Concordia University
COMP 471 / CART 498 C – Fall 2006

Computer Graphics

Final Project Report --- Due December 11, 2006

Prof. Sha Xin Wei

Julien BRICHE – 6129439 – onisdream@hotmail.com
Aymeric FOURCADE - 6129641 - fourcade@polytech.unice.fr

Table of Contents

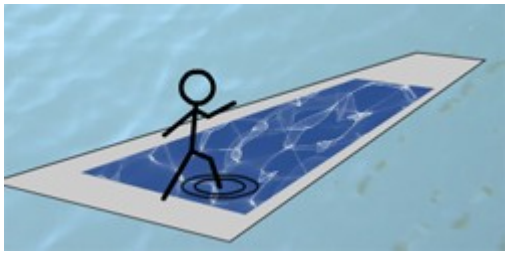
I. Introduction

Thanks to the spread of new technologies, digital arts found new ways to explore. For instance this month there is an exhibition in Montréal called « Touching the Invisible » by the Smart Studio where you can see (and experiment) their projects dealing with arts and brain frequencies (for example the project called « brain bar », by scaning your brain frequencies mixes a cocktail according to what it scaned). As a result we took this course dealing with realtime video processing to learn a little bit more about responsive digital arts, and that's what we tried moreover to explore through our project.

II. Project overview

1. Concept

What is Crossing a River ? It's a realtime video project that generates a realistic water effect on the ground which reacts as people walking above it. As a result the walker could think that he is walking on a fluid floor (not necessary water because the effect relies on the nature of the surface).



The idea of the name came to us when we were exploring the EV Building (Fine Arts Faculty building of Concordia) to find the better place to show our project. As the bridge above escalators on the ground floor seemed perfect, we imagined a virtual river lying on it with people crossing it. Finally the exhibition didn't occur there (for equipment conveniance).

2. Team

We are two exchange students from France, we belong to an engineer school in Nice (in the Computer Science department) and we decided to do our last year in Concordia in order to attend other kind of courses and more particularly computer graphics oriented ones.

We have both solid backgrounds in programing so it was quite easy to learn OpenGL, for Max MSP and Jitter it was a little bit more difficult as this is streaming processing programing but we managed togother to develop our skills in realtime video and we learnt a lot in that field.

3. Goal

Crossing The River has both aesthetic and interacive purpose. Indeed we designed it for places where people are using to pass without stopping as metro corridors or malls hall. Nevertheless such an application could only handle few people to interact with it. Moreover it doesn't require artistic sense or the attention of the audience to be appreciated. In fact it just consists in entertaining people for a short moment in a common place where they are used to walk

4. What the project explores

Basically Crossing The River captures people walking on the camera field and then projects on the floor near the last position of the walker a wave spreading through the floor.

So on the one hand we had to deal with edge detection to track a person walking in front of the camera.

On the other hand we had to create the complicated wave spreading effect which means wave mathematics or actually convolution operations. See the technical interest sections for more details.

5. Roles

To fullfill this interesting project we were only two persons and as we are roomates (so we didn't face assiduity trouble to set up meetings) we really did this project together so it's quite hard to say exactly which part we each dealt with.

As a result individual contributions are not very distinct. Indeed we did the patch together, the report together and also the web site (actually only the researches were done separately).

6. Deliverable

Attached with this project, you can find the web site on :

http://hybrid.concordia.ca/~j_briche/COMP471

With also a sets of pictures of the exhibition in the EV Building and some videos too.

At last there is the patch which manages the whole.

III. Methodology

1. What we planed to do

At the begining of our project, we went through an intense brainstorming and set up a planning as a result.

At first we wanted to detect very precisely the position of the walker on the floor thanks to couple of camera (we wanted also to generate a sound in the same time that the person hit the floor). Indeed using two cameras (lying in the floor) would have enabled us to access to the tridimensional coordinates of the walker position on the surface.

Concerning the wave effect, we decided at first to realise it with OpenGL and C++ as we had just learnt it and as programing wasn't a problem for us.

2. What we really did

As the experimentations with two cameras ran wrong (we faced huge trouble of calibration) we decided then to use one camera and to give up finding the exact position of walker's feet (we gave up in the same time to add sounds to our project) instead we performed a shape detection by using a camera above the walker which was easier to implement and which gave us a quite satisfying result (see « jitter_overview.avi » video).

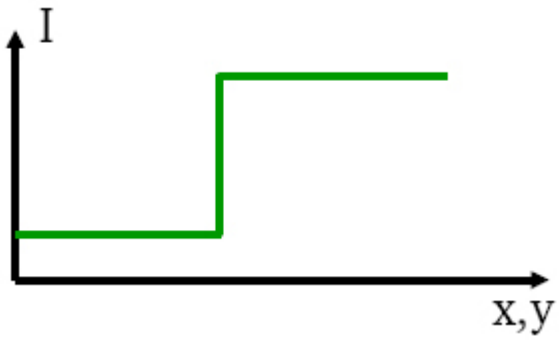
For the wave effect we didn't face any problem, the result was quite good (see « opengl_overview.avi » video). Unfortunately we failed to make communicate our OpenGL application and our Jitter patch. Of course through many tutorials we saw that it existed many ways to link those technologies but in our case our researches failed. As a result we decided to explore the convolution operation way by using a patch we saw during a course lecture but more details are explained in the following section.

IV. Technical interest

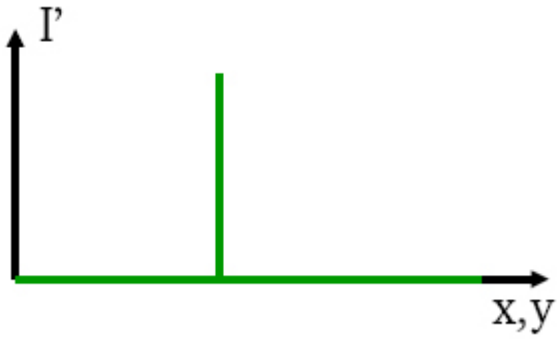
We saw earlier that Crossing The River was performing an edge detection and a wave spreading effect, let's see how does it work.

1. Edge detection

At first if we are considering a 2D image which pixel values are contained in a matrix what is an edge ?
In this view an edge is a place of the image where there is a discontinuity of the intensity (the intensity is the sum of the four Red Green Blue and sometimes Alpha components). Such a definition can be represented below with I the intensity and x, y the spatial coordinates of the 2D image.

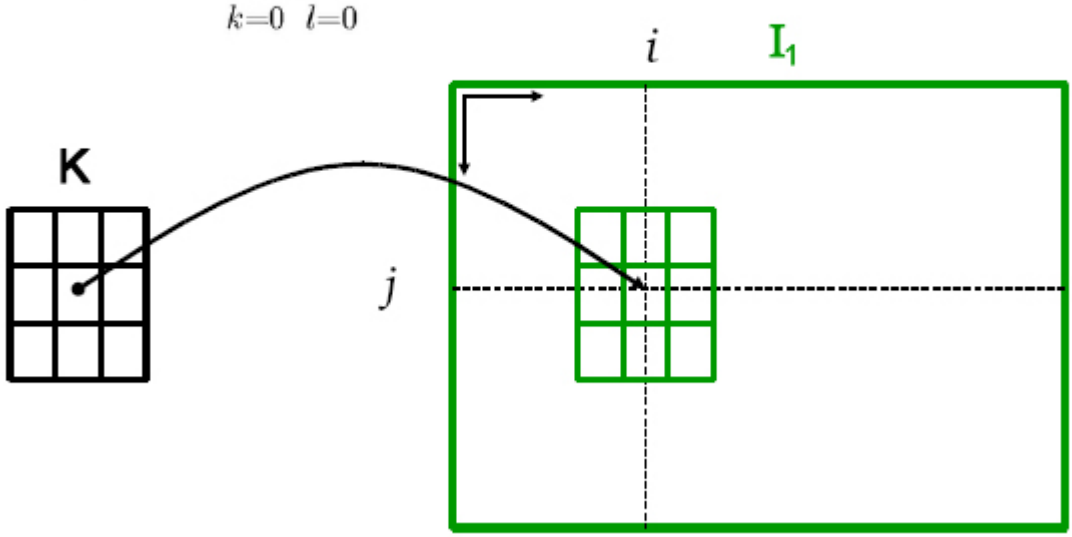


The edge is where there is a jump of intensity and in mathematics such a jump means the maximum value of the spatial derivation, as below :



So in order to detect edge we have to find those jumps of intensity. To do that we are going to use a convolution product. Basically such an operation means mutiplying a kernel with a matrix as below (I1 is the original matrix intensity, K the convolution Kernel and I the resulting intensity at the pixel (i,j) in I1) :

$$I\ (i,j) = \sum_{k=0}^{2}\sum_{l=0}^{2} I_1(i+k-1, j+l-1)K(k,l)$$



And we apply that kernel for each pixel of the image (contained in the matrix).
It exists several different kernels to perform edge detection (Roberts, Gaussian, Laplacian, average) as we decided to use a Sobel edge detection the convolution kernels look like :

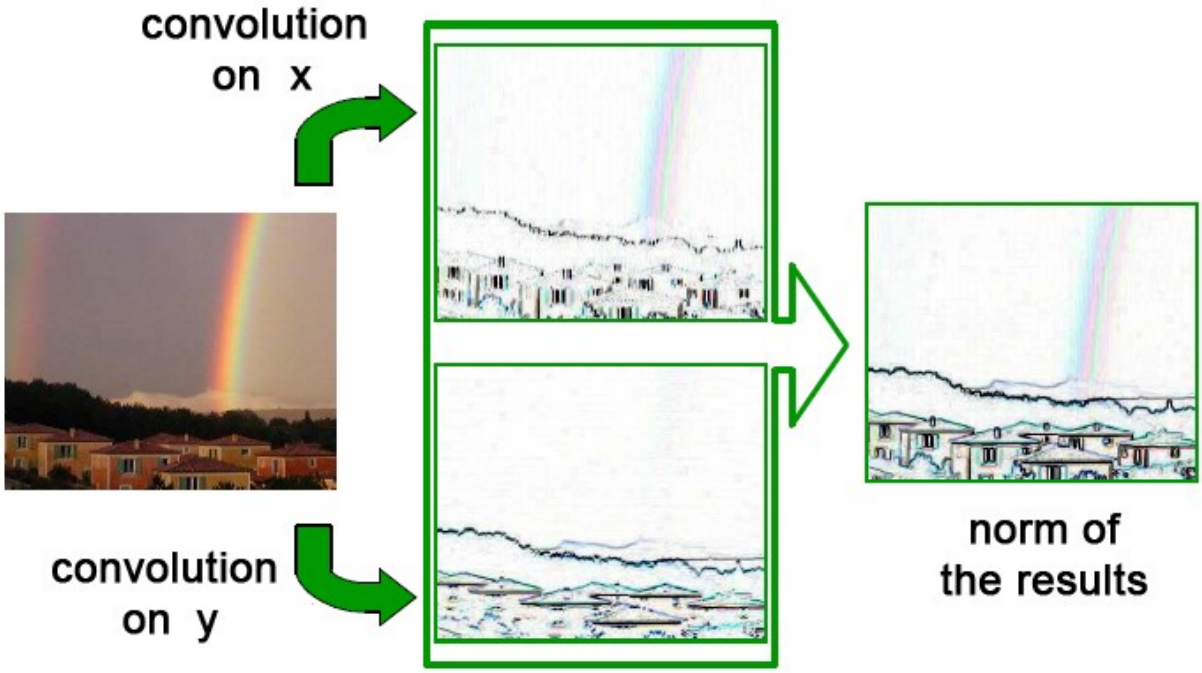
$$\frac{dI}{dx} = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} / 4$$

$$\frac{dI}{dy} = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} / 4$$

So the first convolution matrix detect edge following the x axis whereas the second one is following the y axis. To have the detection on both direction we have to compute the norm as below :

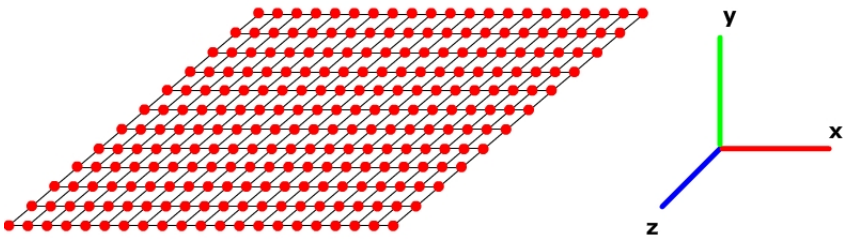
$$\sqrt{(\frac{dI}{dx})^2 + (\frac{dI}{dy})^2}$$

And the result is the new value of the pixel which the convolutions were applied. See below an example of the Sobel edge detection with the two convolutions emphased :

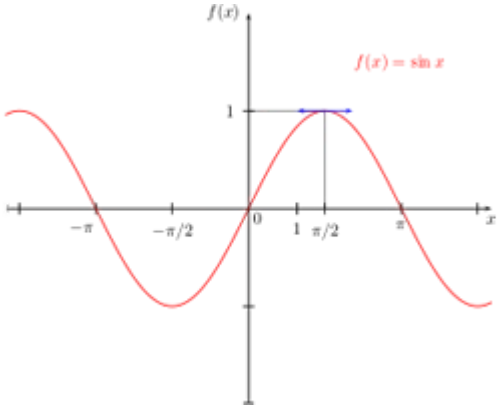


2. Wave spreading effect

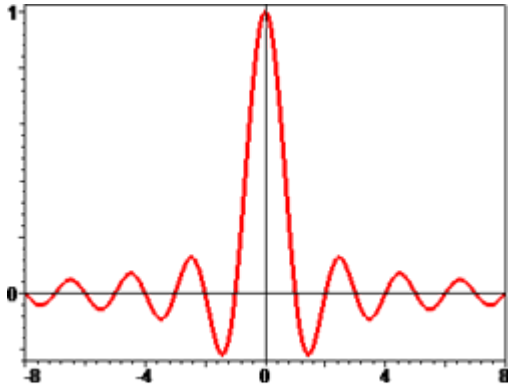
As we explained on the methodology section, we implemented by ourself the 3D wave spreading effect. To do that we took an OpenGL plan of vertices, as below :



Then our researches lead us to trigonometric functions, as the sinus function below :



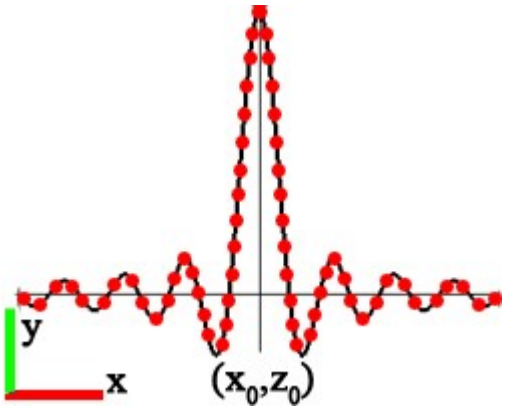
But a wave spreading effect means a spatial attenuation so we modified the sinus function by composing it with the inverse function. Which lead to the function below (called « sine cardinal ») :

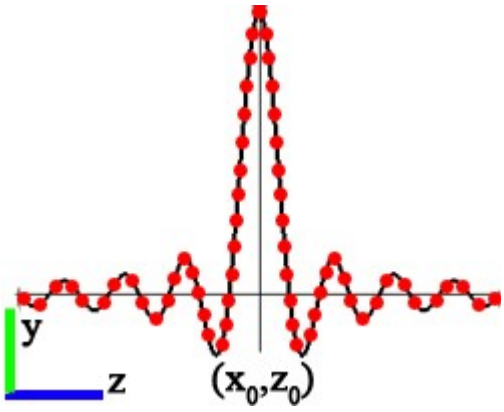


Of course we adapted it for our OpenGL vertices plan :

$$f(x,z)=\sin\frac{(x^2+z^2+k\pi)}{(x^2+z^2)} \quad with \quad k\in\mathbb{R}$$

And this is the result on the vertices plan with those side and front view :





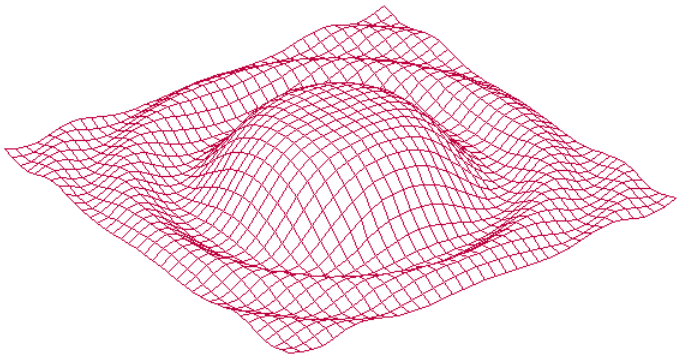


Illustration 1: sine cardinal

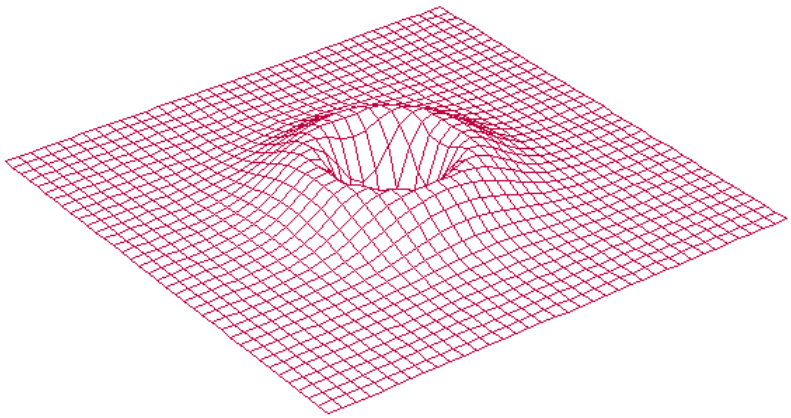


Illustration 2: exponential enveloppe

As a result the point (x0, z0) is the location where the wave begins to spread, this is the point that the edge dectection gives us and this is were we failed to match OpenGL and Jitter.

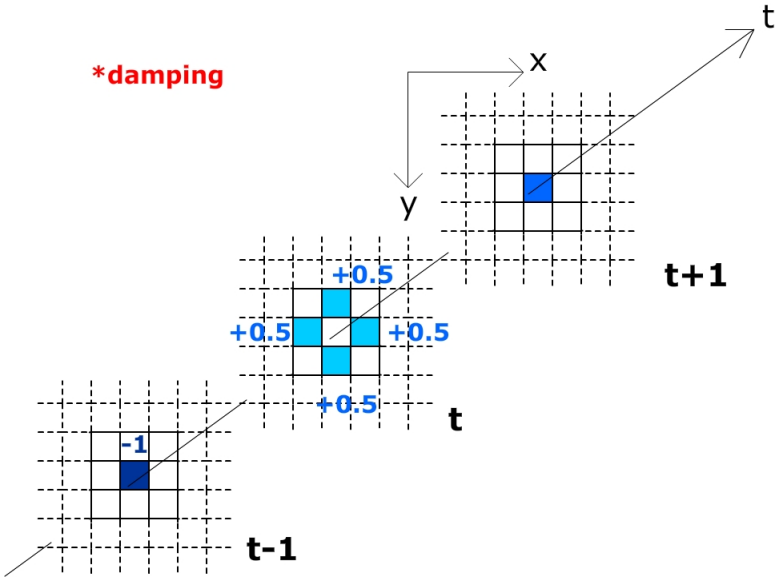
Notice : In this state waves were looping from this point so we composed one more time the « sine cardinal » with a decreasing exponential to have a wave spreading and attenuating once a time.

As we didn't manage to use our OpenGL application, we decided to search for another way. Fortunately we saw in a course lecture a patch that produced the effect we were looking for so we started to study it (and our previous work wasn't for nothing as it enabled us to see how such an effect can work).

So this patch uses the fluid equation below to spread the wave :

$$h(x,y,t+1)=damping\left[\frac{1}{2}\left[h(x+1,y,t)+h(x-1,y,t)+h(x,y+1,t)+h(x,y-1,t)\right]-h(x,y,t-1)\right]$$

Which can be represented with the following sketch :



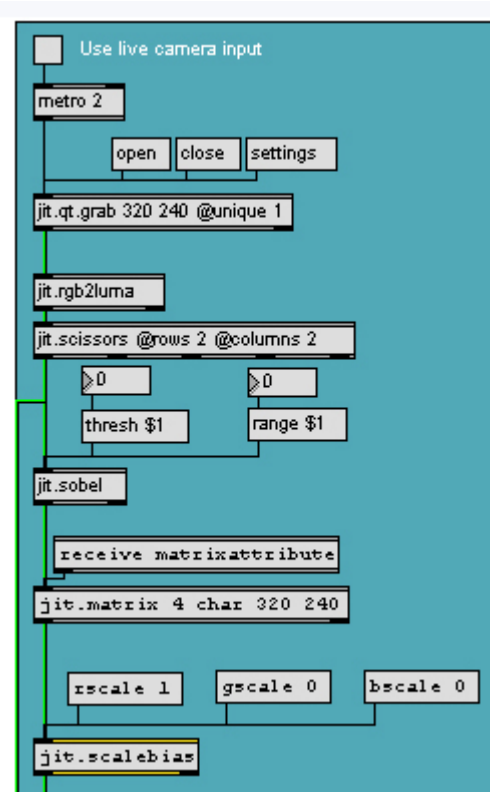
The damping is the fluid resistance force, the higher it is, the shorter are the waves.
Moreover the **jit.convolve** obecjt is also used to add neighboring pixel values in the fluid equation with the following kernel :

$$K = \frac{1}{10}\begin{pmatrix} 1 & 4 & 1 \\ 4 & 0 & 4 \\ 1 & 4 & 1 \end{pmatrix}$$

We can notice it's a kind of reversed gaussian kernel.

3. Patch explanation

In this section we will explain basically how our patch works :



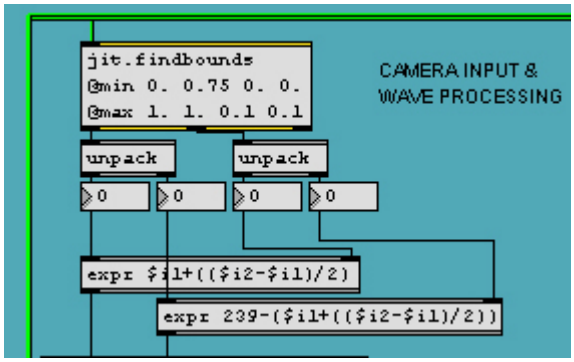
At first we use the **jit.rgb2luma** object to set the image in the monochrome luminosity mode (following the equation L = (0.299 * red value) + (0.587 * green value) + (0.114 * blue value).

Then we use the **jit.scissors** in order to match the origine of the projected plan and the origine of the camera, we added this object during the exhibition as we noticed that the range of the camera were bigger than the range of the surface projected.

Then we perform the edge detection by using the **jit.sobel** object (described above). Moreover we adjust the threshold attribute of this object in order to have only the top of the person who is walking in front of the camera (as a result the camera doesn't detect the floor and the wave effect spreading above it).

As the **jit.findbounds** object takes a range of color values we use **jit.scalebias** to turn the white shape of the person walking into red (on a black background).

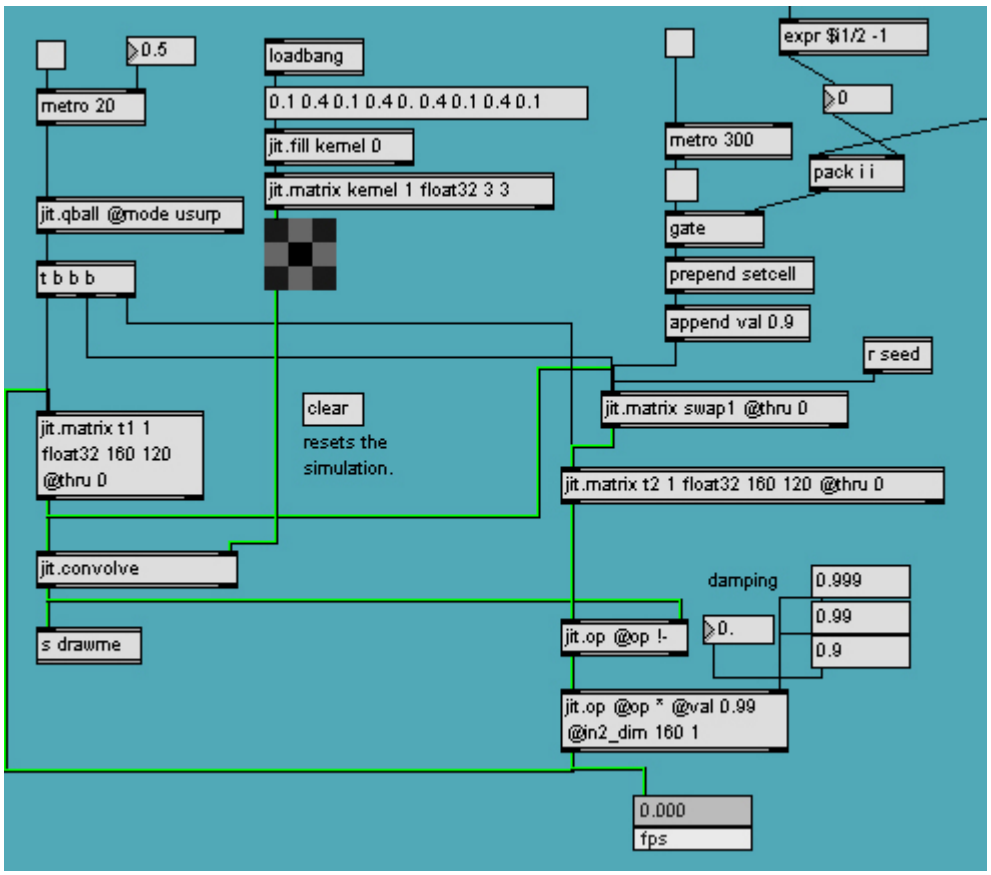
As a result **jit.findbounds** returns the bounding box surrounding the previous red shape. As it returns the xmin, ymin, xmax and ymax of the box we make a little computation to obtain the center of the bounding box.



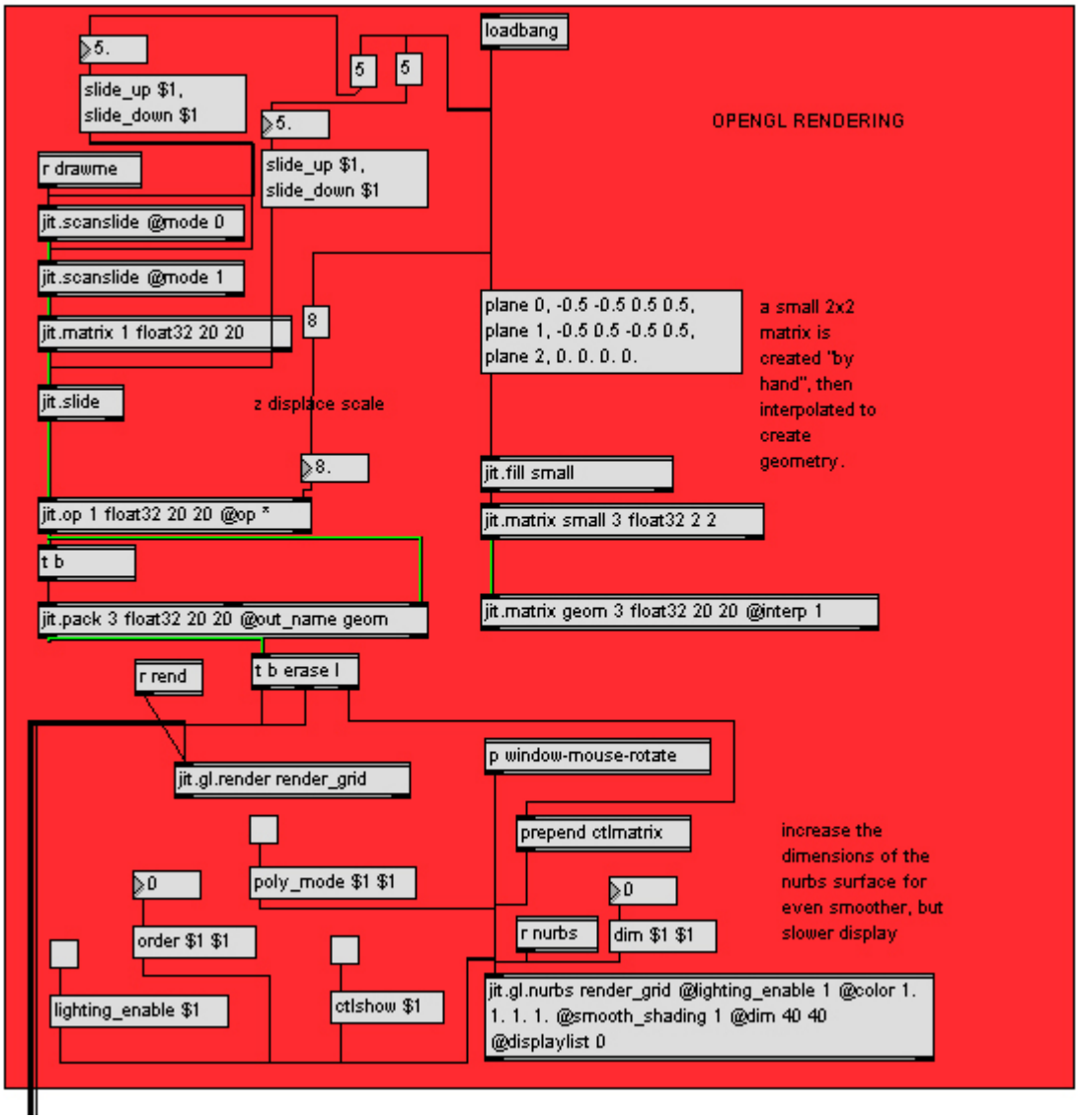
As a result we have the coordinates where the wave ripples have to begin.

Notice : At this point of the patch we are also undersampling what we obtained by putting it in a 20 lower size matrix in order to see how the motion detection is going on (it shows a black dot discribing the motion followed by the patch). This is not necessary but it was very useful during the exhibition.

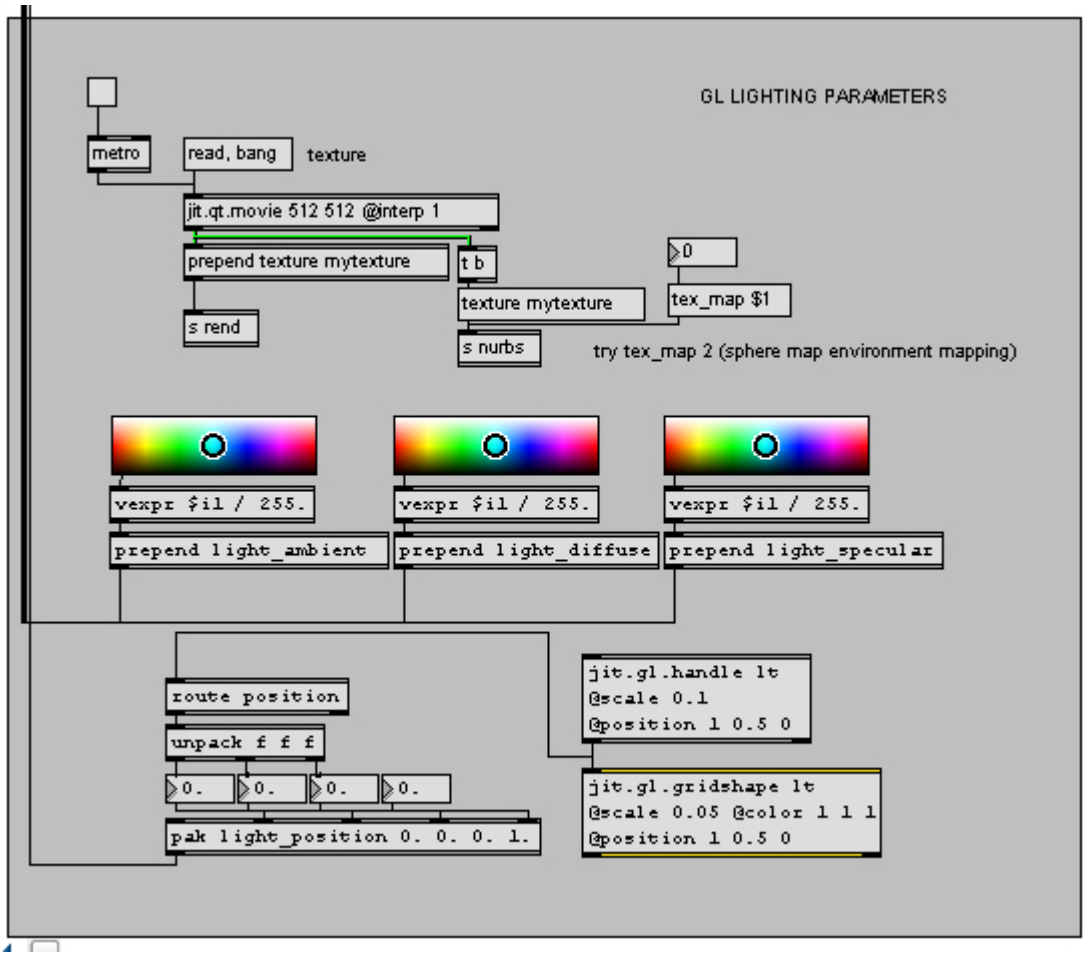
Then we use the **jit.convolve** object to perform the wave effect spreading thanks to the fluid equation and we send what to draw to the **jit.gl.render** object.



The rest of the patch (the red and grey part) deals with the OpenGL rendering of the plan where the wave spread effect occurs. We can adjust here, the light and color attributes and choose a texture (image or video) to map on the plan.

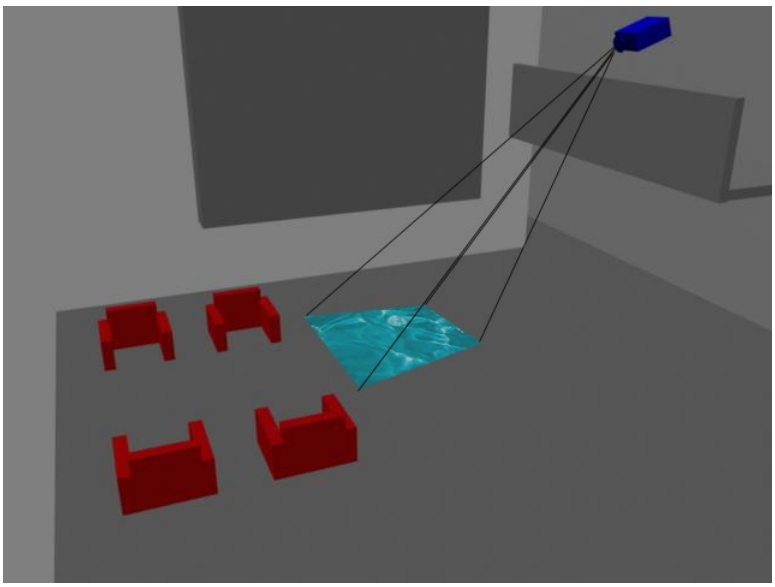


In our case we have used a video of almost still water to map on the plan. We thought at the begining to design this video with OpenGL but because of a lack of time we made it with Maya (which is actually using OpenGL).



V. Exhibition

The exhibition took place on the EV Building (1515 Ste-Catherine West Montréal) on december 7th and 8th between 3 and 6 pm. We showed our project in the 8th floor atrium, we were projected from the 9th floor balcony as below :



We faced some probems of equipment at first because we were given a computer which was rebooting randomly so it was hard to perform tests. So we had to wait that another team finished with its equipement. Finally we received a working computer and as a result we had a short amount of time to adjust everything. Consequently the first exhibition wasn't very satisfying.

Fortunately the second day things were better and we succeed to have a good result. However it would have been better if we could have switch off the lights of the atrium (we reduce this problem by putting a blue sheet on the floor).

See the web site for significant videos (better than the pictures).

VI. Conclusion

This project and this course match our expectations on digital arts, we were really happy to have the opportunity to work in such an environment (EV building) with high quality equipments. Indeed we couldn't have such an opportunity in France as our school doesn't provide such courses.
Thank you to Nick Gauthier, Olivia Tang and most of all Sha Xin Wei for their precious help and support.

VII. References

COMP471 Course Web Site

COMP6761 Course Web Site

Cycling'74

Smart Studio

Wikipedia.org

Google

Diane Lingrand

Ecole Polytechnique Universitaire de Nice Sophia Antipolis