

## PROTEIN EXPRESSIONS

### Study N. 1

Here we describe some of the technical and scientific aspects of the video, scene by scene. We are aware of the fact that the terminology used can be difficult for non-professional biologist or graphic experts. Hopefully the interested reader will take occasion to get some insights from many excellent cell biology and 3D computer graphics textbooks.

#### 1. Red Blood cells with macrophage

The set is a fine capillary. It is populated by red blood cells, platelets and a few white blood cells. The scene was built in Autodesk Maya. The capillary is a cylinder on whose surface (inside normals) is mapped a microscopic photograph of epithelial cells grown to confluence in a Petri dish.

Red cells are all particle instances of a simple polygonal sphere deformed to become biconcave; they are emitted at 200 particles/sec, set at high speed and collide with the capillary wall. The white cell with the aspect of a macrophage is a Nurbs sphere converted to soft body and is moved by a turbulence field inside. On its surface are mapped two hair systems (short and long 'tentacles') whose motion is governed by the sphere geometry.

#### 2. White Blood cell (T cell).

In the same scene (with less red cells and platelets) the T cell is a polygonal sphere converted in Maya to N-cloth with a another N-rigid sphere inside to keep it in shape; its surface geometries are modelled by hand and moved randomly.

#### 3. Flight over cell surface.

The membrane was obtained imposing properties derived from membrane biochemistry to several particle systems (in Maya): single lipids, groups of lipids (rafts) and groups of proteins distributed on the surface. This was converted to a grey scale texture attributed to the displacement node of the surface of a plane using Blender SuperTexture Node and Texture Plugin.

#### 4. The membrane.

Based on the texture above, the rafts were constructed as a Game Engine (GE) structure in Blender, with proteins loosely connected with rigid body joints. Proteins in the raft were modelled (with very low resolution) on the basis of PDB files: CD4 (1WIO), Thy-1/CD59 (2UX2), TCR (1KCG), CD3 trimer (1XIW, 1SY6, 1SY6), Toll Like Receptor (2Z7X), ICAM-1 (1IAM for domains 1 and 2, and 1P53 for domains 3,4 and 5).

The group of proteins was set to follow the centre of a raft. This motion was recorded and provided the 2D surface sliding of proteins on the membrane. The Z (vertical) motion to follow the wavy surface was obtained by dropping the proteins at each frame on the distorted plane. An additional rotatory motion was attributed with a random generator.

Each protein also has a skeleton that provides the inner motion. Also oligosaccharides chains are modelled from PDB structures and moved with an internal skeleton by another random motion

generator, faster and with wider angles relative to proteins.

The Potassium channel is modelled from the structure in PDB file 1LNQ (side chains added with Swiss PDBviewer).

## 5. Calmodulin.

The movement of Calmodulin (CaM) is the most scientifically relevant part of the entire video. CaM is a small protein (148 aminoacids, 2262 atoms) structurally organized in two parts connected by a flexible linker. The two parts are capable of capturing Calcium ions, and when this happens the lobes open and become reactive towards other proteins, thus transducing the signal delivered by the Calcium wave.

We meet apoCalmodulin once inside the cell, immediately after falling into the K<sup>+</sup> channel. Protein movements were obtained elaborating on an NMR collection (1CFC) using the Game Engine of Blender according to a flow as follows:

all atoms of the protein (including hydrogens) are imported in Blender (we wrote a specific script to import and export .pdb files) in two different positions, chosen as the farthest apart as RMSD, separated by X frames (say 500). Atoms are connected with rigid body joints (chemical bonds). Blender GE interpolates between conformations evaluating joints and collisions. Intermediate conformations so generated are automatically exported, adjusted using the GROMOS force field implemented in Swiss-PDBviewer and compared (RMSD) with other conformations present from the original NMR collection. Any NMR conformation closer than 2 Å is imposed as intermediate keyframe in the correspondent time between the initial positions, and the process is repeated.

With a few cycles of this procedure we have generated a continuous motion that covers a geometrical distance of 21 Å in steps of <1 Å, 'visiting' a dozen conformations experimentally derived. These are set in Blender at 50 frames distance, and again interpolated by the GE. Finally, each frame of the global movement is exported as a pseudo-pdb file and sent to biophysical programs that introduce some necessary minor adjustments to the position of atoms.

For visualization of chemical and physical properties, we have developed a visual code applied to the surface of the molecule without making use of colors.

Each frame from the animation (.pdb), after adjustments of atomic positions, is converted to .pqr and elaborated with a series of libraries and other programs and scripts (VMD and pyMLP) that calculate electrostatic and lipophilic potential (EP and LP respectively) values. Using a home-made program, these fields are mapped on the protein surface (obtained as an .obj file from Chimera) and associated to all vertices of the mesh. These values are fed to the texture nodes of Blender and are finally expressed as scale of smooth-shiny to bumpy-dull for MLP and with particles emitted from and attracted to the surface for EP.

## 6. Microtubules.

Microtubules are built with an array of tubulin dimers (1TUB) displaced by the appropriate distance to form a 13 filaments pipe. To represent the EP on the microtubules, a system of particle was set to emit 300.000 light points. Proteins in the scene (except Calmodulin) are NFkB (from our colleagues at Cineca, Bologna), Diaphanous (1Z2C) and Aurora (2BFX), rendered according to our procedure and displaced as projections on black (invisible) billboards.

Calcium waves are yet another particle system emitting every 3 seconds (75 frames). We travel along the microtubule until we came close to one such wave, and see Calmodulin during its transition from apoCalmodulin to Calcium-loaded (CaCaM, PDB entry 1X02).

## 7. ApoCaM to CaCaM

We observe the transition above from different points of view, revealing the opening of the

protein when Calcium is bound. We agree that calcium ions deserve a better (more scientific, with proper size and charge) representation than they have now.

### 8. Skipping to the plasma membrane.

Travelling fast along the microtubules, we reach the plasma membrane during the cytokinesis, when the contractile ring pulls the membrane that will eventually divide the cell in two. Here a bundle of actin filaments are pulled by myosin, as described in the next section

The underlying membrane is plane converted to cloth, contracting along the middle line through pinning to a cylinder scaling in time, and covered with 50.000 particles (hair) that follow the plane geometry to simulate presence of many proteins.

### 9. The contractile ring

To create the actin filaments, we positioned the actin monomers according to the crystal structure in PDB file 1ALM. This model, containing 5 monomers, was imported in the Blender scene via Chimera. Since the models in this file only contain the alpha-carbon atoms, another, atomic resolution crystal structure of actin (1ATN) was used to obtain a high resolution actin filament model. In Blender an array modifier was created, to multiply the single actin model by values derived by carefully adjusting the offset and rotation settings of the monomers in the crystal. The array-based actin filament can be elongated in a snap.

Myosin filaments were created using of PDB files 1I84 and 1SR6, (two extreme conformations of the heavy chain motor domain) and 1NKN (coiled-coil tail structure from two intertwined myosins), which were loaded into Chimera, together with 1ALM (5 aligned actin monomers). The latter file also contains a myosin head domain bound to the actin filament, and was used to align the assembly of proteins. The other myosin structures were aligned to the 1ALM file and the surface models were created and exported. The models also include the myosin essential light chain and regulatory light chain molecules, wrapped around the neck domain of myosin heavy chain. Once imported into Blender, the myosin motor domain movement was animated using an armature for the pre-power stroke model. The armature was connected and subsequently posed to closely match the myosin surface model in the power stroke position. Care was taken to not only animate the large displacement of the head group, but also the more subtle displacements of smaller subdomains of the proteins, through an additional armature into the bumps of the protein surface model. The resulting rigged model could now be moved from the pre-power stroke conformation to the powerstroke conformation by moving from one pose of the armature to the other. The final movement of the myosin molecule is composed of five partly overlapping steps:

- 1- power stroke (first 20 frames); the motion of the myosin head while attached to an actin filament; this motion causes the displacement of the filament.
- 2- release and lift (frame 20-40). The myosin head is released from the actin filament and the head is lifted away from the actin.
- 3- prepare for power stroke (frame 29-55). The myosin head is rotated around the neck to reposition it for the next power stroke.
- 4- drop (frame 41-66). The myosin head moves towards the actin filament.
- 5- connect (frame 66-67). The myosin again grasps an actin filament; to visualize this, the motion of the bones stops for just one frame.

The tail domains of two myosin heavy chains twist around each other to form a coiled coil as described in PDB file 1NKN, whihc only contains 89 amino acids per strand, whereas the tail domain of a myosin heavy chain molecule extends over hundreds of amino acids. Therefore, the model was imported into Blender and multiple copies of the small model were carefully aligned and connected manually until the correct tail length was reached. Myosin is active as bipolar filaments, composed of 10-20 pairs of myosin molecules. To mimic this, a bundle of myosin tails

was created by joining a number of tail models together. Twenty myosin head models were placed pairwise at the two ends of this filament model. The conformational movement of the headgroups was subsequently shifted with respect to each other to create a more realistic, shuffled animation. For controlling the entire animation, each armature object (which drive the movement of the myosin head groups) was given a so-called time ipo, allowing for acceleration or deceleration of the movement of the head groups. As all the head groups were given the same time ipo, the speed of movement can be globally changed easily.

### PRINCIPAL AUTHORS:

Monica Zoppè	Scientist, IFC – CNR, Pisa. <i>Corresponding author</i>
Raluca Andrei	PhD student, Scuola Normale Superiore and IFC – CNR, Pisa
Stefano Cianchetta	3D graphic, IFC – CNR, Pisa
Tiziana Loni,	3D graphic, Big Bang Solutions, Navacchio (Pisa)
Maria Francesca Zini	Scientist, IFC – CNR, Pisa
Leonard Bosgraaf	Dept. of Biochemistry, Univ. of Groeningen, Netherland
Marco Callieri	Scientis, Visual Computing Lab, ISTI – CNR, Pisa

### PROGRAMS USED

Graphics and image processing:

Blender [www.blender.org](http://www.blender.org)

Maya Autodesk <http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=7635018>

The Gimp [www.gimp.org](http://www.gimp.org)

Djv\_view [http://djv.sourceforge.net/djv\\_view.html](http://djv.sourceforge.net/djv_view.html)

ImageMagick [www.imagemagick.org](http://www.imagemagick.org)

Scientific

VMD <http://www.ks.uiuc.edu/Research/vmd/>

Swiss-PDBviewer (Gromos43B1) <http://spdbv.vital-it.ch/>

Chimera <http://www.cgl.ucsf.edu/chimera/>

Reduce - MolProbity <http://molprobity.biochem.duke.edu/>

PDB2PQR <http://pdb2pqr.sourceforge.net/>

PyMLP.py <http://code.google.com/p/pymlp/>

APBS: <http://sourceforge.net/projects/apbs>

Several home made scripts and programs

We have used wherever possible open source programs. All our scripts and programs will be packaged in user friendly form, and made available in open format.

For more information, or comments, please email us [mzoppe -at- ifc . cnr . it](mailto:mzoppe@ifc.cnr.it)

Visit our website to stay updated with the latest news [www.scivis.ifc.cnr.it](http://www.scivis.ifc.cnr.it)