

Time-Lapse microscopy and image processing for stem cell research modeling cell migration

Tomas Gustavsson^a, Karin Althoff^a, Johan Degerman^a, Torsten Olsson^a,
Ann-Catrin Thoreson^b, Thorleif Thorlin^b, Peter Eriksson^b

^aDepartment of Signals and Systems, Chalmers University of Technology, S-412 96
Göteborg, Sweden

^bInstitute of Clinical Neuroscience, Göteborg University, S-412 96 Göteborg, Sweden

ABSTRACT

This paper presents hardware and software procedures for automated cell tracking and migration modeling. A time-lapse microscopy system equipped with a computer controllable motorized stage was developed. The performance of this stage was improved by incorporating software algorithms for stage motion displacement compensation and auto focus. The microscope is suitable for *in-vitro* stem cell studies and allows for multiple cell culture image sequence acquisition. This enables comparative studies concerning rate of cell splits, average cell motion velocity, cell motion as a function of cell sample density and many more. Several cell segmentation procedures are described as well as a cell tracking algorithm. Statistical methods for describing cell migration patterns are presented. In particular, the Hidden Markov Model (HMM) was investigated. Results indicate that if the cell motion can be described as a non-stationary stochastic process, then the HMM can adequately model aspects of its dynamic behavior.

1. INTRODUCTION

Until a few years ago the prevailing dogma in neuroscience stated that no neurons could be created in the brain after the developmental period. We now know that this is not true. The birth of new neurons from neuronal stem cells, a process called neurogenesis, has been seen in adult brains of both rodents and humans^{1,2}. The discovery of neurogenesis in the brain has awakened a hope for treatment and cure of many severe brain diseases where the loss of brain cells is the cause of the damaged brain functions. However, little is known about the basic regulatory mechanisms of neurogenesis as yet, and intense research efforts are made to better understand these mechanisms. In the attempts to understand this regeneration of brain cells experimental designs using cell cultures are of uttermost importance. Under culture conditions the properties of neuronal stem cells can be characterized and the cells can be manipulated under well controlled conditions. One crucial question in these studies is what becomes of the neural stem cells? What new cell types are generated from the stem cells and can these be forced to increase the amount of a specific cell type that is considered desirable and beneficial for therapeutically applications? These kinds of studies are called cell lineage analyses and they are important for the future clinical use of stem cells. In these experiments large amounts of cells have to be followed over time and there is an urgent need for the development of refined forms of time-lapse techniques and analysing methods. During a basic time-lapse session many cells are followed for a specific time. Traditionally time-lapse studies are time consuming in that images from one cell culture at a time are recorded during an extended period of several days and a series of identical experiments are then performed to obtain consistence and repeatability in the findings. Technical solutions that enable the time-lapse system to record simultaneously from several culture wells during one single experiment will dramatically increase the data available and speed up considerably the process of lineage analysis. Furthermore, the time-lapse data can be used for statistical analyses of cell migration based on cellular movement patterns during different culture conditions. The engineering challenges are numerous and this paper will describe approaches and solutions to many of the technical problems involved as well as some preliminary results of significant biological relevance. This includes the *time-lapse microscopy hardware setup, software methods for controlling the motorized stage, segmentation procedures, cell tracking algorithms, and statistical procedures for analyzing cell migration.*

2. MATERIALS AND METHODS

2.1 Materials

The clonal population was received at passage 4 as a gift from Dr Fred Gage (Laboratory of Genetics, The Salk Institute, La Jolla, CA, USA). The cells were cultured in Dulbecco's Modified Eagle's medium (MEM)/Nut Mix F12 plus 2.5 mM L-glutamine and N2 supplement (N2 medium, GIBCO Invitrogen Corp., Paisley, Scotland, UK). Twenty ng/ml human basic FGF-2 (Peprotech Inc., Rocky Hill, NJ, USA) was used to keep the cultures proliferating. Cells were used between passages 5 and 20 post-cloning.

2.2 Time-lapse microscopy setup

The time-lapse microscopy hardware setup consists of a microscope (Leica DM IRB, Leica Microsystems) equipped with a programmable motorized stepper stage (H 101, Prior Scientific Inc.), a sample holder for six (or up to 24) culture wells, an incubator with temperature and gas controls (Solent Scientific), a cooled (thermoelectric) high-resolution digital camera (Microimager II, Quantitative Imaging Inc) with FireWire (IEEE1394) interface, and a conventional PC for steering the camera and stage control as well as storing the images on disk for subsequent processing. Figure 1 presents a flow chart of the time-lapse microscopy hardware setup.

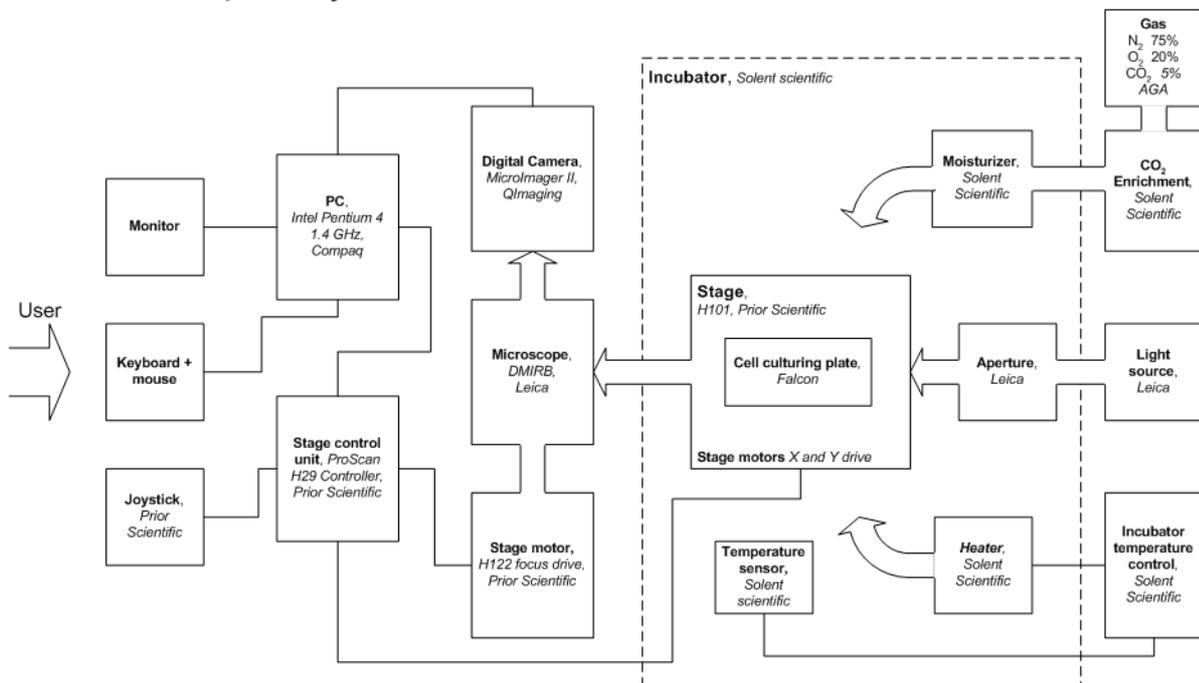


Figure 1. The time-lapse microscopy hardware setup.

2.2.1 Stage control

Since the sample holder in our experiments allowed for six (or up to 24) culture wells, it was possible to move the stage around performing almost simultaneous image capture from each of these wells. In order to evaluate the stage performance, the concepts of accuracy, repeatability, and resolution needs to be considered. *Accuracy* is defined as the difference between requested and actual motion performed by the linear motion device. *Repeatability* is defined as the ability to reproduce a given linear motion. *Resolution* is defined as the minimal movement or step size that the device is capable of performing. Usually, the resolution of the x and y axes are different from that of the z axis or focus resolution. According to the manufacturer's General Specifications, the repeatability of our stage is +/- 1 micron, the minimal step size (resolution) is 0.04 micron, and the accuracy is +/- 3 microns.

In a series of preliminary experiments, the stage was moved in systematic motion patterns as well as randomly. For each start and stop positions, images presenting easily detected geometrical landmarks were captured by the high-resolution camera. It was observed that there is a consistent linear positioning error of the order of a few pixels. In order to compensate for this, we applied an optical flow technique³ that used two time-consecutive images as input for calculating the image displacement. The images were then aligned by a simple translation procedure.

2.2.2 Autofocusing

It is well known that moving the stage in the xy-plane may lead to a focus distortion in the z-direction. Hence, for computer controlled moving stage applications there is a need for a real-time auto focus algorithm. Our auto focus algorithm was based on an analysis of the magnitude of the Fourier spectrum. A quantitative auto focus function was determined by a summation of the high frequency components of the Fourier magnitude. For each stop position in the xy-plane, the stage was moved along the z axis repeatedly until the focus function was maximized. In that position, the image was captured for storage.

2.2.3 Frame capture and storage

The real-time image capture and storage software was implemented using the C programming language. This software enabled the operator to specify the number of frames to capture, the size of each frame, the inter-frame delay, as well as a number of camera parameters such as gain and offset. The image capture and storage software was communicating with the auto focus software resulting in a fully automated image sequence acquisition program.

2.2.4 Digitized cell image sequences

The first data set, *stem1*, consists of 400 frames and in each frame 94 cells were tracked. The second data set, *stem2*, consists of 400 frames and 26 cells were tracked. For both data sets, the image size was 1280/2 x 1024/2 (2 x 2 binning was applied), the temporal resolution was 30 frames per hour, and the exposure time was 50 ms. *Stem2* was captured at a higher degree of microscopic resolution. Figure 2 shows example frames from the two data sets.

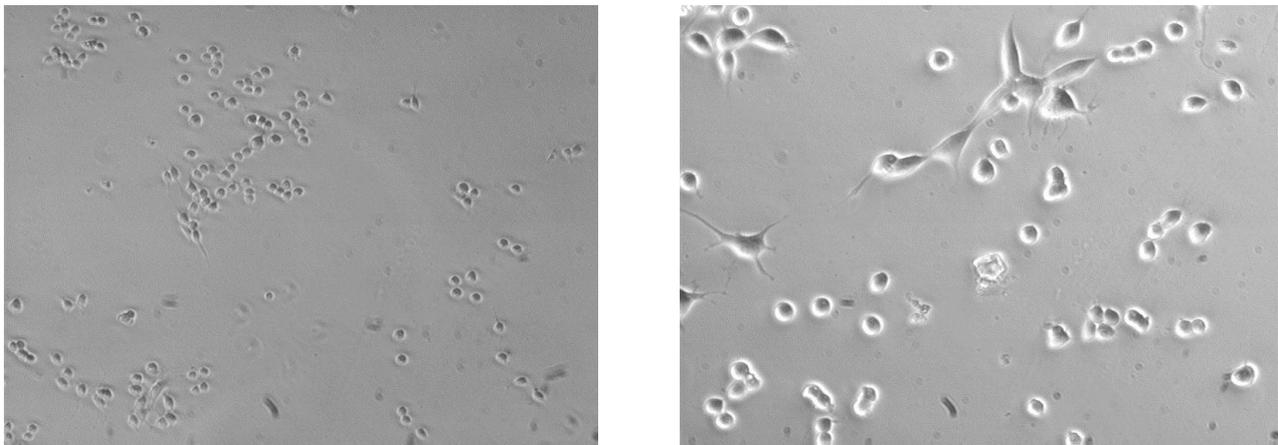


Figure 2. Example frames of the data sets *stem1* (left image) and *stem2* (right image)

2.3 Image segmentation

For details about this section see [4].

2.3.1 Removal of uneven illumination

As an image preprocessing step, each frame of the time-lapse sequence was subtracted by a background image. This background image was constructed by modeling the image intensities by a second-order polynomial.

2.3.2 Image intensity and variance thresholding

The thresholding methods of Otsu⁵, and Kittler & Illingworth⁶ were applied. Based on the results of Wu et. al⁷. and Glasbey⁸, the method of Kittler & Illingworth was chosen for intensity thresholding. This was because the intensity distributions of object and background in our images do not have identical variances. Variance thresholding was carried out by first computing a local image variance measure for each pixel position and then applying a threshold. In this case, we used the method of Otsu simply because it produced a more plausible result for our images. The intensity and variance thresholded images were combined by (binary) summation, small objects were removed and holes were filled. The boundaries of the binary objects were detected and overlaid the original image. The resulting image is shown in Figure 3.

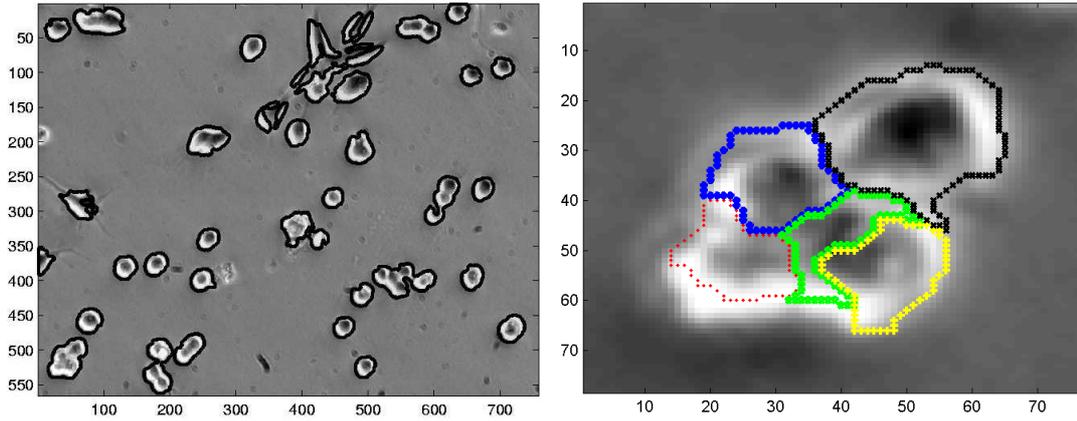


Figure 3. Original image with overlaid detected object contours (left image) and a cell cluster with overlaid detected contours extracted by the watershed algorithm (right image).

2.3.3 Segmenting individual cells

Several different methods including snakes⁹, the watershed¹⁰ algorithm and methods based on the cells' circularity were investigated for segmenting individual cells. The most promising so far was the watershed algorithm. The basic version should then be supplemented by a subsequent merging step in case of over segmentation. The algorithm has the advantage of not requiring any information on the cell's position in a previous image. This is crucial in cases when cell splits or suddenly enters the scene. Figure 3 presents an example of a segmented cluster using the watershed algorithm.

2.4 Cell tracking algorithm

For each segmented cell in frame j , a Region-Of-Interest (ROI) ω is determined to search for the cell in frame $j+1$. Those cells that are found inside ω constitute the candidates.

For each cell i of frame j , and every candidate m of frame $j+1$, there is a weight, w . This weight is defined as the product of two statistically independent contributions:

$$w_i^m = \underbrace{C(p_{i,j}, c_{m,j+1})}_{\text{Correlation Contribution}} \cdot \underbrace{P(p_{i,j} \rightarrow c_{m,j+1})}_{\text{Movement Contribution}} \quad (1)$$

The correlation contribution is a measure of the similarity determined by correlating the pixels of a rectangular region including cell i at position $p_{i,j}$ in frame j and the pixels of identically sized regions including $c_{m,j+1}$ in frame $j+1$ and positioned inside ω_i .

The movement contribution corresponds to the probability that the cell has moved from position $p_{i,j}$ to position $c_{m,j+1}$. The numerical value of this contribution is determined from a statistical model of the cell movement. This model may vary between experiments depending on the types of cells that are investigated. Typically, the movement contribution is inversely proportional to a function of the distance between $p_{i,j}$ and $c_{m,j+1}$. Once the weights linking all cells of frame j

to every candidate of frame $j+1$ have been established, the optimal and unique linking is obtained by the Viterbi algorithm. In order to make the tracking algorithm applicable for routine use, it was implemented in a user-friendly Java programming environment making extensive use of the Java Advanced Imaging libraries¹¹. Figure 4 illustrates the tracking algorithm.

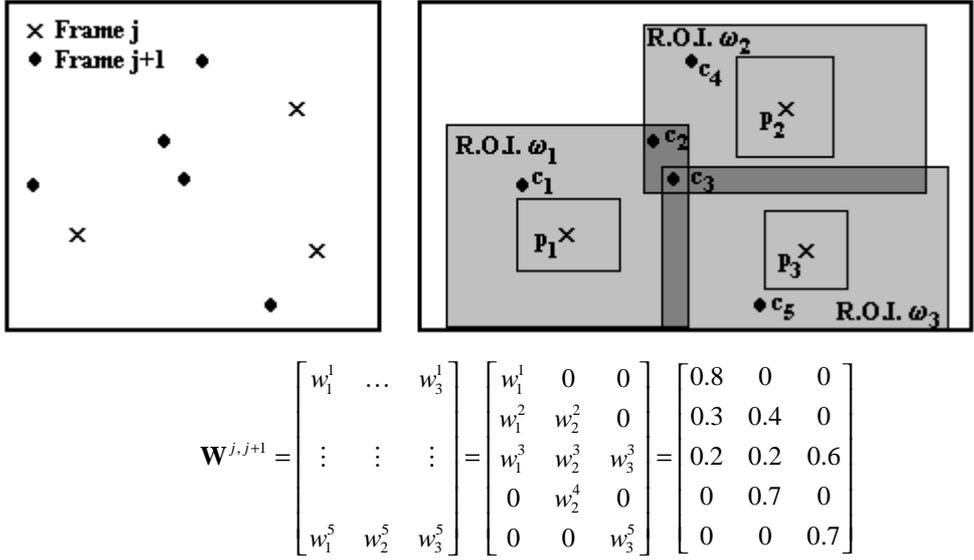


Figure 4. An example with three cells, five candidates, and the cell linking weight matrix, illustrating the tracking algorithm.

2.5 Cell migration modeling

This section describes a number of motion modeling techniques each of which may be applicable for characterizing cell migration or as an aid for accurately tracking the cells. Preliminary studies carried out by our group¹² indicated that the Hidden Markov Model (HMM) might be appropriate for describing the sudden change in cell motion behavior. Therefore, this modeling technique is described in more detail.

2.5.1 Brownian motion modeling

By a Brownian motion, the motion components in the x- and y- direction are statistically independent and normally distributed with zero mean.

2.5.2 The ARMA model

Given that the cells are being tracked over time, we obtain for each cell a description of the cell migration in terms of speed, speed persistence, movement direction, etc. Each of these aspects of cell migration can be described by a time-varying signal $y(t)$. For the purpose of statistical modeling, we can view the signal $y(t)$ as being the output of a causal linear time invariant (LTI) filter $T(q)$ driven by an independent stationary white noise source $e(t)$. Then, we can model the signal $y(t)$:

$$y(t) = T(q) \cdot e(t) \tag{2}$$

An ordinary LTI system can be described by the recurrence equation:

$$y(t) = a_1 \cdot y(t-1) + \dots + a_n \cdot y(t-n) = c_0 \cdot e(t) + c_1 \cdot e(t-1) + \dots + c_m \cdot e(t-m) \tag{3}$$

$$y(t) = \frac{C(q)}{A(q)} \cdot e(t) = T(q) \cdot e(t) \tag{4}$$

where $A(q) = 1 + a_1 \cdot q^{-1} + a_2 \cdot q^{-2} + \dots + a_n \cdot q^{-n}$, and $C(q) = c_0 + c_1 \cdot q^{-1} + c_2 \cdot q^{-2} + \dots + c_m \cdot q^{-m}$. If the signal $e(t)$ is a white stochastic sequence, then the equation above is called an autoregressive-moving average ARMA(n,m) model, where n is

the order of the autoregressive part and m is the order of the moving average part of the model. The parameters of the models can be estimated by conventional least squares techniques.

2.5.3 First-order autoregressive model

For the case when $m = 0$ we have the autoregressive process of order n , (i.e., an AR(n) process).

$$y(t) = A(q) \cdot e(t) \quad (5)$$

Intuitively, the first-order autoregressive model, i.e. AR(1) (i.e., $A(q) = 1 + a_1 \cdot q^{-1}$), should tell us something about the short-term “memory” of the cell’s motion behavior.

2.5.4 Change detection algorithms

When a signal undergoes a sudden change, it is important to be able to detect this change without too much delay. The design of change detection algorithms splits into two tasks:

Generating residuals The idea to exploit is that the spectrum of the signal change is white when there is no significant change, whereas coloured (with a higher mean value) otherwise.

Design of decision rules This task is to design suitable decision rules for determining the changes as reflected by their residuals.

The advantage with applying decision rules is that *a priori* information about the signal statistics is not necessarily required. Hence, the class of CUSUM (short for *cumulative sum*) algorithms is built on statistical properties which are retrieved directly from the input data lending themselves for real-time applications. The following CUSUM algorithm due to Gustafsson¹³ was implemented:

$$\begin{aligned} \hat{\theta}_t &= \frac{1}{t - t_0} \cdot \sum_{k=t_0+1}^t y_k \\ \varepsilon_t &= y_t - \hat{\theta}_{t-1} \\ s_t^1 &= \varepsilon_t \\ s_t^2 &= -\varepsilon_t \\ g_t^1 &= \max(g_{t-1}^1 + s_t^1 - \nu, 0) \\ g_t^2 &= \max(g_{t-1}^2 + s_t^2 - \nu, 0) \end{aligned} \quad (6)$$

Alarm if $g_t^1 > h$ or $g_t^2 > h$.

After each alarm, there is a reset operation: $g_t^1 = 0$, $g_t^2 = 0$ and $t_0 = t$. The design parameters ν and h should be tuned to the properties of the signal at hand.

2.5.5 Kalman filtering

A model, represented in state-space form, is described by the equations

$$\begin{aligned} x(k+1) &= A \cdot x(k) + G \cdot e(k) \\ y(k) &= C \cdot x(k) \end{aligned} \quad (7)$$

where the first equation is known as the *state equation* and the second as the *measurement equation*. The system describes an input-output relation where $x(k)$ is the input and $y(k)$ is the output of the system. The vector $x(k)$ also contains the states that describe the system (e.g. position and velocity). The matrix A determines how to predict the next state vector from the current state vector, C describes the linear combination of $x(k)$ which forms the output $y(k)$, and

G describes the noise characteristics. For tracking applications, it is possible to combine multiple Kalman models corresponding to different motion patterns.

2.5.6 Hidden Markov modeling

Hidden Markov Modeling (HMM) may be used for describing the statistics of non-stationary stochastic processes. They consist of a finite number of states and at each time instance the process belongs to one of these states. Transition probabilities govern the likelihood that the process is changing from one state to the other or remains in the same state. Also, each state includes parameters of the probability density functions associated with the stochastic process that belongs with that state. Figure 5 presents an illustration of a two state HMM.

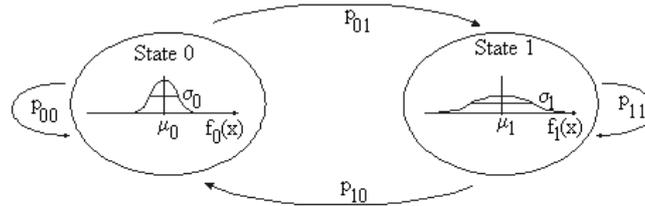


Figure 5. The figure presents a two state HMM with the transition probabilities, and the probability density functions associated with the two states.

In order to define an HMM completely, the following elements are needed.

- The number of states of the model, N .
- The number of possible observations, M . If the observations are continuous then M is infinite.
- A set of state transition probabilities, $\Lambda = \{p_{ij}\}$, where

$$p_{ij} = P\{x_{t+1} = j | x_t = i\}, \quad (8)$$

$$i, j \in [1, 2, \dots, N]$$

and x_t denotes the state at time t . Transition probabilities should satisfy the conventional constraints:

$$\sum_{j=1}^N p_{ij} = 1, \quad (9)$$

$$p_{ij} \geq 0, \quad (9)$$

$$i, j \in [1, 2, \dots, N]$$

- A probability distribution associated with each of the states,

$$B = \{b_j(k)\}, \quad (10)$$

$$b_j(k) = P\{o_t = v_k | x_t = j\}, \quad (10)$$

$$\begin{cases} j \in [1, 2, \dots, N] \\ k \in [1, 2, \dots, M] \\ t \in [1, 2, \dots, T] \end{cases}$$

where v_k denotes the k^{th} possible observation, and o_t the observation at time t . The probability distribution should satisfy the conventional constraints:

$$\begin{aligned} \sum_{k=1}^M b_j(k) &= 1, \\ b_j(k) &\geq 0, \\ j &\in [1, 2, \dots, N], k \in [1, 2, \dots, M] \end{aligned} \quad (11)$$

- The initial state distribution, $\boldsymbol{\pi} = \{\pi_i\}$, where

$$\pi_i = P\{x_1 = i\}, i \in [1, 2, \dots, N] \quad (12)$$

Applying these definitions, we can use the compact notation $\boldsymbol{\lambda} = (\Lambda, B, \boldsymbol{\pi})$ to denote an HMM with discrete probability distributions. Furthermore, we apply the following assumptions:

Markov assumption: We assume that the next state is dependent only upon the current state (i.e., the model is a first order HMM).

Stationary assumption: We assume that state transition probabilities are independent of time

$$P\{x_{t_1+1} = j | x_{t_1} = i\} = P\{x_{t_2+1} = j | x_{t_2} = i\} \quad (13)$$

for any t_1 and t_2 .

Output independence assumption: We assume that the current output (observation) is statistically independent of the previous outputs (observations).

Given an HMM, $\boldsymbol{\lambda}$, and a sequence of observations $O = o_1, o_2, \dots, o_T$, we need to solve the following three problems:

Learning problem: How can we adjust the model parameters, $\{\Lambda, B, \boldsymbol{\pi}\}$, in order to maximize $P\{O | \boldsymbol{\lambda}\}$?

Evaluation problem: What is the probability that the observations are generated by the model, $P\{O | \boldsymbol{\lambda}\}$?

Decoding problem: What is the most likely state sequence of the model that produced the observations?

To solve the evaluation problem we apply the forward algorithm proposed by Vaseghi¹⁴. This recursive algorithm brings down the complexity from a N^T (using conventional techniques) to a N^2T problem. The learning problem is solved by applying a *Maximum Likelihood* criterion for determining the model parameters, so that we obtain an optimal fit between training data (observations) and model. The iterative Baum-Welch algorithm¹⁴ does this with guaranteed convergence. Basically, the transition probabilities $p_{i,j}$ are estimated as:

$$p_{i,j} = \frac{\text{Expected number of transitions from state } i \text{ to state } j}{\text{Expected number of transitions from state } i}, \quad (14)$$

and the state probabilities $b_j(k)$ as:

$$b_j(k) = \frac{\text{Expected number of times in state } i \text{ and observing } v_k}{\text{Expected number of times in state } i} \quad (15)$$

To solve the decoding problem we apply the Viterbi algorithm. Given the observed sequence $O = \{o_1, o_2, \dots, o_T\}$ and the model $\boldsymbol{\lambda} = (\Lambda, B, \boldsymbol{\pi})$, the recursive Viterbi algorithm finds the most likely state sequence $X = \{x_1, x_2, \dots, x_T\}$ by applying a maximum likelihood optimization criterion.

3. RESULTS

Since the descriptions of time-lapse microscopy hardware and software design, segmentation techniques, and implementation of a cell tracking algorithm may be considered results on their own right, this section will mainly focus on the cell migration modeling aspects. Indeed, the performance of these modeling techniques is strongly dependent on the performance of the image acquisition, cell segmentation, and tracking modules. Nevertheless, here we will consider the cells as already being accurately digitized, segmented, and tracked, enabling us to focus solely on extracted cell position data given as a function of time.

3.1 Brownian motion modeling

Cell motion from both data sets (*stem1*, *stem2*) was tested for normality using Pearson's χ^2 test. Although the p-values of this test were not consistently in favor of a Gaussian motion distribution, they at least gave an indication that this model would be a plausible starting point. Figure 6 shows a realization of one single cell migration and their corresponding displacement histogram and correlation diagram. Although the quantization is coarse, it can be seen that motion distributions seem to have zero, or close to zero, means with a certain deviation. Also, correlation is close to zero but for lag zero, indicating that there is very little correlation between two consecutive movements. Both observations are in favor of a Gaussian distribution, i.e. a Brownian motion.

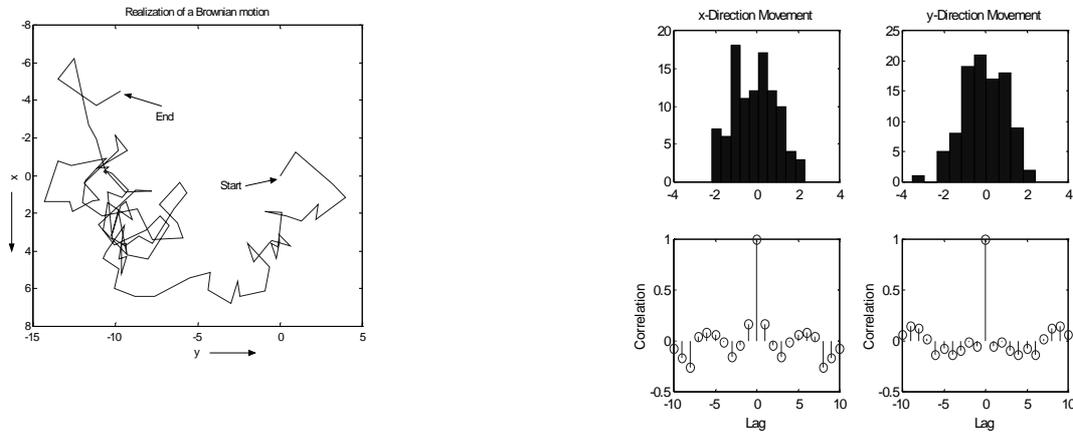


Figure 6. Realization of a single cell migration (left image), displacement histograms (upper right image) and correlation diagrams (right lower image) for the x and y movement components, respectively.

3.2 Change detection algorithm

In order to test the CUSUM algorithm described in Section 2.5.4, 10 synthetic data sets including 60 samples each was created as follows:

$$\begin{aligned} p(t) &= 0.45 + \varepsilon(t), \quad 1 \leq t < 30 \\ p(t) &= 0.55 + \varepsilon(t), \quad 30 \leq t \leq 60 \end{aligned} \quad (16)$$

where $\varepsilon(t) \sim N(0, 0.1)$.

Notice that the level of the abrupt change in the signal mean value is of the order of the variance of the signal noise. Running the algorithm using the 10 data sets as input, the algorithm gave a total of 20 false alarms (on average two alarms per test). The response time (the time between the true and detected change) never exceeded three samples. The result of a typical test run is shown in Figure 7. It can be seen that a total of three false alarms were detected (i.e., alarms appearing before the true change).

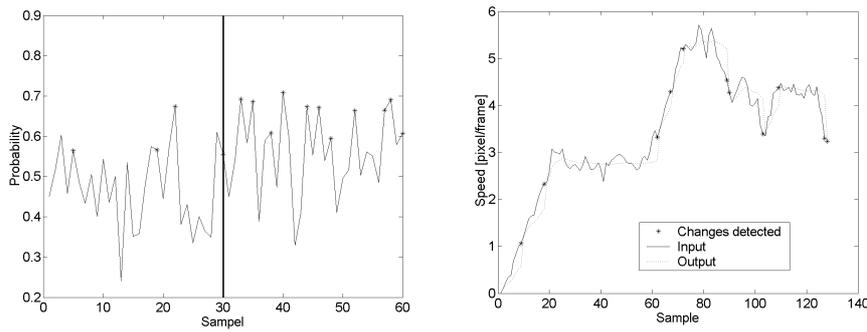


Figure 7. The left image illustrates of the performance of the CUSUM algorithm applied to a noisy signal. The level of the abrupt change in the signal mean value is of the order of the variance of the signal noise. Still, the algorithm gives only three false alarms before the change occurs. The right image shows the results of the algorithm as being applied to a signal representing the velocity of the cell.

3.3 Kalman filtering

A summary of the results applying Kalman filtering for predicting cell positions is given in Table 1.

Filter	Mean Square Prediction Error
No filter (just predicting position using the last position)	9.55 pixels ²
Filter with one state/dimension (position)	9.55 pixels ²
Filter with two states/dimension (position and velocity)	13.0 pixels ²
Two filters with model switching	9.51 pixels ²
Filter based on ARMA-model	8.91 pixels ²

Table 1. Prediction errors for different types of Kalman filters as applied to cell position and velocity data. For this experiment, a total number of 50 cells from 118 frames each were used.

As can be seen from the results, the Kalman filter with only position included in the state space description produces results identical to the case in which no filter at all was applied. This is because the best estimate one can make with a state space only including position and no measurement noise is to use the last position as the estimated position. Since the cell motion pattern was almost Brownian, it is believed that Kalman filtering for predicting positions is of limited value for this particular application. For details on testing the Kalman filtering, the reader is referred to [15].

3.4 Hidden Markov modeling

For details about this section see [16].

3.4.1 Verifying the modeling software using synthetic data

The synthetic data were constructed by generating 100 sequences of Poisson processes ($\lambda_1 = 1, \lambda_2 = 3$) including 400 samples each. The initial state and transition probabilities were chosen as:

$$\pi = [0.8 \quad 0.2]$$

$$\Lambda = \begin{bmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{bmatrix}$$

Two sequences were chosen to provide the initial state probabilities B:

$$B = \begin{bmatrix} 0.3000 & 0.2725 & 0.2025 & 0.1250 & 0.0575 & 0.0425 \\ 0.1150 & 0.1950 & 0.2350 & 0.1900 & 0.1100 & 0.1550 \end{bmatrix}$$

for the value range [0 1 2 3 4 5-inf]. The initial state and transition probabilities were chosen to be:

$$\pi = [0.67 \quad 0.33]$$

$$\Lambda = \begin{bmatrix} 0.95 & 0.05 \\ 0.10 & 0.90 \end{bmatrix}$$

After training with another 88 sequences the state probabilities B changed to

$$B = \begin{bmatrix} 0.3899 & 0.3383 & 0.1961 & 0.0648 & 0.0080 & 0.0103 \\ 0.0480 & 0.1245 & 0.3363 & 0.1884 & 0.1384 & 0.1882 \end{bmatrix}$$

whereas π and Λ changed to

$$\pi = [0.7618 \quad 0.2382]$$

$$\Lambda = \begin{bmatrix} 0.9667 & 0.0333 \\ 0.0443 & 0.9557 \end{bmatrix}$$

Figure 8 shows the initial state probabilities and the same probabilities after training.

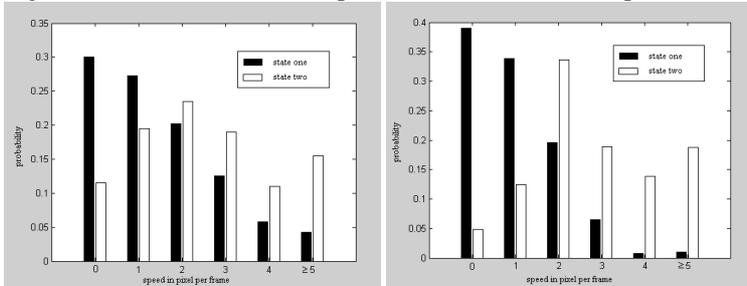


Figure 8. Initial state probabilities (left image) and state probabilities after training (right image) of the synthetic data set.

With the trained model parameters the states of the remaining 10 sequences of the synthetic data set were decoded. Figure 9 shows the values of one of these synthetic sequences as well as its original states and the states recognized by the trained HMM. As can be observed, the states recognized by the model fit the original states for all 10 evaluation sequences very well. Only when two state transitions occur quickly after one another they are not recognized by the model because they hardly influence the output signal. In summary, the modeling software offers excellent results when it is trained with data based on two different states.

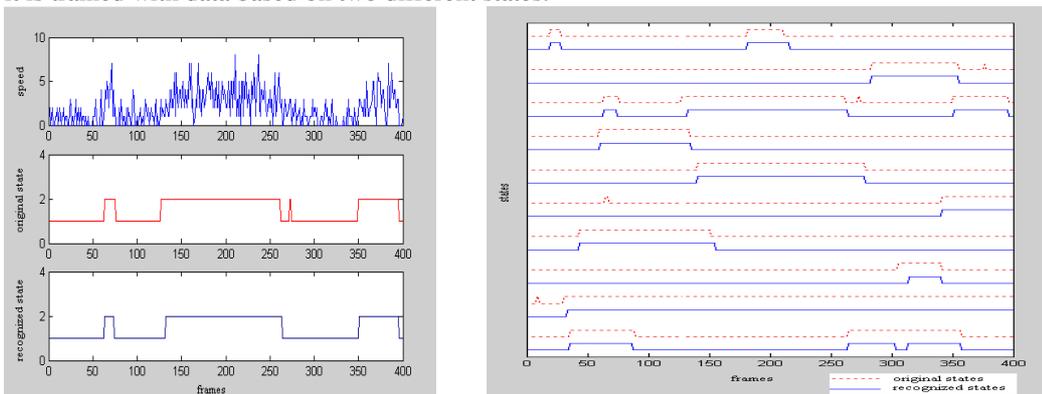


Figure 9. One sequence of the synthetic data, its original states and the states recognized by the trained HMM (left image). Original states of 10 synthetic data compared with the states recognized by the HMM.

3.4.2 Hidden Markov modeling of cell velocity

The HMM was applied to velocity data of the data set *stem1*. Figure 10 shows the initial state probabilities and the state probabilities after training.

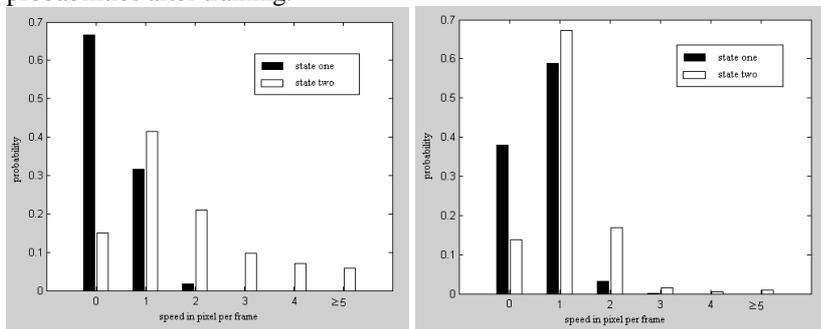


Figure 10. The initial state probabilities (left image) and the state probabilities after training (right image) of the velocity data of *stem1*.

Although the difference between state 1 and state 2 probabilities decreased as a result of the training, it was still possible to decode the sequence. Figure 11 presents the decoded states.

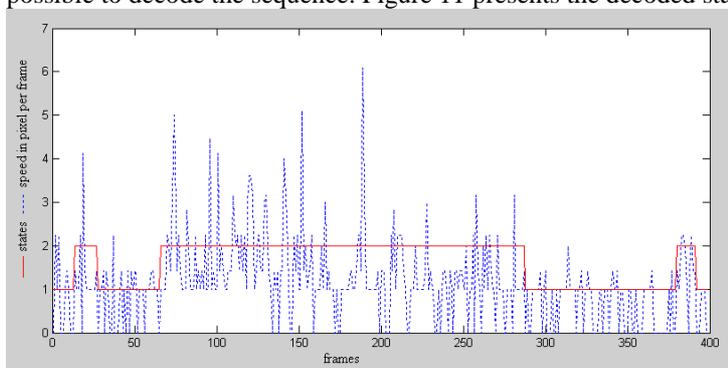


Figure 11. The decoded states of the velocity data from one cell of *stem1*.

The HMM was then applied to velocity data of the data set *stem2*. Figure 12 shows the initial state probabilities and the state probabilities after training.

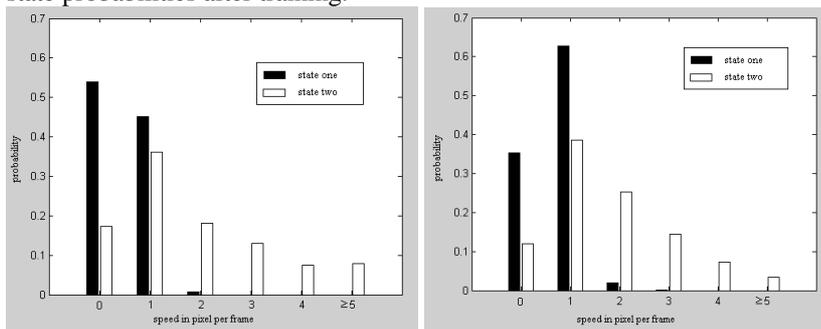


Figure 12. The initial state probabilities (left image) and the state probabilities after training (right image) of the velocity data of *stem2*.

The difference between the state 1 and state 2 probabilities for *stem2* decreased as a result of the training, but not as much as for *stem1*. The reason for this improved performance may be that the image resolution is higher for *stem2*. Figure 13 presents the decoded states.

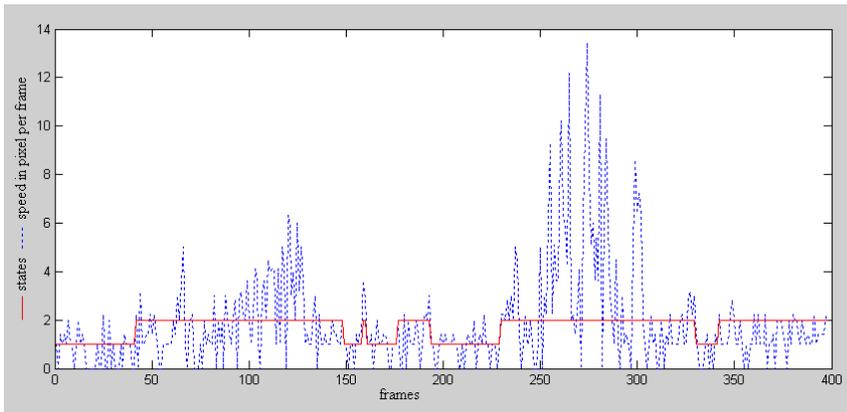


Figure 13. The decoded states of the velocity data from one cell of *stem2*.

The results appear to be in agreement with a subjective interpretation of the potential state changes.

In order to estimate the AR(1) coefficients, the number of samples per estimation needs to be specified. Using 5 samples per estimate resulted in a somewhat noisy signal but it was still possible to decode the state sequence. As could have been expected, using 100 samples per estimate gave a smoother signal. The range of coefficient values was $[-1,0]$. The training and state sequence decoding results are presented in Figure 14 and Figure 15, respectively.

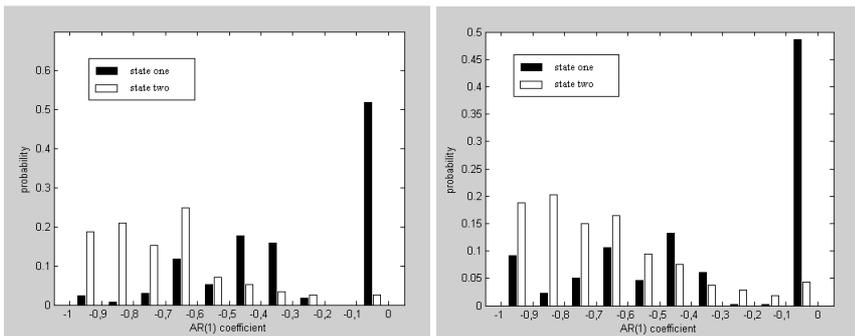


Figure 14. The initial state probabilities (left image) and the state probabilities after training (right image) of the AR(1) velocity data of *stem1*.

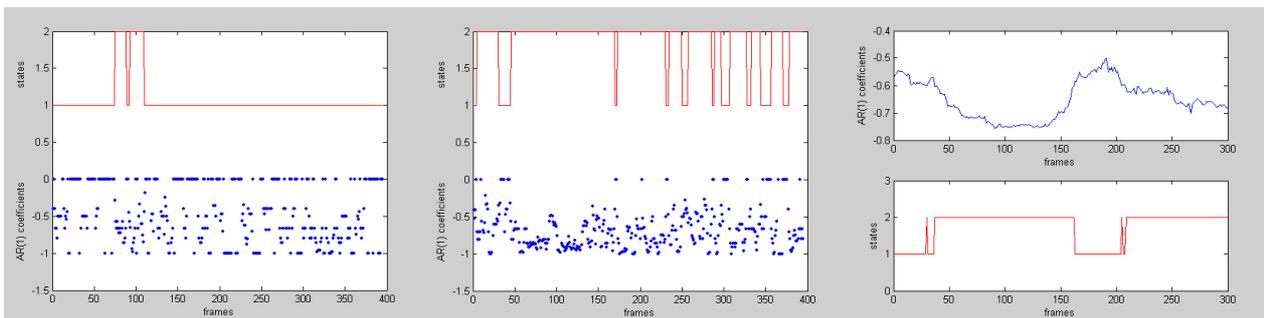


Figure 15. The decoded states (given the training results presented in Figure 14) of velocity AR(1) for a cell that is mostly in state 1 (left image) and for another cell that is mostly in state 2 (middle image). The same type of results but using 100 samples per coefficient estimate (right image).

Similar experiments as for velocity and AR(1) data were carried out using motion direction as input. The results did not indicate states associated with a preferred direction. Even if the average velocity was clearly different between the two states, there were no preferred movement directions for cells in neither of the two states. This observation is in agreement with the Brownian motion hypothesis.

4. DISCUSSION

Cell tracking and migration modeling is a very complex task. It ranges from hardware for data acquisition to software for advanced statistical modeling. Each of the intermediate processing modules (preprocessing, segmentation, detection of cell splits, etc.) is crucial to the overall system performance. In this paper, several techniques have been presented for solving many of the problems in relation to cell tracking and migration modeling.

In one of our previous reports¹², we presented the hypothesis that stem cells during real-time experiments under culture conditions have the potential to undergo sudden changes in migration behavior. By subjectively inspecting a few sets of cells, it became clear that some of the cells suddenly changed their motion pattern from slow to fast or the other way around. Whereas motion direction appeared to be random, the average step length seemed to increase significantly as a result of the immediate change. Following some preliminary statistical tests supporting the idea of applying a Brownian motion model for describing the cell motion, we followed up by introducing the Hidden Markov Model as a means for describing the two different motion states and the transition between them. In order to get to the point where these hypothetical models could be tested in practice, a number of hardware and software procedures had to be developed.

The performance of the computer-controllable motorized stage was improved by incorporating algorithms for stage motion displacement compensation and auto focus. Both procedures are prerequisites for an accurate image sequence acquisition.

Several cell segmentation algorithms have been presented in this paper. Because cell segmentation is strongly linked to cell tracking, our tracking algorithm guided the segmentation procedure by predicting the cell position. This showed to be the key to success most of the time. Nevertheless, the segmentation problem has not been solved entirely. There are cases of touches, overlaps, and clustering that still challenge most of the state-of-the-art segmentation algorithms that we investigated. However, for a cell tracking system to be used by biologists in their routine work, we strongly believe that the tracking algorithm should allow for some sort of user interaction for resolving difficult tracking ambiguities. As long as this interaction can be done easily, we foresee no problem with a semi-automated system rather than a fully automated system.

The cell tracking algorithm presented works reasonably well for most of the sequences we have tried so far. It is easy to use due to the Java user interface. Although tracking errors do sometimes occur, it is easy to correct these manually. The algorithm can be improved by incorporating more effective motion models. Hitherto, the movement contribution discussed in section 2.4 was based on a simple inter-cell distance measure. By replacing this measure with a more powerful motion model such as any of those described in Sections 3.2-3.4, an improved tracking performance can be anticipated.

The Hidden Markov Model seems to have a great potential for describing the statistical variations seen in the stem cell motion pattern. The validity of the model is conditional to the hypothesis that the cell motion signal has got the appropriate statistical properties (i.e., it should be possible to describe it as a non-stationary stochastic process). The experiments carried out supported this hypothesis. The HHM model presented in this paper should be evaluated further so that its pros and cons can be explained more explicitly. We plan to carry out many more experiments using multiple cell cultures that will be systematically modified (e.g. by different types of growth hormones) for enabling comparative concerning rate of cell splits, average cell motion velocity, cell motion as a function of cell sample density and many more. This will give us the necessary test data for evaluating the HMM and possibly other modeling techniques.

5. CONCLUSIONS AND FUTURE WORK

We have presented a computerized system for automated cell tracking and cell migration modeling. The time-lapse microscopy setup including a motorized stage allowed for capturing image sequences of multiple cell cultivations. Several algorithms for cell segmentation, tracking, and motion modeling have been presented. The HMM model proved to be capable of accurately describe the statistical differences between stem cells featuring different migration patterns.

There are many exciting possibilities for developing our microscopy and motion modeling techniques. Concerning microscopy sample preparation, cells from different time-lapse experiments can be stained for cell type specific markers. These can identify the number of different cell types, such as neurons, astrocytes or oligodendrocytes, which are present in the culture at the end of the experiment. The data sequences from the time-lapse studies can then be used to track backwards events on the single cell level combined with the knowledge of end point cell type specificity. As for the mathematical modeling of cell migration, further laboratory experiments will certainly bring new ideas to be tested and verified. The HMM modeling presented in this paper represents the beginning of a most exciting series of investigations into advanced mathematics and statistics for applications to modern stem cell research.

ACKNOWLEDGEMENTS

This project was partly funded by the Swedish Foundation for Strategic Research (SSF) under the VISIT program, No. A30-960626, and the Swedish Research Council (VR), No. 621-2001-3531. The authors are grateful for the contributions made by the following undergraduate students: Torbjörn Ahlqvist, Rikard Nordgren, Johan Lindgren, Peter Pap de Pestény, Martina Speck, and André Bieberle.

REFERENCES

1. P.S. Eriksson, E. Perfilieva, T. Bjork-Eriksson, A.M. Alborn, C. Nordborg, D.A. Peterson, F.H. Gage, "Neurogenesis in the adult human hippocampus", *Nat. Med.*, **4**,1313-1317, 1998.
2. E. Gould, A.J. Reeves, M.S. Graziano, C.G. Gross, "Neurogenesis in the neocortex of adult primates", *Science*, **286**, 548-552, 1999.
3. B. Horn, B. Schunck, "Determining Optical Flow", *Artificial Intelligence*, **17**,185-203, 1981.
4. K. Althoff, *Implementation and evaluation of stem cell segmentation techniques*, Technical report Ex001/2003, Department of Signals and Systems, Göteborg, 2003.
5. N. Otsu, "A Threshold Selection Method from Gray-level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-9:1**, 62-66, 1979.
6. J. Kittler, J. Illingworth, "Minimun Error Thresholding", *Pattern Recognition*, **19:1**, 41-47, 1986.
7. K. Wu, D. Gauthier, M.D. Levine, "Live cell image segmentation", *IEEE Transactions on Biomedical Engineering*, **42:1**, 1-12, 1995.
8. C.A. Glasbey, "An analysis of histogram-based thresholding algorithm", *Graphical Models and Image Processing*, **55:6**, 532-537, 1993.
9. M. Kass, A. Witkin, D. Terzopoulos, "Snakes: Active Contour Models", *International Journal of Computer Vision*, **1:4**, 321-331, 1987.
10. S. Beucher, "The Watershed Transformation applied to Image Segmentation", *Scanning Microsc. Suppl.*, **6**, 299-314, 1992.
11. <http://java.sun.com/products/java-media/jai/>
12. T. Ahlqvist, *Automated Neural Cell Tracking & Modelling*, Master Thesis, Department of Signals and Systems, Chalmers University of Technology, Göteborg, 2001.
13. F. Gustafsson, M. Millnert, L. Ljung, *Signalbehandling*, Studentlitteratur, Lund, 2000.
14. S.V. Vaseghi, *Advanced Signal Processing and Digital Noise Reduction*, John Wiley & sons Ltd., New York and B.G. Teubner, Leipzig, 1996.
15. J. Lindgren, P. Pap de Pestény, *Stem Cell Tracking and Modelling*, Master Thesis LiU-IMT-Ex-329, Department of Biomedical Engineering, Linköpings University, Linköping.
16. M. Speck, A. Bieberle, K. Althoff, T. Gustavsson, *An investigation into Hidden Markov Modeling for applications to cell migration*, Technical report Ex002/2003, Department of Signals and Systems, Göteborg, 2003.