FluoroSim: A Visual Problem-Solving Environment for Fluorescence Microscopy

Cory W. Quammen, Alvin C. Richardson, Julian Haase, Benjamin D. Harrison, Russell M. Taylor II, and Kerry S. Bloom

1Department of Computer Science, UNC Chapel Hill, USA
2Department of Biology, UNC Chapel Hill, USA

Abstract

Fluorescence microscopy provides a powerful method for localization of structures in biological specimens. However, aspects of the image formation process such as noise and blur from the microscope’s point-spread function combine to produce an unintuitive image transformation on the true structure of the fluorescing molecules in the specimen, hindering qualitative and quantitative analysis of even simple structures in unprocessed images. We introduce FluoroSim, an interactive fluorescence microscope simulator that can be used to train scientists who use fluorescence microscopy to understand the artifacts that arise from the image formation process, to determine the appropriateness of fluorescence microscopy as an imaging modality in an experiment, and to test and refine hypotheses of model specimens by comparing the output of the simulator to experimental data. FluoroSim renders synthetic fluorescence images from arbitrary geometric models represented as triangle meshes. We describe three rendering algorithms on graphics processing units for computing the convolution of the specimen model with a microscope’s point-spread function and report on their performance. We also discuss several cases where the microscope simulator has been used to solve real problems in biology.

1. Introduction

Fluorescence microscopy is an indispensable tool for imaging biological specimens. A traditional brightfield microscope records the image formed by the absorption and transmission of an external light source as it travels through a specimen. In contrast, a fluorescence microscope records the image from light emitted by fluorescing molecules, called fluorophores, attached to or embedded within a specimen. When illuminated with light of a specific excitation frequency, these molecules fluoresce, emitting light at a lower frequency. A dichroic mirror filters out the excitation frequency, allowing only the light at the fluorescence frequency to be registered by the image sensor. Figure 1 shows a conventional fluorescence microscope setup.

Fluorescence microscopy has three key benefits. Scientists can label only the parts of the specimen in which they are interested with fluorophores, making tasks such as determining the location and structure of specific subcellular components much easier than with conventional bright field microscopy. Additionally, fluorescence microscopy enables in vivo experiments impossible with other higher resolution imaging modalities that require conditions fatal to the specimen, such as the vacuum required in a transmission electron microscope. Finally, fluorescence microscopy allows optical sectioning of specimens by adjusting the focal plane through a series of positions along the optical axis (conventionally denoted as the z-axis), forming a stack of 2D images that constitutes a 3D image of the specimen.

While the benefits of fluorescence microscopy have propelled it into widespread use in microbiological research, artifacts from the image formation process pose challenges for qualitative and quantitative analysis. A 3D point-spread function (PSF) characterizes the optical response of a fluorescence microscope to a point source of light. The PSF can be thought of as a blurring kernel, producing the fuzzy images characteristic of fluorescence microscopy. An xy-slice in the PSF represents how light passes from a point source emitter through the focal plane associated with that slice.
comes from stray photons in the system, and read-out noise detected by a charge-coupled device (CCD), background noise number of photons emitted from the fluorophores and de-imaging process: shot noise arises from fluctuations in the light from the specimen, even from out-of-focus fluorophores, is collected in each image [MKCC99], causing the pinhole aperture to block much of the out-of-focus light. The confocal microscope is thereby truncated, having roughly an hourglass shape. Focus decreases with increasing $z$ distance from the center of the hourglass, resulting in an overall widening and dimming of the PSF. In fact, all of the light from the specimen, even from out-of-focus fluorophores, is collected in each image [Cas79]. The PSF can be calculated from theory [GL92], but it is preferable to measure the PSF from the microscope under the same conditions used to acquire experimental images. A measured PSF can be obtained by imaging a fluorescent bead smaller than the spatial extent of a pixel in the resulting image.

In a widefield fluorescence microscope, the PSF has roughly an hourglass shape. Focus decreases with increasing $z$ distance from the center of the hourglass, resulting in an overall widening and dimming of the PSF. In fact, all of the light from the specimen, even from out-of-focus fluorophores, is collected in each image [MKCC99], causing the characteristic blur found in widefield fluorescence images. Confocal microscopes reduce image blurriness by using a pinhole aperture to block much of the out-of-focus light. The PSF from a confocal microscope is thereby truncated, having an approximately elliptical shape with primary axis along $z$.

Besides blur from the PSF, noise further distorts fluorescence images. Noise is introduced in several parts of the imaging process: shot noise arises from fluctuations in the number of photons emitted from the fluorophores and detected by a charge-coupled device (CCD), background noise comes from stray photons in the system, and read-out noise is inherent in CCDs [SGGR06]. The combination of these noise sources fits an approximately Gaussian distribution.

Figure 3 shows examples of the unintuitive transformation fluorescence microscopy induces on objects such as spheres and tubes. The surprising results from these simple specimen geometries make clear that understanding the imaging process is vital for scientists who use fluorescence microscopy in the lab. Toward that end, we have developed FluoroSim, an interactive visual problem-solving environment that generates images of geometric specimen models as they would appear in a fluorescence microscope (available for download at http://www.cs.unc.edu/Research/nano/cismm/download/microscopesimulator/). We see several applications for FluoroSim:

- Training inexperienced scientists to understand the imaging artifacts in fluorescence microscopy.
- Testing and refining hypotheses of specimen model geometry by comparing simulated images from a specimen model to experimental images from a real specimen.
- Planning experiments by determining whether fluorescence microscopy is a suitable modality for distinguishing among multiple hypotheses.
- Generating fluorescence images of known geometry to test new image processing and analysis algorithms.
- Automatically registering a parameterized specimen model to experimental images obtained in the lab to determine the model parameters that best explain the image.

We aim for interactive rendering rates for several reasons. Interactivity allows a scientist to manipulate a specimen model and watch in real time how changes in orientation
Quammen et al. / FluoroSim: A Visual Problem-Solving Environment for Fluorescence Microscopy

Figure 3: Examples of unintuitive images in fluorescence microscopy. (a) Models of surface-labeled small tubes. The side-by-side tubes are half the diameter of the larger tube. (b) Simulated noise-free fluorescence image of the tubes convolved with a calculated widefield PSF generated by XCOSM [MC02]. The two tubes can easily be misconstrued as a single tube while the single tube appears as two tubes. (c) A surface-labeled bead (in wireframe) with 1 micron radius superimposed with experimental images at focal planes with 2 micron spacing. The top-most image could be interpreted as the top of the sphere, but it is four times further away from the sphere center than the true top.

and position relative to the focal plane appear in the resulting fluorescence image. The real-time feedback improves the scientist’s intuition for artifacts in fluorescence microscopy. Interactivity also encourages a scientist to quickly try out interesting scenarios involving specimen models, possibly leading to insight on future experiments. Finally, in future work, we plan to use the simulator for registering specimen models to 3D fluorescence images; many 3D fluorescence image stacks will be generated during registration, so reducing the time to generate each 2D image will ultimately improve registration speed.

To make fluorescence image simulation interactive, FluoroSim incorporates convolution rendering algorithms we have developed to run on graphics processing units (GPUs). These algorithms compute the convolution of an arbitrary model geometry with the PSF of the microscope at interactive frame rates for a single focal plane.

2. Related Work

Convolving a model of a fluorophore distribution with a PSF is a fundamental building block in iterative deconvolution algorithms that attempt to remove blurring induced by the PSF. Such image restoration methods attempt to invert the convolution by searching for the underlying fluorophore distribution in the specimen [MKCC99]. In these algorithms, the specimen model is represented implicitly by an image containing an estimate of the fluorophore density in each voxel.

Aside from deconvolution, convolving a representation of fluorophores with a PSF is an integral part of the image simulation technique called model convolution, on which our work builds. In this technique, the distribution of fluorophores is modeled either explicitly or as the result of a simulation of biological processes [SGP’04] [GOB07]. Fink et al. used simulated fluorescence images of subcellular component models represented by constructive solid geometry to estimate fluorophore densities inside real cells [FML98]. Lehmussola et al. developed a parametric random shape model for generating simulated fluorescence microscope images of populations of cells to test image analysis algorithms [LRS’07]. Sprague et al. used model convolution to evaluate models of kinetochore-attached microtubule dynamics in yeast during metaphase by statistical comparison of simulated and experimental images [SPM’03].

FluoroSim advances the model convolution technique in several ways. First, FluoroSim supports convolution of triangle mesh models widely used in 3D modeling. Second, FluoroSim incorporates algorithms we have developed for the GPU that enable interactive convolution of models. Finally, the FluoroSim modeling environment enables the creation and manipulation of specimen models with real-time updates of the simulated image.

3. Image Generation

FluoroSim generates 2D fluorescence images at a single focal plane. This approach has two performance advantages over generating full 3D images. First, GPUs are designed specifically for rendering 2D images, so rendering a single focal plane at a time fits their capabilities well. Second, it is efficient for mimicking real microscopes during the exploration phase prior to acquisition of a stack; an entire 3D convolution need not be computed to extract and display a single section. When a full 3D specimen image is desired, the focal plane position can be adjusted along the -dimension in incremental steps just as in optical sectioning microscopy.

3.1. Geometric Representation and Virtual Fluorophore Placement

FluoroSim uses a standard triangle mesh for representing geometric specimen models. This representation is ubiquitous in geometric modeling and graphics applications and is well supported in the Visualization Toolkit (VTK) [SML06] and TetGen [SG05], software libraries on which FluoroSim is built. FluoroSim’s modeling environment allows placement and manipulation of predefined primitive models such as...
spheres and tubes, and it also loads OBJ, PLY, and VTK files created in external modeling tools.

In biological applications, fluorophores are attached to a specimen’s surface or embedded within the specimen material. FluoroSim emulates both kinds of specimen labeling with methods for generating uniform random samplings of geometry surfaces and the volumes contained inside surfaces. Figure 4 shows examples of each kind of labeling.

In surface labeling, sampling a fluorophore location according to a uniform random distribution involves selecting a random triangle in the surface mesh with probability equal to the ratio of triangle area to total surface mesh area, followed by sampling a random point in that triangle. We compute an array containing the triangle probabilities and then compute its prefix sum, which represents the cumulative distribution function (CDF) over the triangle probabilities. Given a pseudo-random number \( r \) in the range \([0, 1]\), we invert the CDF to find the index of the randomly selected triangle. Inversion entails searching for the index of the value in the CDF array closest to but less than \( r \); the CDF array is sorted so an efficient binary search is used. The resulting index indicates the randomly selected triangle. From this triangle, we draw a uniform random sample with a method described by Turk [Tur90]. For volume labeling, we first use TetGen [SG05] to tetrahedralize a triangle mesh; fluorophore generation follows the same approach but operates on tetrahedra in the volume mesh.

FluoroSim gives users the ability to change the fluorophore density on a model and regenerate fluorophore samplings. For specimens with known fluorophore density, generating multiple images with different fluorophore samplings can reveal how the appearance of multiple real specimens in experimental fluorescence images might differ due to variability inherent in how fluorophores bind to the specimen. For specimens with unknown fluorophore density, we start with a low density and increase it until the appearance of generated images across successive samplings does not change significantly. In most cases, sampling tens of thousands of fluorophores is sufficient. This approach is equivalent to averaging images produced by successive fluorophore samplings, resulting in an expectation image of the specimen model. If the intensity range of the expectation image is different from the intensity range in experimental images, the discrepancy can be corrected by rescaling intensities in the expectation image.

3.2. Convolution Rendering on the GPU

We implemented three convolution rendering algorithms for fluorescence microscope simulation as extensions to the VTK library. The algorithms are partially expressed as programs in the OpenGL Shading Language and NVIDIA’s CUDA Toolkit. Each algorithm takes a list of 3D points representing fluorophore locations and a 3D image representing the PSF of the microscope. For all algorithms, we store the PSF in a 3D texture because it provides an easy mechanism for looking up trilinearly interpolated values in the PSF. The camera model is orthographic because of the small depth of field in fluorescence microscopes, and the view direction is fixed parallel to the \( z \)-axis, looking in the \(-z\) direction.

3.2.1. Billboarding Algorithm

In this approach, we use a standard method for producing volumetric effects in rasterized computer graphics. The basic approach involves drawing rectangular billboards aligned with the image plane that are textured with images. By enabling the GPU’s blending mode, the billboards can be blended together in interesting ways to produce volumetric effects such as fog. In the case of fluorescence microscope simulation, one billboard is drawn for every fluorophore so that the billboard’s center is at the projected \((x, y)\) location of the fluorophore in image space. The billboard’s extent matches the extent of the PSF in \(x\) and \(y\) image space, and the texture on each billboard is an \((x, y, z)\) slice through the PSF (see Figure 2a for an example) at a specific \(z\)-depth. The \(z\)-depth is the difference between the fluorophores position’s \(z\)-component and the focal plane position; the center of the PSF, where the point-source is most in focus, corresponds to a \(z\)-depth of 0. Rendering the billboards with 32-bit floating-point additive blending yields the sum of the fluorophore contributions.

In our OpenGL implementation, each corner of the billboard has a 3D texture coordinate used to look up the \(xy\)-slice from the PSF texture. In particular, the four texture coordinates are \((0, 0, z), (0, 1, z), (1, 1, z), \) and \((1, 0, z)\), where \(z\) is the difference computed above scaled and biased to fit in the texture coordinate range.

3.2.2. Per-Pixel Gather Algorithm

Modern GPUs have many programmable streaming processors that support typical computational patterns such as accessing arbitrary memory locations and executing dynamic length loops. To exploit these streaming processors, we have
implemented a fragment program that, for every pixel in the output image, iterates through the list of fluorophore locations and sums the light contribution from each fluorophore to the pixel. A fluorophore’s contribution to a pixel is determined by computing a 3D offset from the fluorophore location to the pixel center in world space and using that vector, after appropriate scaling and biasing, as an index into the 3D PSF texture. The value returned by the texture lookup is computed by hardware-accelerated trilinear interpolation of the eight PSF voxels that surround the center of the image pixel. Fluorophores falling outside the boundary of the PSF centered at the pixel contribute no light to the pixel.

There are some implementation challenges with this approach. We store the list of fluorophore locations as a 32-bit texture, and for every pixel, we look up the value at each point, after trilinear interpolation, and add those values together to get the total light contribution to the pixel. The two optimizations increase the speed of this algorithm. In the first optimization, a screen-space bounding rectangle of the projected fluorophores diluted by a rectangle half the screen-space extent of the PSF limits the computation domain. The bounding rectangle is large enough to ensure that all pixels potentially having fluorophore contributions are processed while excluding pixels that cannot receive contributions from fluorophores. The second optimization comes from considering that fluorophores on a specimen model are likely to be close together while specimen models may be far apart. A multi-pass approach is used where each pass gathers the light contributions from a subset of the fluorophores and adds the results to the framebuffer with additive blending.

Two optimizations increase the speed of this algorithm. In the first optimization, a screen-space bounding rectangle of the projected fluorophores diluted by a rectangle half the screen-space extent of the PSF limits the computation domain. The bounding rectangle is large enough to ensure that all pixels potentially having fluorophore contributions are processed while excluding pixels that cannot receive contributions from fluorophores. The second optimization comes from considering that fluorophores on a specimen model are likely to be close together while specimen models may be far apart. A multi-pass approach is used where a separate bounding rectangle is computed and processed for each specimen model. This optimization potentially reduces the total number of pixels covered by the bounding rectangles, particularly for small specimens separated by large distances in the rendered image. It also reduces the ratio of fluorophores that contribute to a pixel to those that do not because fluorophores far from the pixel are not examined in the fragment program.

3.2.3. Fourier Domain Algorithm

The previous two convolution methods operate in the spatial domain. Using the Fast Fourier Transform (FFT), convolution via point-wise multiplication in the Fourier domain is asymptotically faster than the spatial domain methods. We have implemented a Fourier domain based algorithm that bins fluorophores into an image, then convolves that image with the PSF. Rather than compute a full 3D convolution, we follow Sprague et al. and implement a method that sums together partial 2D convolutions of subsets of the fluorophores to produce a 2D image at a particular focal plane depth [SPM*03]. The algorithm proceeds as follows:

1. With additive blending enabled, rasterize as points the fluorophores within a thin slab normal to the z-axis into a fluorophore image.
2. Render the PSF slice corresponding to the slab into a second image.
3. Convolve the two images using the FFT algorithm and component-wise multiplication in the Fourier domain.
4. Add the convolution result to an accumulation image.
5. Repeat for all slabs in z within the z-extent of the PSF.

On the GPU we compute Step 1 by adjusting camera clipping planes and rendering all the fluorophores into a texture target with additive floating-point blending. A slab thickness 0.25 times the z-spacing of the PSF offers a tradeoff between accuracy and speed; a smaller fraction of the z-spacing would produce more accurate results at the cost of computing the convolution of more slabs. We compute Step 2 by rendering the PSF into a second texture target, cyclically shifting the PSF slice so that its center is at the image origin. As in the billboarding algorithm, the PSF slice is determined by converting the world space difference between the slab and the focal plane into the third coordinate of the 3D PSF texture lookup. Step 3 makes use of the CUFFT library function that computes the FFT on the GPU as well as a custom CUDA kernel function for computing the pointwise multiplication in the Fourier domain. Finally, Step 4 is computed by rendering the convolution result into an accumulation texture with additive blending.

When specimen models are small relative to the extent of the PSF in the z-dimension, many of the slabs will contain no fluorophores and hence there is no need to compute the convolution. To prevent unnecessary computation, we check how many fluorophores were rendered into the slab image with OpenGL occlusion queries. If none are rendered, we can skip the convolution for that slab.

Because rasterizing the fluorophores quantizes the fluorophore locations, the method is less accurate than the two spatial domain methods. To increase accuracy, thinner slabs and higher resolution of the fluorophore and PSF images could be used at the cost of increased memory and computation.

3.3. Gaussian Noise Generation

To indicate expected variability due to noise in fluorescence images, we implemented Gaussian noise generation on the GPU. We generate uniform random numbers in parallel using the method described in [TW08]. For each pixel, two
of the four 32-bit pseudo-random numbers generated with this method are used to generate a sample from a normalized Gaussian distribution using the Box-Muller transform [BM58]. The generated noise is added to the image following the convolution rendering step.

4. Performance Results

We tested the performance of the three convolution rendering algorithms under three modeling scenarios:

- Rendering an image of a single small specimen model where the spatial extent of the image is small relative to the spatial extent of the PSF image.
- Rendering an image of a single small specimen model where the spatial extent of the image is large relative to the spatial extent of the PSF image.
- Rendering a large image of three small specimens separated in \(z\).

The specimen models were surface-labeled one micron spheres, the PSF was \(512 \times 512\) image of three specimens separated in \(z\). The generated noise is added to the image following the convolution rendering step.

Table 1: Rendering times (in milliseconds) of the three convolution algorithms as a function of fluorophore count.

<table>
<thead>
<tr>
<th># fluorophores</th>
<th>Billboarding algorithm</th>
<th>Gather algorithm</th>
<th>Fourier algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>50,000</td>
<td>242.1</td>
<td>112.0</td>
<td>125.7</td>
</tr>
<tr>
<td>100,000</td>
<td>483.5</td>
<td>217.5</td>
<td>133.2</td>
</tr>
<tr>
<td>200,000</td>
<td>964.8</td>
<td>429.5</td>
<td>146.7</td>
</tr>
<tr>
<td>400,000</td>
<td>1,932.0</td>
<td>852.9</td>
<td>174.3</td>
</tr>
<tr>
<td>800,000</td>
<td>3,857.8</td>
<td>1,711.5</td>
<td>231.4</td>
</tr>
<tr>
<td>512 × 512 image of a single specimen</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50,000</td>
<td>1,307.5</td>
<td>499.5</td>
<td>1,398.9</td>
</tr>
<tr>
<td>100,000</td>
<td>2,613.6</td>
<td>993.4</td>
<td>1,404.4</td>
</tr>
<tr>
<td>200,000</td>
<td>5,226.0</td>
<td>1,982.0</td>
<td>1,428.2</td>
</tr>
<tr>
<td>400,000</td>
<td>10,452.6</td>
<td>3,976.6</td>
<td>1,445.6</td>
</tr>
<tr>
<td>800,000</td>
<td>failed</td>
<td>7,949.0</td>
<td>1,503.7</td>
</tr>
<tr>
<td>512 × 512 image of three specimens separated in (z)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50,000</td>
<td>1,307.2</td>
<td>505.3</td>
<td>4,092.2</td>
</tr>
<tr>
<td>100,000</td>
<td>2,617.1</td>
<td>1,000.1</td>
<td>4,131.7</td>
</tr>
<tr>
<td>200,000</td>
<td>5,230.6</td>
<td>1,985.6</td>
<td>4,170.4</td>
</tr>
<tr>
<td>400,000</td>
<td>10,458.8</td>
<td>3,961.5</td>
<td>4,182.5</td>
</tr>
<tr>
<td>800,000</td>
<td>failed</td>
<td>7,894.6</td>
<td>4,299.7</td>
</tr>
</tbody>
</table>

The specimen models were surface-labeled one micron spheres, the PSF was \(512 \times 512\) image of a single specimen model where the spatial extent of the image is small relative to the spatial extent of the PSF image.

5. Applications

We have used FluoroSim to answer questions in several real-world applications. All images in these applications were generated by the per-pixel gather algorithm with no Gaussian noise. In all applications, we used the same PSF which was calculated by X COSM [MC02] with the microscope parameters used to take the experimental images in Figures 5c and 5e.

Is a model of the mitotic spindle in yeast plausible?

During cell division, the mitotic spindle is a structure that ensures both the mother and daughter cells receive a copy of the DNA. We created a model of the mitotic spindle in \(S.\) cerevisiae featuring a hypothesized cylindrical arrangement of chromatin surrounding interpolar microtubules that form the backbone of the spindle apparatus [YHP+08]. Our collaborators (authors Haase and Bloom) take the qualitative match of the simulated and experimental images (see Figure 5) as evidence that supports the model.

Can fluorescence microscopy distinguish between branching and adjacent fibrin fibers?

Fibrin is a polymerized protein that forms into a mesh in blood clots. A challenge in understanding the structure of fibrin meshes and their mechanical properties is distinguishing between fibers that branch and fibers that are merely adjacent to each other. We used FluoroSim to create models of both configurations to determine whether a fluorescence microscope with this PSF can distinguish between them.
Figure 5: The mitotic spindle model \[YHP^*08\]. (a) Side-on view; (b) end-on view. Helices represent the chromatin labeled with green fluorescing protein. Experimental and noise-free simulated images in side-on orientation (c-d) and end-on orientation (e-f). Differences in background between experimental and simulated images are caused by external chromatin not accounted for in the model. Structural variation in real specimens account for shape discrepancies.

Figure 6c-d shows simulated images from the two volume-labeled model configurations where the focal plane is centered in the horizontal fiber. The images appear indistinguishable. However, intensity profiles along a line running down the center of the horizontal fiber in the simulated images show that where the two fibers meet, the branching fiber model is brighter than the adjacent fiber model. Assuming the two fibers have uniform diameters, the result is unexpected; where the fibers overlap, the two adjacent fibers have twice the number of fluorophores that are in the branching fiber model and should therefore appear brighter there. Our branching model, however, happens to have slightly more volume where the two fibers meet than would the union of two fibers, accounting for some of the increased intensity. Moreover, the additional fluorophores in that extra volume are nearer the focal plane than the top adjacent fiber, increasing their light contributions and accounting for the rest of the increased intensity. This investigation provides evidence that distinguishing between the two fiber configurations depends on the thickness of the fibrin at the branch points.

6. Conclusions and Future Work

FluoroSim is a new software tool for training, hypothesis testing, and experiment planning involving fluorescence microscopy. Our main contribution is a method for generating simulated fluorescence images from arbitrary geometric models represented by triangle meshes at interactive rates. As part of this contribution, we have described three convolution algorithms that exploit GPUs and analyzed their performance in several modeling scenarios. We also showed three biological applications where FluoroSim has been used to answer relevant questions.

In future work, we intend to add model registration to FluoroSim. This will enable the automatic optimization of a geometric model’s parameters to a fluorescence image such that the model parameters best explain the image. As part of this effort, we plan to validate the simulated images by comparing them to experimental images of specimens with known geometry such as spherical polystyrene beads. We
also intend to further improve rendering speed by parallelizing image generation across multiple GPUs.

Acknowledgements

We gratefully acknowledge David Feng, Brian Eastwood, Ryan Schubert, and the anonymous reviewers for their helpful suggestions for this paper. We also thank David Borland for guidance on aspects of the OpenGL implementation.

C. W. Quammen, A. C. Richardson, and R. M. Taylor II were supported by National Institutes of Health grant GM32238 and National Science Foundation grant MCB 0451240.

References


© The Eurographics Association 2008.