

# Detection of Molecular Particles in Live Cells via Machine Learning

Shan Jiang,<sup>1,2</sup> Xiaobo Zhou,<sup>2,3\*</sup> Tom Kirchhausen,<sup>4</sup> Stephen T. C. Wong<sup>2,3</sup>

<sup>1</sup>Department of Mathematical Sciences, Tsinghua University, Beijing, 100084, People's Republic of China

<sup>2</sup>HCNR-Center for Bioinformatics, Harvard Medical School and Brigham and Women's Hospital, Boston, Massachusetts 02215

<sup>3</sup>Functional and Molecular Imaging Center, Brigham and Women's Hospital, Massachusetts 02115

<sup>4</sup>Department of Cell Biology and the CBR Institute of Biomedical Research, Harvard Medical School, Boston, Massachusetts 02115

Received 25 July 2006; Revision Received 29 January 2007; Accepted 28 February 2007

Grant sponsor: NIH/NLM; Grant number: R01 LM008696; Grant sponsor: Center for Bioinformatics Research Program, Harvard Center of Neurodegeneration and Repair, Harvard Medical School, Boston, USA.

\*Correspondence to: Dr. Xiaobo Zhou, HCNR-Center for Bioinformatics, Harvard Medical School and Brigham and Women's Hospital, Boston, MA 02215, USA.

Email: zhou@crystal.harvard.edu

Published online 12 April 2007 in Wiley InterScience (www.interscience.wiley.com)

DOI: 10.1002/cyto.a.20404

© 2007 International Society for Analytical Cytology

## • Abstract

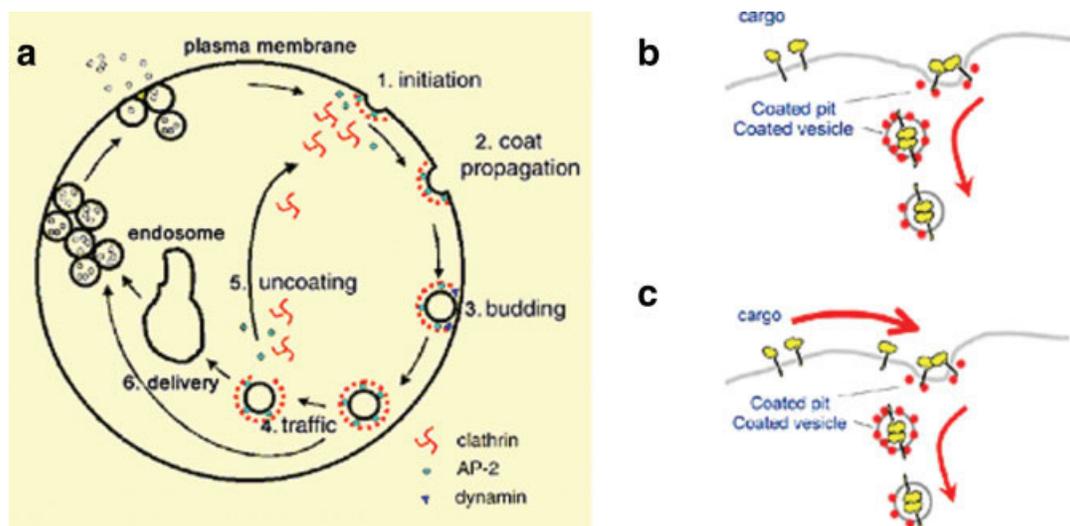
Clathrin-coated pits play an important role in removing proteins and lipids from the plasma membrane and transporting them to the endosomal compartment. It is, however, still unclear whether there exist "hot spots" for the formation of Clathrin-coated pits or the pits and arrays formed randomly on the plasma membrane. To answer this question, first of all, many hundreds of individual pits need to be detected accurately and separated in live-cell microscope movies to capture and monitor how pits and vesicles were formed. Because of the noisy background and the low contrast of the live-cell movies, the existing image analysis methods, such as single threshold, edge detection, and morphological operation, cannot be used. Thus, this paper proposes a machine learning method, which is based on Haar features, to detect the particle's position. Results show that this method can successfully detect most of particles in the image. In order to get the accurate boundaries of these particles, several post-processing methods are applied and signal-to-noise ratio analysis is also performed to rule out the weak spots. © 2007 International Society for Analytical Cytology

## • Key terms

particle detection; machine learning; Haar features; signal-to-noise ratio

**THROUGHOUT** the past decade, cell biologists have been acquiring large volumes of light microscopy images from cells and tissues for the study of cellular dynamics at different biological levels of complexity and resolution, including cell movement, changes in cell shape in response to the environment, intracellular traffic of vesicles, pathogens, nucleic acids, proteins and lipids, biogenesis of organelles, and more recently, the movement and behavior of single molecules within a cell (1–16). There are, however, significant challenges in such high-content bioimaging studies (17), such as accurate segmentation and tracking of the dynamic cellular behavior of cells in a large population or of thousands of moving particles or molecules within a cell. Existing imaging analysis tools derived from these methods are extremely limited in their scope and capacity to analyze high-content, live-cell imaging (18,19). Currently, scientists have to resort to slow, manual analysis to extract information. Image processing and modeling has become the rate-limiting factor to realize the potential of dynamic cellular and molecular imaging studies. In this study, we apply a machine learning approach based on Haar features to detect the particles and extract information of the clathrin-coated pits and vesicles in the cell.

Clathrin-coated pits and vesicles can be found in all nucleated cells, from yeast to humans. They represent an important means by which proteins and lipids are removed from the plasma membrane (endocytosis) and transported to an internal compartment (endosome). The vesicles are also carriers of proteins and lipids from the trans-Golgi network to the endosome and often also participate in many other pathways (5–7). Our understanding of the clathrin-dependent pathway of membrane vesicular traffic (Fig. 1) owes much to the advances in visualization techniques. In the optical microscope, fluorescently labeled versions of a variety of marker proteins



**Figure 1.** (a) cycle of events for clathrin-mediated endocytosis (20); (b,c) Two different models of initiation describing how the assembly of the coat might be coupled to the entrapment of a membrane. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

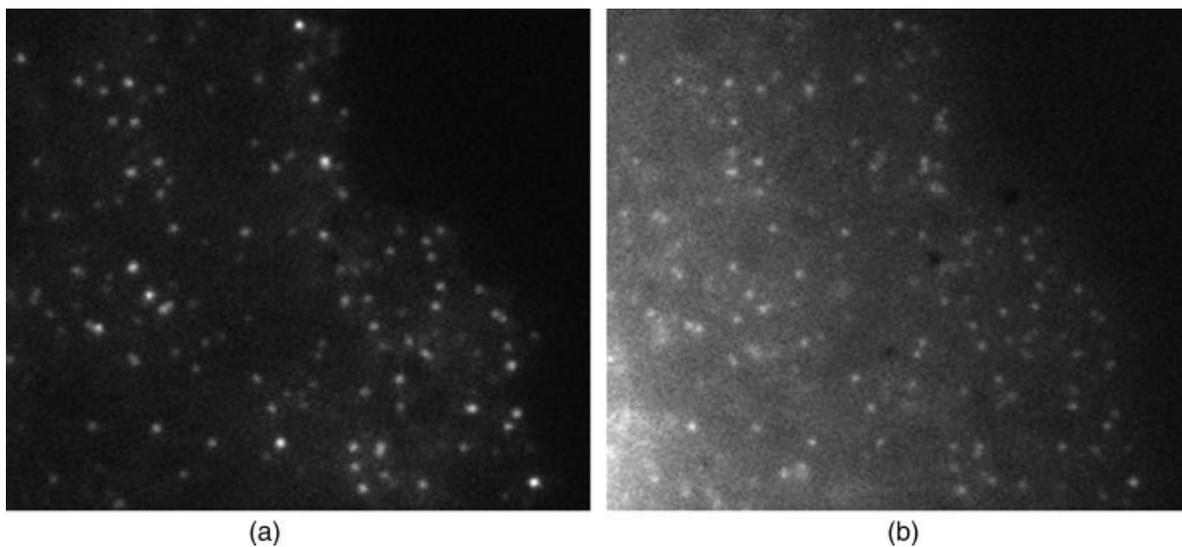
have given us a tantalizing glimpse of the dynamics of the system in living cells. The clathrin pathway has thus acquired special status for analyzing molecular mechanisms in membrane traffic.

Figure 1a shows a simplified cycle of events for clathrin-mediated endocytosis. The main roles of clathrin, the clathrin adaptor (AP) complexes, and dynamin are illustrated. Clathrin is the structural protein involved in coat formation. APs are protein complexes linking the clathrin coat with cargo recruitment. Dynamin is a molecule involved in membrane pinching. The coated vesicles that participate in this process are of reasonably uniform diameters (900–1,500 Å). This figure also shows the assembly and disassembly cycle of a clathrin coat.

On the other hand, Figures 1b and 1c illustrate two different models of initiation describing how the assembly of the coat might be coupled to the entrapment of a membrane-anchored receptor. In Figure 1b, the membrane cargo is part of the nucleation complex that mediates the initiation of coat assembly. In Figure 1c, the membrane cargo diffuses on the plane of the membrane until it finds a partially assembled coat. Coat propagation then proceeds. Many of these steps can now be resolved with the aid of recently developed live-cell imaging acquisition tools (21,22). However, it is difficult to perform the kind of global analysis required for a statistically significant interpretation of the data using the existing software tools. A single image may capture hundreds or thousands of coated pits and vesicles forming at the surface of a cell, which not all behave in the same fashion in time and space, and information about these objects may need to be extracted and assembled at different time points and correlated with information from other images obtained at different spectral frequencies. Biologists can tag clathrin light chains with fluorescent proteins, such as EGFP (enhanced green fluorescent protein) or YFP (yellow fluorescent protein), as first described in

Ref. 5, and also add fluorescent tags, with a distinct spectrum, to cargos, such as LDL (low-density lipoprotein) or virus. We can acquire long time-series data using wide-field, confocal, and total internal reflection fluorescence (TIRF) microscopy. If a software designed for such molecular imaging analysis was available, we would be able to ask questions such as: Are there “hot spots” for the formation of clathrin-coated pits, or do pits and arrays form randomly on the plasma membrane? To answer this question, we would like to track many hundreds of individual pits as they form, by visualizing clathrin light chain, and track whether the pit is successful in forming a vesicle, by simultaneously visualizing the cargo. First of all, we need to detect all the individual pits accurately and extract information for tracking.

Computerized analysis of cellular microscopy images is dependent upon the development of automated segmentation and feature extraction for this new class of image data types. In this article, we will focus on the image data shown in Figures 2 and 3, which will be introduced in the next section. The general framework of analysis in molecular dynamics studies is summarized in Figure 4. Detection of particles is the most critical step in molecular or subcellular image analysis where knowledge of the morphology of particles and the distribution of fluorescence signals in the particles is required. Many methods in detection based on the segmentation technologies (15,18,22–23) are discussed in Ref. 20. All these methods did not work in particle identification in our movies due to the following three reasons: first, there is a strong nonuniform background noise; second, the images have very low contrast; and third, many particles have only a few pixels and are even amorphous. An adaptive threshold segmentation method is proposed to solve this problem (20). Although this method could detect most of the particles accurately, its results of segmentation of the particles are not smooth while under-seg-

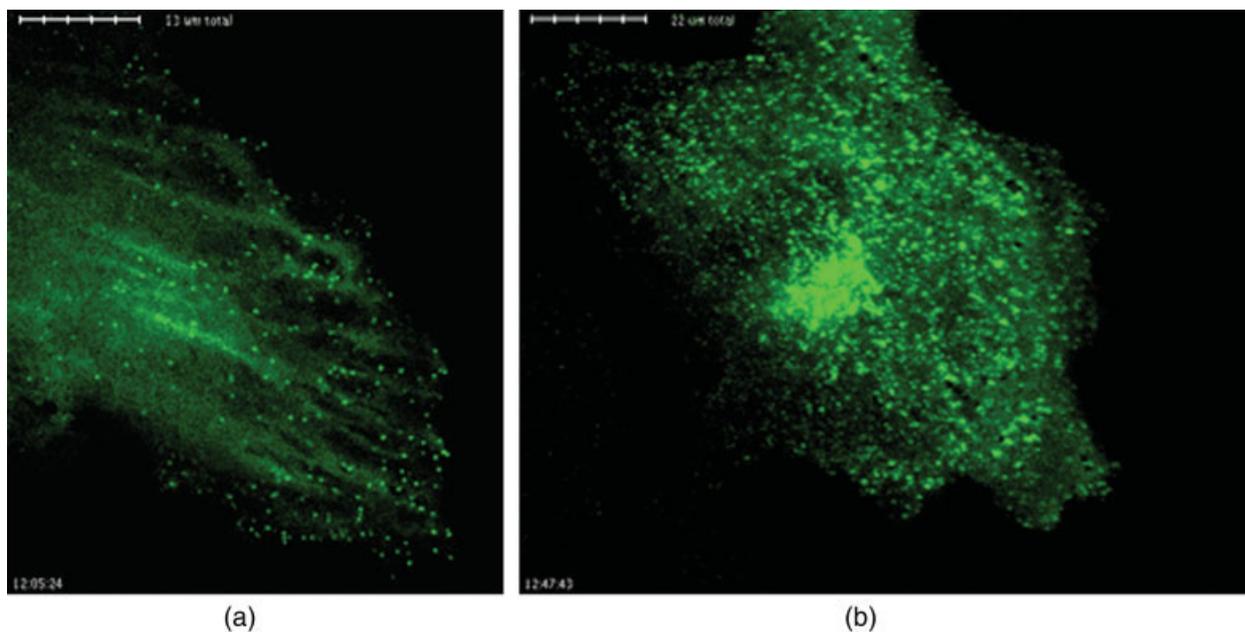


**Figure 2.** The first frame of TIRF (a) and EPI (b) data.

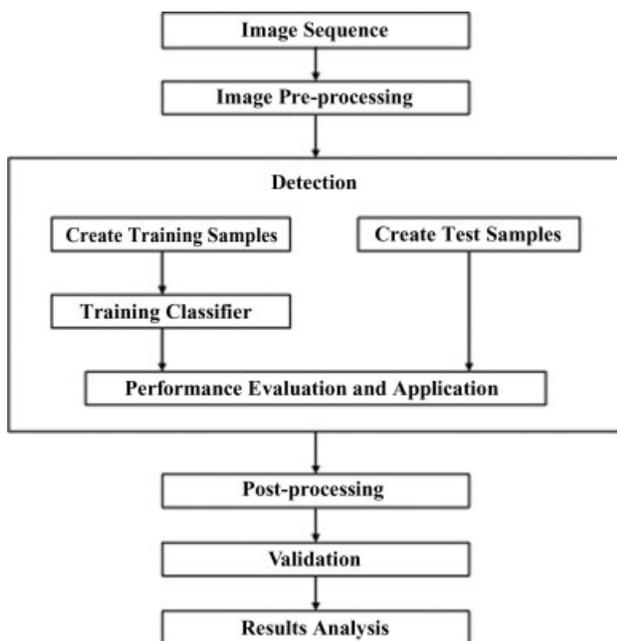
mentation and over-segmentation also exist because this method is based merely on the intensity of the images. Another method of detection is based on artificial neural network (24). This algorithm relies on the combined features of the color texture and temporal texture. Its disadvantage is that it only addresses the issues of the particles moving in the blood vessels. For our molecular dynamics movies, many texture features cannot be used in detection because most of the particles are frequently deformed by the noise.

The article concerns resolving the detection problem described above, also shown in Figure 4. The major contribu-

tions of this article are in the following aspects. First, we apply a novel machine learning approach based on Haar features to detect particles. Since the Haar features include weak and simple defined features, they are the combination of the intensity, shape, and scale information of the objects. Second, since the fluctuation of fluorescence intensity has an inherently random distribution that could result in biased segmentation, we also study the signal-to-noise (SNR) ratio of the spots detected by using the proposed adaptive threshold method and tracking method. The idea is that if the SNR of the spot is too low, that particular spot can be eliminated. We studied two issues: one



**Figure 3.** The first frame of Adaptor (a) and Clathrin (b) data. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]



**Figure 4.** General description of our detecting system.

is the calculation of SNR of each spot, and the other one is the definition of a cutoff value of the SNR; in other words, at what confidence level we can rule out the spot.

In the next section, we introduce the materials acquisition and the machine learning method based on Haar features. The experimental results are presented in Experimentation Results section. Finally, the last two sections provide the discussion and conclusion of this article.

## MATERIALS AND PARTICLE DETECTION

### Materials Acquisition

Images were acquired with a spinning disk confocal coupled to a fully motorized epifluorescence microscopes using  $63\times$  or  $100\times$  lenses under control of SlideBook software. 12-bit digital images were obtained with a cool CCD camera with  $2 \times 2$  binning and a spatial resolution of 140 or 200 nm/pixel for the two lenses, respectively. The 12-bit digital images were transformed into 8-bit images in our study. Images were acquired with exposure times between 800 and 1200 ms from cells maintained at  $37^\circ\text{C}$  by using a heated stage (3). We studied four movies to show that the proposed approach is effective for automatic particle segmentation and tracking. These four movies are the TIRF (total internal reflection fluorescence) movie, EPI (fluorescence microscopy) movie, Adaptor movie, and Clathrin movie. TIRF and EPI movies are shown in Figure 2. The other two movies, Adaptor and Clathrin discussed in Ref. 3, are available in the corresponding *Cell* website and are shown in Figure 3. Time lapse series (60 frames, 10-s intervals acquired at  $37^\circ\text{C}$ ) corresponding to a confocal optical section is obtained from the top surface of a BSC1 (African Green Monkey Kidney) cell constitu-

tively expressing LCa-YFP (green) and is pretreated for 30 min with hypertonic media.

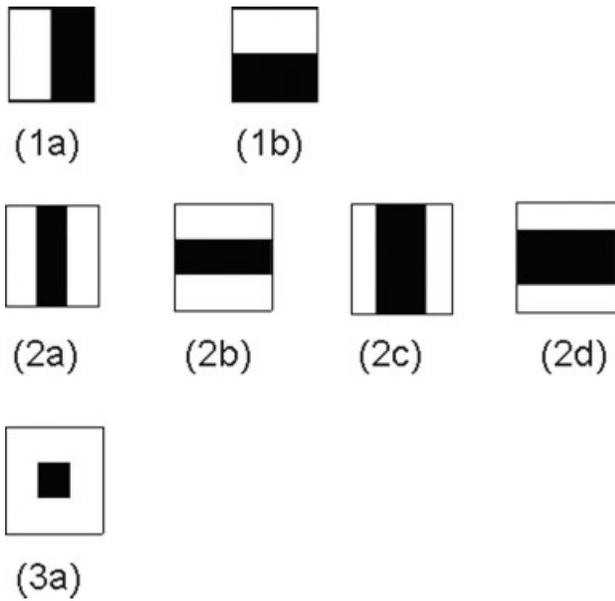
### Particle Detection and Segmentation Using Machine Learning Approach

In this article, we describe a system for fully automated tracking of microparticles and the retrieval of their behavioral characteristics. The task of automated spot tracking in video is particularly challenging because of the spots' indistinct shapes, the instability of the videos, and the abundance of the moving objects, and their frequent superposition. To tackle this problem, we need to detect the spots accurately. Many methods are tried to detect all the particles through working with the entire image, but it is hard to find a good criterion to judge whether it is a particle or not, especially in regards to the touching problem.

One method is the single threshold method for the segmentation of the particles. Otsu (23) proposed a method, which selects the appropriate threshold, to minimize the intra-class variance of the black and white pixels, or conversely to maximize the between-class variance. Iterative algorithm (22) is also used to find the threshold, which serves as the optimal value for maximizing the between-class variance. However, single threshold does not always hold for segmentation of different kinds of images accurately. Figure 3b shows the first frame of a clathrin movie that captures the dynamic behavior of clathrin coated pits and vesicles labeled with LCa-YFP. Figure 12a shows the single threshold segmentation result. Obviously, such a threshold has underestimated the true value. The segmentation result using the method in Ref. 3 is discussed in Ref. 24. The segmentation still needs much improvement as the background noises are too strong (24). To overcome the nonuniform background shade, noises, and shading, global threshold usually cannot meet such requirement. Another method is to apply the watershed algorithm to separate attached objects (25), and Figure 12b shows the result of the watershed segmentation. Obviously, the algorithm failed to isolate attached particles because there is no special fixed shape in the particles.

All the aforementioned methods reported in the literature fail in our movies, because only single threshold is used to resolve the segmentation problem. However, if we can focus on the local region, which has the largest probability to include the object, we can develop an efficient algorithm to find the particles and get the useful information about the particles for tracking and statistical analysis, such as shape, boundary, and roughness. Since many methods based on the intensity fail to analyze our data satisfactorily, we seek new methods that combine the intensity feature with other features.

**A machine learning approach based on Haar features.** A machine learning approach based on Haar features for visual object detection was developed by Viola and Jones (26) and was then extended by Lienhart and coworkers (27,28). This method was implemented in the public domain software, OpenCV, for human face detection. Since this machine learn-



**Figure 5.** Seven feature prototypes of simple Haar-like and center-surround features. (1a) and (1b) are edge features; (2a–2d) are line features; (3a) is center-surround feature.

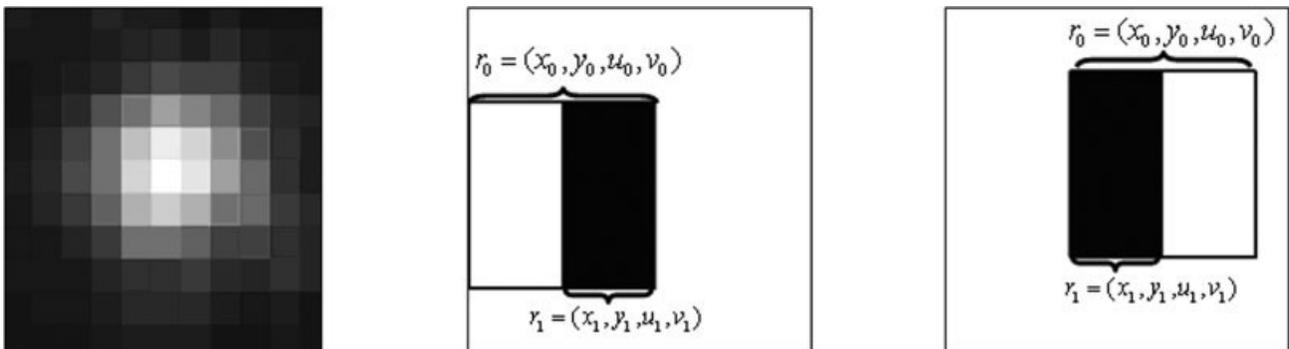
ing approach is effective in detecting different sizes of human faces in one image, we extend it to detect cathrin particles, a different type of objects. Unlike the human face detection, there is a lack of variety of the object class because many applications of human face detection necessitated thousands of instances of objects, which are used to describe the difference between the human face and nonface objects. Furthermore, there are an inadequate number of features available to describe the particles. Based on our knowledge, the particles are predicted to be spots that are slightly lighter than its background and are circular in shape. Many features that can be used to detect cells, such as textures and contours, cannot be directly used to detect the particles in our movies because their sizes are too minute (the smallest one is only 4 pixels in our Clathrin data), while the background noises are too strong. For such objects, a different statistical model (classi-

fier) has to be trained instead and then used to detect the objects.

Haar features are some simple features that are computed in a way similar to the coefficients in Haar wavelet transforms. Each Haar feature is represented by the template (shape of the feature), and its coordinate relative to the search window origin and the size (scale factor) of the feature, as shown in Figure 5. To build up this model for detecting particles, we first construct the training sample sets; it is a considerably important step in the machine learning approach. Then, we deploy a learning algorithm, based on the AdaBoost algorithm, to select a small number of critical visual features from a larger set of features to yield a cost-effective classifier.

*Prepare the training samples.* Statistical model-based training takes multiple “positive” samples, classes of interests, and multiple “negative” samples, images that do not contain the objects of interest, to form one training set. The number of training samples used for training is directly proportional to the accuracy of the results obtained. Moreover, the positive samples must be resized to the same window size for training. This window size cannot be too small, because all the features are extracted from this subwindow. Further, it is also necessary to build a remarkably large and general pool consisting of simple Haar-like features combined with feature selection to increase the capacity of the learning algorithm. After experimenting different sizes of the subwindow, we find that the  $10 \times 10$  size of the subwindow provides the best performance of detection and a moderate computation time to achieve 98% hit rate. When the subwindow size is determined, the feature pool is also determined. A total of 5,725 Haar-like features are created in the feature pool in the  $10 \times 10$  size subwindow. This large data pool ensures the accuracy of the detection results.

The estimation method of the feature number is based on the property of the Haar features, which is discussed in Refs. 27 and 28. Seven feature templates or prototypes are shown in Figure 5. Each feature is represented by a template (shape of the feature) and its coordinate relative to the search window origin and the size (scale factor) of the feature. Therefore, each Haar-like feature template,  $\eta(r_0, r_1)$  with  $i, \dots, 5725$ ,



**Figure 6.** Examples of Haar features in a positive training sample of particles.

is a combination of the “white” rectangle and the “black” rectangle, where  $r_0$  indicates the “white” rectangle, which is also defined as the entire region for each feature, and  $r_1$  indicates the “black” rectangle. Figure 6 shows the examples of Haar feature in a positive training sample. Each rectangle in the feature can be specified by the position information  $(x, y)$  and the size information  $(u, v)$

$$r = (x, y, u, v) \quad (1)$$

with  $0 \leq x, x + u \leq 10, 0 \leq y, y + v \leq 10, x, y \geq 0$ , and  $u, v \geq 0$ , where  $u$  and  $v$  indicate the width and height of the rectangle, respectively. Different feature sizes can be obtained by changing the size of the rectangle in the features. The feature value  $\xi(\eta_i)$  is the difference between the sums of the pixels within two different color rectangle regions. The formulation of computing the upright rectangle region is

$$I(x, y) = \sum_{x' < x, y' < y} \mathfrak{I}(x', y') \quad (2)$$

$$S(r) = S_r(x, y, u, v) = I(x + u, y + v) - I(x + u, y) - I(x, y + v) + I(x, y) \quad (3)$$

where the original image  $\mathfrak{I}$  is transformed to the integral image  $I$  by Eq. (2) (Fig. 7),  $r$  is the region defined for the different rectangles in the feature. The response value  $\xi(\eta_i)$  of each feature  $\eta_i(r_0, r_1)$  is computed by

$$\xi(\eta_i(r_{i,0}, r_{i,1})) = w_{i,0}S(r_{i,0}) + w_{i,1}S(r_{i,1}), \quad \text{with } i = 1, \dots, 5725 \quad (4)$$

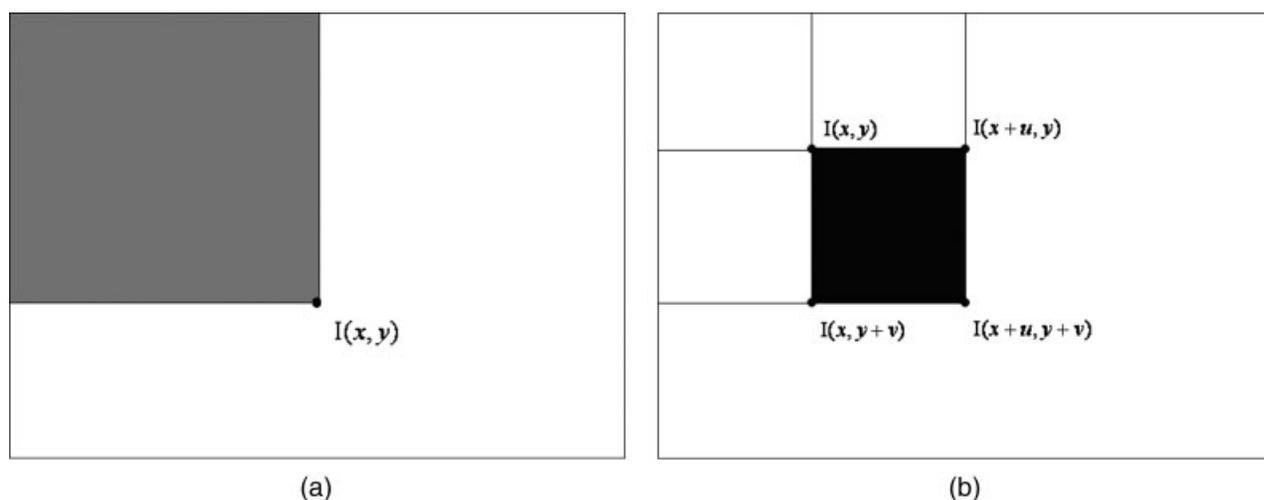
where the weights  $w_{i,0}$  and  $w_{i,1}$  have opposite signs and are used to compensate for the difference in areas between the two rec-

tangles  $r_{i,0}$  and  $r_{i,1}$ . For example, for feature template (1a),  $w_0 = -1$  and  $w_1 = 2$ . Since the Haar-like features include the intensity, shape, and scale information, they can be easily set to be scale invariant to detect different sizes of objects.

*Haar training classifier trained by AdaBoost algorithm.* After the training samples are constructed, the feature pool is built. During training, different features are extracted from the training samples, and distinctive features that can be used to classify the object are selected from the feature pool. Assume that there are  $n$  training samples and  $m$  feature values are computed in each sample image. Therefore, we can attain a feature response matrix  $\xi$ ,

$$\xi = \begin{array}{c} \begin{array}{cc} \text{positive samples} & \text{negative samples} \\ \text{sample } X_1 \dots \text{sample } X_j & \text{sample } X_{j+1} \dots \text{sample } X_n \end{array} \\ \left( \begin{array}{ccc|ccc} \xi_{11} & \dots & \xi_{1j} & \xi_{1j+1} & \dots & \xi_{1n} \\ \xi_{21} & \dots & \xi_{2j} & \xi_{2j+1} & \dots & \xi_{2n} \\ \vdots & \dots & \vdots & \dots & \dots & \vdots \\ \xi_{i1} & \dots & \xi_{ij} & \dots & \dots & \xi_{in} \\ \vdots & \dots & \vdots & \dots & \dots & \vdots \\ \xi_{m1} & \dots & \xi_{mj} & \xi_{mj+1} & \dots & \xi_{mn} \end{array} \right) \begin{array}{l} \text{feature 1} \\ \vdots \\ \text{feature } i \\ \vdots \\ \text{feature } m \end{array} \end{array}$$

where  $n = 2,000$  is the sample size and  $m = 5,725$  is the feature number in this study; each element  $\xi_{ij}$  of the matrix  $\xi$  is the feature response value  $\xi(\eta_i)$  for the sample  $x_j$  computed by Eq. (4). Here,  $x_j$  is a  $10 \times 10$  pixel subwindow of an image. For each feature, we train a classifier which chooses the optimal threshold  $t$  to minimize the number of samples that are misclassified by this feature. This classifier is restricted to use a single feature. Recall that each sample  $x_j$  is  $10 \times 10$  in size, and we can use each feature extracted from it to determine



**Figure 7.** Computational scheme of the rectangle region of the features in integral image. (a) Each element  $I(x, y)$  in integral image is the sum of all pixels above and to the left at position  $(x, y)$  in original image  $\mathfrak{I}$ ; (b) The value of the “black” region can be computed with only four elements in integral image  $I$ .

whether it is a particle or not. Therefore, we can obtain a classifier matrix  $\mathbf{F}$ ,

$$\mathbf{F} = \begin{array}{c} \begin{array}{c} \text{positive samples:} \\ y_j = 1 \end{array} \quad \begin{array}{c} \text{negative samples:} \\ y_j = 0 \end{array} \\ \begin{array}{c} \text{sample } x_1 \quad \dots \quad \text{sample } x_j \quad | \quad \text{sample } x_{j+1} \quad \dots \quad \text{sample } x_n \end{array} \\ \left( \begin{array}{ccccccc} f_{11} & \dots & f_{1j} & f_{1j+1} & \dots & f_{1n} \\ f_{21} & \dots & f_{2j} & f_{2j+1} & \dots & f_{2n} \\ \vdots & \dots & \vdots & \dots & \dots & \vdots \\ f_{i1} & \dots & f_{ij} & \dots & \dots & f_{in} \\ \vdots & \dots & \dots & \dots & \dots & \vdots \\ f_{m1} & \dots & f_{mj} & f_{mj+1} & \dots & f_{mn} \end{array} \right) \begin{array}{l} \xrightarrow{\text{feature 1}} \mathbf{F}_1 = (f_{11} \quad f_{12} \quad \dots \quad f_{1n}) \\ \vdots \\ \xrightarrow{\text{feature } i} \mathbf{F}_i = (f_{i1} \quad f_{i2} \quad \dots \quad f_{in}) \\ \vdots \\ \xrightarrow{\text{feature } m} \mathbf{F}_m = (f_{m1} \quad f_{m2} \quad \dots \quad f_{mn}) \end{array} \\ \begin{array}{l} m \times n \end{array} \end{array}$$

where  $f_{ij}$  is the result of the classifier  $f_i$  based on  $\xi_{ij}$ :

$$f_{ij} = f_i(\xi_{ij}) = \begin{cases} 1, & \xi_{ij} \geq t_i \\ 0, & \xi_{ij} < t_i \end{cases}, \quad \text{with } i = 1, \dots, m \quad (5)$$

where  $\xi_{ij}$  is the element of the feature matrix  $\xi$  and  $t_i$  is the threshold for Haar-like feature  $\eta_i(r_0, r_1)$ , shown in Figure 8; the response 1 means the object, and 0 means the nonobject. Every such classifier, called a weak classifier, is not able to detect an object; rather, it reacts to some simple feature in the image that may relate to the object.

The AdaBoost algorithm (29) is used to augment the performance of these weak classifiers. Considering that some readers may not be familiar with this algorithm, we summarized it as follows:

- Step 1, label training samples  $(x_1, y_1), \dots, (x_m, y_m)$ , where  $y_j = 0, 1$  for negative and positive samples, respectively,

and obtain the labeled training samples matrix  $\mathbf{Y} = (y_1, y_2, \dots, y_n)'$ .

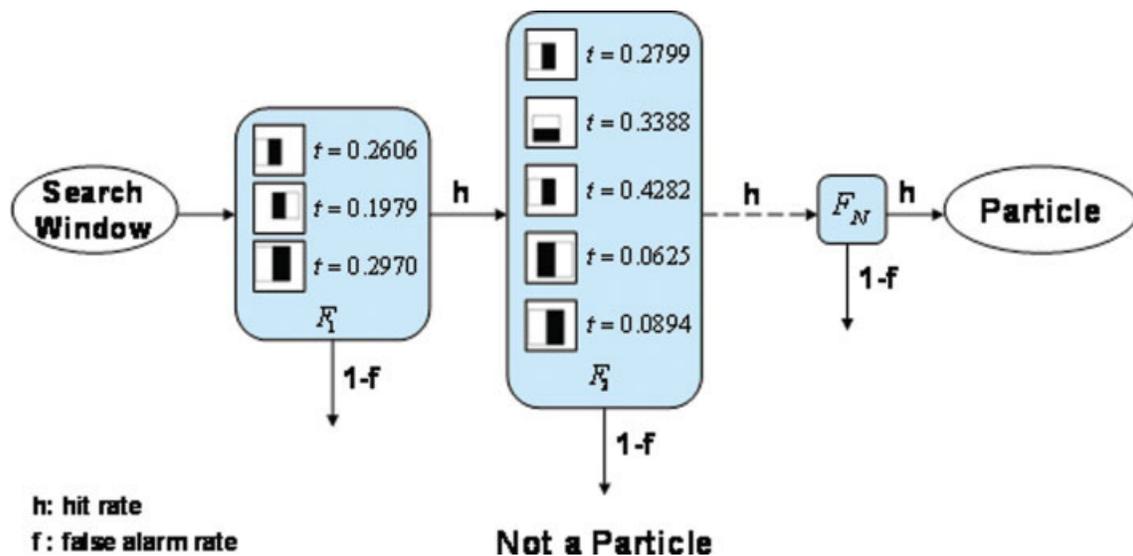
- Step 2, initialize weights matrix  $\omega = (\omega_1 \quad \omega_2 \quad \dots \quad \omega_n)$ , where  $\omega_j = \frac{1}{n}$  for each sample  $(x_j, y_j)$ .
- Step 3, For  $t = 1, \dots, T$ ,

- Evaluate the performance of each feature, or weak classifier. For each feature  $\eta_i(r_0, r_1)$ , a weak classifier  $f_i$  is trained and the error is evaluated with respect to  $\omega_i$ ,

$$\varepsilon_i = \omega_i \cdot \text{abs}(\mathbf{F}'_i - \mathbf{Y}), \quad i = 1, \dots, m \quad (6)$$

where  $\mathbf{F}_i = (f_{i1} \quad f_{i2} \quad \dots \quad f_{in})$  is the row vector in matrix  $\mathbf{F}$  and  $\omega_1 = \omega$ ;  $\text{abs}(\mathbf{F})$  involves taking the absolute value of each matrix or vector element.

- Choose the classifier  $f_i$  with the lowest error  $\varepsilon_t$  and update the weight matrix  $\omega_t = (\omega_{t1} \quad \omega_{t2} \quad \dots \quad \omega_{tn})'$  by  $\omega_{t+1,j} = \omega_{t,j} \beta_t^{f_j}$  with  $j = 1, \dots, n$  and  $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ , and renormalize weights so that  $\|\omega_t\|_1 = 1$ .



**Figure 8.** Particle detection cascade of classifiers, where rejection can happen at any stage.  $t$  is the threshold value for each weak classifier.  $F_1, \dots, F_N$  are the strong classifiers trained by Adaboost algorithm in each stage. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

- Step 4, the final strong classifier consisting of all selected weak classifiers is

$$F(\xi_{1j}, \dots, \xi_{Tj}) = \sum_{t=1}^T \alpha_t f_{tj} = \sum_{t=1}^T \alpha_t f_t(\xi_{tj}), \quad (7)$$

where  $\alpha = -\log\beta_t$  is the final weight of the weak classifier  $f_t$ . The sample image  $x_j$  is determined as including a particle when  $F(\xi_{1j}, \dots, \xi_{Tj}) \geq (1/2) \sum_{t=1}^T \alpha_t$ , otherwise not.

Theoretically, the classifier  $F$  should have achieved an arbitrarily high ( $<1$ ) hit rate and an arbitrarily small ( $>0$ ) false alarm rate, if the number of weak classifiers in the sum (ensemble) is large enough. However, in practice, that would require a substantially large training set as well as a substantially large number of weak classifiers, resulting in a slow processing rate. Therefore, we use the required hit rate and false alarm rate as the stop criterion in our program.

From the boosting process, the feature number becomes an important factor in relation to the computational time. Although the computation of the feature's value based on the integral image is very fast, the large number of features in the feature pool will also increase the time for the training classifier to achieve a high hit rate ( $0.999^{20}$ ) and a low false alarm rate ( $0.5^{20} = 10e-7$ ). Therefore, we exclusively choose those features that resemble the particles in shape the most for computation. Intuitively, we assume the particles are almost circular. We restrict the features extracted from training samples to the square shape, based on the congruous circular shape assumption made previously. Recall that there are 5,725 rectangle features associated with each image sub-window. Subsequent to the restriction of the feature's shape, the number of features that are used to train classifier decreases significantly. Yet, we still combine a small number of these features to form an effective classifier. The technique, called "Cascade" algorithm (26), can also be applied to improve the performance of the boosted classifier. The whole process of the training classifier is shown in Figure 8.

**Detection process.** When the classifier is trained and tested, detection can be done by sliding a search window of different size through the image and by checking whether an image region at a certain location "looks like a particle" or not. Fifty-seven features are selected from the feature pool to build up the classifier. Initially, the rectangular search region in the given image is estimated by the classifier to find out whether it contains the object. The program scans the image several times at different scales. Each time it considers overlapping regions in the image and applies the classifiers to the regions.

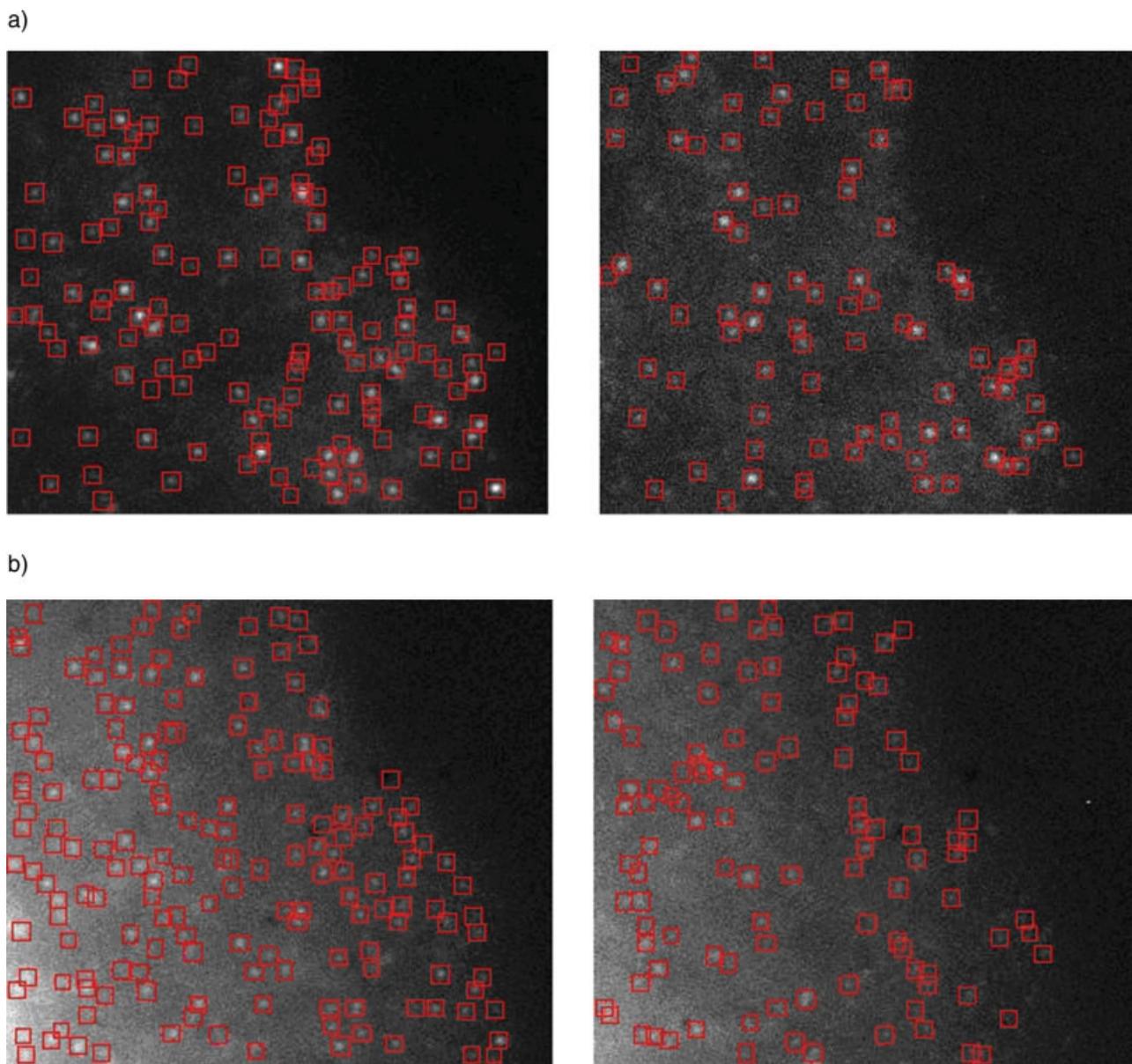
To improve the detection performance, we use the Canny edge detector to reject some image regions that contain too few or too much edges and indeed cannot contain the searched object. Furthermore, we fix the searching window sizes from  $8 \times 8$  to  $11 \times 11$  and eliminate the large window which contains a small window inside, because the interpolation technique may reduce the accuracy of detection. We use the Haartraining classifier to detect four movies, i.e. the TIRF (total internal reflection fluorescence) movie, EPI (fluores-

cence microscopy) movie, Adaptor movie, and Clathrin movie (shown in Figs. 9 and 10). The results show that our classifier works well with all these four movies. The movies of Adaptor and Clathrin can be downloaded from the website of the original paper (3).

**Postprocess the detection results.** After the detection, some postprocessing methods should be applied to these rectangle regions to extract appearance information of particles, such as boundary and area, for accurate tracking. It is difficult to accurately segment all the particles in the whole image because there is not any special fixed shape in the particles. Therefore, many methods, such as watershed algorithm, fail to segment all the particles in the whole image. Fortunately, not too much postprocessing work is needed in the local region because more attention is paid to the information in the rectangular region.

First, the intensity and the texture features are extracted from the rectangle region. To augment the accuracy of segmentation, we classify all rectangle regions in accordance to their corresponding features because there are discrepancies between the background noise and the contrast in the image. The mean, maximum and minimum intensity, contrast, and energy texture features are selected to determine the standards of classification of the rectangular regions. Consequent to the classification, three classes are identified. The first class is a region with high intensity and high contrast; the second class is a region with high intensity features and low contrast; a very low intensity results in a low texture, which will not be considered in this experiment. Contrast, which is also known as variance or inertia, is calculated by  $r_c = \sum_{i,j} (i-j)^2 P(i,j)$ ; energy is also known as uniformity and angular second moment, and is calculated by  $r_E = \sum_{i,j} P(i,j)^2$ .  $P(i,j)$  is the co-occurrence matrix of the rectangle region (30,31).

For each rectangle region, a single threshold method, which is dependent upon the fuzzy c-means, is used to segment the objects from the background. Fuzzy c-means (32) is one of the most powerful algorithms in clustering algorithms because of its convergence property and low complexity. For the first class of the rectangular regions, we use the fuzzy c-means clustering to partition the image pixels into three distinct classes. Assume that the first class is the background, the second class is the noise and the third class is the object of interest. The threshold value is set to be the median of the maximum value of the second class and the minimum value of the third class. For the second class of the rectangular regions, the threshold value is set to be the median of the maximum value of the first class and the minimum value of the second class to include more candidates. Morphological operations and shape and size constrain are applied to the separated objects in every region after segmentation. We first create a circle mask and then apply the morphological erosion and dilation operations to smooth the object and remove the noise in regards to the shape and size constrain. Consequently, those objects, whose sizes are less than 4 pixels, shapes resemble lines, and widths are 1-pixel wide, are discarded. We then apply the signal-noise



**Figure 9.** The detection results from Haartraining Classifier to the TIRF and EPI data. (a) The detection results from TIRF data; (b) The detection results from EPI data. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

ratio method to eliminate the weak spots detected by using the above-proposed detecting method and postprocessing results.

The basic idea of the SNR is that a spot will be eliminated if the SNR of the spot is too low. Global SNR is calculated by the difference in mean intensity between the object  $I_s$  and background  $I_b$ , divided by a representative noise level

$$\text{SNR}_g = (I_s - I_b)/\sigma \quad (8)$$

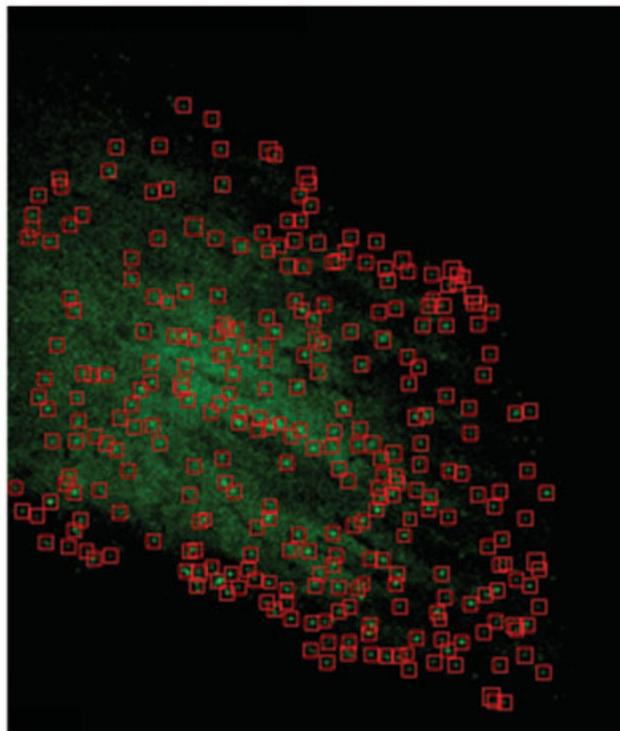
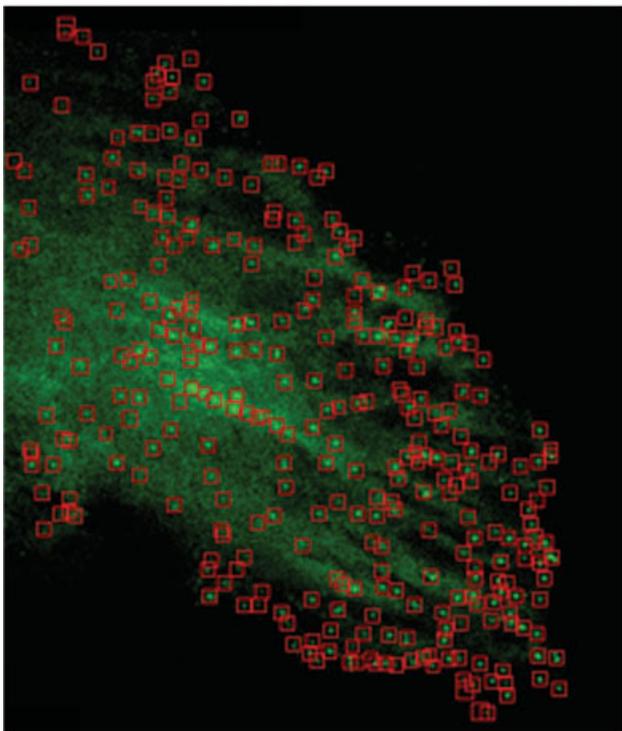
where  $I_s$  is defined as the average of the detected spots,  $I_b$  is defined as the average of the background pixels, and the  $\sigma$  is defined as the standard deviation of the background ( $\sigma_b$ ).

To estimate the local SNR for each particle, assume that we are studying the particle with the center defined as  $P$  without the loss of generality. Following the Haartraining classifier approach, we use the same rectangular region as the local region for spot  $P$ . Denote  $I_{sp}$  as the average of the detected signals,  $I_{bp}$  as the average of the detected background, and  $\sigma_{sp}$  as the standard deviation of the background of this region. Then the local SNR for spot  $P$  is estimated by

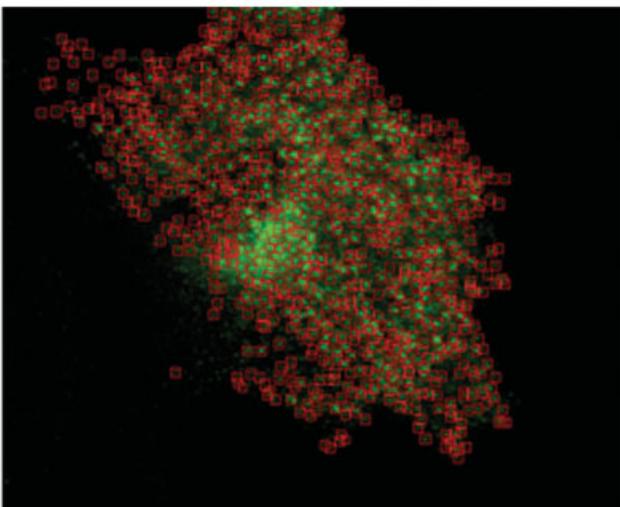
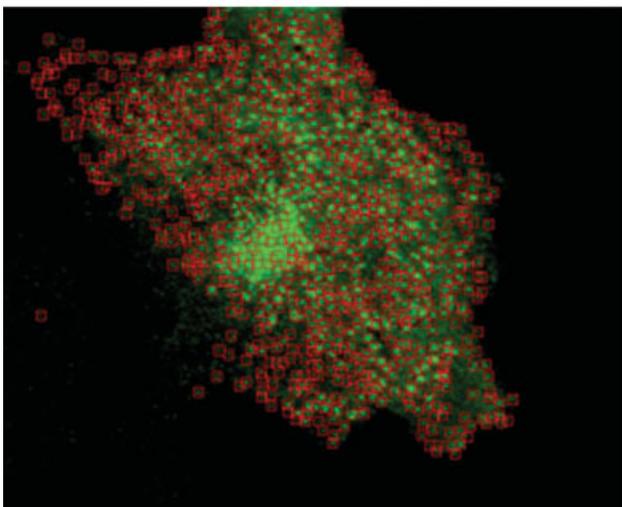
$$\text{SNR}_p = (I_{sp} - I_{bp})/\sigma_{bp} \quad (9)$$

One problem for estimating  $\sigma_{bp}$  is that this value is estimated from the local region, but there are a limited number of pixels

a)



b)



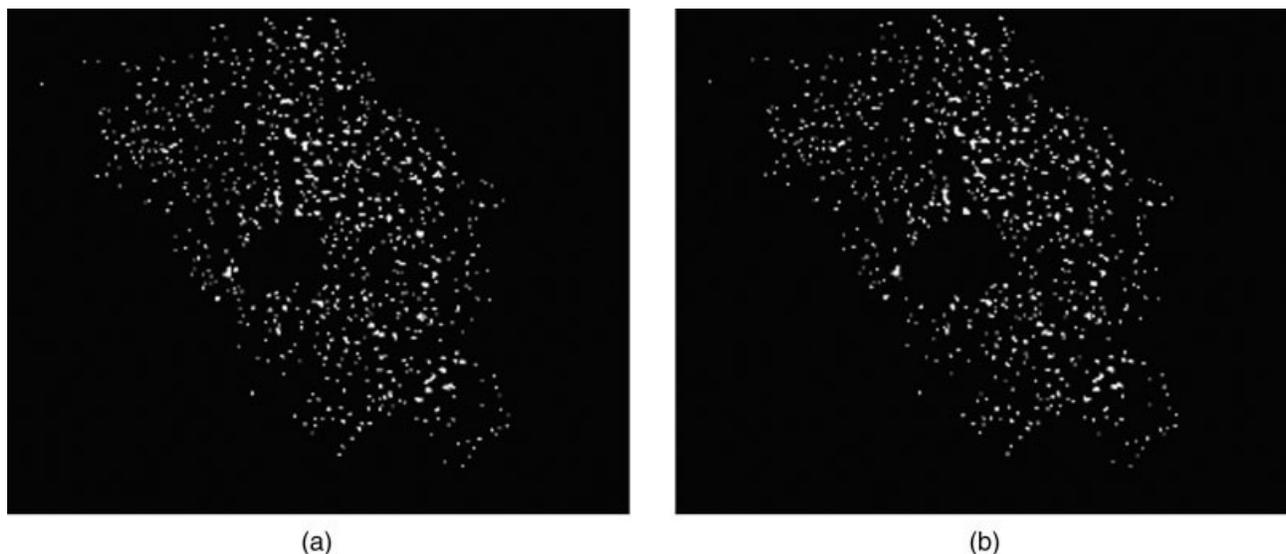
**Figure 10.** The detection results from Haartraining Classifier to the Adaptor and Clathrin data. (a) The detection results from Adaptor data; (b) the detection results from Clathrin data. [Color figure can be viewed in the online issue, which is available at [www.interscience.wiley.com](http://www.interscience.wiley.com).]

available for comparison in the entire image. So, one possible solution to replace the  $\sigma_{bp}$  by  $\lambda\sigma_{bp} + (1 - \lambda)\sigma_b$ , where  $\lambda(0 < \lambda \leq 1)$  is a weight to balance the estimated bias from local region. In this study, we set the parameter as 0.9.

Ehrlich et al. (3) considered a standard normal distribution and calculated the confidence level based on real signals.

Given a standard normal distribution assumption, the confidence level of the signal with S/N is  $\text{SNR}_p$ , which is defined by a Q-function as

$$\begin{aligned} \text{CL}(\text{SNR}_p) &= p(|z| \leq \text{SNR}_p) = 1 - p(|z| > \text{SNR}_p) \\ &= 1 - 2Q(\text{SNR}_p) \end{aligned} \quad (10)$$



**Figure 11.** (a) Segmentation result by using the Haar-training classifier followed by the postprocessing; (b) The segmentation result after ruling out the spots using SNR-based criterion.

where  $Q$ -function is the tail integral of a unit-Gaussian probability of function and is defined as  $Q(x) = (2/\sqrt{2\pi}) \times \int_x^\infty \exp(-t^2/2)dt$ . The  $Q$ -function can be calculated by

$$Q(x) = 0.5 \left[ 1 - \operatorname{erf}(x/\sqrt{2}) \right] = 0.5 \operatorname{erfc}(x/\sqrt{2}) \quad (11)$$

where  $\operatorname{erf}(x) = (2/\sqrt{\pi}) \int_0^x \exp(-t^2)dt$  and  $\operatorname{erfc}(x) = (2/\sqrt{\pi}) \int_x^\infty \exp(-t^2)dt$  can be calculated by using the Matlab function  $\operatorname{erf}(x)$  and  $\operatorname{erfc}(x)$  separately.

Assume that we want to calculate the confidence level being 99.9%, then we can calculate the lowest SNR required such that we have 99.9% confidence to say that this signal is real. Given a confidence level  $\alpha$ , the lowest SNR required can be calculated by

$$\operatorname{SNR}_0 = \sqrt{2} \operatorname{erf}^{-1}(2\alpha - 1) \quad (12)$$

where  $y = \operatorname{erf}^{-1}(x)$  is the inverse error function of  $Q$ -function for parameter  $x$ . The inverse error function satisfies  $x = \operatorname{erf}(y)$ , for  $-1 \leq x \leq 1$ , and  $-\infty \leq y \leq \infty$ . According to Eq. (10), if we want to have a 99.9% confidence to say that this signal is real, then the SNR is expected at least to be 3.09. It is always a challenge on how to determine the confidence level or the corresponding SNR cutoff value. In our clathrin movies, the global SNR is estimated to be 3.38, and the corresponding confidence level is 99.93%. It indicated that the results were fairly good, but some objects with a low local SNR, which should have been eliminated, might have remained or left off in our calculations. We set the SNR cutoff value to be 1.5 and obtain a new global SNR value, 3.4081. Finally, the global SNR value terminates at the value of 3.41 after several iterations. After the above process of elimination, the particle segmentation results are shown in Figure 11. We also compare our segmentation

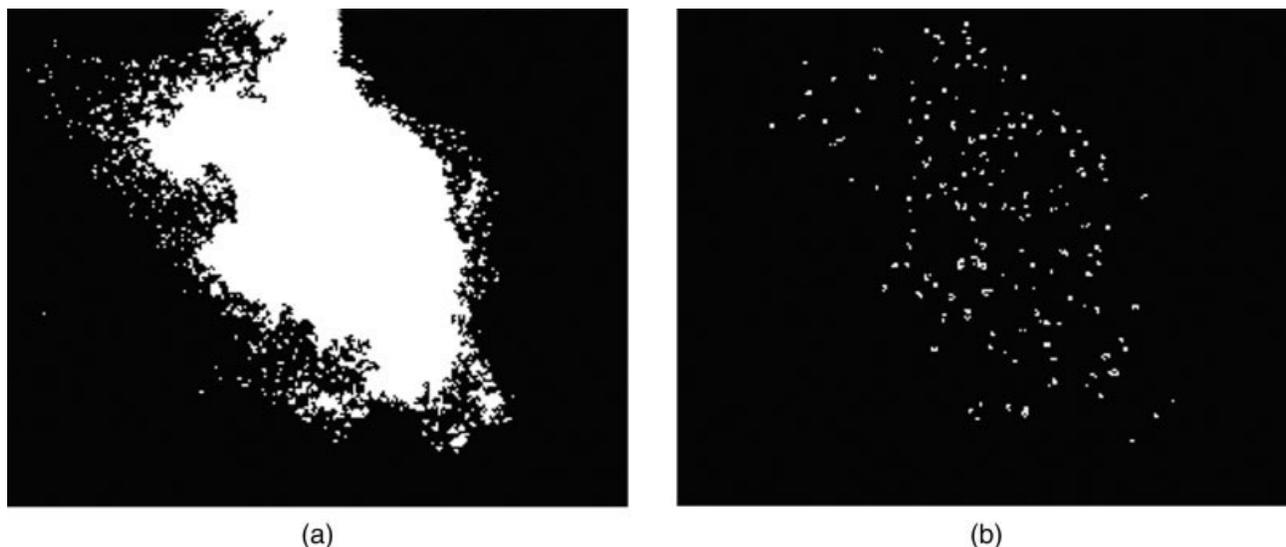
results with the results obtained from the Otsu's method and the watershed segmentation (Fig. 12). We find out that our method has improved the detection performance.

## EXPERIMENTATION RESULTS

Images we studied are discussed in Materials and Particle Detection section. Before analyzing the experimental results, we remind that the movies we studied are the TIRF (total internal reflection fluorescence) movie, EPI (fluorescence microscopy) movie, Adaptor movie, and Clathrin movie, which are shown in Figures 2 and 3. Time lapse series (60 frames, 10-s intervals acquired at 37°C) corresponding to a confocal section is obtained from the top surface of a BSC1 (African Green Monkey Kidney) cell constitutively expressing LCa-YFP (green) and is pretreated for 30 min with hypertonic media. The study of these four movies shows that the proposed approach is effective for automatic particle segmentation and tracking. We discuss the accuracy of particle detection in this section. The proposed methods are implemented by Matlab.

Figure 3b is the first frame of Clathrin movie that captures the dynamic behavior of the clathrin-coated pits and vesicles labeled with LCa-YFP. This example is the time-lapse series (120 frames, 10-s intervals acquired at 37°C) obtained from a cell constitutively expressing LCa-YFP using the spinning disk confocal microscope (3).

To evaluate the accuracy of particle detection, we define four numbers to demonstrate the performance of our detection method. The false positive (FP) is defined as the erroneous detection of a spot as particle, whereas actually the spot is not a particle; the false negative (FN) is defined as the erroneous detection of a spot as nonparticle, whereas actually the spot is a particle. Another reminder is that there is always a tradeoff between false positives and false negatives. In terms of



**Figure 12.** (a) The segmented image using Otsu's method; (b) Result of watershed segmentation.

statistical hypothesis testing, the tradeoff is a matter of balancing the risk of Type I errors (false positives that reject the null hypothesis when it is true) against Type II errors (false negatives that fail to reject the null hypothesis when it is false). True positive (TP) is defined as the true detection of a spot as particle; True negative (TN) is defined as accurate detection of the spot not to be a particle. In the total number of detected spots, the number of true particles is (TP + FN), and the number of nonparticles is (FP + TN). In this article, we use the true positive rate,  $TPR = TP/(TP + FN)$ , and the false positive rate,  $FPR = FP/(TP + FN)$ , to investigate the performance of the proposed approach.

We first analyzed the TIRF and EPI movies. Ninety-seven particles were counted manually in the first frame of the TIRF movie and 108 particles were counted manually in the first frame of the EPI movie. A high global SNR (4.69) of the TIRF movie represents good detection results; in other words, it shows that only a few weak spots are discarded, whereas the EPI movie gets a very low SNR value (1.28) and its average SNR value in the whole movie is merely 1.35. These values can be attributed to the strong background noise in the movies (Table 1). We then analyzed the Adaptor and Clathrin movies. Hundred and seventy one particles were counted manually in

the first frame of the Adaptor movie and 653 particles were counted manually in the first frame of the Clathrin movie. As shown in Materials and Particle Detection section, Otsu's method failed to segment such images. The segmented method used in Ref. 3 only works for weak background noise. The proposed detection method only needs a few, simple post-processing steps. The cutoff value of the SNR is set to be 1–2.

Table 1 summarizes the performance of the proposed segmentation approach, where step 1 means the proposed detection approach; step 2 means the postprocessing methods; step 3 means ruling out the spots-based SNR analysis; and  $SNR_g$  means global SNR. From this table, we can see that the detection accuracy is quite robust, even the property of the movies that we deal with are quite different because the true positive rate is between 97.08 and 99.07%, as well as the false positive rate is between 2.14 and 7.41%. According to the above four examples, the proposed approach works very well in low contrast and nonuniform background noise environment.

## DISCUSSION

For particle detection, the machine learning approach based on Haar feature is a novel technique. Also, we know that

**Table 1.** Performance of the proposed detection approach with the four movies

MOVIE	STEP 1	STEP 2	STEP 3	$SNR_g$	FP	FN	TP	TPR (%)	FPR (%)
TIRF	132	106	99	4.6892	4	2	95	97.94	4.12
EPI	181	144	115	1.2798	8	1	107	99.07	7.41
Adaptor	323	325	173	3.1184	7	5	166	97.08	4.09
Clathrin	997	716	654	3.4094	14	13	640	98.01	2.14
Average								98.02	4.44

Step1: The result from the Haar-training classifier; Step 2: The result after postprocessing; Step 3: Rule out spots-based SNR analysis;  $SNR_g$ : global SNR; FP: false positive number; FN: false negative number; TP: true positive number; TPR: The rate of true positive number to the total number of particles; FPR: The rate of false positive number to the total number of particles.

one of the important steps in training classifier is to prepare training data sets and the efficiency of the classifier strongly depends on the quality of the training samples. Although we could have improved the Haartraining classifier, it would not have impact to the touching problem of particles. It is because the particles are too small and nonrigid in shape. One may wonder why not choosing a small search window to detect the particles. Even though we could have used a small search window to do the detection, such as  $4 \times 4$  subwindow size, it would mistakenly detected the absent object in three aspects. First, the majority of the spots are objects within 200–300 nm in diameter (3 by 3 pixels at 130 nm/pixel resolution), corresponding to the way a coated pit/vesicle 100–200 nm in diameter would appear. The particles' sizes are too small and the particles' boundaries are amorphous and blurred. If we only use the  $4 \times 4$  subwindow for training, there are only 121 features that can be used for training classifier. This number is too small to be used for training the classifier based on Haar features with a high accuracy of detection. To attain a powerful classifier, we need to choose a larger subwindow size for training. However, it will reduce accuracy and increase the possibility of getting more false detections by using the interpolation method to make the object as large as the enlarged window size. A small window size will result in a lack of features available for training the classifier and the enlarged window size will reduce the accuracy of detection. Secondly, in a small region, it is hard to judge whether it is a particle or it is merely noise. Finally, the center of the rectangle may not be the exact center position of the spot because there is more or less background in the rectangular region. Therefore, we conclude that  $10 \times 10$  will be the most desirable size for our experimentation. Furthermore, additional postprocessing methods are needed to resolve the touching problem in the local region.

## CONCLUSION

To deal with strong background noise and low contrast dynamic images of living cells, the traditional segmentation methods, such as single threshold, edge detection, and morphological transform, do not work. In this study, we introduced a novel machine learning method based on Haar feature to detect the fast moving spots. After the detection, several postprocessing methods are used to segment the spots out of the rectangle regions, and the SNR criterion is used to rule out the weak spots. Results show that the proposed method worked well for representative movies with the segmentation of clathrin-coated pits, vessels, and receptors. The method presented here thus has the potential to provide a cost-effective solution to resolve detection of subcellular molecular particles in living cells. The next step of this research will focus on tracking the molecular particles based on the information extracted in this article.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the collaboration with their biology collaborators in the Department of Cell Biology and The CBR Institute of Biomedical Research at Har-

vard Medical School in this research effort. They would like to thank other research members of the Life Science Imaging Group of the Center for Bioinformatics, Brigham and Women's Hospital and Harvard Center for Neurodegeneration and Repair, Harvard Medical School for their technical comments.

## LITERATURE CITED

- Gaidarov I, Santini F, Warren RA, Keen JH. Spatial control of coated-pit dynamics in living cells. *Nat Cell Biol* 1999;1:1–7.
- Boland MV, Murphy RF. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics* 2001;17: 1213–1223.
- Ehrlich M, Boll W, Oijen AV, Hariharan R, Chandran K, Nibert ML, Kirchhausen T. Endocytosis by random initiation and stabilization of clathrin-coated pits. *Cell* 2004;118:591–605.
- Ghosh RN, Webb WW. Automated detection and tracking of individual and clustered cell surface low density lipoprotein receptor molecules. *J Biophys* 1994;66:1301–1318.
- Hirst J, Robinson MS. Clathrin and adaptors. *Biochim Biophys Acta Mol Cell Res* 1998;1404:173–193.
- Kirchhausen T. Clathrin. *Ann Rev Biochem* 2000;69:699–727.
- Kirchhausen T. Three ways to make a vesicle. *Nat Rev Mol Cell Biol* 2000;1:187–198.
- Kirchhausen T. Clathrin adaptors really adapt. *Cell* 2002;109:413–416.
- Kanda T, Sullivan KF, Wahl GM. Histone-GFP fusion protein enables sensitive analysis of chromosome dynamics in living mammalian cells. *Curr Biol* 1998;8:377–385.
- Lamaze C, Fujimoto LM, Yin HL, Schmid SL. The Actin Cytoskeleton is required for receptor-mediated endocytosis in mammalian cells. *J Biol Chem* 1997;272:20332–20335.
- Merrifield CJ, Feldman ME, Wan L, Almers W. Imaging actin and dynamin recruitment during invagination of single clathrin-coated pits. *Nat Cell Biol* 2002;4:691–698.
- Puertollano R, Wel NNvd, Greene LE, Eisenberg E, Peters PJ, Bonifacino JS. Morphology and dynamics of clathrin/GGA1-coated carriers budding from the trans-Golgi network. *Mol Biol Cell* 2003;14:1545–1557.
- Rappoport JZ, Simon SM. Real-time analysis of clathrin-mediated endocytosis during cell migration. *J Cell Sci* 2003;116:847–855.
- Rappoport JZ, Taha BW, Simon SM. Movement of plasma-membrane-associated clathrin spots along the microtubule cytoskeleton. *Traffic* 2003;4:460–467.
- Won CS. *Stochastic Image Processing*. New York: Kluwer Academic/Plenum; 2004.
- Zhou X, Wang X. FIR-Median hybrid filters with polynomial fitting. *J Dig Signal Process* 2004;14:112–124.
- Abraham VC, Taylor DL, Haskins JR. High content screening applied to large-scale cell biology. *Trends Biotechnol* 2004;22:15–22.
- Chen S, Haralick M. Recursive erosion, dilation opening and closing transform. *IEEE Trans Image Process* 1995;4:335–345.
- Christophe Z, Elisabeth L, Vannary MY, Nancy G, Jean-Christophe OM. Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: A tool for cell-based drug testing. *IEEE Trans Med Imag* 2002;21:1212–1221.
- Zhou XB, Wong STC. Dynamic subcellular behavior study in high content imaging using a multi-threshold approach. Presented at IEEE International Symposium on Circuits and Systems, Kobe, Japan, May 2005.
- Guo R, Pandit SM. Automatic threshold selection based on histogram modes and a discriminate criterion. *Mach Vis Appl* 1998;10: 331–338.
- Lin KC. Fast image thresholding by finding the zero(s) of the first derivative of between-class variance. *Mach Vis Appl* 2003;13:254–262.
- Otsu N. A threshold selection method from gray level histogram. *IEEE Trans Syst Man Cybern* 1978;8:62–66.
- Zhou X, Wong S. Dynamic sub-cellular behavior study in high content imaging using a novel approach. Presented at IEEE ISCAS 2005, Kobe, Japan, May 2005.
- Bleau A, Leon JL. Watershed-based segmentation and region merging. *Comput Vis Imag Understand* 2000;77:317–370.
- Viola P, Jones MJ. Rapid object detection using a boosted cascade of simple features. 2001; Proc Conf Computer Vision and Pattern Recognition. Kauai, HI, USA. 1:511–518.
- Lienhart R, Maydt J. An extended set of haar-like features for rapid object detection. Presented at the Proceedings of the IEEE Conference on Imaging Processing (ICIP'02), New York, 2002.
- Kuranov A, Lienhart R, Pisarevsky V. An empirical analysis of boosting algorithms for rapid objects with an extended set of Haar-like features. Intel Technical Report MRL-TR-July 02-01, 2002.
- Freund Y, Schapire RE. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, Morgan Kaufman, San Francisco, 1996. pp 148–156.
- Haralick RM, Shanmugam K, Dinstein I. Textural features for image classification. *IEEE Trans Syst Man Cybern* 1973;SMC-3:610–621.
- Walker RF, Jackway P, Longstaff ID. Improving co-occurrence matrix feature discrimination. Presented at the Proceedings of DICTA '95, Third Conference on Digital Image Computing: Techniques and Applications, Brisbane, Australia, 1995.
- Zhou X, Wang X, Dougherty ER, Russ D, Suh E. Gene clustering based on cluster-wide mutual information. *J Comput Biol* 2004;11: 151–165.