Transactions on Mass-Data Analysis
of Images and Signals
Vol. 11, No 1 (2020) 27-53
© P-ISSN1868-6451, E-ISSN 2509-9353
ISBN 978-3-942952-80-4

ibai Publishing
www.ibai-publishing.org

Expert Opinion on Automatic Image Analysis for Cellular and Molecular Biology and Drug Discovery

Petra Perner

FutureLab Artificial Intelligence IBaI-2, Radeberg, Germany
pperner@futurelab-ai-ibai-2.de

Abstract. In the rapidly expanding fields of cellular and molecular biology, fluorescence illumination and observation is becoming one of the techniques of choice to study the localization and dynamics of proteins, organelles, and other cellular compartments, as well as a tracer of intracellular protein trafficking. The automatic analysis of these images and signals in medicine, biotechnology, and chemistry is a challenging and demanding field.
In this paper we present our newly developed methods and techniques for the analysis of microscopic cell images. We present methods for static 2-D image analysis and dynamic 3-D image analysis.
We start we a description of the challenges and requirements to the systems. Then we move further with a description of any processing unit in the full image analysis chain. We start with image segmentation followed by feature extraction, image mining, and image interpretation. Our new method for meta learning of image segmentation is described. Automatic and symbolic feature extraction is described as well as our novel texture descriptor that is effective for the description of the texture on cells. The image mining methods for learning the interpretation knowledge is described by a supervised method based on decision tree induction and an unsupervised method by conceptual clustering. Finally, we give a brief overview of our novel method for live cell tracking. We show on results the extraordinary performance of the methods.
At the end of the paper we give a summary of our expert opinion on microscopic image analysis for cellular and molecular biology. Finally, we summarize our work in the conclusion section.

Keywords: Automation and Standardization of Visual Inspection Tasks; High-Content Analysis of Images HCA; Cellular and Molecular Image Analysis and Interpretation; Image-Mining, Systems for Knowledge Discovery and Interpretation; Microscopic Cell Image Analysis

1 Introduction

In the rapidly expanding fields of cellular and molecular biology, fluorescence illumination and observation is becoming one of the techniques of choice to study the localization and dynamics of proteins, organelles, and other cellular compartments, as well as a tracer of intracellular protein trafficking. Quantitative imaging of fluorescent proteins and patterns is accomplished with a variety of techniques, including widefield, confocal and multiphoton microscopy, ultrafast low-light level digital cameras and multitasking laser control systems. These microscopic images can be of 2-dimensional or 3-dimensional nature, or even videos recording the life cycle of a cell. Currently the interpretation of the resulting pattern in these digital images is usually done manually. However, the huge amount of data created and the growing use of these techniques in industry for pharmacological aspects or diagnostic purposes in medicine require automatic image interpretation procedures. These image interpretation procedures should allow to interpret these images automatically, and also to detect automatically new knowledge to study the cellular and molecular processes. The continuation of mass image analyses on the basis of the classical procedures leads to investments of proportions that are not feasible. New procedures based on image mining and case-based reasoning are therefore required.

We are developing methods that allow the automatic analysis of these images for the discovery of patterns, new knowledge and relations. The present work is applied to 2-dimensional microscopic fluorescent image and to 3-d-image analysis (video analysis). The aim of our work is to provide the system with image-analysis, feature-extraction and knowledge-discovery functions that are suited for mining a set of microscopic cell images for the automatic detection of image-interpretation knowledge and then applying this knowledge within the same system for automatic image interpretation of the HEp-2 cell images. At the end, the system can work online in a pharmaceutical drug discovery process or in a medical and biological laboratory process and automatically interpret the patterns on the cells in the image and calculate quantitative information about the cell pattern. It can also track the cells in the microscopic images and calculate features about the dynamics of the cells. The developed processing functions should make the system flexible enough to deal with different kinds of cell-images and different image qualities and require a minimal number of interactions with the user for knowledge mining. The image-interpretation process is running fully automatically, based on the image-analysis and feature-extraction procedures developed for this kind of image analysis and the learned interpretation knowledge by the developed knowledge-mining procedures.

In Section 2 of the paper we describe the challenges and the requirements to the system. The architecture of the system is described in Section 3. The architecture is two divided. It has facilities for image mining and learning the interpretation knowledge and for automatic image interpretation and object tracking. Section 4 is dealing with image segmentation. A novel case-based reasoning image segmentation unit based on meta-learning and thresholding is presented as well as a case-based object recognition unit. Section 5 describes the features that are used to describe the appearances of the cells. A novel texture descriptor based on random sets

is presented. The image mining facilities and the methods used for image mining are described. Our decision tree induction, case-based reasoning, and conceptual clustering methods are described. Section 6 gives results when applying the system to Hep-2 cell pattern recognition for the recognition of 32 pattern. Section 7 describes our novel cell tracking algorithm and presents that feature that are calculated to describe the moving cells. Section 8 gives results for cell tracking. In Section 9 we give expert opinion on the recent architecture and methods. Finally, we give conclusions in Section 10.

2 Challenges and Requirements to the Systems

Application-oriented systems that can only solve one specific task are very costly and it takes time to develop them. The success of automatic image-interpretation systems can only be guaranteed when the development effort is as low as possible and when they can be adapted quickly to different needs and tasks. It is preferable that the automatic system not only calculates image features from the images but also maps the measurements to the desired information the user wants to obtain with his experiment. This views High-Content Image Analysis as a pattern recognition and image interpretation problem rather than as an image measurement problem where all possible image features are extracted from the images for further analysis. The pattern or the final information, such as e.g. “do the bacteria co-localize with the lysosomes”, is the central focus of the image analysis and the system should provide all functions that are necessary to achieve this result.

That requires developing systems that can run on a class of applications such as microscopic fluorescent images. Such systems should have functions that are able to:

- Automatically detect single cells in the image regardless of the image quality with high accuracy, robustness and flexibility;
- Automatically describe the properties of the cell nucleus and the cytoplasm by image features (numerical and symbolical);
- Automatically interpret the images into cell patterns or other decisions (prediction);
- Automatically detect new knowledge from image data and apply it to automatic interpretation;

The challenges are:

- New strategies are necessary that are able to adapt the system to changing environmental conditions during image capture, user needs and process requirements;
- Introduction of Case-Based-Reasoning (CBR) strategies and Data-Mining strategies [1] into image-interpretation systems on both the low-level and high-level to satisfy these requirements.

3 The Architecture

Our answer to this problem is a system architecture [2] named Cell Interpret (Figure 1) that is comprised of two main parts:

- The on-line part that is comprised of the image analysis and the image interpretation part;
- The off-line part that is comprised of the database and the data mining and knowledge discovery part;

These two units communicate over a database of image descriptions, which is created in the frame of the image-processing unit. This database is the basis for the image-mining unit.

The on-line part can automatically detect objects, extract image features from the objects and classify the recognized objects into the respective classes based on the prior stored decision rules. The interface between the off-line and the on-line part is the database where images and calculated image features are stored. The off-line part can mine the images for a prediction model or discover new groups of objects, features or relations. These similar groups can be used for learning the classification model or just for understanding the domain. In the later case the discovered information is displayed on the terminal of the system to the user. Once a new prediction model has been learnt the rules are inputted into the image interpretation part for further automatic interpretation after approval of the user. Besides that, there is an archiving and management part that controls the whole system and stores information for long-term archiving.

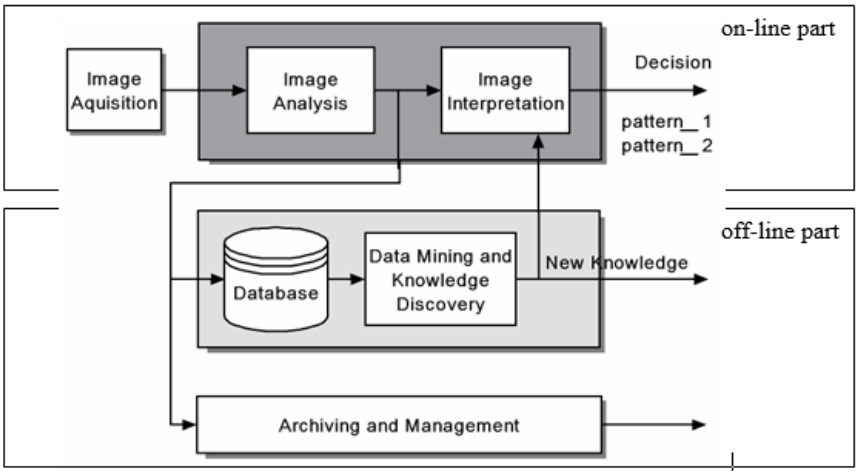


Fig. 1. Architecture of Cell Interpret.

Images can be processed automatically or semi-automatically. In the first case, a set of images specified by the expert is automatically segmented into background and objects of interest and the feature extraction procedures installed in the image analysis

system are used for each object to automatically calculate all features. All features are extracted regardless of their applicability for the specific application. This requires executing feature subset selection methods later on. For semi-automatic processing, an image from the image archive is selected by the expert and then is it displayed on the monitor. To perform image processing an expert communicates with a computer. In this mode he has the option to calculate features based on the feature extraction procedures and/or record symbolic features based on his expert knowledge. This procedure ensures that also complicated image features, which are difficult to name, articulate or develop automatic feature extraction procedures, can also be taken into account and further evaluated by image mining. After the feature has been established by evaluating the acquired data base, the proper automatic feature extraction procedure can be developed and included into the system and made available for High-Content Analysis. The intelligence of the system will therefore incrementally improve.

4 Case-Based Image Segmentation

Image segmentation is a process of dividing an image into a number of different regions such that each region is homogeneous with respect to a given property, but the union of any two adjacent regions is not. Image thresholding is a well-known technique for image segmentation. Because of its wide applicability to many areas of digital image processing, a large number of thresholding methods have been proposed over the years [3-5]. Image thresholding has low computational complexity, which makes it an attractive method, but does not take into account spatial information and is mostly suitable for images where the gray-levels constitute well defined peaks, separated by not too broad and flat valleys. Another common approach to image segmentation is based on feature space clustering, which has sometimes been regarded as the multidimensional extension of the concept of thresholding. Clustering schemes using different kinds of features (multi-spectral information, mean/variation of gray-level, texture, color) have been suggested [6-8]. This approach can be successfully used if each perceived region of the image constitutes an individual cluster in the feature space. This requires a careful selection of the proper features, which depends on image domain.

Segmentation can also be accomplished by using region-based methods, or edge-detection-based methods, or methods based on a combination of those two approaches [9-11]. Region-based methods imply the selection of suitable seeds from which to perform a growing process. In general, region-merging and region splitting are accomplished to obtain a meaningful number of homogeneous regions. Seed selection and homogeneity criterion play a critical role for the quality of the obtained results. Edge-detection-based methods follow the way in which human observers perceive objects, as they take into account the difference in contrast between adjacent regions. Edge detection does not work well if the image is not well contrasted, or in the presence of ill-defined or too many edges.

Watershed-based segmentation [12] exploits both region-based and edge-detection-based methods. The basic idea of watershed-based segmentation is to identify in the gray-level image a suitable set of seeds from which to perform a growing process. If the main feature taken into account is gray-level distribution, the seeds are mostly detected as the sets of pixels with locally minimal gray-level (called regional minima). The growing process groups each seed with all pixels that are closer to that seed than to any other seed, provided that a certain homogeneity in gray-level is satisfied. Thus, watershed-based segmentation limits the drawbacks of region-based and edge-detection-based methods.

To overcome the drawbacks of the algorithms mentioned above, learning methods are applied to image segmentation. These learning methods are applied to learn the mapping between image features and semantically meaningful parts, to learn the parameters of the segmentation algorithm or to learn the mapping between rank performance of the segmentation algorithm and the image features. There are statistical learning methods, machine learning methods, neural-net-based learning methods, and learning methods using a combination of different techniques. The main drawbacks of these methods are:

1. The need of a sufficiently large training set, and
2. The need of training again the whole model, when new data come in.

Therefore, it seems to be useful to use Case-based Reasoning (CBR) for a flexible image segmentation system, since CBR can be used as a reasoning approach as well as an incremental knowledge-acquisition approach. We propose a novel image-segmentation scheme based on case-based reasoning. We use CBR for meta-learning of the segmentation parameters (see Section 4.1) and for case-based object recognition (see Section 4.2).

4.1 CBR Meta Learning for Image Segmentation

The case-based reasoning unit for meta learning of image segmentation parameters [13] consists of a case base in which formerly processed cases are stored. A case is comprised of image information, non-image information (e.g. image-acquisition parameters, object characteristics and so on), and image-segmentation parameters. The task is now to find the best segmentation for the current image by looking up the case base for similar cases. Similarity determination is done based on non-image information and image information. The evaluation unit will take the case with the highest similarity score for further processing. In case there are two or more cases with the same similarity score, the case appearing first will be taken. After the closest case has been chosen, the image-segmentation parameters associated with the selected case will be given to the image-segmentation unit and the current image will be segmented (Figure 2). It is assumed that images having similar image characteristics will show similar good segmentation results when the same segmentation parameters are applied to these images. The image segmentation algorithm is in our case a histogram-based image-segmentation algorithm [13] and a watershed-based image-segmentation algorithm [14].

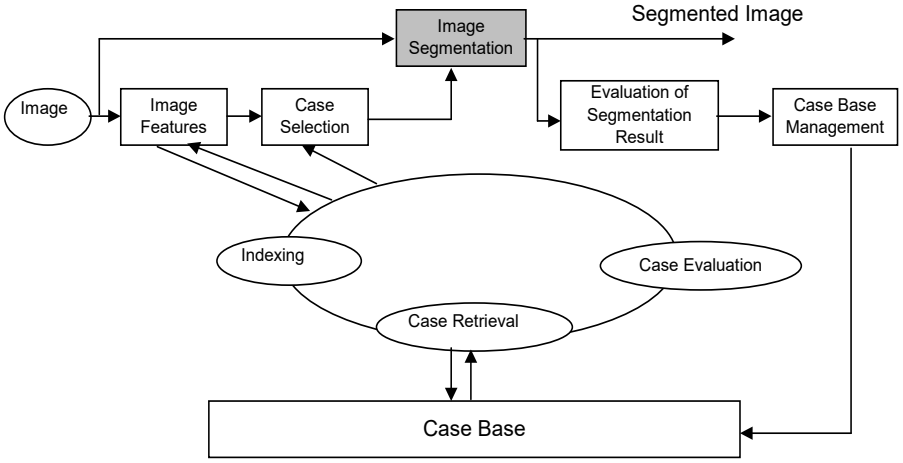


Fig. 2. CBR Image Segmentation Unit.

The result of the segmentation process can be observed by the user or an automatic evaluation procedure. When the evaluation is done by the user, he compares the original image with the labeled image on display. If he detects deviations of the marked areas in the segmented image from the object area in the original image, which should be labeled, then he will evaluate the result as incorrect and case-base management will start. This will also be done if no similar case is available in the case-base. The proposed method is close to the critique-modify framework described by Grimnes et al. [15]. The evaluation procedure can also be done automatically. However, the drawback is that there is no general procedure available. It can only be done in a domain-dependent fashion. Once the chosen evaluation procedure observes a bad result, the respective case is tagged as bad case. The tag describes the critique in more detail.

In an off-line phase, the best segmentation parameters for the image are determined by an optimization procedure and the attributes, which are necessary for similarity determination, are calculated from the image. Both, the segmentation parameters and the attributes calculated from the image, are stored into the case-base as a new case. In addition to that the non-image information is extracted from the file header and stored together with the other information in the case-base. During storage, case generalization will be done to ensure that the case base will not become too large.

4.2 Case-based Object Recognition

We propose our case-based object recognition method to recognize objects by their shape. In contrast to traditional object recognition methods [16] our method is comprised of a case mining part and the object recognition part [17]. The case mining part

can learn the desired contour of the object and the number of contours necessary for recognizing a particular class of objects. The learnt contours make up the case base and are the basis for the case-based object-recognition method. The objects in the image may be occluded, touching, or overlapping. It can also happen that only part of the object appears in the image.

A case-based object-recognition method uses cases that generalize the original objects and matches them against the objects in the image, see Figure 3. During this procedure a score is calculated that describes the quality of the fit between the object and the case. The case can be an object model which describes the inner appearance of the object as well as its contour. In our case the appearance of the whole object can be very diverse. The shape seems to be the feature that generalizes the objects. Therefore, we decided to use contour models. We do not use the gray values of the model, but instead use the object's edges. For the score of the match between the contour of the object and the case we use a similarity measure based on the scalar product. It measures the average angle between the vectors of the template and the object.

The acquisition of the case is done semi-automatically. Prototypical images are shown to an expert. The expert manually traces the contour of the object with the help of the cursor of the computer. Afterwards the number of contour points is reduced for data-reduction purposes by interpolating the marked contour by a first-order polynom. The marked object shapes are then aligned by the Procrustes Algorithm [18]. From the sample points the direction vector is calculated. From a set of shapes general groups of shapes are learnt by conceptual clustering which is a hierarchical incremental clustering method [19]. The prototype of each cluster is calculated by estimating the mean shape [19] of the set of shapes in the cluster and is taken as a case model.

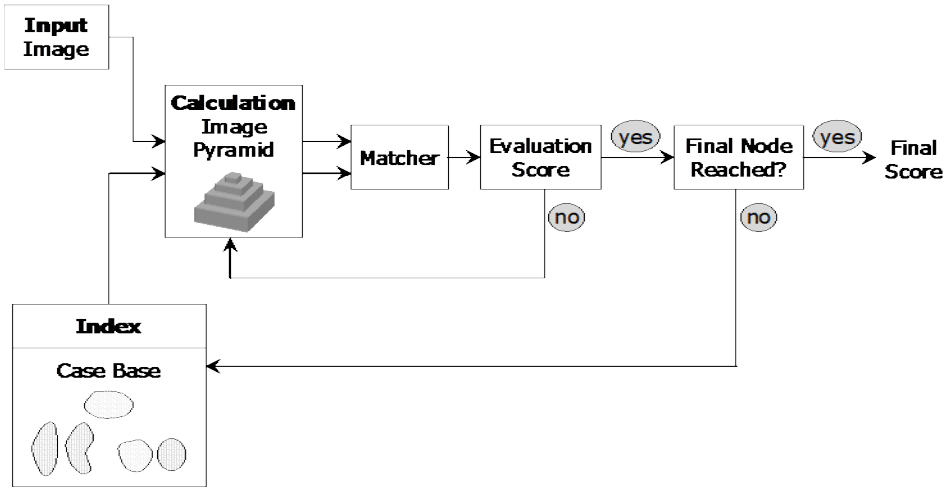


Fig. 3. Principle of case-based object-recognition architecture.

5 Automatic and Symbolic Feature Extraction

The system can now, based on the feature-extraction filter data base (Figure 4) installed in the system, calculate image features for the labeled objects. These features are composed of statistical gray-level features, the object contour, square, diameter, shape [20] and a novel texture feature based on random sets [21] that is flexible enough to describe different textures of cells.

The novel texture-feature descriptor is flexible enough to describe different textures inside the cells that reflect the appearance or location of subcellular particle`s (vesicles, bacteria moving into the cells, or chromosomes etc.). The texture descriptor is based on Random Sets that were invented by Matheron [22]. An in-depth description of the theory can be found in Stoyan, et al. [23]. The Boolean model allows to model and simulate a huge variety of textures e.g. for crystals, leaves, etc. The texture model X is obtained by taking various realizations of compact random sets, implanting them in Poisson points in R^n, and taking the supremum. The functional moment $Q(B)$ of X, after Booleanization, is calculated as:

$$PB \subset Xc = QB = \exp -\theta Mes(X \oplus B) \forall B \in k \qquad (1)$$

K where K is the set of the compact random set of Rn, θ the density of the process and $\overline{\text{Mes}}$ $(X' \oplus \check{X})$ is an average measure that characterizes the geometric properties of the remaining set of objects after dilation. Formula (1) is the fundamental formula of the model. It completely characterizes the texture model. $Q(B)$ does not depend on the location of B, i.e., it is stationary. One can also provide that it is ergodic so that we can peak the measure for a specific portion of the space without referring to the particular portion of the space.

Formula 25 show us that the texture model depends on two parameters:

- The density θ of the process and
- a measure $\bar{}$Mes$(X \oplus B\check{})$ that characterizes the objects. In the one-dimensional space, it is the average length of the lines and in the two-dimensional space $\bar{}$Mes $(X \oplus B\check{})$ is the average measure of the area and the perimeter of the objects under the assumption of convex shapes.

We consider the two-dimensional case and develop a proper texture descriptor. Suppose now that we have a texture image with 8 bit gray levels. Then we can consider the texture image as the superposition of various Boolean models, each of them having a different gray level value on the scale from 0 to 255 for the objects within the bit plane. To reduce the dimensionality of the resulting feature vector, the gray levels ranging from 0 to 255 are now quantized into S intervals t. Each image f(x,y) is classified according to the gray level into t classes, with t=□0,1,2,..,S□. For each class a binary image is calculated containing the value “1” for pixels with a gray level value falling into the gray level interval of class t and value “0” for all other pixels. The resulting bit plane f (x,y,t) can now be considered as a realization of the Boolean model. The quantization of the gray level into S intervals was done at equal distances. In the following, we call the image f(x,y,t) a class image. In the class image we can

see a lot of different objects. These objects get labeled with the contour-following method [20]. Afterwards, features from the bit-plane and from these objects are calculated. Since it does not make sense to consider the features of every single object due to the curse of dimensionality, we calculate the mean and standard deviation for each feature that characterizes the objects such as the area and the contour. In addition to that, we calculate the number of objects and the areal density in the class image.

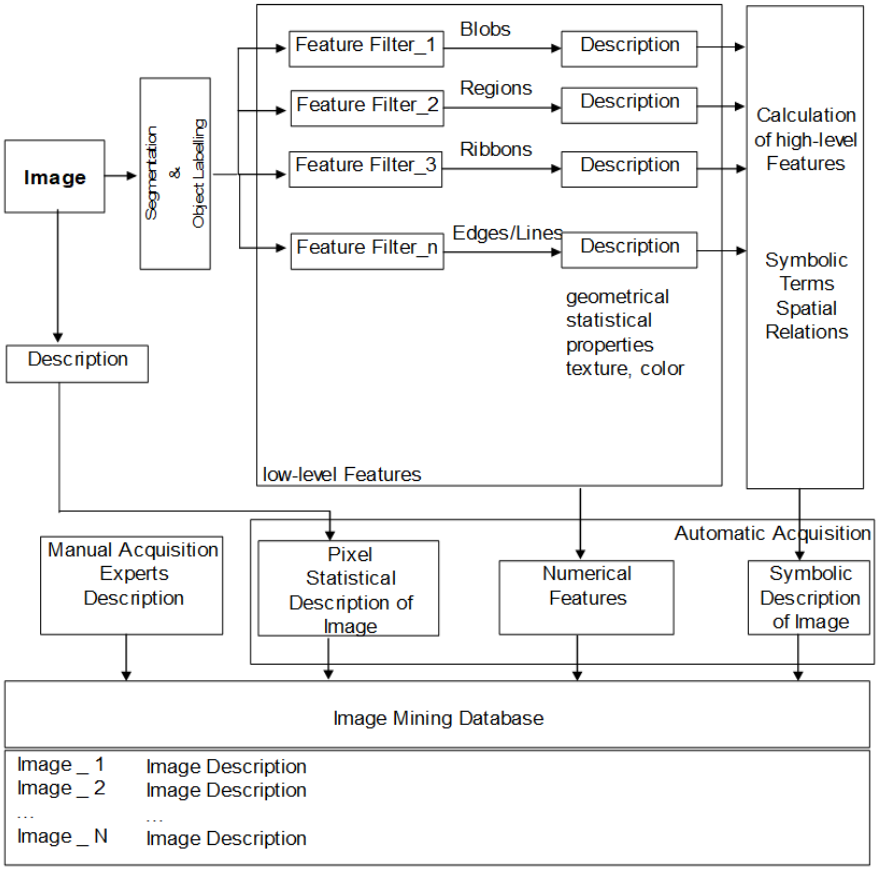


Fig. 4. Feature Filter Data Base.

The list of features and their calculation are shown in Table 1. The first one is the areal density of the class image t which is the number of pixels in the class image, labeled by "1", divided by the area of the image. If all pixels of an image are labeled by "1", then the density is one. If no pixel in an image is labeled, then the density is zero. From the objects in the class image t, the area, a simple shape factor, and the length of the contour are calculated. Per the model, not a single feature of each object is taken for classification due to the curse of dimensionality, but the mean and the standard deviation of each feature are calculated over all the

objects in the class image t. We also calculate the frequency of the object size in each class image t.

Depending on the number of slices S we get a feature set of 42(S=6), 84(S=12), 112(S=16) features.

Table 1. Texture Features based on Random Set.

Description	Name	Type	Formula
Area in class image *t*	Area_t	num	$\mathrm{Area_t} = \begin{cases} \mathrm{Area_t} = \mathrm{Area_t} + 1 & \text{if } f(x,y,t) = 1 \\ \mathrm{Area_t} = \mathrm{Area_t} & \text{if } f(x,y,t) + 0 \end{cases}$
Density in class image *t*	Dens_*t*	num	$\mathrm{Dens} = \begin{cases} \mathrm{Dens_t} = \mathrm{Dens_t} + 1 & \text{if } f(x,y,t) = 1 \\ \mathrm{Dens_t} = \mathrm{Dens_t} & \text{if } f(x,y,t) + 0 \end{cases}$ with $\mathrm{A} = \sum_{t=1}^{s} \mathrm{Area}_t$
Number of objects	Count_*t*	num	n(t)
Mean area of objects in class image *t*	Area Mean_*t*	num	$\overline{A(t)} = \frac{1}{n(t)} \sum_{i=1}^{n(t)} A_i(t)$
Standard deviation of the contour length of objects in class image *t*	Cont Std Dev_*t*	num	$S(t) = \sqrt{\frac{1}{n(t)} \sum_{i=1}^{n(t)} (u_i(t) - \overline{A(t))^2}}$
The contour length of a single object is $u = l + \sqrt{2} \cdot m$ with *l* being the number of contour pixels having odd chain coding numbers and *m* being the number of contour pixels having even chain coding numbers.			
Mean contour length of objects in class image *t*	Cont Mean_*t*	num	$\overline{u}(t) = \frac{1}{n(t)} \sum_{i=1}^{n(t)} u_i(t)$
Standard deviation of the contour length of objects in class image *t*	Cont Std Dev_*t*	num	$S(t) = \sqrt{\frac{1}{n(t)} \sum_{i=1}^{n(t)} (u_i(t) - \overline{u(t))^2}}$

The system evaluates or calculates image features and stores their values in a database of image features. Each entry in the database presents features of the object of

interest. These features can be numerical (calculated on the image) and symbolical (determined by the expert as a result of image reading by the expert). In the latter case the expert evaluates object features according to the attribute list, which has to be specified in advance for object description or is based on a visual ontology available for visual content description. Then the user feeds these values into the database. When the expert has evaluated a sufficient number of images, the resulting database can be used for the image-mining process.

6 Image Mining and Knowledge Discovery

The image mining part should allow extracting knowledge or making observations from different perspectives. Therefore, we have included methods for predictions and methods for knowledge discovery [1]. Knowledge discovery methods allow us to summarize data into groups and patterns or observe relations among groups. Usually they are prior to prediction. We prefer conceptual clustering [1] for this task since the discovery process is incremental and therefore fits perfectly to case-based reasoning and decision tree induction as prediction methods.

6.1 Decision Tree Induction

Decision tree induction allows one to learn from a set of data samples a set of rules and basic features necessary for decision-making in a specified diagnostic task, see Figure 5. The induction process does not only act as a knowledge discovery process, it also works as a feature selector, discovering a subset of features that is the most relevant to the problem solution. Decision trees partition decision space recursively into sub-regions based on the sample set. In this way the decision trees recursively break down the complexity of the decision space. The outcome has a format which naturally presents a cognitive strategy that can be used for the human decision-making process. For any tree all paths lead to a terminal node, corresponding to a decision rule that is a conjunction (AND) of various tests. If there are multiple paths for a given class, then the paths represent disjunctions (ORs). The developed tool allows choosing different kinds of methods for feature selection, feature discretization, pruning of the decision tree and evaluation of the error rate. It provides an entropy-based measure, a gini-index, gain-ratio and chi square method for feature selection [1].

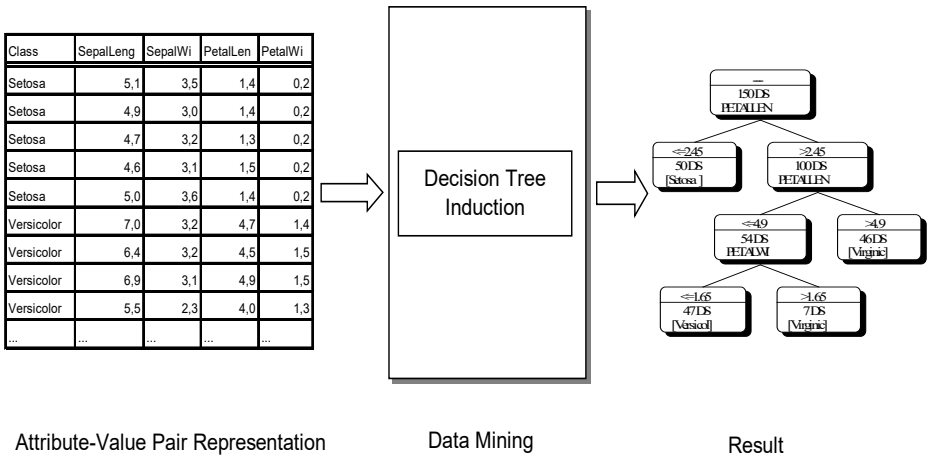


Fig. 5. Basic Principle of Decision Tree Induction.

The following methods for feature discretization are provided: cut-point strategy, chi-merge discretization, minimum description length, histogram-based discretization method and lvq-based method [1]. These methods allow one to make discretization of the feature values into two and more intervals during the process of decision-tree building. Depending on the chosen method for attribute discretization, the result will be a binary or n-ary tree, which will lead to more accurate and compact trees. The tool allows one to choose between cost-complexity pruning, error-reduction-based methods and pruning by confidence-interval prediction. The tool also provides functions for outlier detections. To evaluate the obtained error rate one can choose test-and-train and n-fold cross validation. Missed values can be handled by different strategies [1].

The user selects the preferred method for each step of the decision tree induction process. After that the induction experiment can start on the acquired database. A resulting decision tree will be displayed to the user. He/she can evaluate the tree by checking the features used in each node of the tree and comparing them with his/her domain knowledge. Once the diagnosis knowledge has been learnt, the rules are provided either in txt-format or XML format for further use in the image interpretation part or the expert can use the diagnosis component of the tool for interactive work. It has a user-friendly interface and is set up in such a way that non-computer specialists can handle it very easily.

6.2 Case-based Reasoning for Image Interpretation

It is difficult to apply decision trees in domains where generalized knowledge is lacking. But often there is a need for a prediction system, even though there is not enough generalized knowledge. Such a system should

1. Solve problems using the already stored knowledge and

2. Capture new knowledge, making it immediately available to solve the next problem.

To accomplish these tasks case-based reasoning is useful. Case-based reasoning explicitly uses past cases from the domain expert´s successful or failing experience. Therefore, case-based reasoning can be seen as a method for problem-solving as well as a method to capture new experience in an incremental fashion and make it immediately available for problem-solving. It can be seen as a learning and knowledge-discovery approach, since it can capture from new experience some general knowledge such as case classes, prototypes and some higher-level concepts. We find these methods especially applicable for inspection and diagnosis tasks. In the case of these applications people store prototypical images into a digital image catalogue rather than a large set of different images [22].

We have developed a unit for Cell Interpret that can perform similarity determination between cases, as well as prototype selection [23] and feature weighting [24]. We call $x_n \in \{x_1, x_2, \ldots, x_n\}$ a nearest-neighbor to x if $\min d(x_i, x) = d(x_n', x)$, where i= 1,2, … , n. The instance x is classified into category C_n, if x_n is the nearest neighbor to x and x_n belongs to class C_n. In the case of the k-nearest neighbor we require k-samples of the same class to fulfill the decision rule. As a distance measure we use the Euclidean distance. Prototype Selection from a set of samples is done by Chang`s Algorithm [23]. Suppose a training set T is given as $T = \{t^1, \ldots, t^m\}$. The idea of the algorithm is as follows: We start with every point in T as a prototype. We then successively merge any two closest prototypes $p^1$and p^2 of the same class by a new prototype p, if the merging will not downgrade the classification of its patterns in T. The new prototype p may simply be the average vector of p^1 and p^2. We continue the merging process until the number of incorrect classifications of the patterns in T starts to increase.

The wrapper approach is used for selecting a feature subset from the whole set of features. This approach conducts a search for a good feature subset by using the k-NN classifier itself as an evaluation function. The 1-fold cross-validation method is used for estimating the classification accuracy and the best-first search strategy is used for the search over the state space of possible feature combination. The algorithm terminates if we have not found an improved accuracy over the last k search states. The feature combination that gave the best classification accuracy is the remaining feature subset. After we have found the best feature subset for our problem, we try to further improve our classifier by applying a feature-weighting technique.

The weights of each feature $W_1$are changed by a constant value δ: $w_i := w_i \pm \delta$. If the new weight causes an improvement of the classification accuracy, the weight will be updated accordingly; if not, the weight will remain as it is. After the last weight has been tested the constant δ will be divided into half and the procedure repeats. The procedure terminates if the difference between the classification accuracy of two iterations is less than a predefined threshold.

6.3 Conceptual Clustering

The intention of clustering as another image mining function is to find groups of similar cases among the data according to the observation. This can be done based on one feature or a feature combination. The resulting groups give an idea how data fit together and how they can be classified into interesting categories. Classical clustering methods only create clusters but do not explain why a cluster has been established. Conceptual clustering methods build clusters and explain why a set of objects confirm a cluster. Thus, conceptual clustering is a type of learning by observation and it is a way of summarizing data in an understandable manner [1]. In contrast to hierarchical clustering methods, conceptual clustering methods build the classification hierarchy not only based on merging two groups. The algorithmic properties are flexible enough to dynamically fit the hierarchy to the data. This allows incremental incorporation of new instances into the existing hierarchy and updating this hierarchy according to the new instance.

A concept hierarchy is a directed graph in which the root node represents the set of all input instances and the terminal nodes represent individual instances. Internal nodes stand for sets of instances attached to the nodes and represent a super-concept. The super-concept can be represented by a generalized representation of this set of instances such as the prototype, the medium or a user selected instance. Therefore a concept C, called a class, in the concept hierarchy is represented by an abstract concept description and a list of pointers to each child concept $M(C)=\{C_1, C_2, ..., C_i, ..., C_n\}$, where C_i is the child concept, called subclass of concept C.

Our conceptual clustering algorithm presented here is based on similarities, because we do not consider logical but numerical concepts [19]. The output of our algorithm for applying eight exemplary shape cases of fungal strain *Ulocladium Botrytis* is shown in (Figure 6). On top level the root node is shown which comprises the set of all input cases. Successively the tree is partitioned into nodes until each input case forms its own cluster. The main advantage of our conceptual clustering algorithm is that it brings along a concept description. Thus, in comparison to agglomerative clustering methods, it is easy to understand why a set of cases forms a cluster. During the clustering process the representative case, and also the variances and maximum distances in relation to this representative case, are calculated, since they are part of the concept description. The algorithm is of incremental fashion, because it is possible to incorporate new cases into the existing learnt hierarchy.

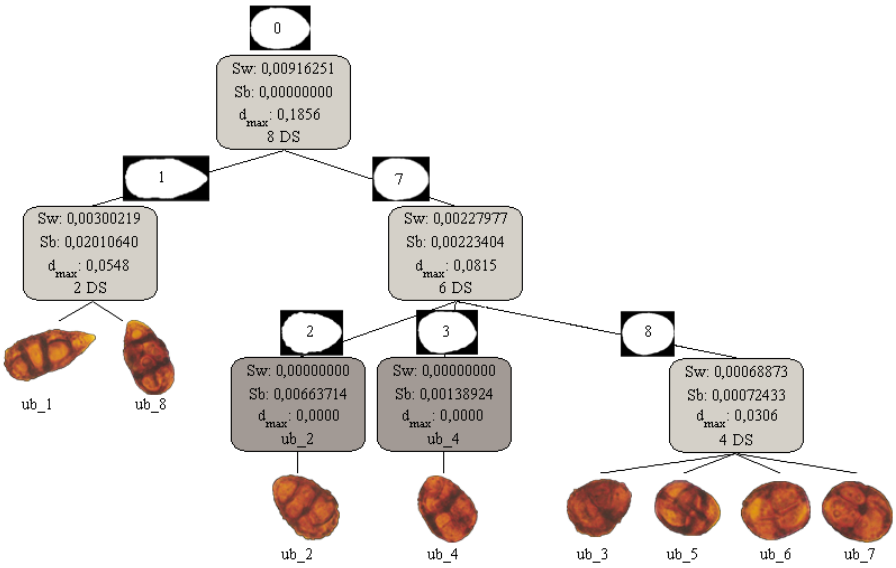


Fig. 6. Output of the Conceptual Clustering Algorithm for 2-D Shapes obtained from Fungal Spores.

7 Results for Static 2D Image Analysis

The kinds of cells that are considered in this application are HEp-2 cells, which are used for the identification of Antinuclear Autoantibodies (ANA). ANA testing for the assessment of systemic and organ-specific autoimmune diseases has increased progressively since immunofluorescence techniques were first used to demonstrate antinuclear antibodies in 1957. HEp-2 cells allow for recognition of over 30 different nuclear and cytoplasmic patterns, which are given by upwards of 100 different autoantibodies. The identification of the patterns has up to now been done manually by a human inspecting the slides with the help of a microscope. The lacking automation of this technique has resulted in the development of alternative techniques based on chemical reactions, which do not have the discrimination power of the ANA testing. An automatic system would pave the way for a wider use of ANA testing. Prototypical images of HEp-2 cell patterns for six different classes are shown in Figure 7. The images were taken by an image-acquisition unit consisting of a microscope AXIOSKOP from Carl Zeiss Jena, coupled with a video camera.

In a knowledge-acquisition process [25] with a human operator, using an interview technique and a repertory grid method, we acquired the knowledge of this operator, while classifying the different cell types. Some of this knowledge is shown in Table 2. The symbolic terms show that a mixture of different image information is necessary for classification. The operator uses the intensity as well as some texture information.

In addition, the appearances of the cell parts within the cells are of importance, like "dark nuclei", which also requires spatial information.

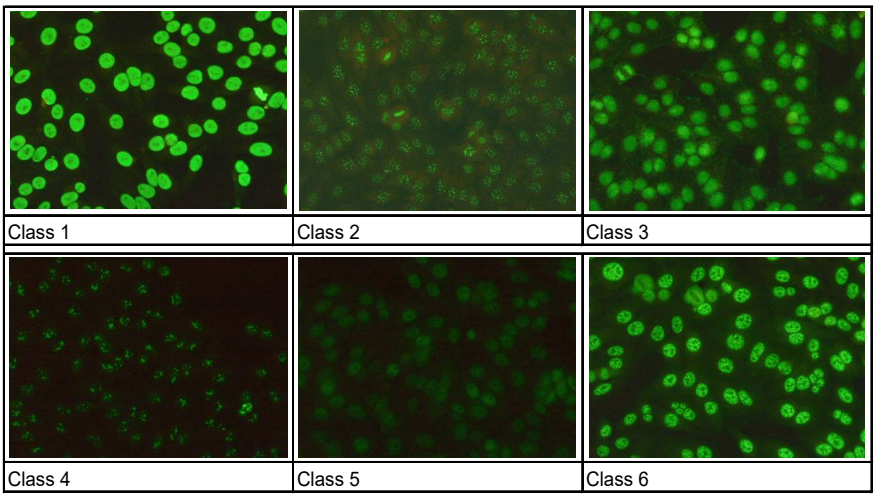


Fig. 7. Prototypical Images of Six Classes.

Table 2. Some knowledge about the class description given by a human operator.

Class	Class Name	Description
Homogeneous nuclei fluores-cence	Class_1	Smooth and uniform fluorescence of the nuclei. Nuclei appear sometimes dark. The chromosome fluorescence is from weak to very intense
Fine speckled nuclei fluores-cence	Class_2	Dense fine speckled fluorescence
...	...	...
Nuclei fluores-cence	Class_9	Nuclei are weakly homogenous or fine-grained and can hardly be discerned from the background

Each image is processed by the image-analysis procedure described in the previous section. The color image is transformed into a gray-level image. The image is normalized to the mean and standard gray level calculated from all images to avoid invariance caused by the inter-slice staining variations. Automatic thresholding has been performed by the algorithm described in Section 4.1. For the objects in each slice, features based on the texture descriptor described in Section 5 are calculated for classification [26]. The first one is a simple Boolean feature which expresses the occurrence or non-occurrence of objects in the slice image. Then the number of objects in the slice image is calculated. From the objects, the area, a shape factor, and the length

of the contour are calculated. The mean value for each feature is calculated over all the objects in the slice image. This is done in order to reduce the dimension of the feature vector. Since the quantization of the gray level was done in equal steps and without considering the real nature, we also calculated for each class the mean value of the gray level and the variance of the gray level. A total of 192 features were calculated that make up a very intelligent structure and texture descriptor for cells [26]. The data base created from 7-10 images per class which made up 30 cells per class is given to our decision tree unit. This unit learns the classification knowledge based on decision tree induction. Finally, the system was evaluated based on cross validation. The final result is shown in Table 2. The overall classification accuracy is 92.73%. The class specific classification accuracy [1] is shown for each class in Table 3 on the right side of the table and the classification quality for each class in the bottom line of the table. In most of the classes we achieved good classification accuracy. There are only few classes where the classification accuracy is not as good as the other ones. It is interesting to note that in case of class_5 four cases got misclassified as class_14 "U1-RNP" but when checking with the expert it tended out that the classifier put these samples in the right class. The case was that the expert mislabeled the cases as class_5 while the automatic system recognized that these samples belong not to class_5 but to class_14. This example shows nicely that an automatic system can lead to standardization of cell image classification. It provides objective results, it works constantly without getting tired and the results are reproducible.

The computation time of an image for the Hep-2 application is 20 seconds by an image size of 1600x1200. This computation time is fast enough for the considered application and for most other applications. Users who like to have a faster computation time can easily speed up the computation time by parallelization. Parallelization can be done in the simplest case by using more than one computer. In the hardest case, the whole algorithm can be set up in parallel fashion.

The methods developed within the framework Cell Interpret have been applied to many different applications of microscopic cell images including Hep-2 cell, Hela-cells and Malaria diagnosis. They showed to be flexible enough for different kind of cell images diagnosis tasks and they efficiently enabled the mining of the relevant knowledge for the development of an automatic image interpretation system. The Hep-PAD version developed based on Cell Interpret has been licensed to qualified industries and is meanwhile a commercial application in usage at different medical laboratories e.g. by Prof. Landenberg from the University Clinic in Mainz/Germany. We are currently further developing the framework of Cell Interpret to video microscopy and developing more feature extraction and image mining procedure that can further support the image mining process.

Table 3. Results for Hep-2 Pattern Analysis.

Example: Result LDS6 and DM4																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		Class Specific Quality
	AmaCent	Actin	AMA Who	Centromer	CoarseSp	Homogen	Jo-1	Nucleolaer	PMSCL	SCL70	Speckled	SS-A	SS-B	U1-RNP	Vimentin	Sum	CSQ
AmaCent	6															6	100,00%
Actin		7														7	100,00%
AMA Who			7													7	100,00%
Centromer				7												7	100,00%
CoarseSp					5						2					7	71,43%
Homogen						8										8	100,00%
Jo-1							6									6	100,00%
Nucleolaer								7								7	100,00%
PMSCL									7							7	100,00%
SCL70										8						8	100,00%
Speckled											6					6	100,00%
SS-A							1					7				8	87,50%
SS-B													7			7	100,00%
U1-RNP					4						1			7		12	58,33%
Vimentin															7	7	100,00%
Sum	6	7	7	7	9	8	7	7	7	8	9	7	7	7	7	110	
Cl. Qual.	100,00%	100,00%	100,00%	100,00%	55,56%	100,00%	85,71%	100,00%	100,00%	100,00%	66,67%	100,00%	100,00%	100,00%	100,00%		94,48%

Classification Quality

Total Number of samples			110	110
Correct classied samples			102	106
Correctness			92,73%	96,36%
Error rate			7,27%	3,64%

8 Live Cell Tracking

Live Cell Tracking is another important task in cellular and molecular biology as well as in drug discovery. Here we briefly present our novel algorithm for life cell tracking [41] and the features we calculate to describe the cell movement.

The flowchart of the algorithm is given in Figure 8.

At first, the image gets threshold by Otsu`s well-known segmentation procedure. Afterwards the morphological filter opening by a 3x3 window is applied to close the contour and the inner holes. Fragmented cells at the image borders as well as small remaining objects of a size of ten pixels are deleted. The cells at the image borders are only tracked when they fully appear inside the image. Around the object is drawn the convex hull and remaining holes or open areas inside the cell area are closed by the operation flat-fill. The resulting images after these operations are shown in Figure 3 for five time frames (see Figure 9). This resulting area is taken as the cell area and the area with its grey levels is temporarily stored as template in the data base.

Then the center of gravity of the object is determined and the search window is tentatively spanned around the object. A raw estimation of the cell's movement direction is calculated by the mean-shift filter over 3 frames. In the resulting direction is started the search for similar cells. Cells fragmented by the window border are not considered for further calculation. Each cell inside the window is compared by the similarity measure to the respective template of the cell under consideration. Before the similarity is determined the cells are aligned, so that they have the same orientation. The cell having the highest similarity score to the template is labeled as the same

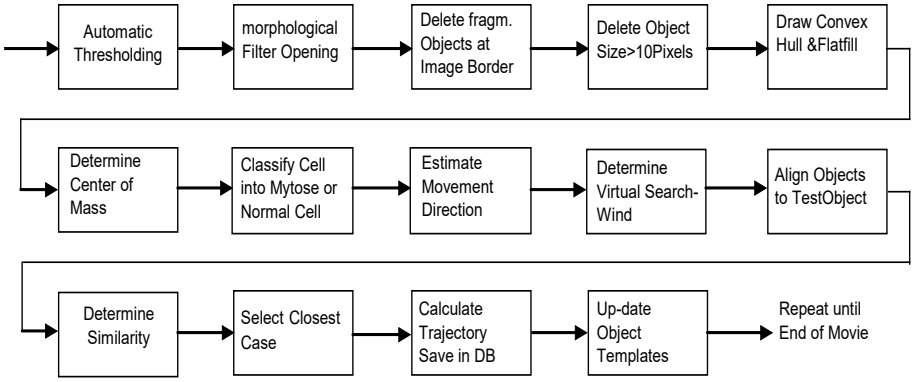
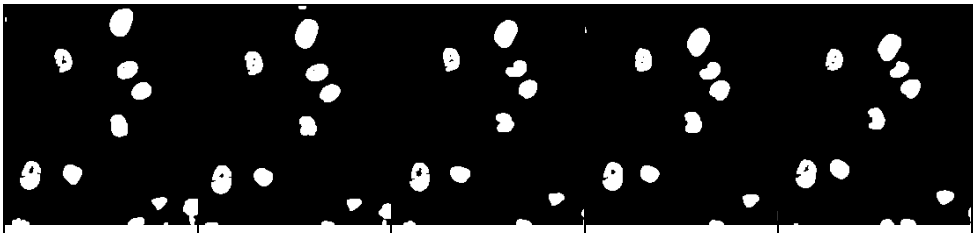
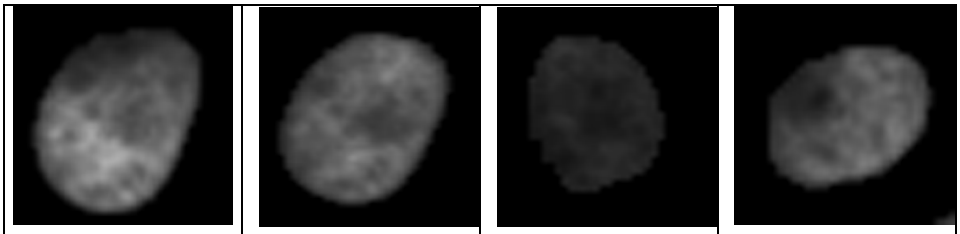


Fig. 8. Flowchart of the algorithm



Im-age_tTM_002	Im-age_tTM_003	Im-age_tTM_004	Im-age_tTM_005	Im-age_tTM_006

Fig. 9. Image after Threshold and Morphological Filtering



Cell_1 Image002 ***Template***	Cell_1 Image003	Cell_2_Image003	Cell_3_Image003
Similarity value	**0,0554918423**	0,1190173551	0,0833996087

Fig. 10. Cells used for comparison

cell moved to the position *x,y* in the image *t+1*. The template is updated with the detected cell area for the comparison in the next time frame (see Figure 10). The position of the detected cell is stored into the data base under the label of the template.

Mitotic cells are detected by classifying each cell based on the texture descriptor given in Perner and Attig [17] and a learnt decision tree classifier.

Let C_t be the cell at time-point t and C_{t+1} the same cell at time point $t+1$. Then the rule for labeling a cell as "disappeared" is: **IF** C_t has no C_{t+1} **THEN** Disappearing Cell.

9 Results of Tracking Algorithm and Measures from the Path

Results in Figure11 a-e show the tracking path of six different cells. We compared the manually determined path of a cell from an operator of the High-Content Screening Process-line with the automatic determined path by the tool *IBaI-Track* for 10 videos with 117 frames each. If both methods gave the same path we evaluated it as positive otherwise as negative. We observed a correspondence between these two descriptions of 98.2 %. That is a very good tracking result and means the method can nearly track all cells accurately. The computation time for a sequence of 117 images of 674x516 pixels each and on average 80 cells per image is 11 minutes 42 seconds on PC with 1.8 GHz.

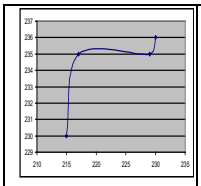


Fig. 6a Tracking Path Cell_1

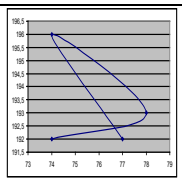


Fig. 6b Tracking Path Cell_2

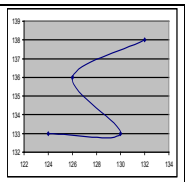


Fig. 6c Tracking PathCell_3

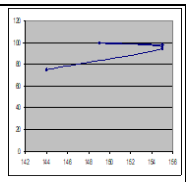


Fig. 6d Tracking Path Cell_4

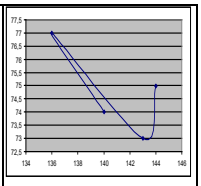


Fig. 6e Tracking Path Cell_5

Fig. 11. Trajectory examples

The output of the cell-tracking algorithm is a tuple of coordinates for each cell that describes that path of the cell (see Fig. 11a-11e and Fig. 12). Biologists want to study the kinetics of the cells. Therefore, we have to extract features from this path that describe the motility and velocity of a cell. Table 4 shows features that are used to describe the path of a cell. These features (see table 5) are provided in a table to the biologist for further study. Please note, one image contains many cells. As result, we obtain a bunch of numerical values for one image that is hard to overlook for a human. More high-level descriptions are necessary that summarize these features and their feature values in a more compact information about the kinetics of the cells in one image. Recently, biologist use statistical tools to study these features. More complex image mining methods such as decision tree induction and conceptual clustering can be of help in order to bring out the higher-level descriptions.

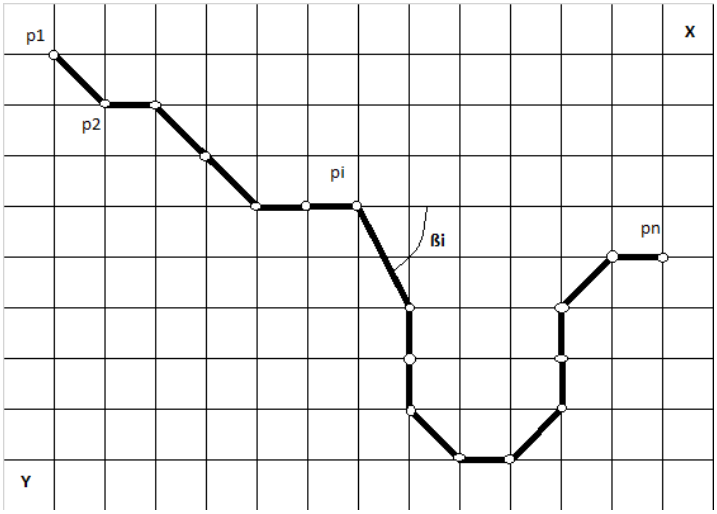


Fig. 12. Path of a Cell marked with coordinate points

Table 4. Measures for motility and velocity of a cell

Measure	Definition
Total distance traveled	$d_{\text{tot}} = \sum_{i=1}^{N-1} d(\mathbf{p}_i, \mathbf{p}_{i+1})$
Net distance traveled	$d_{\text{net}} = d(\mathbf{p}_1, \mathbf{p}_N)$
Maximum distance traveled	$d_{\text{max}} = \max_i d(\mathbf{p}_1, \mathbf{p}_i)$
Total trajectory time	$t_{\text{tot}} = (N-1)\Delta t$
Confinement ratio	$r_{\text{con}} = d_{\text{net}}/d_{\text{tot}}$
Instantaneous angle	$\alpha_i = \arctan(y_{i+1} - y_i)/(x_{i+1} - x_i)$
Directional change	$\gamma_i = \alpha_i - \alpha_{i-1}$
Instantaneous speed	$v_i = d(\mathbf{p}_i, \mathbf{p}_{i+1})/\Delta t$
Mean curvilinear speed	$\bar{v} = \frac{1}{N-1}\sum_{i=1}^{N-1} v_i$
Mean straight-line speed	$v_{\text{lin}} = d_{\text{net}}/t_{\text{tot}}$
Linearity of forward progression	$r_{\text{lin}} = v_{\text{lin}}/\bar{v}$
Mean squared displacement	$\text{MSD}(n) = \frac{1}{N-n}\sum_{i=1}^{N-n} d^2(\mathbf{p}_i, \mathbf{p}_{i+n})$

Table 5. Output from the Celltracking Tool

Cell Number	Total Distance traveled	Maximum distance traveled	…	…	…	…	Mean squared displacement
Cell_1	500	200	…	…	…	…	50
…	…	…	…	…	…	…	
Cell_n	Xn1	Xn2	…	…	…	…	Xn12

10 Expert Opinion

Recent developments are highly application oriented. Often the system works only in a semi-automatic modus [27,28] that puts a lot of work to the user using the system. Standard image processing methods are applied to specific tasks combined with a lot of heuristics [27-31] to make the methods more or less automatically work on the specific images. One such method is the Watershed-Transformation for image segmentation [31]. We have developed a flexible and automatic Case-Based Watershed Transformation method where the WT can be adapted to the image characteristics of the image under consideration.

Standard texture feature extraction procedures are used as well [32] but the random set approach as described here does have the flexibility to describe the different particles appearing in a cell and their randomness. Application-oriented systems that can only solve one specific task are very costly and it takes time to develop them. The success of automatic image-interpretation systems can only be guaranteed when the development effort is as low as possible and when they can be adapted quickly to different needs and tasks. The proposed architecture of Cell Interpret will help to overcome this problem. There are commercial High-Content Analysis developments where data mining capabilities are included in the system. However, a better understanding of when and how to apply these methods and how to interpret the results are necessary for the user. Therefore we are constantly working on a methodology of data mining that is presented in our data mining tutorial (www.data-mining-tutorial.de) and copied in our data mining tools included in Cell Interpret.

Our novel cell tracking method can automatically track the cell in the image without heavy interaction of the user. There are no parameters to adjust, except for the threshold of the similarity. This can be easily done, or the user can take the predefined threshold of the system developer.

Another interesting observation in high-content analysis is that of images are created by using different staining to make specific cell details/objects visible [33,34]. It is obvious that in the resulting images the specific object details/parts are most visible and the analysis of these images can be simply made. However, for a computer vision expert arises the question if this approach is really necessary in all case studies or would it be better to consider the whole task as a pattern recognition problem as has

been done in the HEp-2 cell application and study the different patterns that appear when treating the cells in different ways. This statement might be a bit provocative and we have to admit that we do not know all applications in HCA but we would be happy to further discuss this with experts from the domain.

We also think that a better categorization of the different image analysis tasks is necessary to ensure a standardization of the image analysis procedures in HCA or cellular and molecular biology. A first study in that direction has been given in [35] [36]. Biologists, computer scientists and all other people involved in this field need to further discuss this and find a common basis of understanding. The case-based reasoning approach in our system architecture Cell Interpret we are recently being further developing for cell-tracking and 3D image analysis.

11 Conclusion

In this paper we have presented our architecture, ***Cell_Interpret,*** for High-Content Image Analysis (HCA) and cellular and molecular biology and the methods used for the different tasks such as image segmentation, feature extraction, image mining, classification and interpretation, and cell tracking. Most of the methods are based on case-based reasoning. CBR solves problems using already stored knowledge, and captures new knowledge, making it immediately available for solving the next problem. Therefore, case-based reasoning can be seen as a method for problem solving, and also as a method to capture new experience and make it immediately available for problem solving. It can be seen as a learning and knowledge-discovery approach, since it can capture from new experience some general knowledge, such as case classes, prototypes and some higher-level concepts. In this work we used CBR for meta-learning of parameter for the different processing units and for learning the interpretation knowledge. The idea of case-based reasoning originally came from the cognitive science community which discovered that people are reasoning on formerly successfully solved cases rather than on general rules. Our interest is to build intelligent flexible and robust data-interpreting systems [37-41] that are inspired by the human case-based reasoning process and by doing so to model the human reasoning process when interpreting the cell images.

Other methods that are used are decision tree induction, case-based reasoning, and conceptual clustering. Our decision tree induction unit can learn binary tree and n-ary trees that is novel, and it is not known to us that there is a commercial tool available that can do it. N-ary trees when having a better accuracy than binary trees are often more compact and present the knowledge in a better way to the user. Also, deep learning is nowadays the preferred method, we still rely on decision trees since they give the user an understanding how the decision has been made.

Case-based reasoning is to prefer when no generalized knowledge is available. That is often the case when starting a new HCA or biological experiment.

Conceptual clustering is our choice for learning groups since it is an incremental learning method and brings out the concept of the cluster. We have developed own conceptual learning methods and included in our tool.

Our feature descriptor based on random sets if flexible enough to describe the different texture and has also knowledge explanation capabilities.

Finally, we think that a better categorization of the different image analysis tasks is necessary to ensure a standardization of the image analysis procedures in HCA or cellular and molecular biology. This task must be done together with the domain experts. It will help to develop more specific methods and algorithm that can be easily used by the domain expert without having heavy knowledge in image analysis and interpretation.

References

1. Perner P (2002) Data Mining on Multimedia Data. New York 2558.
2. Perner P, Utility model, computer system for the automatic data analysis, classification, interpretation and data mining of cells, cell structures, microorganism, biotic particle, parts and products in digital images, DE 20206003294 U1.
3. Wang L, Bai J (2003) Threshold selection by clustering gray levels of boundary. Pattern Recognition Letters 24: 1983-1999.
4. Demirkaya O, Asyali MH (2004) Determination of image bimodality thresholds for different intensity distributions. Signal Processing: Image Communication 19: 507-516.
5. Patricio MA, Maravall D (2007) A novel generalization of the gray-scale histogram and its application to the automated visual measurement and inspection of wooden Pallets. Image and Vision Computing 25: 805-816.
6. Pauwels EJ, Frederix G (1999) Finding Salient Regions in Images. Computer Vision and Image Understanding 75: 73-85.
7. Cutrona J, Bonnet N, Herbin M, Hofer F (2005) Advances in the segmentation of multi-component microanalytical images. Ultra microscopy 103: 141-152.
8. Filin S, Pfeifer N (2006) Segmentation of airborne laser scanning data using a slope adaptive neighbourhood. ISPRS Journal of Photogrammetry and Remote Sensing 60: 71-80.
9. Kermad CD, Chehdi K (2002) Automatic image segmentation system through iterative edge–region co-operation. Image and Vision Computing 20: 541-555.
10. Munoz X, Freixenet J, Cufí X, Marti J (2003) Strategies for image segmentation combining region and boundary information. Pattern Recognition Letters 24: 375-392.
11. Voss TC, Demarco IA, Day RN (2005) Quantitative Imaging of Protein Interactions in the cell nucleus. Bio techniques 38: 413-424.
12. Beucher S, Meyer F (1993) 'The morphological approach of segmentation: the watershed transformation', in Dougherty E (Ed.) Mathematical Morphology in Image Processing New York 433-481.
13. Perner P (1999) An architecture for a CBR image segmentation system. Journal of Engineering Application in Artificial Intelligence, Engineering Applications of Artificial Intelligence 12: 749-759.
14. Frucci M, Perner P, Sanniti di Baja G (2007) Case-based Reasoning for Image Segmentation by Watershed Transformation In: Case-Based Reasoning on Signals and Images, Springer Verlag 419-432.

15. Grimnes M, Aamodt A (1996) A two-layer case-based reasoning architecture for medical image understanding. Advances in Case-Based Reasoning Springer Verlag Berlin 164-178.

16. Knowles DW, Sudar D, Bator-Kelly C, Bissell MJ, Lelievre SA (2006) Automated local bright feature image analysis of nuclear protein distribution identifies changes in tissue phenotype. PNAS 103: 445-445.

17. Perner P, Perner H, Janichen S (2006) Recognition of Airborne Fungi Spores in Digital Microscopic Images, Journal Artificial Intelligence in Medicine AIM Special Issue on CBR 36: 137-157.

18. IL Dryden, KV Mardia (1998) Statistical Shape Analysis. John Wiley & Sons Inc 121: 120-1131.

19. Jaenichen S, Perner P (2006) Conceptual Clustering and Case Generalization of two-dimensional Forms. Computational Intelligence 22: 178-193.

20. Zamperoni P (1996) Feature Extraction in H Maitre J Zinn-Justin (Eds.) Progress in Picture Processing 121-184 Elsevier Science.

21. Perner P, Perner H, Muller B (2002) Mining Knowledge for Hep-2 Cell Image Classification. Journal Artificial Intelligence in Medicine 26: 161-173.

22. Matheron G (1975) Random Sets and Integral Geometry. J Wiley&Sons, New York London.

23. Stoyan D, Kendall WS, Mecke J (1987) Stochastic Geometry and Its Applications. Akademie Verlag 345.

24. Wettschereck D, Aha DW (1995) "Weighting Features," in Case-Based Reasoning Research and Development Springer-Verlag: Berlin Heidelberg 347-358.

25. Perner P, Image Mining: Issues, framework, a generic tool and its application to medical-image diagnosis. Journal Engineering Applications of Artificial Intelligence 15: 193-203.

26. Perner P, Perner H, Muller B (2002) Texture Classification based on Random Sets and its Application to Hep-2 Cells. ICPR IEEE Computer Society 2: 406-411.

27. Gokay KE, Wilson JM (2000) Targeting of an Apical Endosomal Protein to Endosomes in Madin-Darby Canine Kidney Cells Requires Two Sorting Motifs. Traffic 1: 354-365.

28. Beil M, Durschmied D, Paschke St, Schreiner B, Nolte U, et al. (2002) Spatial Distribution Patterns of Interphase Centromeres During Retinoic Acid-Induced Differentiation of Promyelocytic Leukemia Cells. Cytometry 47: 217-225.

29. Velliste M, Murphy RF (2002) Automated determination of protein subcellular locations from 3D fluorescence microscope images. IEEE Press: 867-870.

30. Irinopoulou T1, Vassy J, Beil M, Nicolopoulou P, Encaoua D, et al. (1997) Three-Dimensional DNA Image Cytometry by Confocal Scanning Laser Microscopy in Thick Tissue Blocks of Prostatic Lesions. Cytometry 27: 99-105.

31. Swedlow JR, Goldberg I, Brauner E, Sorger PK (2003) Informatics and Quantitative Analysis in Biological Imaging. Science 300: 100-102.

32. Tran D, Pham T, Zhou X (2005) Cell Phase Identification using Fuzzy Gaussian Mixture Models. International Symposium on Intelligent Signal Processing and Communication Systems China 465- 468.

33. JD Lieb, Ortiz de Solorzano C, Garcia Rodriguez E, Jones A, Angelo M (2000) The Caenorhabditis elegans Dosage Compensation Machinery Is Recruited to X Chromosome DNA Attached to an Autosome. Genetics 156: 1603-1621.
34. Ecker RC, Steiner GE (2004) Microscopy-Based Multicolor Tissue Cytometry at the Single-Cell Level. Cytometry 59: 182-190.
35. Swedlow JR, Goldberg I, Brauner E, Sorger PK (2003) Informatics and Quantitative Analysis in Biological Imaging. Science 300: 100-102.
36. Berlage T (2005) Analyzing and mining image databases. Drug Discov Today 10: 795-802.
37. Perner P, Holt A, Richter M (2005) Image Processing in Case-Based Reasoning. The Knowledge Engineering Review 20: 311-331.
38. De Mantaras RL, Cunningham P, Perner P (2005) Emergent case-based reasoning applications. The Knowledge Engineering Review 20: 325- 328.
39. Holt A, Bichindaritz I, Schmidt R, Perner P (2005) Medical applications in case-based reasoning. The Knowledge Engineering Review 20: 289-292.
40. Perner P (2008) Prototype-Based Classification. Applied Intelligence 28: 238-246.
41. Chang CL (1974) "Finding Prototypes for Nearest Neighbor Classifiers." IEEE Trans. on Computers 23: 1179-1184.
42. Perner P., Tracking Living Cells in Microscopic Images and Description of the Kinetics of the Cells, Procedia Computer Science 60 (2015), p. 352-362.