An Automated Method for Analysis of Flow Characteristics of Circulating Particles From *In vivo* Video Microscopy

Eran Eden*, Dan Waisman, Michael Rudzsky*, Haim Bitterman, Vera Brod, and Ehud Rivlin, *Member, IEEE*

***Abstract*—The behavior of white and red blood cells, platelets, and circulating injected particles is one of the most studied areas of physiology. Most methods used to analyze the circulatory patterns of cells are time consuming. We describe a system named CellTrack, designed for fully automated tracking of circulating cells and micro-particles and retrieval of their behavioral characteristics. The task of automated blood cell tracking in vessels from *in vivo* video is particularly challenging because of the blood cells' nonrigid shapes, the instability inherent in *in vivo* videos, the abundance of moving objects and their frequent superposition. To tackle this, the CellTrack system operates on two levels: first, a global processing module extracts vessel borders and center lines based on color and temporal patterns. This enables the computation of the approximate direction of the blood flow in each vessel. Second, a local processing module extracts the locations and velocities of circulating cells. This is performed by artificial neural network classifiers that are designed to detect specific types of blood cells and micro-particles. The motion correspondence problem is then resolved by a novel algorithm that incorporates both the local and the global information. The system has been tested on a series of *in vivo* color video recordings of rat mesentery. Our results show that the synergy between the global and local information enables CellTrack to overcome many of the difficulties inherent in tracking methods that rely solely on local information. A comparison was made between manual measurements and the automatically extracted measurements of leukocytes and fluorescent microspheres circulatory velocities. This comparison revealed an accuracy of 97%. CellTrack also enabled a much larger volume of sampling in a fraction of time compared to the manual measurements. All these results suggest that our method can in fact constitute a reliable replacement for manual extraction of blood flow characteristics from *in vivo* videos.**

***Index Terms*—Blood cells, leukocyte, microcirculation, microscopy, motion correspondence, segmentation, tracking.**

Manuscript received February 6, 2005; revised May 4, 2005. This work was supported by a research grant from the Albert Godstein Research Fund, The Technion–Israel Institute of Technology, Haifa, Israel. The Associate Editor responsible for coordinating the review of this paper and recommending its publication was E. Meijering. *Asterisk indicates corresponding author.*

*E. Eden and *M. Rudzsky are with the Faculty of Computer Science, The Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: eraneden@cs.technion.ac.il; rudzsky@cs.technion.ac.il).

D. Waisman, V. Brod, and H. Bitterman are with The Ischemia–Shock Research Laboratory of the Carmel Medical Center and the Bruce Rappaport Faculty of Medicine, The Technion–Israel Institute of Technology, Haifa 32000, Israel.

E. Rivlin is with the Faculty of Computer Science, The Technion–Israel Institute of Technology, Haifa 32000, Israel.

Digital Object Identifier 10.1109/TMI.2005.851759

I. Introduction

Analysis of the behavior of circulating blood cells and injected particles in *in vivo* video microscopy is paramount to micro-hemodynamics research. Manually performed analysis is time consuming and limited in the number of measurements that can be made and in the reproducibility of the results. Hence, a robust automated method for identification, tagging and tracking of blood cells as well as measurements of cell velocities and microvascular shear rates may prove valuable for *in vivo* video microscopy analysis. The difficulty in developing a method for blood cell tracking stems from the cells' nonrigid shapes and the noise, cluttering and abundance of moving objects that disappear and reappear sporadically. Another level of difficulty is added by the video instability and the defocusing that occurs in moving organs, which is intrinsic to videos taken during *in vivo* microscopic studies.

A. Background

Research dealing with image processing of blood vessels and blood circulation can be classified into algorithms performed on the vessel level and algorithms performed on the cell/particle level. Analysis on the vessel level includes tasks such as segmentation of blood vessel regions [1]–[3] and vessel centerline extraction [4], [5]. Analysis and tracking of circulating cells/particles includes the detection and/or tracking of leukocytes (white blood cells) [1], [6]–[10], and erythrocytes (red blood cells) [11]. Cell tracking methods can be further divided into edge-detection-based methods that use active contours in order to track cells, and feature-based methods. In [6] and [8] a shape and size constrained snake model is introduced for tracking leukocytes. Good performance was reported. However, the algorithm is based on user interaction for initialization of the leukocyte center position, which is not feasible for investigation of hundreds of objects in video sequences. Another shortcoming of the system is that it tracks one leukocyte at a time. In a recently published work [12], a method for multicell detection and tracking is described. In contrast to other methods, here the initialization of cell position is performed automatically. This is done by exploiting the characteristic shape and intensity of cells. In another method for leukocyte tracking, video images are registered using template matching with normalized cross correlation, followed by segmentation of vessel regions using a pixel's gray level temporal variance [1], [13]. Hopfield networks are then used for coupling fragmented leukocytes trails. Other methods for detection of leukocytes and fluorescent lymphocytes include [7], [9], [10]; however, they do not perform tracking.

B. General Description of Our System

The CellTrack system is an automated software system for robust tracking of multiple particles in blood flow. It has been designed and tested on a variety of *in vivo* color video recordings of rat microcirculation using fluorescent reflected light microscopy, or transmitted light microscopy. The system extracts the following information from *in vivo* video microscopy: vessel topology and diameter, vessel center-line, blood flow direction, leukocyte or microsphere velocity and shear rate. The number of cells or particles in the studied vessels are also observed and tagged. The CellTrack system has two main modules.

1) Global processing module, performing analysis on the vessel level:
   - image registration (Section III-A);
   - vessel segmentation (Section III-B);
   - computation of the virtual flow (Section III-C).
2) Local processing module, performing analysis on the cell level:
   - cell detection (Section IV-A);
   - resolving the motion correspondence problem (Section IV-B);
   - tracking of leukocytes/microspheres and computing their velocity (Section IV-B).

The main modules in the system are summarized in Fig. 1. The contributions of this paper are in the following aspects.

- *In vivo* color video sequences of rat mesentery were generated and analyzed as opposed to gray scale videos, which are analyzed by most automated cell tracking methods. The advantage of using color video sequences is twofold. First, the color information can be used to augment the accuracy of the cell and particle detection algorithm. Second, using different colorings for different types of objects (e.g., microspheres and leukocytes) may enable simultaneous tracking of more than one type of object in the same video sequence.
- A description of a new vessel segmentation algorithm that is based on artificial neural network (ANN) classification. The algorithm utilizes a combination of color texture as well as temporal texture-based features. We tested the algorithm on a variety of *in vivo* color video sequences and compared it to other segmentation algorithms. The results demonstrate the superiority of the method described here.
- An algorithm for efficiently computing an approximation of the blood flow direction (termed 'virtual flow') in each location inside a vessel based on the vessel topology and axial direction is described.
- The Chetverikov particle image velocimetry method [14], [15] was enhanced and applied to resolve the motion correspondence problem of circulating leukocytes and microspheres.
- The tracking accuracy was tested on two types of targets, leukocytes and microspheres, in a variety of scenarios. In addition to standard tracking characteristics such as object location and velocity, the method can also compute other characteristics, for example, shear rate between the fluid layers and the vessel wall. These computer generated measurements were compared to manual measurements, yielding an accuracy of 97%.

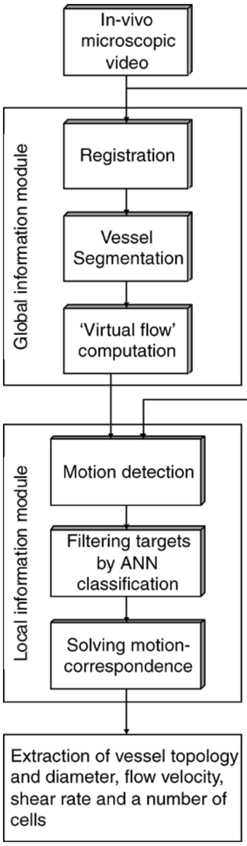

In-vivo microscopic video
Global information module
Registration
Vessel Segmentation
'Virtual flow' computation
Local information module
Motion detection
Filtering targets by ANN classification
Solving motion-correspondence
Extraction of vessel topology and diameter, flow velocity, shear rate and a number of cells

Fig. 1. System flow chart.

II. *In Vivo* Video Microscopy – Experimental Setup

A microcirculation study was performed on rat mesentery. The mesentery is a thin transparent membrane that supports the blood vessels feeding the bowel. In anesthetized rats a section of intestinal mesentery was exteriorized through a midline abdominal incision, draped over a heated transparent pedestal, and covered with a nylon wrap. Capillaries and postcapillary venules were selected for observation.

A. Videomicroscopy

All observations were done using an Olympus trinocular fluorescent microscope BX-60 (Tokyo, Japan) equipped with trans- and epi-illumination. Experiments were recorded with a Sony 3CCD color video DXC-950P camera coupled to the microscope and connected to a Sony digital videocassette recorder (DVCAM DSR-30P), and in parallel to a color video monitor (Mennen Medical). Two different types of video recordings were generated, as described in [16].

1) **Type I videos:** Green fluorescent microspheres (1 μm diameter) (Molecular Probes – USA) were injected into the systemic circulation at a rate of 0.1 ml/min by an infusion pump. To enable the visualization and identification of the blood vessels, they were counter-stained by injection of fluoresceine or Rhodamine G6 (vital fluorescent colorants). In order to see the microcirculatory behavior of the circulating particles in a nontransparent solid organ, epi-illumination with a fluorescent blue light was used (480 nm wave length); for an example, see Fig. 2.

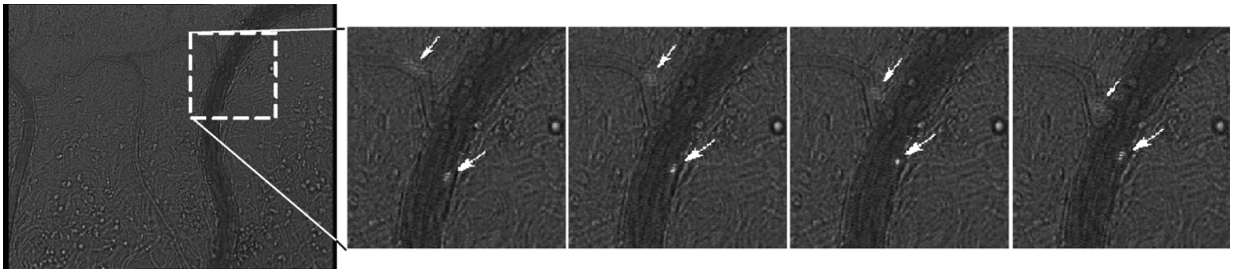


Fig. 2. Small region extracted from four consecutive frames of a Type I *in vivo* color video sequence. The circulating microspheres are colored in green fluorescent. The white arrows were added to mark the locations of the microspheres.

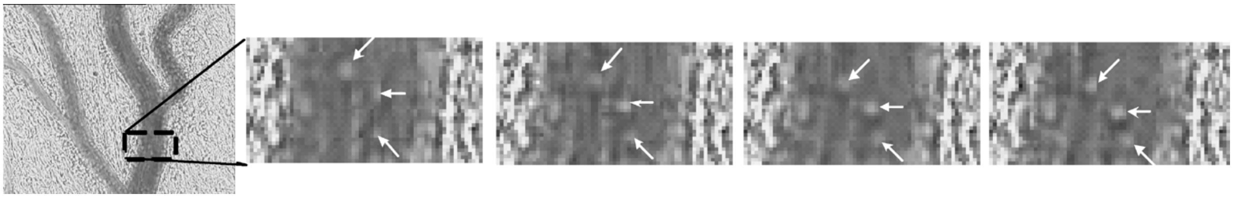


Fig. 3. Small region extracted from four consecutive frames of a Type II *in vivo* color video sequence. The white arrows were added to mark the locations of circulating leukocytes.

2) **Type II videos:** These videos recorded a series of experiments that studied the behavior of unstained circulating leukocytes in the vascular bed of the mesenteric microcirculation. Their rolling adhesion to the vessel's walls was measured; for an example, see Fig. 3.

All the algorithms described in this paper were implemented and tested on both types of video recordings (five Type I videos and four Type II videos). One video sequence of each type was used in order to fine tune the CellTrack parameters. When applied on the rest of the respective video sequences, the same set of parameters was used. We have made examples of both types of *in vivo* video sequences publicly available at http://www.cs.technion.ac.il/rudzsky/~celltrack1.html.

III. Global Processing Module

The global processing module deals with the extraction of information on the vessel level. It incorporates three submodules: registration, vessel segmentation, and "virtual flow" computation.

A. Registration

Prior to tracking, it is important to compensate for the image background motion, which occurs between sequential frames as a result of respiratory movements, and compute the offset between the frames. Let I_t be an image frame corresponding to a moment t in time. We performed the following procedure to calculate the global offset between a reference frame R and another frame I. Let $B'_{(u,v)}$ and $B_{(a,b)}$ be quadratic $N \times N$ blocks extracted from R and I with centers at (u, v) and (a, b), respectively (in the application we use $N = 30$ pixels; however, tests with other values of N yielded similar results as long as $\mathrm{N} > 20$). An estimation of the offset between R and I can be computed by tracing $B'_{(u,v)}$ in I. This is done by finding a block $B_{(a,b)}$ in I that maximizes the normalized cross correlation (NCC) between the two blocks [1]. Let $\mu_{(u,v)}$ and $\mu_{(a,b)}$ be the mean of the pixel values in block $B'_{(u,v)}$ and block $B_{(a,b)}$, respectively. The normalized cross correlation between two blocks is

$$\mathrm{NCC}(B_{(a,b)}, B'_{(u,v)}) = \frac{\sum\limits_{k,l}[I(a+k, b+l) - \mu_{(a,b)}][R(u+k, v+l) - \mu_{u,v}]}{\left\{\sum\limits_{k,l}[I(a+k, b+l) - \mu_{(a,b)}]^2 \sum\limits_{k,l}[R(u+k, v+l) - \mu_{(u,v)}]^2\right\}^{0.5}} \quad (1)$$

where indexes k and l are used for running over all the pixels in the blocks. The offset $\mathbf{r} = (a - u, b - v)$ between $B'_{(u,v)}$ and $B_{(a,b)}$, which maximizes $\mathrm{NCC}(B_{(a,b)}, B'_{(u,v)})$, is an estimation of the offset between R and I. The search for block $B_{(a,b)}$ is not done on the entire image I. Instead, we assume that the offset between two frames does not exceed a maximal offset. Notice that some blocks are more appropriate for registration than others; e.g., a block that contains contrasted colors is likely to yield better registration accuracy than a monotonic block. To accommodate for this, the block selection procedure includes the following steps. The Canny edge detector [17] is run on the reference image, R. For a block to be valid, the number of pixels that belong to edges must be larger than a predefined threshold. The Canny edge detector threshold is set automatically as follows: 1) A derivative magnitude image is computed. 2) The histogram of this image is extracted. 3) The upper threshold is set to the minimal number for which more than 70% of the pixels in the histogram are classified as not having an edge. The lower threshold is 0.4 × the upper threshold. This procedure reduces the chance of nonoptimal thresholding. We also performed this block selection procedure using the Harris corner detector [18]. Our motivation was to test whether the uniqueness of corner motion versus ambiguity of edge motion improves the results. In practice, both approaches yielded similar results. In order to make the registration more robust, the final offset between frames R and I is the median of offsets between n different pairs of blocks (in the application $n = 10$ was used). Differences that might occur between frames, in addition to background motion, include image blurring as a result of the camera getting out of focus. This is not resolved by the above registration procedure. Algorithms for solving this have been suggested [19], but due to their substantial running time [8] (several minutes for registration of two frames), they are not practical for registration of entire video sequences. Other registration algorithms, such as the ones described in [20], [21], might further enhance registration accuracy. In practice, the global registration procedure described here was able to compensate for most of the motion effects in the video sequences under study.

B. Vessel Segmentation

Accurate vessel segmentation can enhance the precision of cell tracking. It enables the restriction of regions where targets should be tracked, thus reducing both running time and target false positives. The task of image segmentation has received much attention in the literature [22]–[24]. Most of the proposed algorithms attempt to identify specific regions in an image using edge-based and/or region-based methods [25]. A recent study described a snake model approach for detection of vessel borders [3]. These algorithms rely mostly on the texture, color and intensity of the image. However, when performing segmentation of frames taken from a video sequence it is also possible to utilize temporal information to augment the accuracy of segmentation. A vessel segmentation approach that utilizes the temporal nature of *in vivo* video was suggested in [1]. It is based on the assumption that the gray level of a pixel where blood flows will vary from frame to frame while that of regions outside the vessel, e.g., tissues, will basically stay constant. In [1], the temporal variance of a pixel's gray level was used in order to quantify the amount the pixel's intensity changed through a sequence of frames. The temporal variance of a pixel's gray value in a sequence of images is defined as

$$\sigma_T^2(x,y) = \frac{1}{n}\sum_{t=1}^{n}(I_t(x,y) - \overline{I}_T(x,y))^2 \tag{2}$$

where I_t is the tth frame and the temporal mean of a pixel is defined as

$$\overline{I}_T(x,y) = \frac{1}{n}\sum_{t=1}^{n} I_t(x,y). \tag{3}$$

A temporal information-based segmentation can now be performed in the following way: $\sigma_T^2(x,y)$ is computed for each pixel in the image, and if it exceeds a predefined threshold, the pixel is classified as a vessel region; otherwise, it is identified as a tissue region. When tested on our video sequences this temporal segmentation yielded unsatisfactory results (see Section V-A). This may be explained by local tissue deformations during respiratory motion that were not compensated for by the global registration process depicted in Section III-A. Another factor leading to wrong pixel classification could be the motion of fluids outside the vessel that contributes to a high temporal variance in regions outside the vessel. We, therefore, opted for a synergistic approach that aims to utilize temporal patterns combined with the information inherent in the image's color and texture.

Our algorithm uses artificial neural networks (ANN) [26]–[28] which are trained to classify a pixel into either vessel or tissue classes, based on color as well as temporal features. The trained ANN can then be used for segmentation by simply traversing over all the pixels in an image and classifying them. More formally, let I_t, $1 \leq t \leq n$, be a sequence of frames and $\overline{I}$ be the respective mean image in which each pixel is computed as in (3). $\overline{I}_R$, $\overline{I}_G$, $\overline{I}_B$ and $\overline{I}_{GR}$ are the respective RGB and gray level mean images. $\overline{I}_{GR}$ is created by converting the original RGB image into an intensity image, i.e., eliminating the hue and saturation while retaining the luminance. σ_T^2 is the temporal gray value variance over all pixels in I_t, $1 \leq t \leq n$, as defined in (2). $W_R(x,y)$, $W_G(x,y)$, $W_B(x,y)$, $W_{GR}(x,y)$ and $W_{tmp}(x,y)$ are $l \times l$ window patches, ($l = 7$ pixels), obtained from $\overline{I}_R$, $\overline{I}_G$, $\overline{I}_B$, $\overline{I}_{GR}$ and σ_T^2, respectively with window centers at position (x, y). For training and testing of the ANN classifier a data set, *Segmentation Set*, was created in which each element $\mathcal{V}$ consisted of five window patches:

$$\mathcal{V} = [W_R(x,y), W_G(x,y), W_B(x,y), W_{GR}(x,y), W_{tmp}(x,y)]. \tag{4}$$

These window patches were used for extraction of classification features. The first four windows were used for obtaining color features; the fifth was used for obtaining temporal features. The *Segmentation Set* contained a total of 400 elements, 200 of the "vessel" class and 200 of the "tissue" class, depending on the type of region in which the window's center pixel resided. Half of the elements in each class were used as a training set and the other half as a test set. It has been argued that high level features may substantially improve classification accuracy [29]. Consequently, we did not train the ANN classifier on the "raw data," i.e., on the pixel intensity values in each element. Instead, the ANN was trained on high level features that were extracted from each element. The feature extraction was performed using a co-occurrence matrix, a statistical tool for extraction of second order texture information from images [30], [31]. For an image I with N gray scale values $[0, 1, \ldots, N-1]$, the corresponding co-occurrence matrix P_v is an $N \times N$ matrix. In this matrix each element $P_v(i,j)$ represents the probability of the co-occurrence of values i and j in image I separated by a vector $v = (\delta x, \delta y)$. A matrix P_v with a vector v, which connects adjacent pixels, can describe a texture fairly well. Hence, we used the following compositional co-occurrence matrix, in which each element $P(i,j)$ was computed as follows:

$$P(i,j) = \frac{P_{v_1}(i,j) + P_{v_2}(i,j) + P_{v_3}(i,j) + P_{v_4}(i,j)}{4} \tag{5}$$

where $v_1 = (0,1)$, $v_2 = (1,1)$, $v_3 = (1,0)$, $v_4 = (1,-1)$. We did not include the matrices for vectors $(0,-1)$, $(-1,-1)$, $(-1,0)$ and $(-1,1)$, since co-occurrence of values in an image is symmetrical. The co-occurrence matrix is commonly used for extraction of color texture features. However, we expand the idea to enable extraction of temporal texture features. Initially, the compositional co-occurrence matrix was computed for each window patch in each element in the *Segmentation Set* and a vector of 60 high level features was extracted (a complete list of features is given in the Appendix). We implemented the greedy forward selection algorithm described in [29] to reduce dimensionality. Our motivation for dimensionality reduction was twofold: first, to improve the running time and second, to discard irrelevant features that may introduce noise and reduce classification accuracy. The reduced feature subset contained seven color features and four temporal features, all of which are listed in Table I. Features (2)–(4) are color texture features that were computed using the compositional co-occurrence

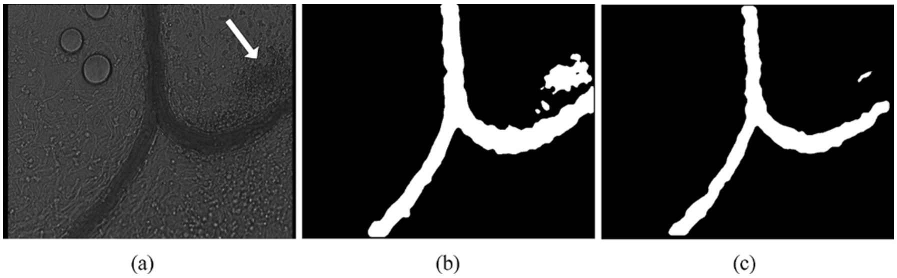


Fig. 4. This figure shows an example where the utilization of temporal features, in addition to color features, improves the quality of segmentation. (a) Video frame taken from mesenteric blood vessels. Notice that the outer vessel region marked by the arrow contains color and texture composition similar to the inner vessel regions. (b) Segmentation results of a ANN classifier that only utilizes the color features (described in Table I). White and black regions correspond to inner and outer vessel regions, respectively. It can be seen that part of the outer vessel region was wrongly classified as being an inner vessel region. (c) Segmentation results of a ANN classifier that is based both on color and temporal features (described in Table I). Most of the outer region is now classified correctly.

TABLE I
COLOR AND TEMPORAL–BASED FEATURES CHOSEN BY THE FEATURE SELECTION PROCESS. P AND Q ARE THE CO-OCCURRENCE MATRICES EXTRACTED FROM $W_{GR}(x,y)$ AND $W_{tmp}(x,y)$, RESPECTIVELY. A FULL DESCRIPTION OF THE FEATURES AND HOW THEY WERE CALCULATED IS GIVEN IN THE APPENDIX

Color features		Temporal features	
(1) Median	$med(W_{GR})$	(8) Median	$med(W_{tmp})$,
(2) Energy	$E(P)$	(9) Variance	$\sigma^2(W_{tmp})$
(3) Entropy	$S(P)$	(10) Homogeneity	$H(Q)$
(4) Homogeneity	$H(P)$	(11) Mean	$\mu(W_{tmp})$
(5) Mean R	$\mu(W_R)$		
(6) Mean G	$\mu(W_G)$		
(7) Mean B	$\mu(W_B)$		

matrix P extracted from $W_{GR}(x,y)$. Feature (10) is a temporal texture feature that was computed using the compositional co-occurrence matrix Q extracted from $W_{tmp}(x,y)$ (for details on how to calculate the features, see Appendix). It is not surprising that the subset of the features that yielded the highest ANN accuracy in terms of correlation coefficient [see (10)] also contained, in addition to the standard color-based features, temporal-based features. For example, in some video sequences local vessel and tissue regions have similar color intensity and texture. Therefore, these regions are not separable by methods that rely solely on differences in color and texture. Utilization of temporal patterns, however, enables their accurate separation (Fig. 4). We elaborate on this in Section V. In the training of the ANN we evaluated many ANN architectures. To prevent data overfitting, training was halted when performance on the test set declined. A feed-forward back-propagation neural network consisting of an input layer, a hidden layer of 10 neurons and an output layer of one neuron was chosen since it yielded the highest accuracy in terms of the correlation coefficient [see (10)]. Once a classifier is selected, the task of locating the vessel regions in an image can be performed by traversing over all the pixels in the image and classifying them into "vessel" and "tissue" pixels.

Reducing Segmentation Algorithm Running Time: The ANN classification operation is time expensive: about 24 min for processing a 400 × 400 pixel image on a Pentium 4, 2.5-GHz PC using Matlab implementation. To improve the running time, we assume a minimal vessel diameter (in pixels). Instead of classifying each pixel in the image, we use an iterative approach that starts with a rough segmentation that is then further refined. This is done as follows.

1) *Rough segmentation:* Traverse over I by SI (Sample Interval) steps where SI is the maximal integer that is smaller than the minimal vessel diameter. The reason for constraining the initial size of SI is to prevent "skipping over" a vessel region as a result of sparse sampling. Each sampled pixel is classified and its neighboring pixels in a SI × SI window are assigned the same class. The result is a bitmap where a pixel is assigned the value 1 if it is of "vessel" class , otherwise it is assigned 0 [see Fig. 5(b)].
2) *Refined segmentation:* Traverse over I by $\lfloor \mathrm{SI}/2 \rfloor$ steps. If the sampled pixel is on the border between a "vessel" region and a "tissue" region in the bitmap (i.e., the $\lfloor \mathrm{SI}/2 \rfloor \times \lfloor \mathrm{SI}/2 \rfloor$ window is not entirely 1's or 0's), then classify the pixel and assign its $\lfloor \mathrm{SI}/2 \rfloor \times \lfloor \mathrm{SI}/2 \rfloor$ neighborhood to the same class [Fig. 5(c)]. The refined segmentation step is repeated until the size of the traversal steps is smaller than one pixel. The final result is smoothed by a median filter that removes salt and pepper noise and by morphological filtering to smooth segmentation edges [Fig. 5(c), (d)]. This sampling procedure does not lower the accuracy of the segmentation, however, the running time is reduced considerably. The average running time dropped from about 24 min to 79 s when tested on 400 × 400 pixel video sequences.

For each video sequence, the segmentation algorithm is applied only once as a tracking preprocessing step. The color features are extracted from the mean of the first m frames in the video sequence. The temporal features are computed over the first n frames. We experimented with different values for n and m. When the feature extraction was performed on a small number of frames, $5 \leq m \leq 10$ and $5 \leq n \leq 40$, the method accuracy increased as more frames were used, as might be expected. The accuracy becomes roughly constant for $m > 10$ and $n > 40$.

C. *"Virtual Flow" Computation*

We present a procedure for computing an approximation of the blood flow direction in a vessel based on the vessel topology. The algorithm receives as input a bitmap, as in Fig. 5(d). For each location inside the vessel the algorithm returns a unit vector – a prediction of the direction in which the blood will flow.

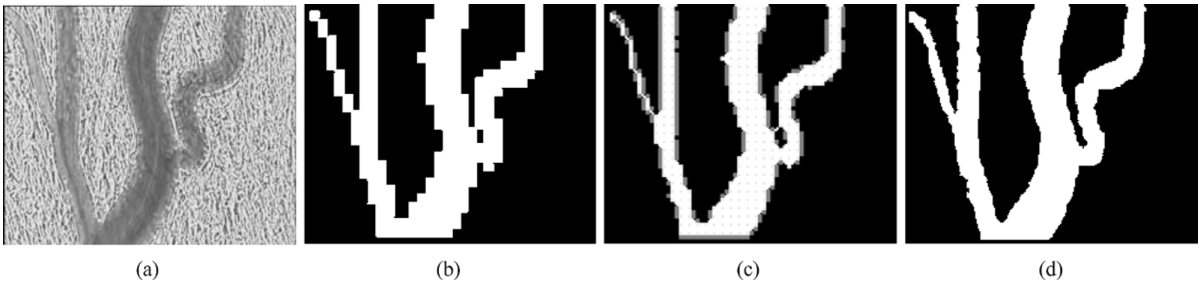


Fig. 5. Segmentation of a video frame taken from mesenteric blood vessels. (a) Original frame. (b) Rough segmentation. (c) Finer segmentation after one iteration. (d) Final segmentation after four iterations and morphological filtering.

The term virtual flow is used because the blood flow direction is calculated based on the vessel shape, without actually measuring the "real blood flow direction" through consecutive frames. We assume a model where the blood flow inside a vessel is laminar and parallel to the vessel centerline, which is correct for unbranched vessels $D < 100\ \mu\mathrm{m}$. As a pre-processing step, the vessel centerline is extracted. A good review on vessel center-line extraction is given in [4]. After testing several approaches we found that the skeleton algorithms proposed in [5], [32] are simple for implementation and yield robust and accurate results (see Fig. 6). Once the vessel center-line is computed, we compute the virtual flow in location (x, y) in the image as follows.

1) Locate n adjacent pixels located on the center-line and nearest to location (x, y). The notation $NCLP(x, y)$ is used to refer to the coordinates of these nearest center-line pixels (in the application $n = 10$ was chosen; larger values of n yielded similar results).
2) Compute the direction of the $NCLP(x, y)$ centerline section. This is done by averaging over the sum of angles between each two adjacent pixels in $NCLP(x, y)$

$$\alpha(x, y) = \frac{1}{n}\sum_{i=0}^{n-1} \tan^{-1}\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i}\right)$$

where $(x_i, y_i), (x_{i+1}, y_{i+1}) \in NCLP(x, y)$ are adjacent pixels.

The angle of the virtual flow at location (x, y), $VF(x, y)$, can have one of two values: $\alpha(x, y)$ or $-\alpha(x, y)$. It is not possible to determine which of the values corresponds to the real blood flow direction solely based on vessel topology. To resolve this ambiguity the following automatic initialization is performed: the actual movement of flowing targets is tracked during the first several frames (in the application we used five frames). The image is divided into regions, and in each region the median target flow direction is computed. This yields a map in which each region is associated with a vector indicating the median flow direction in that region. The ambiguities can now be easily resolved in the following way: a $VF(x, y)$ in a particular region is assigned the angle, either $\alpha(x, y)$ or $-\alpha(x, y))$, which is closer to the median flow direction in that region. Although this median flow direction is often imprecise, it will solve the $VF(x, y)$ ambiguity as long as it is within $-\pi/2$ to $\pi/2$ of the correct flow direction, which is almost always the case (see Fig. 7). The virtual flow is used for enhancing the accuracy of the motion correspondence algorithm (we elaborate on this in Section IV-B).

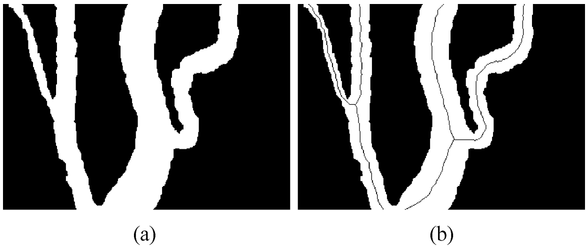


Fig. 6. (a) Vessel segmentation bitmap generated by our segmentation algorithm. (b) The black lines inside the white vessel regions were generated using the center-line extraction algorithm described in [32].

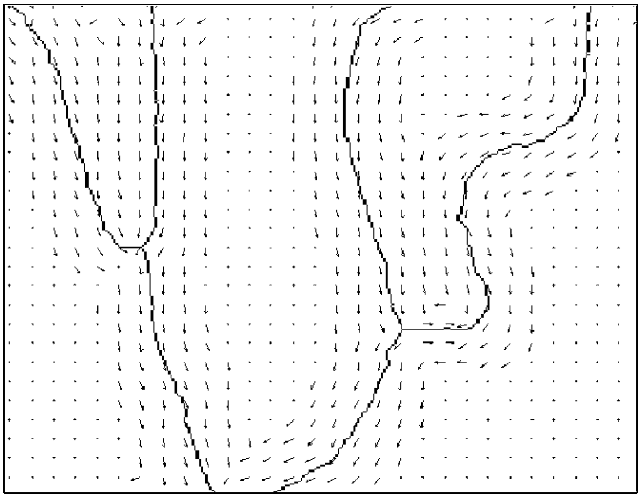


Fig. 7. Vessel centerline and unit vectors indicating the virtual flow direction at various locations in the image.

IV. Local Processing Module

In this section we present a general algorithm for multi-object tracking in circulating blood. The tracker was designed to track two types of targets: "artificial fluorescent microspheres" and "leukocytes." For each target in each frame the tracker returns its velocity, direction and shear rate between the fluid layers and the vessel wall. In the rest of this section, unless specifically mentioned, we will not make a distinction between the tracking of leukocyte cells and fluorescent microspheres and will use the general term "cell" in reference to both.

A. Cell Detection

The cell detection submodule is responsible for locating all the cells in a given frame; it returns their center coordinates and their boundaries. Cells in circulating blood are in constant motion.

A popular approach for locating moving objects is background subtraction and temporal differencing of consecutive frames [33], [34]. Yet, *in vivo* video sequences contain many moving elements besides circulating cells such as vessel boundaries and tissue elements that shift as a result of respiratory movements, fluids that flow outside the vessels as well as circulating erythrocytes within the vessel. Hence, an approach that relies solely on motion detection is likely to suffer from excessive noise. To overcome this problem we execute the following four target detection steps:

1) motion detection;
2) location-based filtering;
3) leukocyte and microsphere recognition performed by ANNs;
4) clustering of adjacent pixels in order to identify cell objects.

The motion detection in Step 1) is performed using background subtraction. In Step 2), any object moving outside a vessel region is assumed to be a false target and is therefore, filtered out. This is done by utilizing the output of the vessel segmentation algorithm described in Section III-B. The result is a list of pixels that belong to moving objects located inside a vessel. This list still contains many false targets and hence further filtering is required. This is done in Step 3) by two ANN classifiers that are trained to classify a pixel as a target or a non-target (one ANN is trained to identify leukocytes and the other, microspheres). The classification is based on color and texture features extracted from the pixel's surrounding 7×7 window. The optimal feature set for training each of the ANNs was automatically selected from a large set of features using the same feature selection procedure as described in Section III-B. In the ANN training and testing a constant threshold was used. After testing a wide range of thresholds, a threshold of 0.5 was chosen since it yielded the highest performance in terms of the correlation coefficient. When cell detection is performed on the video sequences, the classification accuracy is augmented by using a dynamic threshold that is based on posterior image information. A search cone proportional in size and of the same direction as the velocity computed in the previous two frames is applied, similar to [14]. An object within the cone boundaries is more likely to be a real cell and, therefore, the threshold in the cone is lowered. In our application the threshold was lowered by 25%. However, the exact choice for this value is not critical, since lowering the threshold by more than 25% yielded similar results. Step 3) results in a list of pixel locations that are classified as belonging to targets. In Step 4), adjacent pixels are clustered into cell entities. This is efficiently done by scanning the entire image and aggregating all the proximal target pixels using the union-find algorithm. In order to approximate the distance between two pixels that are likely to belong to the same cell, we take into account *a priori* knowledge of target sizes: leukocytes diameter (7 μm – 20 μm), microsphere diameter (1 μm) and knowledge of the microscope magnification ($\times$ 1000) that was used to create the video.

Once the clusters are identified, the cell center is located by computing the center of mass of all the pixels in the cluster [Fig. 8(a)–(c)].

B. Motion Correspondence

Motion correspondence is the task of connecting a target in frame i to the same target in the preceding and following frames. This task is especially difficult when multiple objects with overlapping trajectories and changing appearance all move simultaneously, as is the case in leukocyte/microsphere circulation. Methods for computing motion correspondence of particles and cells such as [1], [9], [14], and [15] have been proposed. In [9], leukocytes were traced through consecutive frames by appearance-based features. It was assumed that the shape of a certain leukocyte changes only slightly between consecutive frames. Many of the leukocytes/microspheres in our video sequences have nonunique and amorphic shapes, and therefore, in order to track them reliably, other features besides appearance should be taken into account. In [15], a motion correspondence algorithm, which maximizes the smoothness of trajectory and velocity of particles, was used in order to perform particle image velocimetry. The algorithm is based on the premise that the trajectory direction and velocity of a moving particle changes only slightly between consecutive frames. For instance, in Fig. 9, the path through $\mathrm{A1} \rightarrow \mathrm{B1} \rightarrow \mathrm{C1}$ is smoother than the path through $\mathrm{A2} \rightarrow \mathrm{B1} \rightarrow \mathrm{C2}$, and is consequently, more likely to be the correct one. This approach can be adopted in order to solve the motion correspondence problem for circulating leukocytes and microspheres. However, it relies solely on local information, i.e., on the locations of the cells in each frame. Its accuracy can be augmented by combining the local information with the global information extracted on the vessel level. The idea is to utilize the virtual flow (described in Section III-C) in the following way: given a cell located at position (x, y) in frame i, we compute the virtual flow $VF(x, y)$ at that location. This gives the general direction in which the cell is most likely to flow in frame $i + 1$. For example, Fig. 10 shows a leukocyte at position A in frame i and two leukocytes at positions B and C in frame $i + 1$. This scenario can be interpreted in two ways:

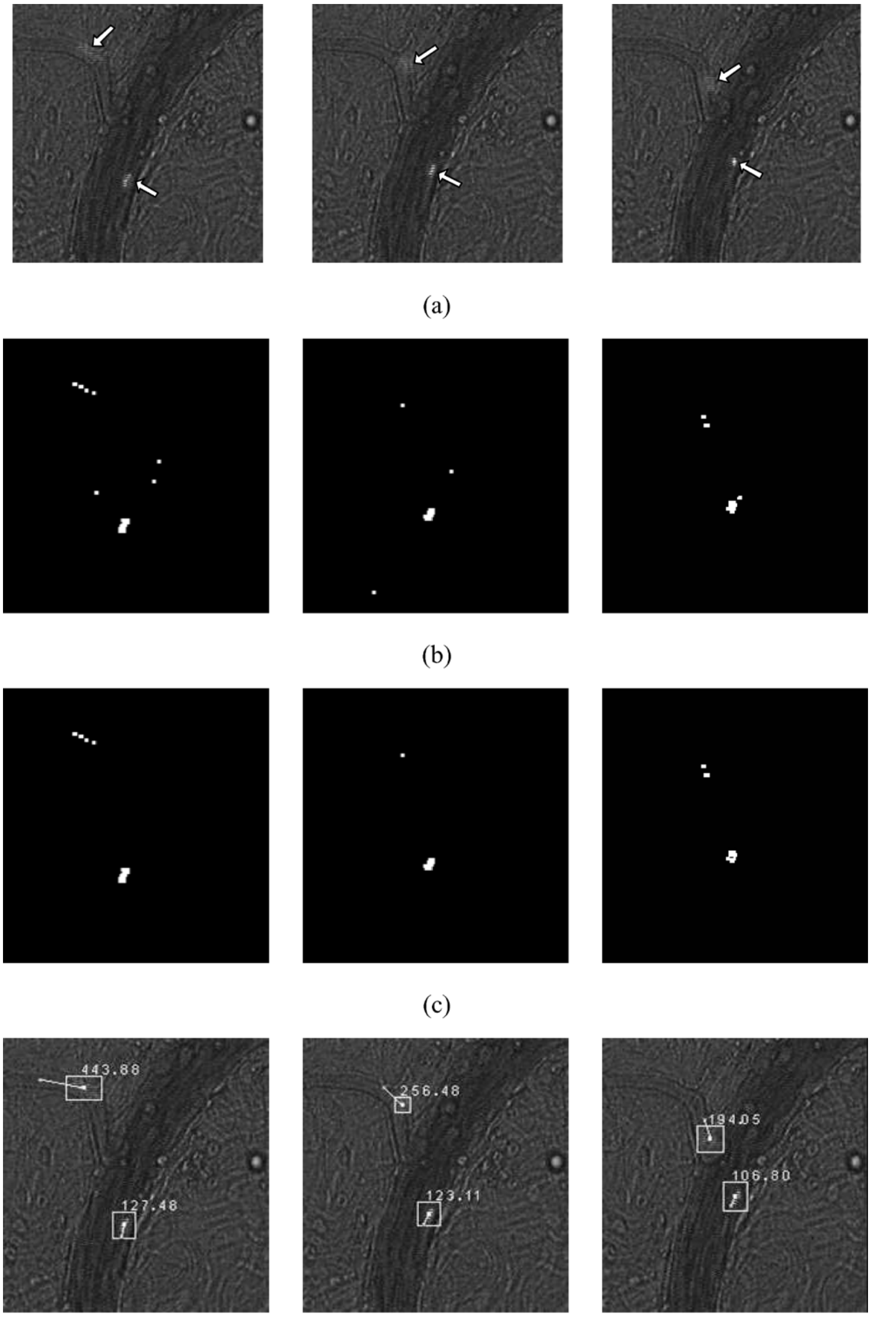
- leukocyte moved from position A to B and a new leukocyte appeared in C;
- leukocyte moved from position A to C and a new leukocyte appeared in position B.

Since the shapes of the leukocytes are similar, the motion correspondence cannot be solved based on target appearance. Nevertheless, it is possible to solve this ambiguity by invoking global information regarding vessel topology. In this case trajectory $\mathrm{A} \rightarrow \mathrm{C}$ is more compatible with the virtual flow (black arrow) than trajectory $\mathrm{A} \rightarrow \mathrm{B}$, and consequently, it is more likely that the leukocyte at position A moved to C. Hence, when trying to determine the path of a cell the algorithm should, in addition to finding the smoothest trajectory and velocity as described in [15], also take into consideration the compatibility of the trajectory with the virtual flow.

Consider three image frames taken in consecutive moments of time: $i - 1$, i and $i + 1$. Our algorithm for motion correspondence is as follows.

- Given a cell B at position (x_B, y_B) in frame i and a circle $\mathcal{C}$ of radius R around it, we define two sets

$$S_{i-1}(B) = \{\text{Cells in frame } i-1 \text{ that appear within the circle } \mathcal{C}\}.$$



(a)

(b)

(c)

(d)

Fig. 8. (a) Small region extracted from three consecutive frames of an *in vivo* video sequence where circulating fluorescent microspheres were studied. (b) Binary images obtained after background subtraction and thresholding, followed by filtering out of any motion that occurred outside a vessel. White areas indicate locations where movement has occurred between sequential frames. Notice that this includes both true as well as false microsphere detections. (c) Binary images after filtering with the microsphere specific ANN. Only the true microspheres remain. (d) The motion correspondence algorithm output. The motion correspondence algorithm output. Each identified microsphere is surrounded by a square. Each line indicates a microsphere trajectory between the current and preceding frames. The line indicates the trajectory between the current and preceding frames. The numbers above the squares are the microsphere velocities ($\mu\mathrm{m/s}$).

$$S_{i+1}(B) = \{\text{Cells in frame } i+1 \text{ that appear within the circle } \mathcal{C}\}.$$

- $\theta_1 = VF(x_A, y_A)$ and $\theta_2 = VF(x_B, y_B)$ are the virtual flow angles at locations (x_A, y_A) and (x_B, y_B) of cells A and B, respectively.
- A legal path of a cell B through frames $i-1$, i and $i+1$ is a triplet A $\rightarrow$ B $\rightarrow$ C where A $\in S_{i-1}(B)$ and C $\in S_{i+1}(B)$.
- In order to resolve the motion correspondence of cell B, that is to find the correct path of traversal through frames $i-1$, i and $i+1$, we perform an evaluation of all possible paths. For each cell A $\in S_{i-1}(B)$ and for each cell C $\in S_{i+1}(B)$ we calculate a motion correspondence functional $J(A, B, C, \theta_1, \theta_2)$

$$J(A, B, C, \theta_1, \theta_2) = \mathrm{AS}(A, B, C) + VS(A, B, C) + \mathrm{VFS}(A, B, C, \theta_1, \theta_2). \quad (6)$$

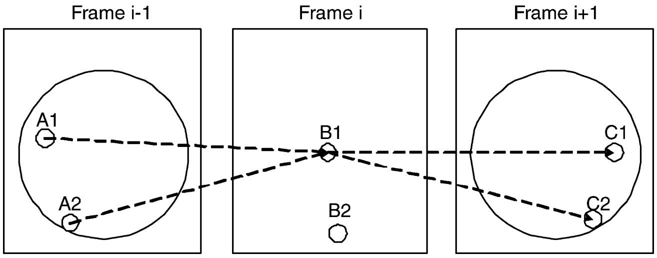


Fig. 9. Motion of cells through consecutive frames. Cell $B1$ in frame i has two possible paths: $\mathrm{A1} \rightarrow \mathrm{B1} \rightarrow \mathrm{C1}$ and $\mathrm{A2} \rightarrow \mathrm{B1} \rightarrow \mathrm{C2}$. The first path is smoother in terms of trajectory angles and should, therefore, be assigned as the correct path by the motion correspondence algorithm.

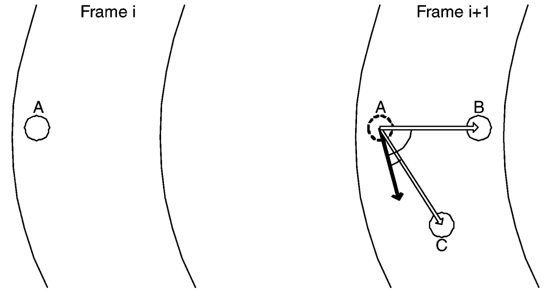


Fig. 10. In frame i a leukocyte appears at position A. In frame i + 1 two leukocytes appear at positions B and C. The trajectory $\mathrm{A} \rightarrow \mathrm{C}$ is more compatible with virtual flow (black arrow) than trajectory $\mathrm{A} \rightarrow \mathrm{B}$ and, therefore, it is more likely that leukocyte A corresponds to C than to B.

- The cell B is assigned the path that attained the minimal score, $\min J(A,B,C,\theta_1,\theta_2)$.

In (6), the angle smoothness (AS), velocity smoothness (VS), and virtual flow smoothness (VFS are

$$\mathrm{AS}(A,B,C) = 1 - \frac{\overline{AB}\cdot\overline{BC}}{\|\overline{AB}\|\cdot\|\overline{BC}\|} \tag{7}$$

$$\mathrm{VS}(A,B,C) = 1 - \frac{2(\|\overline{AB}\|\cdot\|\overline{BC}\|)^{1/2}}{\|\overline{AB}\|+\|\overline{BC}\|} \tag{8}$$

$$\mathrm{VFS}(A,B,C,\theta_1,\theta_2) = 1 - \frac{\cos(\theta_1-\alpha_1)+\cos(\theta_2-\alpha_2)}{2}. \tag{9}$$

The angles α_1 and α_2 are the angles of trajectories AB and AC relative to the positive x-axis. The motivation for choosing the path that achieved the lowest score is that the score function is designed to give low scores for paths with smooth trajectories and constant velocities and whose trajectory direction is compatible with the direction of the virtual flow. The motion correspondence algorithm depends on one parameter R, which is the radius around cell B in which cells are traced in frames $i-1$, and $i+1$. Our tests show that the methods accuracy, and especially its running time, are vulnerable to excessive values of R. This vulnerability is due to the abundance of targets in *in vivo* videos, which can reach up to a thousand circulating cells per frame. A large R means that computing the optimal trajectory of even one cell through three consecutive frames would require computing the scores of over a million possible trajectories. Fortunately, the velocities of leukocytes and microspheres have a physiological upper bound. This enables a reliable restriction of the area in which the optimization should be performed. Although the velocity of a leukocytes/microsphere varies depending on factors such as vessel type, diameter and blood pressure, it is reasonable to assume, based on our measurements in arterioles and venules, that the velocities do not exceed 3000 $\mu\mathrm{m/s}$. Since the camera frame rate and microscope scale are known *a priori*, an upper bound on the initial value of R can easily be computed. Once a target has been tracked over two frames or more, R can be regulated dynamically and more tightly based on the velocity history. Our tests showed that R, which is three times the mean distance travelled by cells in previous frames, gives a sufficiently large safety margin. An example of the output generated by the motion correspondence algorithm when applied on *in vivo* video sequences is given in Fig. 8(d).

V. Experiments

Our method has been tested on nine *in vivo* video-microscopy sequences: five *Type I* video sequences where fluorescent microspheres were tracked and four *Type II* video sequences where circulating leukocytes were tracked. In this section we describe these tests and comparisons to other methods.

A. Testing Vessel Segmentation Accuracy

For assessing the accuracy of the segmentation algorithm we evaluated the performance of the ANN, which was used for segmentation, on the testing examples in the *Segmentation Set* (described in Section III-B). As a measure for quantifying the accuracy of classification we use the correlation coefficient, CC, [28], which is defined as follows:

$$\mathrm{CC} = \frac{T_1T_2 - F_1F_2}{\sqrt{(T_2+F_2)(T_1+F_1)(T_2+F_1)(T_1+F_2)}}. \tag{10}$$

Here, T_1 and T_2 are true classifications of examples from classes 1 and 2, respectively; F_1 and F_2 are false classifications of examples from classes 1 and 2, respectively. The values of the CC range between -1 and 1, where 1 means a perfect classification, 0 means a random classification, and -1 means a completely opposite classification to the correct one. In the segmentation algorithm, T_1 and T_2 correspond to true vessel and true tissue classifications. F_1 and F_2 correspond to false vessel and false tissue classifications.

The ANN performance on the test set was $\mathrm{CC} = 0.82$ (83% of the examples of the vessel class and 98% examples of the tissue class were classified correctly). Examples of two frames taken from two different *in vivo* video microscopy of rat mesenteric microvasculature are shown in Fig. 11(a), (b). They were given as input to the CellTrack segmentation algorithm. The output is shown in Fig. 11(e), (f). We compared the accuracy of the CellTrack segmentation algorithm to two other segmentation algorithms.

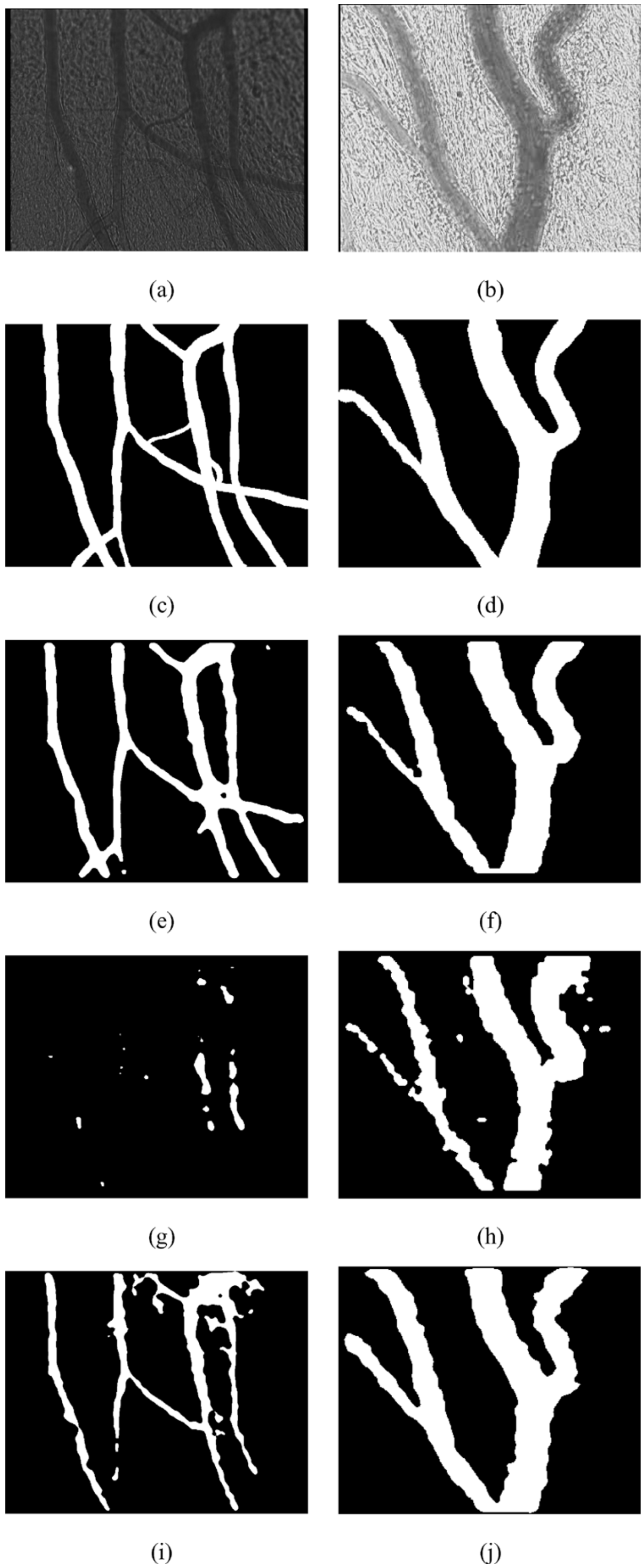


Fig. 11. Comparison of segmentation methods. (a) and (b) are video frames taken from the rat mesenteric microvasculature. (c) and (d) are the manual segmentations of (a) and (b) where black and white represent tissue and vessel regions, respectively. (e)–(j) show a comparison of the output of three different segmentation algorithms: (e) and (f) are the outputs of the CellTrack segmentation algorithm, which utilizes both color- and temporal-based features; (g) and (h) are the output of the temporal-based segmentation algorithm; (i) and (j) are the output of the color texture-based segmentation, which uses Gabor filters [35], [36].

1) Temporal features-based segmentation algorithm.

   We implemented the algorithm for vessel segmentation, presented in [1]. This algorithm is based on the assumption that the variance of gray level intensity computed through multiple frames is higher inside a vessel region than outside it. The algorithm receives a pixels temporal variance (as described in Section III-B) and, using a predefined threshold, classifies it as belonging to either a vessel or tissue region. We applied this algorithm to the testing examples in the *Segmentation Set*. The temporal variance histogram of a video sequence reported in [1] was bi-modal. The inner vessel regions were reported to have distinguishably higher values than outer vessel regions. Hence, for classification, they chose a threshold that best separated this bi-modal histogram. However, when computed on our videos the temporal variance did not yield a bi-modal histogram and, therefore, the threshold could not be determined in this way. In order to keep the comparison fair, we scanned through a wide range of thresholds and calculated the classification accuracy in each instance. The best result was $\mathrm{CC} = 0.09$, which means the classification of pixels was nearly random. These results suggest that the variance assumption does not hold on our video sequences. This may be explained by the inherent noise in the *in vivo* videos caused by the motion of particles and fluids outside the vessel as well as tissue deformations that occur between frames as a result of respiratory movements, all of which cause motion not only inside the vessel but also outside. However, the idea of the temporal variance presented in [1] can still be used to capture more complex temporal patterns that do enable differentiation between vessel and tissue regions. For instance, by computing the standard deviation and homogeneity of the temporal variance in a 7×7 window around a pixel and then feeding the two features into an ANN, it was possible to substantially augment the classification accuracy from $\mathrm{CC} = 0.09$ to $\mathrm{CC} = 0.5628$. This result, however, is still significantly less accurate than that of CellTrack. For a qualitative comparison, the temporal features-based segmentation algorithm was given the same input [Fig. 11(a),(b)] as the CellTrack segmentation algorithm. The corresponding outputs are shown in Fig. 11(g),(h).

2) Color texture features-based segmentation algorithm:

   The second segmentation algorithm with whom a comparison was made is a color texture-based segmentation algorithm that uses Gabor filters [35], [36]. This segmentation is done in a three-step process. In the first step it computes a convolution of a gray level intensity image I with a bank of Gabor wavelets calculated for $s = 4$ different scales and $o = 6$ orientations. Hence, every pixel is transformed into a feature vector of length $\mathrm{o} \times \mathrm{s}$. In the second stage, clustering of these multidimensional vectors is performed by a k-means algorithm [37]. After testing several ks, a $k = 4$ was chosen since it yielded the most accurate results. In the third step a connected component analysis algorithm was applied, followed by morphological operations in order to diminish the influence of noise [38]. As a result, pixels in I were classified into two classes: vessel and tissue.

Unlike the previous two segmentation algorithms, it was not possible to assess and compare the accuracy of this color texture features-based segmentation algorithm in terms of CC on the *Segmentation Set* since it cannot classify individual elements from the set. Fig. 11(a), (b) were given as input to the algorithm. The corresponding outputs are shown in Fig. 11(i), (j).

Fig. 11(e)–(j) shows a qualitative comparison between the output of the CellTrack segmentation, temporal feature-based segmentation and color features-based segmentation algorithms, respectively. In order to perform a quantitative comparison we manually extracted the vessel and tissue regions of Fig. 11(a),(b), which resulted in the two images shown in Fig. 11(c),(d). We assumed that Fig. 11(c),(d) were the correct segmentations and used them as a reference for comparisons. These images contain a total of 516 018 pixels where 98 683 are of the vessel class and 417 335 are of the tissue class. The outputs of the three algorithms were compared to the reference images by subtracting Fig. 11(e),(f), (g),(h), and (i),(j) from Fig. 11(c),(d) and counting the true and false pixel classifications. CellTrack yielded the highest accuracy with a $\mathrm{CC} = 0.87$. The temporal feature-based segmentation yielded the lowest accuracy of $\mathrm{CC} = 0.43$ and the color-based segmentation yielded a $\mathrm{CC} = 0.71$. A detailed comparison of the confusion matrices is given in Table II. The confusion matrices are calculated as follows: $\mathbf{CM} = \begin{pmatrix} T1/(T1+F2) & F2/(T1+F2) \\ F1/(T2+F1) & T2/(T2+F1) \end{pmatrix}$ where T_1 and T_2 are true classifications of elements from class 1 (vessel class) and class 2 (tissue class), respectively. F_1 and F_2 are false classifications of elements from classes 1 and 2, respectively. It can be seen that all of the segmentation algorithms applied on Fig. 11(b) yielded accurate results [Fig. 11(f), (h), (j)], in which case there was no clear advantage to any of the segmentation algorithms. However, segmentation of Fig. 11(a) was substantially more difficult, as a result of similar colors and texture in the tissue and vessel region as well as noisy temporal patterns. All the algorithms yielded less accurate results [Fig. 11(e), (f), (i)]. CellTrack segmentation, however, yielded the most accurate segmentation.

B. Testing Cell Detection Accuracy

Two data sets were created, the *Microsphere Set* and the *Leukocyte Set*. Each element in the sets is a 7 × 7 pixel window. The *Microsphere Set* is made up of 300 elements, each containing a microsphere and 300 elements of regions with no microspheres, i.e., these are either vessels or tissue. The *Leukocyte set* is made up of 300 elements that contain a leukocyte and 300 windows of regions that did not contain a leukocyte. Similar to the segmentation algorithm (Section III-B), the sets were divided into mutually exclusive training and testing sets, of sizes 200 and 100, respectively, on which the ANNs were trained and tested. The accuracy of the classification into leukocyte/microsphere and not-leukocyte/not-microsphere classes, performed on the test sets, in terms of the correlation coefficient is $\mathrm{CC} = 0.8867$ and $\mathrm{CC} = 0.8555$ for the microspheres and leukocytes, respectively. The confusion matrices are given in Table III. The cell detection module incorporates, in addition to the ANNs, other filters such as the motion detection filter and the ANN dynamic threshold (Section IV-A), both of which increase identification accuracy. Therefore, the ANN accuracy given in Table III should be regarded as a lower bound to the method's cell detection performance.

TABLE II
COMPARING THE CELLTRACK SEGMENTATION ALGORITHM, WHICH IS BASED ON BOTH TEMPORAL AND COLOR FEATURES, THE TEMPORAL FEATURES-BASED SEGMENTATION ALGORITHM AND THE COLOR TEXTURE-BASED SEGMENTATION ALGORITHM. IN THE COMPARISON 516 018 PIXELS WERE CLASSIFIED INTO VESSEL (98,683 PIXELS) AND TISSUE (417 335 PIXELS) CLASSES

	CM		CC
CellTrack	0.87	0.13	0.87
segmentation	0.02	0.98	
Temporal features	0.26	0.74	0.43
based segmentation	0.01	0.99	
Color texture features	0.67	0.33	0.71
based segmentation	0.03	0.97	

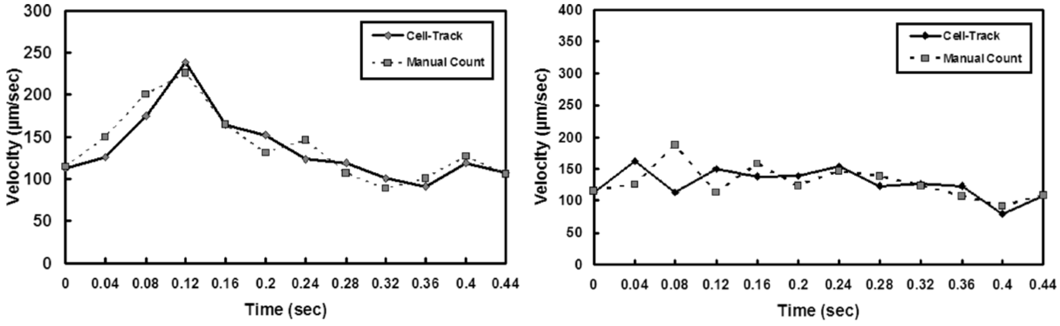
TABLE III
ACCURACY OF THE ANN CLASSIFICATION INTO TWO CLASSES: LEUKOCYTE/MICROSPHERE CLASS AND NOT-LEUKOCYTE/NOT-MICROSPHERE CLASS

	CM		CC
Microsphere	1.00	0.00	0.89
classification	0.07	0.93	
Leukocyte	0.87	0.13	0.86
classification	0.03	0.97	

C. Tracking Accuracy

In order to assess the accuracy of the CellTrack tracker, manual measurements performed on the monitor screen of microspheres and leukocyte flow velocities were compared to measurements extracted by the CellTrack system. We performed the following experiments.

1) Single microspheres were tracked through consecutive frames and manual measurements of their velocities were compared to the computer generated measurements. Results are summarized in Fig. 12.
2) Five different vessel regions were selected from three different video sequences. In each instance the velocities of all the leukocytes or microspheres that flowed through that region were extracted twice, manually and automatically, using CellTrack. Based on these measurements the average flows of the leukocytes and microspheres were calculated for each region. The results are summarized in Fig. 13, showing a compatibility of over 97% between manual and computer extracted statistics.
3) The CellTrack tracker uses a synergistic approach that combines global and local information to decrease tracking errors that occur repeatedly in the inherently noisy *in vivo* video sequences. In particular, it uses the *a priori* knowledge of the vessel topology when trying



(a) (b)

Fig. 12. Two instances of manual versus CellTrack extracted measurements of a single microsphere velocity traced through consecutive frames.

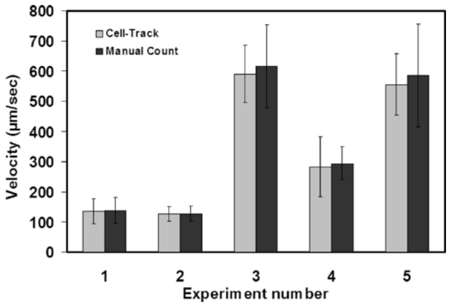


Fig. 13. Flow velocity for fluorescent microspheres measured in differentially sized microvessels from five experiments. No significant statistical differences were found between manually and automatically extracted measurements using CellTrack.

to solve motion correspondence by taking into account the compatibility of the cell trajectory direction and the virtual flow direction. To test the contribution of the virtual flow to the motion correspondence solution, we performed the following test: 171 trajectories of leukocytes were tracked twice – with the virtual flow module enabled ("regular mode") and disabled (in the latter case the motion correspondence algorithm as described in Section IV-B remains the same but the virtual flow smoothness is omitted from the functional in (6)). In both instances we counted the number of times in which the motion correspondence was solved correctly and falsely (compared to the manual solution). The virtual flow enabled tracker yielded an accuracy of 94% (164 correct and 7 false assignments) while the virtual flow disabled tracker had an accuracy of 88% (151 correct assignments and 20 false assignments), hence showing the contribution of the virtual flow to the motion correspondence accuracy.

4) A comparison was made between the CellTrack motion correspondence algorithm and the IPAN general purpose motion correspondence algorithm implemented in [39]. The IPAN algorithm utilizes a cost function that enforces trajectory and velocity smoothness among three points in consecutive frames. Multiple backward-forward testing on the selected best triplet of points is then performed in order to enhance accuracy and resolve ambiguities. The comparison was performed in the following way: first, the CellTrack cell detection module was applied to generate target coordinates. Second, these coordinates were used as input to IPAN and CellTrack motion correspondence algorithms. The predicted trajectories were compared to manually determined trajectories in the two types of videos. In the tested Type I video sequence, where the image is of high quality and the targets sparse, both algorithms predicted the trajectories without any error. In the tested Type II video, IPAN correctly assigned 93% of the trajectories (40 out of 43) while CellTrack correctly assigned 97% of the trajectories (42 out of 43). Overall, both algorithms produce robust trajectory predictions, especially in cases where the quality of the video sequence is high and target coordinates are reliable. However, in cases where target coordinates extraction is noisy, the usage of vessel topology constraints in the functional enabled CellTrack to filter out wrong trajectories that cannot be filtered out by a general purpose motion correspondence algorithm.

5) We demonstrate how the CellTrack system can be applied in order to test a biological hypotheses by extracting a large volume of data. Two *in vivo* video sequences were taken from mesenteric micro vessels before occlusion (closure of a vessel) and following reperfusion (removal of occlusion and restoration of blood flow). Reperfusion injury usually increases the number of leukocytes and slows their velocity due to adhesion. Thus one would expect a relatively low count of leukocytes with high velocities in the video taken before occlusion compared to a high number of leukocytes with low velocities in the video taken following reperfusion. To test this, CellTrack was applied on these two video sequences. A $25 \times 75\ \mu\mathrm{m}$ region inside a vessel was defined. An automated leukocytes count and computation of their mean velocity was performed in this region before occlusion and following reperfusion. Before occlusion the average leukocytes count per frame was 4.4 (1112 leukocytes traced in 250 consecutive frames) compared to 20.1 (4659 leukocytes traced in 232 consecutive frames) following reperfusion. The mean velocity was 32.9 $\mu\mathrm{m/s}$ before occlusion compared to 25.2 $\mu\mathrm{m/s}$ after reperfusion. These results are in agreement to what is expected following reperfusion injury of the gut. Moreover, automatic extraction of these measurements was performed in 2 min, significantly faster compared to the time it would have taken to extract this volume of data using manual counts.

VI. Conclusion

In this paper, we present a system, CellTrack, for analysis of circulating leukocytes and microspheres in *in vivo* video sequences. Tracking targets in *in vivo* video sequences is a very challenging task. The major problems are as follows. 1) The shift that occurs between sequential frames as a result of movements of the tissue caused by respiration. 2) The poor quality of *in vivo* video records. One of the main factors that reduces the quality is video instability and defocusing that occurs as a result of respiratory movement. 3) The abundance of objects, which may include hundreds of leukocytes in each video frame. 4) The targets' amorphic and blurred shapes and their sporadic appearance and disappearance between frames, which makes the motion correspondence problem difficult to solve. Any system aimed at extracting reliable and accurate measurements must incorporate robust solutions to all of the above problems.

To address these problems we adopted an approach that synergizes between the extraction of global information performed on the vessel level and the extraction of local information performed on the circulating cell/microsphere level. The main modules of the CellTrack system described in this paper include: a vessel segmentation algorithm, a target detection algorithm and an algorithm for solving the motion correspondence problem of circulating cells and microspheres. The system was tested on nine *in vivo* video sequences. The CellTrack vessel segmentation algorithm, which uses both temporal and color-based features, was compared qualitatively and quantitatively to two other segmentation algorithms that use either temporal or color-based features. The CellTrack segmentation algorithm yielded the most accurate results, CC $= 0.87$ compared to CC $= 0.43$ and CC $= 0.71$ for the temporal and color feature-based segmentations, respectively. This suggests that the utilization of temporal texture patterns in addition to the color texture patterns leads to significantly more robust and accurate segmentations in *in vivo* videos sequences. We examined the contribution of incorporating vessel topology considerations (i.e., the virtual flow) into the motion correspondence functional. In one test the motion correspondence accuracy rose from 88% to 94% correct assignments once the topology considerations were added, thus supporting their relevancy. A series of tests comparing manually extracted measurements and CellTrack extracted measurements show a compatibility of 97% in the video sequences under study. CellTrack also enables the extraction of a large volume of data in a short period of time (thousands of measurements in ~ 10 min), compared to tedious and time consuming manual extraction ($\sim$ 50-100 measurements in approximately 60 min). We have made examples of the *in vivo* color video sequences used in this study and the corresponding CellTrack output publicly available at www.cs.technion.ac.il/rudzsky/~celltrack1.html

Appendix
Classification Features

In this section the full set of classification features used in Sections III-B and IV-A is given. The features are extracted from a $\mathrm{N} \times \mathrm{N}$ window, W, surrounding the classified pixel (in the application we used $\mathrm{N} = 7$ pixels). We distinguish between two classes of features.

- **Intensity features**, as their name implies, are based on the intensities of the pixels in the window surrounding the classified pixel. We used the following intensity features:
  1) Median $m = med(W)$;
  2) Mean $\mu = (1/N^2)\sum_{x,y} W(x,y)$;
  3) Variance $\sigma^2 = (1/N^2)\sum_{x,y}(W(x,y) - \mu)^2$.
- **Texture features** characterize spatial correlations between values of pixel pairs and are calculated using a co-occurrence matrix, as defined in (5). Let $P(i,j)$ be the (i,j) element of a co-occurrence matrix. We use the following notations and features adopted from [30], [31]:

$$\mu_x = \sum_i i \sum_j P(i,j)$$
$$\mu_y = \sum_j j \sum_i P(i,j)$$
$$\sigma_x^2 = \sum_i (i - \mu_x)^2 \sum_j P(i,j)$$
$$\sigma_y^2 = \sum_j (j - \mu_y)^2 \sum_i P(i,j).$$

For symmetric matrices, $\mu_x = \mu_y = \mu$.

The texture features we used are as follows:

4) Energy $E = \sum_{i,j} P(i,j)^2$;
5) Entropy $S = -\sum_{i,j} P(i,j)\log P(i,j)$;
6) Homogeneity $H = \sum_{i,j}(1/(1 + (i-j)^2))P(i,j)$;
7) Inertia $In = \sum_{i,j}(i-j)^2 P(i,j)$;
8) Prominence $Pr = \sum_{i,j}(i + j - \mu_x - \mu_y)^4 P(i,j)$;
9) Correlation $Corr = -\sum_{i,j}((i-\mu_x)(j-\mu_y)/\sqrt{\sigma_x^2\sigma_y^2})P(i,j)$
10) Shade $Sd = \sum_{i,j}(i + j - \mu_x - \mu_y)^3 P(i,j)$
11), 12) Texture variance σ_x^2, σ_y^2

Each of the above features is computed over five types of window patches $W_R(x,y)$, $W_G(x,y)$, $W_B(x,y)$, $W_{GR}(x,y)$ and $W_{tmp}(x,y)$ corresponding to the RGB, gray level and temporal variance values as described in Section III-B. Hence the full set of features contains $5 \times 12 = 60$ features. This set is then reduced to a smaller feature subset using the feature selection algorithm as described in Section III-B.

References

[1] Y. Sato, J. Chen, R. A. Zoroofi, N. Harada, S. Tamura, and T. Shiga, "Automatic extraction and measurement of leukocyte motion in microvessels using spatiotemporal image analysis," *IEEE Trans. Biomed. Eng.*, vol. 44, no. 4, pp. 225–236, Apr. 1997.

[2] M. E. Martinez-Perez, D. Hughes, A. V. Stanton, S. A. Thom, N. Chapman, A. A. Bharath, and K. H. Parker, "Retinal vascular tree morphology: A semi-automatic quantification," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 8, pp. 912–917, Aug. 2002.

[3] J. Tang and S. T. Acton, "Vessel boundary tracking for intravital microscopy via multi-scale gradient vector flow snakes," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 2, pp. 316–324, Feb. 2004.

[4] H. Greenspan, M. Laifenfeld, S. Einav, and O. Barnea, "Evaluation of center-line extraction algorithms in quantitative coronary angiography," *IEEE Trans. Med. Imag.*, vol. 20, no. 9, pp. 928–941, Sep. 2001.

[5] L. Lam, S. W. Lee, and C. Y. Suen, "Thinning methodologies: A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869–885, Sep. 1992.

[6] N. Ray, S. T. Acton, and K. Ley, "Tracking leukocytes *in vivo* with shape and size constrained active contours," *IEEE Trans. Med. Imag.*, vol. 21, no. 10, pp. 1222–1235, Oct. 2002.

[7] T. Twellmann, T. W. Nattkemper, W. Schubert, and H. Ritter, "Cell detection in micrographs of tissue sections using support vector machines," presented at the ICANN 2001 Workshop on Kernel and Subspace Methods for Computer Vision, Vienna, Austria, Aug. 25, 2001.

[8] S. T. Acton, K. Wethmar, and K. Ley, "Automatic tracking of rolling leukocytes *in vivo*," *Microvasc. Res.*, vol. 63, pp. 139–148, 2002.

[9] M. Egmont-Petersen, U. Shreiner, S. C. Tromp, T. M. Lehmann, D. W. Slaaf, and T. Arts, "Detection of leukocytes in contact with the vessel wall from *in vivo* microscope recordings using a neural network," *IEEE Trans. Med. Imag.*, vol. 47, no. 7, pp. 941–951, Jul. 2000.

[10] T. W. Nattkemper, H. Ritter, and W. Schubert, "A neural classifier enabling high-throughput topological analysis of lymphocytes in tissue sections," *IEEE Trans. Inform. Technol. Biomed.*, vol. 5, pp. 138–149, Jun. 2001.

[11] J. Hektor, J. Janse, H. Heidtmann, H. Schmid-Schonbein, and R. Grebe, "A setup for computer-assisted tracking of erythrocytes in micro vascular networks," *J. Comput. Assist. Microsc.*, vol. 9, no. 3, pp. 135–141, 1997.

[12] D. P. Mukherjee, N. Ray, and S. T. Acton, "Level set analysis for leukocyte detection and tracking," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 562–572, Apr. 2004.

[13] Y. Sato, J. Chen, S. Tamura, N. Harada, and T. Shiga, "Measuring microcirculation using spatiotemporal image analysis," in *Proc. Int. Conf. Computer Vision, Virtual Reality, Robotics in Medicine*, Nice, France, Apr. 1995, pp. 302–308.

[14] D. Chetverikov and J. Verestóy, "Feature point tracking for incomplete trajectories," *Comput.*, vol. 62, pp. 321–338, 1999.

[15] D. Chetverikov, "Particle image velocimetry by feature tracking," *Comput. Anal. Images Patterns*, vol. 2124, pp. 325–332, 2001.

[16] D. Waisman, V. Brod, R. Wolff, E. Sabo, M. Chernin, Z. Weintraub, A. Rotschild, and H. Bitterman, "Effects of hyperoxia on local and remote microcirculatory inflammatory response following splanchnic ischemia and reperfusion," *Am. J. Physiol. Heart. Circ. Physiol.*, vol. 285, pp. 643–652, 2003.

[17] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Jun. 1986.

[18] C. J. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conf.*, 1988, pp. 147–151.

[19] S. T. Acton and A. C. Bovik, "Piecewise and local image models for regularized image restoration using cross-validation," *IEEE Trans. Image Process.*, vol. 8, no. 5, pp. 652–665, May 1999.

[20] B. Zitova and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, pp. 977–1000, 2003.

[21] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, "Mutual information based registration of medical images: A survey," *IEEE Trans. Med. Imag.*, vol. 22, no. 8, pp. 986–1004, Aug. 2003.

[22] M. Mirmehdi and M. Petrou, "Segmentation of color textures," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 2, pp. 142–159, Feb. 2000.

[23] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Ann. Rev. Biomed. Eng.*, vol. 2, pp. 315–337, Aug. 2000.

[24] T. W. Nattkemper, "Automatic segmentation of digital micrographs: A survey," in *Proc. 11th World Congr. Medical Informatics (MEDINFO), Pt. 2*, 2004, pp. 847–851.

[25] J. N. Said, M. Cheriet, and C. Y. Suen, "A recursive thresholding technique for image segmentation," *IEEE Trans. Image Process.*, vol. 7, no. 6, pp. 918–921, Jun. 1998.

[26] B. Warner and M. Misra, "Understanding neural networks as statistical tools," *Am. Statist.*, vol. 50, no. 4, pp. 284–293, Nov. 1996.

[27] B. Cheng and D. M. Titterington, "Neural networks: A review from a statistical perspective," *Statist. Sci.*, vol. 9, no. 1, pp. 2–30, Feb. 1994.

[28] P. Baldi and S. Brunak, *Bioinformatics-The Machine Learning Approach*. Cambridge, MA: MIT Press, 2001.

[29] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[30] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 610–621, Nov. 1973.

[31] R. F. Walker, P. Jackway, and I. D. Longstaff, "Improving co-occurrence matrix feature discrimination," presented at the Proc. DICTA '95, 3rd Conf. Digital Image Computing: Techniques and Applications, Brisbane, Australia, Dec. 6–8th, 1995.

[32] Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Commun. ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[33] I. Haritaoglu, D. Harwood, and L. S. Davis, "W4: Who? When? Where? What? A real time system for detecting and tracking people," in *Proc. AFGR98, IEEE Int. Conf. Auto. Face and Gesture Recognition*, 1998, pp. 222–227.

[34] A. McIvor, Q. Zang, and R. Klette, "The background subtraction problem for video surveillance systems," in *Int. Workshop Robot Vision 2001*, Auckland, New Zealand, 2001, pp. 176–183.

[35] T. S. Lee, "Image representation using 2D Gabor-wavelets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 10, pp. 959–971, Oct. 1996.

[36] Y. Rubner and C. Tomasi, *Perceptual Metrics for Image Database Navigation*. Norwell, MA: Kluwer, 2000, vol. 594.

[37] G. B. Coleman and H. C. Andrews, "Image segmentation by clustering," *Proc. IEEE*, vol. 5, pp. 773–785, 1979.

[38] R. Gonzalez and R. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.

[39] J. Verestóy and D. Chetverikov. Feature point tracking algorithms. [Online]. Available: http://visual.ipan.sztaki.hu