A Gaussian Mixture Model for Automated Vesicle Fusion Detection and Classification

Haohan Li\textsuperscript{1}, Zhaozheng Yin\textsuperscript{1}, and Yingke Xu\textsuperscript{2}

\textsuperscript{1}Department of Computer Science, Missouri University of Science and Technology
\textsuperscript{2}Department of Biomedical Engineering, MOE Key Laboratory of Biomedical Engineering, Zhejiang Provincial Key Laboratory of Cardio-Cerebral Vascular Detection Technology and Medicinal Effectiveness Appraisal, Zhejiang University

hl187c@mst.edu, yinz@mst.edu\textsuperscript{***}, yingkexu@zju.edu.cn

Abstract. Accurately detecting and classifying vesicle-plasma membrane fusion events in fluorescence microscopy, is of primary interest for studying biological activities in a close proximity to the plasma membrane. In this paper, we present a novel Gaussian mixture model for automated identification of glucose transporter 4 (GLUT4) vesicle and plasma membrane fusions and classification of full fusion and partial fusion events in Total Internal Reflection Fluorescence microscopy (TIRFM) image sequences. Image patches of fusion event candidates are detected in individual images and linked over consecutive frames. A Gaussian mixture model is fit on each image patch of the patch sequence with outliers rejected for robust Gaussian fitting. The estimated parameters of Gaussian functions over time are catenated into feature vectors for classifier training. Applied on three challenging datasets, our method achieved competitive results on detecting and classifying fusion events compared with two other state-of-the-art methods.

1 Introduction

Vesicle exocytosis is an essential cellular trafficking process, by which materials (e.g., transporters, receptors and enzymes) are transported from one membrane-bounded organelle to another or to the plasma membrane for growth and secretion. Vesicle exocytosis needs to be highly regulated as the dysregulation of it is related to many human diseases (e.g., neurodegenerative disease, cancer and diabetes, etc. \cite{1}). Total Internal Reflection Fluorescence (TIRF) microscopy is a powerful tool for visualization and analysis of vesicle exocytosis, which selectively records the dynamics of vesicle traffic near the bottom of plasma membrane, with superb spatial resolution. A pH-sensitive mutant of GFP, pHluorin,

\textsuperscript{***} This research was supported by NSF CAREER award IIS-1351049, NSF EPSCoR grant IIA-1355406, ISC and CBSE centers at Missouri S&T.

\textsuperscript{†} This work was supported by the National Basic Research Program of China (2015CB352003), National Natural Science Foundation of China (31301176), Zhejiang Provincial Natural Science Foundation of China (LY13C050001) and SRFDP grant (2013010120172) to Y. X.
is developed and has been widely used in visualization of single vesicle exocytosis [10]. pHluorin is expressed and quenched in the lumen of the vesicle. When a vesicle is exposed to extracellular neutral environment as the vesicle fuses with the plasma membrane, it becomes brightly fluorescent. In this study, we sought to perform automated detection and classification of VAMP2-pHluorin labeled glucose transporter 4 (GLUT4) vesicle exocytosis in 3T3-L1 adipocytes. GLUT4 plays a pivotal role in glucose uptake in human, whose dysfunction results in insulin resistance and diabetes. Usually, a vesicle fusion process includes three stages of vesicle movement, which is shown in Fig.1. In stage 1, vesicles move towards the cell membrane. In the stage 2 and stage 3, upon the stimulation, some vesicles fuse on the cell membrane with a visible explosion phenomenon (called puff), which are considered as a significant feature of full fusion events. Other vesicles transiently fuse with the cell membrane but without fully collapse with the plasma membrane, which are considered as partial fusion events.

**Fig. 1.** The movement process of vesicles and their related fusion events. 3T3-L1 adipocytes were transfected with VAMP2-pHluorin to label the GLUT4 vesicles. pHluorin is a pH-sensitive fluorescent protein that is invisible in the lumen of acidic vesicles, which becomes much more fluorescent when vesicle fuses with the plasma membrane and exposes to a neutral environment. After vesicle fusion, a vesicle either fully collapses and fuses with the plasma membrane (full fusion), or partially fuses with the plasma membrane and then is retrieved rapidly by the clathrin-dependent process (partial fusion).

1.1 Related Work

A typical TIRFM image sequence might consist of hundreds of frames with hundreds of vesicles. Manual analysis is very time-consuming, even for experienced biologists. When computer-based microscopy image analysis is used to relieve human from the tedious manual labeling [7, 8], it is unsurprising that the uncontrollable noise interference of TIRFM images and the high variability of fusion events’ size, duration and maximum intensity hinder the automated image processing. Furthermore, although fusion events could be visualized as bright spots in TIRFM images, they can not be detected by simple thresholding. Some of the
bright spots in TIRFM image sequences are endocytic vesicles or vesicles from other non-acidic compartments. Occasionally, they are moving in and out of the TIRF field. In order to detect bona fide fusion events, one needs to use specific detection algorithms considering both spatial features and temporal features.

In the past a few years, a few approaches have been proposed to perform automated fusion identification. An automated method to identify full fusion events was proposed in [2], which uses a local maximal detector to search fusion event patches, then connects detected patches in the same position as patch sequences. If the total patch intensity increases within 1 second, while the patch's peak intensity decreases during the same time window, then this patch sequence is considered as a full fusion event. However, local maximal detector is sensitive to intensity variation caused by background fluctuation inside the cell, which may generate many false positive detections (Fig.2(a)). The fusion event patch detection may be affected by moving objects too (Fig.2(b)). Lorenz et al.[5] proposed a method to detect pixels with local maximal/minimal intensity in each frame. For each detected pixel, a diffusion model is developed to analyze the quantity of local maximal pixels and the local minimal pixels happened at this pixel’s position, in a time window. The diffusion model method effectively distinguished full fusion events from non fusion regions, leaving a large amount of partial fusion events unrecognized. A Gaussian model was used to fit fusion events in [3]. The standard deviation in the Gaussian model is used to classify fusion events, assuming full fusion events have greater standard deviation than the partial fusion events. However, fusion events generally have large variations on their blob sizes spatially and temporally (Fig.2(c)).

![Fig. 2. Examples of major challenges of fusion event detection. (a) Local maximal detector can obtain lots of false detections. (b) Moving objects are very similar to fusion events, which may cause false detections. (c) Fusion event detection can not be solved by a simple thresholding method due to the variability of their sizes and intensity values.](image_url)

### 1.2 Our Proposal

Although TIRFM images are very noisy, without removing the noise using computational methods, an adaptive detection method based on local contrast is proposed first for searching potential fusion event patches in each frame in our work. Then, a tracking method is developed to link detected patches over consecutive frames into patch sequences as the fusion event candidates. Thirdly,
center-surrounded multi-Gaussian functions are fit on each patch of the patch sequence with an Inlier Selection algorithm to remove outliers during the Gaussian fitting. Finally, the patch sequences are aligned with the same time length and a feature vector is extracted from a series of Gaussian functions fit on the patch sequence, based on which a Support Vector Machine classifier is trained to classify the patch sequence into one of three classes: full fusion, partial fusion or non-fusion event.

2 Detection of Fusion Event Candidates

During the 3-stage vesicle movement in Fig.1, vesicles intend to immobilize at a position near the cell membrane for a few seconds before they either fuse or undock. This visible phenomenon is perceived as a process of intensity changes which suddenly appears in the image and then disappears either gradually (full fusion) or all in a sudden (partial fusion). In this process, the high local contrast is a significant feature to recognize fusion events, which is visible in the stage 2 or 3. According to these observations, we propose a local contrast detector to detect potential fusion event image patches frame-by-frame, and then the detected patches are tracked/linked over consecutive frames into patch sequences as the fusion event candidates.

2.1 Detect Potential Fusion Event Image Patches

During the stage 2 and 3 of a vesicle fusion event, the intensities of pixels around the vesicle location increase. Thus, finding local intensity maximums has been used to detect image patches as the candidates of fusion events [2, 3]. However, due to the illumination variation and intensity fluctuation of background pixels, the local maximal method may find many false positives (Fig.2(a)). Instead, we use a local contrast detector to detect image patches of fusion event candidates. If a pixel’s intensity value, \( I(p) \), is more than \( \gamma \) times of the maximal intensity value among the pixels which are away from pixel \( p \) by a radial distance \( r \), then pixel \( p \) is detected as a high contrast pixel. After detecting all pixels with high contrast, those pixels which are close to each other are clustered together into a group and an image patch around the group centroid is extracted as a fusion event candidate in the current image \( I \).
2.2 Link Potential Fusion Event Image Patches

![Matrix V](image)

Fig. 4. The way to store detected patches into matrix $V$.

![Diagrams](image)

Fig. 5. The method to link image patches into fusion event candidates.

After the frame-by-frame detection, all detected potential fusion event patches are stored in a matrix $V$ (shown in Fig. 4) with the size of $P \times 3$ where $P$ is the number of detected patches in the image sequence and each row of $V$ stores the spatial location and timestamp of a patch. $V$ stores the patches of the 1st frame, then the 2nd frame and so on. We propose a five-step approach to link the patches in $V$ into patch sequences over time, as illustrated in Fig. 5. The 1st step starts from picking the first patch in $V$ as the initial vesicle and the frame in which it appears is the initial frame. In the 2nd step, we determine the spatial searching range as the 5-by-5 neighborhood centered at the location of initial vesicle, and the temporal searching range as the next $N$ frames of the initial frame. Then we search for other detected patches in $V$, which both appear in the spatial searching range and the temporal searching range. In the 3rd step, we connect patches from the initial frame to the last frame in the $N$ frames which contains a potential patch. Although the fusion event appears consecutively on TIRFM image sequence in stage 2 of the vesicle’s three-stage movement shown in Fig. 1, there are still some missing patches inside the connected patch sequences, caused by noise or background intensity fluctuation. So in the 3rd
step, the connected patch sequence will be supplemented by extracting patches at the location of initial vesicle in those frames with missing patches. In the 4th step, the last connected patch will be selected as the new initial vesicle, then the 2nd step and 3rd step are repeated iteratively until there is no detected patches in the next \( N \) frames. In the 5th step, we search the missing patches from the previous \( H \) frames and the next \( T \) frames of the fusion event candidate using a lower threshold \( \gamma \) in the local contrast detector.\(^1\) Thus, a complete set of image patches are linked to form a fusion event candidate (Fig.3(d)). All image patches in this fusion event candidate will be deleted from \( V \), and the five steps are repeated until \( V \) is empty. Therefore, we obtain all image patch sequences which could be considered as fusion event candidates.

Before we classify fusion event candidates, we need to exclude moving objects from them. Moving objects are very similar to fusion events, which intend to suddenly appear as bright spots and then halt at the same location for a while. But the difference between fusion events and moving objects is that the latter will finally move away from the initial location. Our strategy is to record the spatial location of each detected patch during the five-step approach. Then a fusion event candidate will be considered as a moving object from the background if any patch of this candidate is located outside the 5-by-5 neighborhood of the first patch.

### 3 Gaussian Mixture Model for Classification

As observed in \([6, 9]\), differences between full fusion and partial fusion are observable in stage 3 of the movement process of vesicles. Basically, when it undergoes full fusion, the radius of a vesicle will become bigger and bigger and its peak intensity becomes lower and lower. While, when it undergoes partial fusion, the radius of a vesicle will not change very much and its peak intensity becomes lower and lower when it leaves the cell membrane. Based on these visible differences, the pixel intensity values in a fusion event have been modeled by a 2D Gaussian model in \([3, 6]\). However, the pixel intensity in a fusion event may be complicated such that a single Gaussian model is not accurate enough to model it, thus a mixture of Gaussian models might be a good solution. Furthermore, fitting a Gaussian model on observed data is very sensitive to outliers, so a robust mechanism is required to avoid outlier pixels with undesired intensity fluctuation.

In this paper, the region-of-interest of each fusion event is defined by a “peak area” and a “flat area” as shown in Fig.6. The “peak area”, denoted as \( \text{Area}_p \), is a 5 \( \times \) 5 neighborhood which is centered by the highest intensity pixel of the

\(^1\) We set the detector threshold \( \gamma = 1.3 \) for the frame-by-frame detection, and \( \gamma \) is lowered to 1.1 for detecting missing patches in the 5th step. Since fusion events tend to have a short period of appearance at the end of stage 1, but a relatively longer period to fade away in stage 3, we set \( H = 5 \) for head frames and \( T = 10 \) for tail frames. The parameters can also be learned by cross-validation.
impulse. The “flat area”, denoted as $\text{Area}_f$, is a $13 \times 13$ neighborhood surrounding $\text{Area}_p$. In the Gaussian mixture model, two center-surround 2D Gaussian models will fit the pixel values in $\text{Area}_p$ and $\text{Area}_f$, respectively. Meanwhile, to avoid the outlier effect, an Inlier Selection (IS) algorithm is adopted to estimate the parameters of Gaussian models robustly.

![Fig. 6. The region-of-interest of a fusion event consists of a “peak area” and a “flat area”. In stage 2 and stage 3, fusion spreads into the “flat area”]

### 3.1 2D Gaussian Model Fitting

We define our 2D Gaussian function as

$$I(x, y) = \lambda \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) + \beta$$  \hspace{1cm} (1)

where $I(x, y)$ is the intensity value at the position $(x, y)$. Note that $\lambda$ is not necessarily to be $\frac{1}{2\pi\sigma^2}$ since we model the pixel intensities by a Gaussian function rather than a Gaussian probability distribution. We further simplify Eq.1 by defining $\alpha = -\frac{1}{2\sigma^2}$, thus

$$I(x, y) = \lambda \exp(\alpha(x^2 + y^2)) + \beta.$$  \hspace{1cm} (2)

$\beta$ is computed as the minimum of pixel intensity values within the neighborhood. For example, when fitting the Gaussian function to the peak or flat area,

$$\beta = \min_{m \in \text{Area}_{p,f}} \{I_m\}.$$  \hspace{1cm} (3)

Suppose $M$ pixels are selected from the peak area to fit its Gaussian function, the following cost function is defined to estimate $\lambda$ and $\alpha$ for the peak area:

$$L(\lambda, \alpha) = \sum_{m=1}^{M} \left[\log(I_m - \beta) - \log(\lambda \exp(\alpha(x_m^2 + y_m^2)))\right]^2$$  \hspace{1cm} (4)

Taking the partial derivatives regarding to $\lambda$, $\alpha$ and setting them to zero lead to

$$\alpha = \frac{M \sum_{m=1}^{M}(x_m^2 + y_m^2)\log(I_m - \beta) - (\sum_{m=1}^{M}(x_m^2 + y_m^2))(\sum_{m=1}^{M}\log(I_m - \beta))}{M \sum_{m=1}^{M}(x_m^2 + y_m^2)^2 - (\sum_{m=1}^{M}(x_m^2 + y_m^2))^2}$$  \hspace{1cm} (5)

$$\lambda = \exp\left(\frac{\sum_{m=1}^{M}\log(I_m - \beta) - \alpha \sum_{m=1}^{M}(x_m^2 + y_m^2)}{M}\right)$$  \hspace{1cm} (6)

Similarly, the 2D Gaussian fitting procedure can be applied to the flat area.
3.2 IS Algorithm to Avoid Outliers During Gaussian Fitting

When fitting Gaussian functions in the peak and flat areas, the outlier pixels from background may bias the parameter estimation largely. Especially, Area_f is a relatively larger area in TIRFM images compared to Area_p, therefore it may contain more pixels which are interfered by the pixel intensity noise. In some cases, the positions of multiple fusion events are very close to each other such that the Area_f of them may overlap. To avoid these interferences, we propose an Inlier Selection (IS) algorithm to estimate the optimal $\lambda$, $\alpha$ and $\beta$ of Eq.2 in Algorithm 1.

Algorithm 1 Gaussian Fitting with Inlier Selection Algorithm.

Require:
- Maximum iterations $K$, Area_p/f, distance threshold $T$, the maximum number of inliers $O$, the number of pixels randomly picked in Area_p/f $M$;

Ensure:
1: Estimate $\beta$ by Eq.3;
2: $O = -1$;
3: while Iteration less or equal to $K$ do
4: $M = 4$;
5: Estimate $\lambda$ and $\alpha$ by Eq.6 and Eq.5, respectively;
6: Compute all pixel intensities in Area_f by Eq.2: $\hat{I}_m$;
7: Count the number of inliners $O'(\text{those pixels with } (I_m - \hat{I}_m)^2 < T)$;
8: if $O' > O$ then
9: $O = O'$; //The current estimated $\lambda$, $\alpha$ and $\beta$ are the optimal parameters;
10: end if
11: end while
12: Output the optimal parameters $\lambda$, $\alpha$ and $\beta$.

3.3 Feature Extraction from Gaussian Models for Classification

Since the fusion event candidates (represented as image patch sequences) may have different time lengths, we need to align the patch sequences and cut/append them with the same time length in order to have comparable feature vectors. The alignment process is illustrated in Fig.7. For each fusion event candidate, the maximum intensity value of each image patch is computed and the maximums of all image patches of this fusion event candidate formulate a time-series signal, whose climax moment ($t^*$) is selected as the time instant to align the fusion event. We extract features from each image patch in the temporal sliding window $[t^* - F_h, t^* + F_t]$, where $F_h = 10$ and $F_t = 20$ in our experiments. For fusion event candidates whose time lengths are shorter than the sliding window, we will zero-padding them to have the sliding window of $[t^* - F_h, t^* + F_t]$, while the fusion event candidates with longer time lengths will be fit into the time length by dropping frames exceeding the temporal sliding window. Note that, in the datasets we used in this study, the VAMP2-pHluorin fusions in 3T3-L1
adipocytes have an average duration of 5 to 15 frames for the fusion process. Thus, for a typical exocytic fusion event, $F_h$ and $F_t$ provide a suitable temporal sliding window.

![Feature vector alignment process.](image)

For each patch, we obtain a $1 \times 6$ feature vector, including the $\lambda$, $\alpha$ and $\beta$ from both peak area and flat area. The feature vectors of all image patches in a fusion event candidate are catenated sequentially, so the length of our feature vector for a fusion event candidate is $6 \times (F_h + F_t + 1)$, based on which we train a Support Vector Machine classifier to classify the fusion event candidate into one of three classes: full fusion, partial fusion, or non-fusion.

4 Result

![Classification examples of our method on dataset1.](image)

Three datasets from previous experimental study were used to validate our method. Each of these datasets consists of 300 time-lapse images. The ground truth was provided by experienced cell biologists working in the field of vesicle trafficking analysis in TIRFM. Then, single fusion events were manually analyzed and bona fide fusion events were selected using the criteria in [6]. The comparison
**Fig. 9.** Classification examples of our method on dataset 2 (yellow: partial fusion; red: full fusion; dashed line: groundtruth; solid line: our result).

**Fig. 10.** Classification examples of our method on dataset 3 (yellow: partial fusion; red: full fusion; dashed line: groundtruth; solid line: our result).
between our method (Gaussian Mixture Model plus Inlier Selection algorithm, denoted as GMM+IS) and alternative method (Gaussian Mixture Model without Inlier Selection, denoted as GMM w/o IS), was tested in Dataset 1, with the results shown in Table 1. By avoiding outliers during the Gaussian fitting, our method effectively improved the classification accuracy on both full and partial fusion events.

<table>
<thead>
<tr>
<th>Dataset 1</th>
<th>Full Fusion</th>
<th>Partial Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM+IS</td>
<td>Precision: 89.2%</td>
<td>Recall: 91.7%</td>
</tr>
<tr>
<td>GMM w/o IS</td>
<td>Precision: 85.3%</td>
<td>Recall: 80.6%</td>
</tr>
<tr>
<td>SGM[3]</td>
<td>Precision: 79.5%</td>
<td>Recall: 86.1%</td>
</tr>
<tr>
<td>IV[2]</td>
<td>Precision: 64.4%</td>
<td>Recall: 80.6%</td>
</tr>
</tbody>
</table>

Table 1. The comparison of four methods on dataset 1. GMM+IS: Gaussian Mixture Model + Inlier Selection; GMM w/o IS: Gaussian Mixture Model without Inlier Selection; SGM[3]: Single Gaussian Model; IV[2]: Intensity-Variance-Based Method.

We also compare our method with the Single Gaussian model method [3] (denoted as SGM) and Intensity-Variance-Based Method [2] (denoted as IV) in all three datasets, with the results shown in Tables (1-3), respectively. All parameters in algorithms of [2] and [3] are optimized to ensure they can achieve the best results in the three datasets. Compared to the Single Gaussian Model method [3], our method achieves better classification results for both full fusion events and partial fusion events in three test datasets, which validates that the Gaussian Mixture Model with Inlier Selection has a better compatibility to fusion events than the Single Gaussian Model. The algorithm in [2] was developed for full fusion event identification based on intensity thresholds on multiple low level features, e.g., maximum intensity variance and total intensity variance. However, the algorithm in [2] is unable to detect the partial fusion event. Compared to the algorithm in [2], our proposed method has a better full fusion classification result, which proves that the feature we extract from the proposed Gaussian Mixture Model is more effective than low level intensity features. Note that dataset 3 has very low Signal-Noise-Ratio and the frequent background fluctuation generates a strong interference on the performance of all methods. The short fusion duration
in dataset 3, which is as short as 3 frames, makes the feature extraction difficult.

<table>
<thead>
<tr>
<th>Dataset 3</th>
<th>Full Fusion</th>
<th>Partial Fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>GMM+IS</td>
<td>68.2%</td>
<td>71.4%</td>
</tr>
<tr>
<td>SGM[3]</td>
<td>60.9%</td>
<td>66.7%</td>
</tr>
<tr>
<td>IV[2]</td>
<td>53.8%</td>
<td>66.7%</td>
</tr>
</tbody>
</table>

Table 3. The comparison of four methods on dataset 3. GMM+IS: Gaussian Mixture Model + Inlier Selection; SGM[3]: Single Gaussian Model; IV[2]: Intensity-Variance-Based Method.

5 Conclusion

Accurately detecting and classifying vesicle-plasma membrane fusion events from TIRFM images is an important research problem on cellular trafficking processes. We proposed an adaptive detection method based on local contrast to detect image patches of fusion event candidates in individual frames and developed a tracking method to link image patches as candidate patch sequences. A center-surround Gaussian Mixture Model was proposed to fit the image patch intensity with outliers rejected for robust Gaussian fitting. A feature vector is extracted from parameters of the series of Gaussian functions fit on the aligned patch sequence, based on which a SVM classifier is trained. Our method showed better performance and outperformed two other state-of-the-art methods when tested on three real biomedical datasets with variable signal to noise ratio and background fluctuation.

References