of the LV. Our LV shape model consists of  $n$  mid-myocardial points – thus  $3n$  in total. In order to estimate the main strain, individual per-point strains  $l_j$  have to be calculated ( $j = 1, \dots, n - 1$ ). See Figure 6.1. The main strain is thus:

$$L = \frac{1}{n} \sum_{j=1}^n l_j \quad (6.2)$$

Our LV model is divided into six segments – see Figure 6.2. **TODO: Vložit citaci, že toto rozdělení je standardní.**

There are several requirements of good tracker. Since the LV myocardial movement is periodical, strain progress within each beat should be more or less the same – the beat-to-beat variance should be as low as possible.

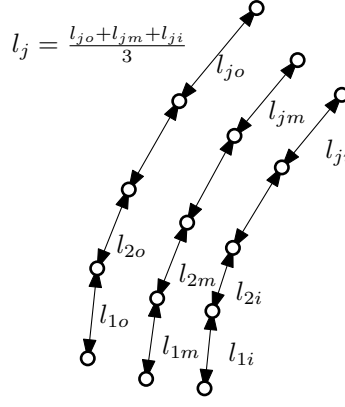


Figure 6.1: Per-point strain

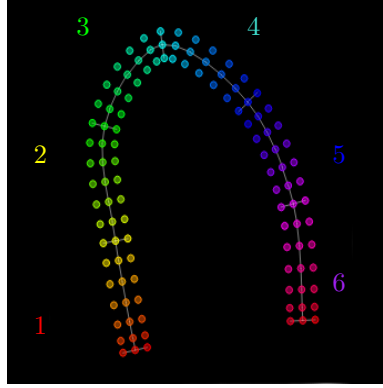


Figure 6.2: LV segmentation

For one person/record, all strain progresses for all beats were aligned together and sampled to 100 values (see Figure 6.3). Beat-to-beat variance is thus the sum of standard deviations for each sample. We have measured the beat-to-beat variance for main LV deformations (main strain, *MS*) as well as for individual segments (per segment strain, *PSS*).

The most important metrics of the tracker is precision. Since the movement of the LV is periodical, we can express the precision as the sum of quadratic differences of corresponding points from the first frame and the last frame of one beat. The shapes can't be compared directly, the last frame should be aligned (translation and rotation) to the first frame shape in terms of least square differences.

All algorithms were compared according to the following performance metrics:

- Main strain beat-to-beat variance
- Segment strain beat-to-beat variance
- Precision

## 6.2 Comparison of implemented techniques

As was mentioned in section 2, our LV tracking algorithm is modular. Individual components are easily interchangeable. We have compared the PyrLK-based point tracking technique with the simple tracker described in section 5.2. Out of the simple trackers based on the arbitrary distance metric, the best results were achieved with the cosine-based tracking technique. Therefore, in the following results, only the PyrLK and cosine trackers will be taken into account.

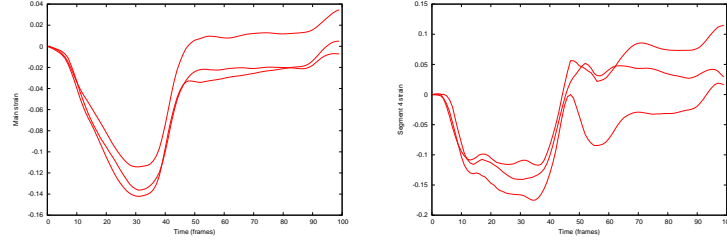


Figure 6.3: Beat-to-beat variance for main LV strain and for selected segment

The strain frequency modulation were performed with the following parameters – the frequency parameters  $f = (1, 2, 3)$ , phase shifts  $\varphi = (0, \frac{1}{3}, \frac{2}{3})$ , myocardial widths  $w = (20, 30)$  pixels. Three overall myocardial shapes were trained. First was directly educed from the fore-mentioned parameters, the second and third LV shape models were trained in the similar way, only the amplitude of the modulation waves were reduced to 0.5 and 0.2 respectively. This modification of the training parameter causes more rigid model.

Our LV strain model consists of individual points divided into 6 segments. We have compared the models where each segment consists of 4, 6, and 11 mid-myocardial points. Thus the first model contains 3 per-point strains for each segment and 57 points total. The second model contains 5 per-point strains in each segment and 93 points total. The last model contains 10 per-point strains in each segment and 183 points total.

We have tested the following point-trackers:

- Tracking utilizing optical flow estimation (PyrLK)
  - Tracking individual points of the shape
  - Tracking all virtual points within certain neighborhood around each shape point. (kernel size  $21 \times 21$  with one virtual point in every  $3 \times 3$  sub-window)
- Tracking based on distance/similarity utilizing
  - Cosine distance (kernel  $21 \times 21$ )

The number of frames and corresponding annotations from which the LV motion is estimated was consecutively selected to 1 (no history), 2 with weights (1, 0.5), and 3 with weights (1, 0.5, 0.25). The post-processing averaging parameter  $n$  were set to (1, 3, 5, 7). It has emerged in our experiments that the image processing filters has no positive impact on the quality of tracking. Therefore, we have not included the image processing in batch testing.,

Nevertheless, the total number of tested individual trackers is 3 (frequency modulation, shape model stiffness)  $\cdot$  3 (point count)  $\cdot$  3 (point trackers)  $\cdot$  3 (input frame weights)  $\cdot$  4 (result averaging) = 324.

In order to find the best tracker, all three main characteristics (MS variance, PSS variance, and precision) have to be combined. For each characteristic  $c$ , we calculate mean value  $\mu_c$  and standard deviation  $\sigma_c$  and perform z-score normalization:

$$c' = \frac{c - \mu_c}{\sigma_c} \quad (6.3)$$

All three characteristics are summed and trackers can be directly compared. The better the tracker is, the lower is the sum of z-score normalized characteristics. The distribution of normalized characteristics for all compared LV trackers can be seen in Figure 6.4.

Table 1 shows the best 5 trackers according to the described characteristics. Several conclusions can be deduced from these results. First, the PyrLK-based trackers outperformed simple trackers based on the minimizing the distance between the neighborhood of tracked point and neighborhood around potential point position in consecutive frame – in our tests represented by trackers based on the cosine metric. Adding additional virtual points around each shape point does not have any positive impact on the tracking performance. The result processing



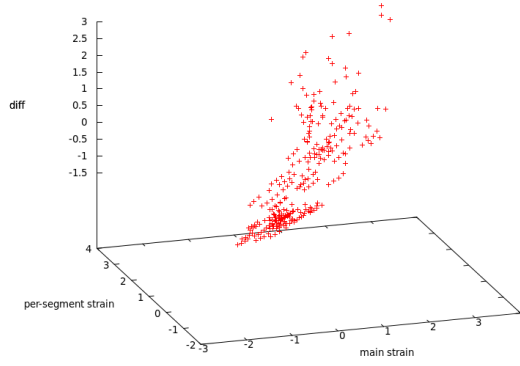


Figure 6.4: Distribution of normalized characteristics for compared LV trackers

Table 1: The best trackers (all are PyrLK-based point trackers with modulation amplitude set to 0.2 and only one input frame)

Rank	Points per segment	moving average filter	Main strain Beat-to-beat variance	Per-segment beat-to-beat variance	Sum of square differences	Sum of normalized characteristics
1	11	7	0.0115	0.0264	19.3349	-4.8452
2	11	5	0.0119	0.0273	19.4738	-4.6489
3	11	3	0.0123	0.0283	19.3920	-4.4637
4	6	7	0.0122	0.0294	21.0413	-4.3603
5	11	7	0.0131	0.0279	20.1497	-4.2462

represented by the moving average filter improves the beat-to-beat variance as well as the target precision. However, too large smoothing parameter  $n$  may vanish some interesting details in myocardium movement. The direct impact of the of the moving average filter can be seen in Figure 6.4 as the quaternions aligned on the lines. It has also emerged that too high amplitude in local strain frequency modulation has negative impact on the beat-to-beat variance. The best results were achieved when the amplitude was set to 0.2. Adding additional weighted input frames to the tracking input does not improve the tracker performance. Our tests shows that better results are achieved when only one (last) frame is used when the optical flow of the points of interest in next frame are calculated.

## 7 Conclusion and further research

We have shown that the utilizing of the probability shape model and PyrLK-based point tracker outperforms simple trackers based on the correlation similarity. We would like to test our method on patients with myocardial movement disruption in our further experiments in order to prove that our method can be used to non-invasive examination and detection, e.g., the detection of myocardial ischemism.

## References

- [1] Jean-yves Bouguet. Pyramidal Implementation of the Lucas Kanade Feature Tracker. Technical report, 2001.
- [2] T. F. Cootes and C. J. Taylor. A mixture model for representing shape variation. *Image and Vision Computing*, 17(8):567–573, 1999.
- [3] T. F. Cootes and C. J. Taylor. Statistical models of appearance for computer vision. *Science and Biomedical Engineering, University of*, 2001.

- [4] B. D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *Proceedings of Imaging Understanding Workshop*, volume 130, pages 121–130, 1981.