A Diffusion Model for Detecting and Classifying Vesicle Fusion and Undocking Events

Lorenz Berger[1], Majid Mirmehdi[2], Sam Reed[3], and Jeremy Tavaré[3]

[1] Department of Computer Science, University of Oxford, UK
[2] Department of Computer Science, University of Bristol, UK
[3] Department of Biochemistry, University of Bristol, UK

Abstract. Fluorescently-tagged proteins located on vesicles can fuse with the surface membrane (visualised as a 'puff') or undock and return back into the bulk of the cell. Detection and quantitative measurement of these events from time-lapse videos has proven difficult. We propose a novel approach to detect fusion and undocking events by first searching for docked vesicles that 'disappear' from the field of view, and then using a diffusion model to classify them as either fusion or undocking events. We can also use the same searching method to identify docking events. We present comparative results against existing algorithms.

1 Introduction

The movement of numerous proteins between the various sub-compartments of a cell is critical in the biological function of a cell. Defects in protein movement can lead to disease, e.g. ineffective movement of a protein called GLUT4 from small intracellular vesicles towards the surface membrane of a fat and muscle cell, and their consequent fusion with that membrane, leads to insulin resistance and type 2 diabetes [1]. Understanding these processes at a molecular level is, therefore, critical to understanding cellular behavior in normal and diseased states.

The docking and fusion of intracellular vesicles with the surface membrane of cells can be visualised using Total Internal Reflection Fluorescence Microscopy (TIRFM). A snapshot of the distribution of vesicles labeled with a fluorescently-tagged GLUT4 in a single fat cell is shown in Fig. 1(left) while Fig. 1(right) shows some of the key events in GLUT4 movement to the surface membrane. These key events in vesicle movement to the surface membrane can be described as follows. First insulin, which is required for the vesicles to fuse with the membrane, signals for the vesicles to move towards the surface membrane. Then some of the vesicles make it to the edge of the cell and 'dock' with the cell membrane. This docking event corresponds to vesicles suddenly halting and vibrating in the same place for a few seconds. After docking for some time, some of the vesicles then fuse at the cell membrane. This can be seen as a 'puff' (see row A in Fig. 2). Other vesicles dock for a few seconds and then undock and leave the vicinity of the membrane, returning back into the main bulk cell (i.e. go out of view; row B in Fig. 2) or move off to a different part of the membrane.

Extracting information and quantifying vesicle dynamics in TIRFM videos has proven difficult and is a major barrier to understanding their molecular basis. It is impractical to manually mark the video data as it would be far too time-consuming and

N. Ayache et al. (Eds.): MICCAI 2012, Part III, LNCS 7512, pp. 329–336, 2012.
© Springer-Verlag Berlin Heidelberg 2012

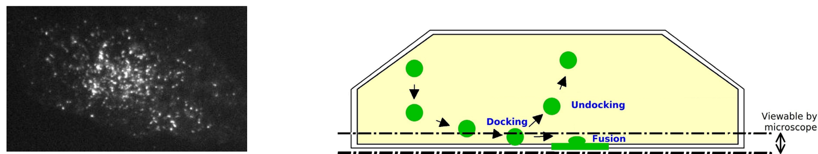

Fig. 1. (left) A fat cell's vesicles (bright dots) tagged with a green fluorescent protein and imaged using TIRFM. (right) Key events of insulin-stimulated GLUT4 translocation (see text for details).

error-prone, making an automatic approach necessary. Developing a robust quantitative vesicle analysis method will be important in many areas of cell biology where vesicle fusion and undocking with the surface membrane occurs, e.g. events such as neurosecretory vesicle fusion [2], which is relevant to Alzheimer's disease and schizophrenia. Current methods that are popularly and extensively used for biological research first segment individual vesicles by analyzing the surrounding gray level distribution, and then use thresholds on the pixel intensity to distinguish between fusion and undocking events, e.g. Bai et al.[3] and Huang et al. [4]. In Vallotton et al. [5], a matched filtering approach is used to identify events that highly correlate against a standard fusion event through space and time. However, this approach is not only computationally expensive, but tends to miss many events due to the high variability in duration, size and noise levels of fusion events. In Mele et al. [6], videos are represented as a 3D image space, where using a threshold for noise, patches of interest are differentiated from noise according to absolute difference in pixel intensity between frames. These patches of interest are then further analysed using a 3D extension of the Maximally Stable Extremal Regions algorithm to detect high intensity and highly variable regions which would correspond to potential fusion events. A set of descriptors including intensity differences, fusion spot size, and degree of fit to a diffusion model are collected for each candidate event, and are then compared against pre-identified fusion events using PCA. *This method however does not consider undocking events.*

2 Proposed Method

We first outline a search algorithm for detecting vesicles that suddenly disappear due to fusion or undocking - we shall call these candidate events. By modifying this algorithm we are also able to detect vesicles that suddenly appear, and to the best of our knowledge this is the first approach that has been proposed to detect docking events. To classify the candidate events, we build on an idea taken from [6] which is to use a mathematical diffusion model to extract features for classification. In [6] an analytical solution to the diffusion equation is derived to explain how the intensity of a vesicle should change over time during fusion (recall as mentioned earlier, [6] cannot handle undocking, and docking, events). We develop this idea by considering a 2D diffusion model which considers the whole vesicle and its surrounding area, and we also introduce a source and sink term to model both fusion and undocking events explicitly. We then fit these fusion and undocking models to the candidate event and use the goodness of the fit

to classify the event as a fusion or an undocking event. We test our algorithm on real TIRFM data and demonstrate its performance for a range of videos.

Detecting Fusion and Undocking Events - Sophisticated algorithms to track the behaviour of vesicles already exist, e.g. [7]. Here, the vesicles of interest are only those that have docked to the membrane and then go on to either fuse or undock, and thus, there is no need to track individual vesicles - this same phenomenon was also looked at in [8] using a patch-based method. Our main assumption is that a vesicle remains stationary for around N frames before it fuses or undocks. Hence, we search for vesicles that have been visible in the same position throughout the past N_p frames but then cannot be found again in this position during N_f future frames. This corresponds to the vesicle first being docked (visible) and then having either undocked or fused (not visible). We also allow for brief periods of vesicle disappearance. This allows us to detect vesicles that shortly go out of view due to noise and TIRFM artifacts but still follow the overall pattern of a disappearing vesicle. As preprocessing, we reduced noise in our videos using a standard Gaussian filter ($\sigma = 1.5$), as in [6]. For further efficiency, we also applied a low threshold to eliminate areas with very low intensity and no activity. We use local maxima to identify the individual vesicles in each frame which gives better results than using an adaptive thresholding approach as performed in [7].

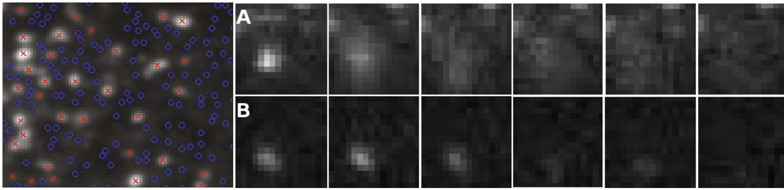

Fig. 2. (Left) Local maxima (red crosses) and local minima (blue circles). (Right) An image sequence showing a very prominent fusion event (row A), and an undocking event (row B).

Let $\mathbf{V}$ denote a 2D matrix that contains the positions of all local pixel maxima (see Fig. 2) in a 5×5 neighbourhood for every frame in the video, such that v_{ij} is the position of the jth maxima (vesicle) in the ith frame, and $\mathbf{v}_{k:} = [v_{k1}, v_{k2}, ..., v_{kj}, ...]$ denote the set of all the local maxima found in the kth frame. Similarly let $\mathbf{W}$ denote a 2D matrix that contains the positions of all the pixels that have a local minima (see Fig. 2) in a 3×3 neighbourhood in each frame such that w_{ij} is the position of the jth minimum in the ith frame. We can then calculate a score a_{ij} for each vesicle v_{ij} which reflects how closely v_{ij} follows the pattern of being stationary and visible for the past N_p frames and then being out of view for the future N_f frames:

$$a_{ij} = \sum_{k=i-N_p}^{k=i-1} h(\mathbf{v}_{k:}, v_{ij}) + \sum_{k=i+1}^{k=i+N_f} g(\mathbf{w}_{k:}, \mathbf{v}_{k:}, v_{ij}). \tag{1}$$

Here $h(\mathbf{v}_{k:}, v_{ij})$ is a function which searches $\mathbf{v}_{k:}$ for a local maxima that is positioned within r pixels of v_{ij}. More specifically,

$$h(\mathbf{v}_{k:}, v_{ij}) = \begin{cases} 1 & \text{if } \exists\, v \in \mathbf{v}_{k:} \text{ s.t } |v - v_{ij}| \leq r\,, \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

where r is the radius of the vesicle. The function $g(\mathbf{w}_{k:}, \mathbf{v}_{k:}, v_{ij})$ searches $\mathbf{w}_{k:}$ for a local minima that is positioned within r pixels of v_{ij} and also checks that there are no local maxima $v \in \mathbf{v}_{k:}$ which are close to v_{ij}, i.e.

$$g(\mathbf{w}_{k:}, \mathbf{v}_{k:}, v_{ij}) = \begin{cases} 1 & \text{if } \exists\, w \in \mathbf{w}_{k:} \text{ s.t } |w - v_{ij}| \leq r, \\ & \quad \wedge \forall\, v \in \mathbf{v}_{k:} \text{ s.t } |v - v_{ij}| > r, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

To decide whether a vesicle should qualify as having gone missing we set a threshold such that if $a_{ij} \geq (N_p + N_f)C$, then a missing vesicle is detected at v_{ij}, where $C \in [0, 1]$ is a threshold representing the amount of 'brief disappearance' that is allowed.

Detecting Docking Events - To detect vesicles that are first not visible but then suddenly appear and stay stationary we can simply rearrange Eq. (1) such that

$$b_{ij} = \sum_{k=i-N_p}^{k=i-1} g(\mathbf{w}_{k:}, \mathbf{v}_{k:}, v_{ij}) + \sum_{k=i+1}^{k=i+N_f} h(\mathbf{v}_{k:}, v_{ij}). \tag{4}$$

To decide whether a vesicle should qualify as having docked we set a threshold such that if $b_{ij} \geq (N_p + N_f)C$, then a docking vesicle is detected at v_{ij}.

3 A Computational Model for Fusion and Undocking Events

TIRFM helps create an evanescent field which illuminates and excites fluorophores in a region ≈100nm below the interface. The evanescent field's intensity decreases exponentially with the distance perpendicular to the interface which directly relates to an exponential decrease in the fluorescence [9]. For a vesicle that goes from being docked at the membrane to fusing and diffusing into the membrane, the total number of fluorophores does not change, however, as the fluorophores diffuse into the membrane they are now collectively closer to the interface which can result in a slight total intensity increase in the video. In the fusion model, this is represented using a diffusive and a source term centred at the vesicle. During an undocking event, there is no diffusion since the vesicle just undocks and returns into the cell. This is modelled using just a sink term centred at the vesicle which is able to explain the sudden decrease in intensity.

Definition of the Models - For each of the previously detected candidate events v_{ij} a sequence of subregions centred at the event spatially and temporally is taken from the video. A sequence length of $N_p/2 + N_f/2$ guarantees that the actual event happens during these frames. The size of the subregion should include the whole vesicle and

some room for a fusion (puff) to happen - we take this to be $4r \times 4r$. Let $\mathbf{I}_k(x, y)$ be the pixel intensity at position (x, y) within the subregion, during the kth frame, with $x \in [-2r, 2r]$, $y \in [-2r, 2r]$ and $k \in [i - N_p/2, i + N_f/2]$. Let $\mathbf{M}_k$ be the circular mask of radius r, centred at the candidate vesicle position $v_{ij} = (v_x, v_y)$ such that

$$\mathbf{M}_k(x, y) = \begin{cases} \mathbf{I}_k(x, y) & \text{if } (x - v_x)^2 + (y - v_y)^2 \leq r^2 \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

This will be the mask on which the source or sink will be able to act. We can then introduce the model for a fusion event as

$$\frac{\partial \mathbf{I}}{\partial t} = D_F \left(\frac{\partial^2 \mathbf{I}}{\partial x^2} + \frac{\partial^2 \mathbf{I}}{\partial y^2} \right) + S_F \mathbf{M}_k, \tag{6}$$

where $S_F \in [0, \infty)$ is the magnitude of the source and $D_F \in [0, \infty)$ is the diffusion coefficient, which model the amount of increase and diffusion of fluorescence respectively. The model for an undocking event is given by

$$\frac{\partial \mathbf{I}}{\partial t} = S_U \mathbf{M}_k, \tag{7}$$

where $S_U \in (-\infty, 0]$ is the magnitude of the sink. As there is no puff, i.e. no diffusion in an undocking event, then $D_U = 0$, causing the diffusion term to disappear.

Using the Models for Classification - We can use Eqs. (6) and (7) to explain how a candidate event evolves temporally and whether it can be classified as a fusion event, an undocking event, or neither. We pick a frame as initial conditions to (6) and (7), evolving the system by one time step and then comparing the predicted result with the actual next frame. The total intensity difference between the predicted frame and the actual frame of each model is then turned into a likelihood ratio which is used for classification. Let us define a function which evolves the fusion model in (6) by one time step and solves for the new intensity distribution $\hat{\mathbf{I}}^F_{k+1}$ of the subregion

$$\hat{\mathbf{I}}^F_{k+1} = \mathbf{F}(\mathbf{I}_k, D_F, S_F). \tag{8}$$

The fusion model in (6) is solved numerically using the Crank-Nicolson method, with homogeneous Neumann boundary conditions and initial conditions $\mathbf{I}_k$. We then optimize (8) to find the optimal D_F and S_F values which let the fusion model best predict the next frame. This optimization step is implemented using the well known Nelder-Mead method. The absolute difference ξ_{Fk} over the whole subregion between the predicted frame $\hat{\mathbf{I}}^F_{k+1} = \mathbf{F}(\mathbf{I}_k, D_F, S_F)$ and the actual frame $\mathbf{I}_{k+1}$ is then given by $\xi_{Fk} = \min_{D_F, S_F} \int_\Omega |\hat{\mathbf{I}}^F_{k+1} - \mathbf{I}_{k+1}| \, d\Omega$, where Ω is the area of the subregion and $k = [i - N_p/2, ..., i + N_f/2]$. A similar calculation can be done for the undocking model using $\xi_{Uk} = \min_{S_U} \int_\Omega |\hat{\mathbf{I}}^U_{k+1} - \mathbf{I}_{k+1}| \, d\Omega$, where $\hat{\mathbf{I}}^U_{k+1} = \mathbf{U}(\mathbf{I}_k, S_U)$ is the function that evolves the undocking model (7) by one time step and solves for a new intensity distribution of the subregion. We can then define a likelihood ratio $\lambda_k = \frac{\xi_{Uk}}{\xi_{Fk}}$ to see which model is better at predicting the next frame. When $\lambda_k > 1$, then the fusion model is better at predicting the next frame $\mathbf{I}_{k+1}$,

and if $\lambda_k < 1$, then the undocking model is better. To finally decide whether a fusion event has happened, we use a simple threshold on $\boldsymbol{\lambda}$ to determine the class, i.e. *Fusion* if $(\max(\boldsymbol{\lambda}) \geq \alpha)$, *Undocking* if $(\min(\boldsymbol{\lambda}) < \gamma \wedge \max(\boldsymbol{\lambda}) < \alpha)$, or *Noise* otherwise. The thresholds can be determined by inspecting the results of a few known events.

4 Results

All the results presented here are evaluated against groundtruth generated by a cell biology expert working in the field of TIRF microscopy and vesicle trafficking. Our data set comprises of four videos[1]. The first part of the proposed method, which searches for 'disappearing' vesicles, achieves an average detection rate of 87.5%. This might not seem as high as expected but is due to the inherent difficulty of the problem. The fusions can be extremely varied in their nature and are often, even for biology experts, difficult to spot and classify. The low spatial and temporal resolution and low SNR in the videos, due to microscopy limitations, also add to the difficulty of the problem. Table 1 shows the number of fusion and undocking events in the groundtruth for each video, the total number of candidate events detected by the proposed method, and the number of candidate events detected by the proposed method corresponding to true events in the groundtruth. In these experiments, we set $N_p = N_f = 20$ and $C = 0.6$. Other advantages of the proposed method are that it is easy to implement, it is fast, and it does not rely on any intensity thresholds, just minima and maxima. This part of the proposed method, i.e. the detection of candidate events alone can be very useful for biologists who analyze such videos. What previously took a full day's work to analyze one video manually can now be achieved in just a few minutes by finding all candidate events in a video (about 0.25 fps) and then manually classifying the events as fusion or undocking events. To automatically classify the events, a threshold on the likelihood ratio produced by the two models is used. The large peak in Fig. 3(left) demonstrates the algorithm's ability to produce a clear signal even for cases where a fusion looks fairly similar to an undocking event. This large peak can then easily be classified as a fusion event.

Table 1. Precision and Recall results for detecting candidate events

	Fusion events groundtruth	**Undocking events groundtruth**	**Candidate events detected**	**Correct candidate events**	**Recall**	**Precision**	**Time taken seconds (s)**
Movie1	16	7	24	19	82.6%	79.2%	42 s
Movie2	10	3	13	13	100%	100%	30 s
Movie3	15	12	35	22	81.5%	62.9%	43 s
Movie4	9	12	23	18	85.7%	78.3%	31 s
Average					87.5%	80.1%	

In Fig. 3(right), an example of an undocking event is shown, where the large trough in the likelihood ratio can also be easily classified as an undocking event. The vesicle in

[1] Note in other works, e.g. [3] and [4], only one long movie was tested.

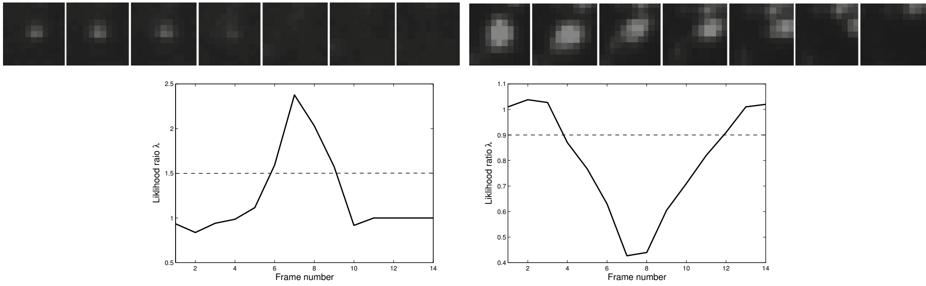

Fig. 3. (Top left) frames 4-11 taken from a fusion event in Movie3.avi, (bottom left) the likelihood ratio between the fusion model and the undocking model for different frames during the fusion sequence, and a threshold of $\alpha = 1.5$. (Top right) an undocking event and its likelihood ratio (bottom right) with a threshold of $\gamma = 0.9$.

the image sequence is stationary and then moves away (undocks) instead of returning straight back into the cell which is more common for an undocking event (see Fig. 2(B)). This is where other methods, such as [3] that use intensity thresholds on features including the maximum intensity increase in the annular area of the vesicle, would fail and incorrectly classify the event as fusion due to the increase in intensity in the annulus, which of course is caused by the vesicle movement, and not by a fusion.

Table 2. Results for the automatic classification of detected events

Fusion	**Accuracy**	**Recall**	**Precision**	**False +ve**	**Undocking**	**Accuracy**	**Recall**	**Precision**	**False +ve**
Movie1	91.7%	100.0%	85.7%	16.7%	Movie1	91.7%	85.7%	85.7%	5.9%
Movie2	76.9%	80.0%	88.9%	33.3%	Movie2	76.9%	33.3%	50.0%	10.0%
Movie3	80.0%	80.0%	61.5%	20.0%	Movie3	54.3%	60.0%	37.5%	43.5%
Movie4	91.3%	83.3%	83.3%	5.9%	Movie4	78.3%	75.0%	81.8%	18.2%
Average	85.0%	85.8%	79.9%	19.0%		75.3%	63.5%	63.8%	19.4%

Table 2 shows the results for the automatic classification of events detected during the searching stage with $\alpha = 1.5$ and $\gamma = 0.9$. It also shows that the diffusion model is able to correctly classify the majority of events and is consistent in detecting fusion events across videos, which are of prime interest to researchers. The complete analysis for a typical set of 200 frames of size 160x160 on a standard 2GHz processor took around 1-2 minutes when implemented in MATLAB. Results for detecting docking events have not been presented here because obtaining the groundtruth for them is extremely cumbersome. Fig. 4 shows a comparison of our method against Bai et al. [3] and Huang et al. [4] using ROC plots. The parameters in algorithms [3] and [4] were optimised for our videos to give their best possible results. Our proposed method outperforms [3] and [4], since it avoids the use of pixel intensity thresholds for classification. This makes it more robust to different videos with different quality and image properties as well as busy regions where closely neighbouring vesicles can interfere with events that are being analyzed. Fig. 4 (right) also shows a basic sensitivity analysis on the classification threshold α which has been performed over all videos. This analysis only looks at the

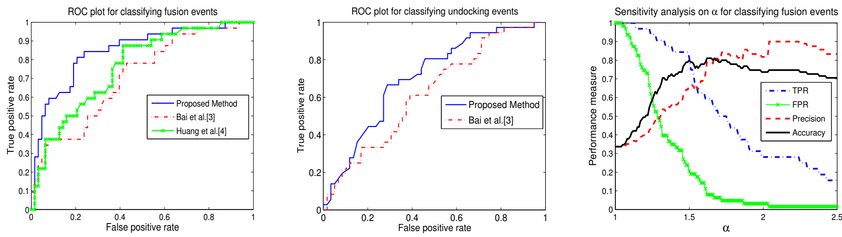

Fig. 4. ROC plots for comparative evaluation of fusion (left) and undocking (middle) events, and a sensitivity analysis (right) of α on the classification of fusion events

classification of fusion events which are of main interest. Choosing α between 1.3 and 1.8 seems to optimize most of the performance measures.

5 Conclusion

Quantitatively measuring the rate of fusions and undocking vesicles is a common problem in cell biology and crucial for making progress in researching the biological function of cells. We proposed a simple, fast and easy to implement search algorithm to find disappearing vesicles. This searching algorithm can also be reformulated to detect docking events. To automatically classify the disappearing events, we proposed a novel computational diffusion model for both fusion and undocking events.

References

1. Leney, S., Tavaré, J.: The molecular basis of insulin-stimulated glucose uptake: signalling, trafficking and potential drug targets. Journal of Endocrinology 203(1), 1–18 (2009)
2. Calì, C., Marchaland, J., Regazzi, R., Bezzi, P.: SDF 1-alpha (CXCL12) triggers glutamate exocytosis from astrocytes on a millisecond time scale: imaging analysis at the single-vesicle level with TIRF microscopy. Journal of Neuroimmunology 198(1-2), 82–91 (2008)
3. Bai, L., Wang, Y., Fan, J., Chen, Y., Ji, W., Qu, A., Xu, P., James, D., Xu, T.: Dissecting multiple steps of GLUT4 trafficking and identifying the sites of insulin action. Cell Metabolism 5(1), 47–57 (2007)
4. Huang, S., Lifshitz, L., Jones, C., Bellve, K., Standley, C., Fonseca, S., Corvera, S., Fogarty, K., Czech, M.: Insulin stimulates membrane fusion and GLUT4 accumulation in clathrin coats on adipocyte plasma membranes. Molecular and Cellular Biology 27(9), 3456–3469 (2007)
5. Vallotton, P., James, D., Hughes, W.: Towards fully automated identification of vesicle-membrane fusion events in TIRFM. In: AIP Proceedings, vol. 952, pp. 3–10 (2007)
6. Mele, K., Coster, A., Burchfield, J., Lopez, J., James, D., Hughes, W., Vallotton, P.: Automatic identification of fusion events in TIRF microscopy image sequences. In: ICCV Workshops Proceedings, pp. 578–584. IEEE (2009)
7. Singh, S., Bhaskar, H., Tavare, J., Welsh, G.: Multiple Particle Tracking for Live Cell Imaging with Green Fluorescent Protein (GFP) Tagged Videos. In: Singh, S., Singh, M., Apte, C., Perner, P. (eds.) ICAPR 2005. LNCS, vol. 3687, pp. 792–803. Springer, Heidelberg (2005)
8. Boulanger, J., Gidon, A., Kervran, C., Salamero, J.: A patch-based method for repetitive and transient event detection in fluorescence imaging. PLoS One 5(10), e13190 (2010)
9. Oheim, M., Schapper, F.: Non-linear evanescent-field imaging. Journal of Physics D: Applied Physics 38, R185 (2005)