USTH MM2.1 – Soft. Eng. for Interactive Media

(1) Example 1: Video compression with auto-refresh

(2) Example 2: Video compression with Region Of Interest (ROI)

Sylvie CHAMBON
schambon@enseeiht.fr

Example 1: Video compression with auto-refresh

- **Context:** Videos on web pages are using too much resources

→ Compression can improve the performance of the digital systems

(a) by reducing time and cost in storage and transmission

(b) without significant reduction of the image quality

- **Idea**
  - whole video is not necessary
  - Updating only the significant part of the current frame of the video

Example 1 : Video compression with auto-refresh

- **Principles**
  - Each frame of the video is displaying block by block
  - Refreshing only blocks B of image i that present a significant difference
  - More precisely $d(B_{t-1}, B_t) > T$
    with d a distance and T a threshold
  - d is defined by

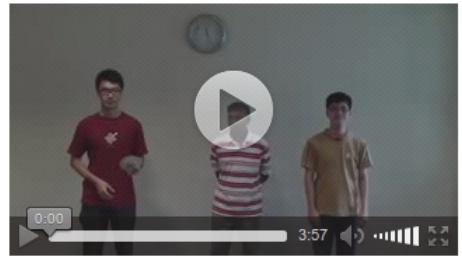
$$d(B_{t-1}, B_t) = \frac{1}{255 \times N_b} \sum_{k=1}^{N_b} |B_{t-1}^k - B_t^k| \quad (1)$$

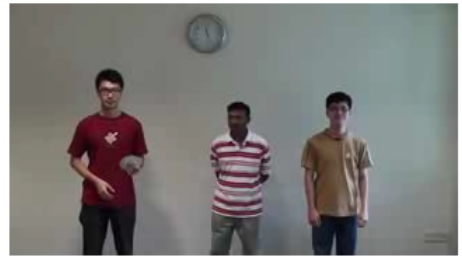
    with B_*^k the pixel k of the block B_*^k
    and N_b the number of pixels of the block
  - $255 \times N_b$ is a normalization, to have a value between 0 and 1

Example 1 : Video compression with auto-refresh

Shape of the web page

Compression by ROI coding





Number of modified blocks 0

Compression rate 0 **%**

Threshold (with or without ROI) : 0.03

Example 1 : Video compression with auto-refresh

HTML5 general page

```
<!DOCTYPE HTML>
<meta charset = utf-8>
<html lan=fr>
        <header>
                <title> Compression by Region Of Interest (ROI) </title>
                <script type="text/javascript" src="autorefresh.js"></script>
        </header>
        <body onload="doLoad()">
        <section>
        ...
        </section>
        </body>
</html>
```

Example 1 : Video compression with auto-refresh

HTML5 general page

```
<!DOCTYPE HTML>
<meta charset = utf-8>
<html lan=fr>
        <header>
                <title> Compression by Region Of Interest (ROI) </title>
                <script type="text/javascript" src="autorefresh.js"></script>
        </header>
        <body onload="doLoad()">
        <section>
        ...
        </section>
        </body>
</html>
```

Example 1 : Video compression with auto-refresh

Canvas

```
<section>
  <header> <h1> Compression by ROI coding </h1>
  </header>

  <article id="mouseInfo">
    <video id="video" src="ct320.ogg" controls width="320" height="180" autobuffer/>
  </article>

  <article>
    <canvas id="compression" width="320" height="180" > </canvas>
  </article>

  <article>
    <canvas id="origin" width="320" height="180" style="display:none"> </canvas>
  </article>

        ...
</section>
```

Example 1 : Video compression with auto-refresh

Canvas

```
<section>
  <header> <h1> Compression by ROI coding </h1>
  </header>

  <article id="mouseInfo">
    <video id="video" src="ct320.ogg" controls width="320" height="180" autobuffer/>
  </article>

  <article>
    <canvas id="compression" width="320" height="180" > </canvas>
  </article>

  <article>
    <canvas id="origin" width="320" height="180" style="display:none"> </canvas>
  </article>

        ...
</section>
```

Example 1 : Video compression with auto-refresh

Canvas

```
<section>
  <header> <h1> Compression by ROI coding </h1>
  </header>

  <article id="mouseInfo">
    <video id="video" src="ct320.ogg" controls width="320" height="180" autobuffer/>
  </article>

  <article>
    <canvas id="compression" width="320" height="180" > </canvas>
  </article>

  <article>
    <canvas id="origin" width="320" height="180" style="display:none"> </canvas>
  </article>

        ...
</section>
```

Example 1 : Video compression with auto-refresh

Input

```
<section>
   ...
  <table>
    <tr> <td> <h5> Number of modified blocks </h5>
         <td> <span id="nbModified"> 0 </span>
         </td>
    </tr>
    <tr> <td> <h5> Compression rate </h5>
         <td> <span id="compressionRate"> 0 </span>
         <td> <h5> % </h5>
         </td>
    </tr>
    <tr> <td> <h5> Threshold (with or without ROI) :  </h5>
         <td> <input id="threshold" type="text" value="0.03" maxlength="5"></input>
         </td>
    </tr>
  </table>
</section>
```

Example 1 : Video compression with auto-refresh

Elements of the script

(1) Description of all the objects used

(2) Function when the web page is loaded

(3) Function for reading the threshold given bu the users

(4) Functions associated to the video

Example 1 : Video compression with auto-refresh

Description of all the objects used

```
// Canvas and context
var video;    // Video
var orig;     // Video without compression
var origCtx;  // Context of video without compression
var comp;      // Compressed video
var compCtx;   // Context of compressed video
// Size of video
var videoWidth;
var videoHeight;
// Size of blocks
var blockSize = 16;
// Number of blocks on the width
var Wblock;
// Number of blocks on the height
var Hblock;
// Threshold in order to decide the copy or not
var threshold = 0;
// Components dynamically changed on the web page
var nbModified;
var compressionElement;
// Rate of compression
var compression = 0;
var totalModified = 0;
var totalFrame = 0;
```

Example 1 : Video compression with auto-refresh

Function when the web page is loaded

```
// Function at the page loading
function doLoad() {
        // Components of the video
        video = document.getElementById("video");
        videoWidth = video.width;
        videoHeight = video.height;
        // Initialization of Wblock et Hblock
        Wblock = Math.floor(videoWidth/blockSize);
        Hblock = Math.floor(videoHeight/blockSize);
        // Context of original video
        orig = document.getElementById("origin");
        origCtx = orig.getContext("2d");
        // Contexte of compressed video
        comp = document.getElementById("compression");
        compCtx = comp.getContext("2d");
        // Initialization of canvas with drawImage
        origCtx.drawImage(video,0,0,videoWidth,videoHeight);
        compCtx.drawImage(orig,0,0,videoWidth,videoHeight,0,0,videoWidth,videoHeight);
        // Values of dynamic elements of web page
        nbModified = document.getElementById("nbModified");
        compressionElement = document.getElementById("compressionRate");
        thresholdElement = document.getElementById("threshold");
        // Action associated to any change of threshold
        thresholdElement.addEventListener("change", handlethreshold,false);
        threshold=thresholdElement.value;
        // Action associated to the video
        video.addEventListener("play", timerCallback, false);
}
```

Example 1 : Video compression with auto-refresh

Function when the web page is loaded

```
// Function at the page loading
function doLoad() {
        // Components of the video
        video = document.getElementById("video");
        videoWidth = video.width;
        videoHeight = video.height;
        // Initialization of Wblock et Hblock
        Wblock = Math.floor(videoWidth/blockSize);
        Hblock = Math.floor(videoHeight/blockSize);
        // Context of original video
        orig = document.getElementById("origin");
        origCtx = orig.getContext("2d");
        // Contexte of compressed video
        comp = document.getElementById("compression");
        compCtx = comp.getContext("2d");
        // Initialization of canvas with drawImage
        origCtx.drawImage(video,0,0,videoWidth,videoHeight);
        compCtx.drawImage(orig,0,0,videoWidth,videoHeight,0,0,videoWidth,videoHeight);
        // Values of dynamic elements of web page
        nbModified = document.getElementById("nbModified");
        compressionElement = document.getElementById("compressionRate");
        thresholdElement = document.getElementById("threshold");
        // Action associated to any change of threshold
        thresholdElement.addEventListener("change", handlethreshold,false);
        threshold=thresholdElement.value;
        // Action associated to the video
        video.addEventListener("play", timerCallback, false);
}
```

Example 1 : Video compression with auto-refresh

Function when the web page is loaded

```
// Function at the page loading
function doLoad() {
        // Components of the video
        video = document.getElementById("video");
        videoWidth = video.width;
        videoHeight = video.height;
        // Initialization of Wblock et Hblock
        Wblock = Math.floor(videoWidth/blockSize);
        Hblock = Math.floor(videoHeight/blockSize);
        // Context of original video
        orig = document.getElementById("origin");
        origCtx = orig.getContext("2d");
        // Contexte of compressed video
        comp = document.getElementById("compression");
        compCtx = comp.getContext("2d");
        // Initialization of canvas with drawImage
        origCtx.drawImage(video,0,0,videoWidth,videoHeight);
        compCtx.drawImage(orig,0,0,videoWidth,videoHeight,0,0,videoWidth,videoHeight);
        // Values of dynamic elements of web page
        nbModified = document.getElementById("nbModified");
        compressionElement = document.getElementById("compressionRate");
        thresholdElement = document.getElementById("threshold");
        // Action associated to any change of threshold
        thresholdElement.addEventListener("change", handlethreshold,false);
        threshold=thresholdElement.value;
        // Action associated to the video
        video.addEventListener("play", timerCallback, false);
}
```

Example 1 : Video compression with auto-refresh

Function when the web page is loaded

```
// Function at the page loading
function doLoad() {
        // Components of the video
        video = document.getElementById("video");
        videoWidth = video.width;
        videoHeight = video.height;
        // Initialization of Wblock et Hblock
        Wblock = Math.floor(videoWidth/blockSize);
        Hblock = Math.floor(videoHeight/blockSize);
        // Context of original video
        orig = document.getElementById("origin");
        origCtx = orig.getContext("2d");
        // Contexte of compressed video
        comp = document.getElementById("compression");
        compCtx = comp.getContext("2d");
        // Initialization of canvas with drawImage
        origCtx.drawImage(video,0,0,videoWidth,videoHeight);
        compCtx.drawImage(orig,0,0,videoWidth,videoHeight,0,0,videoWidth,videoHeight);
        // Values of dynamic elements of web page
        nbModified = document.getElementById("nbModified");
        compressionElement = document.getElementById("compressionRate");
        thresholdElement = document.getElementById("threshold");
        // Action associated to any change of threshold
        thresholdElement.addEventListener("change", handlethreshold,false);
        threshold=thresholdElement.value;
        // Action associated to the video
        video.addEventListener("play", timerCallback, false);
}
```

Example 1 : Video compression with auto-refresh

What is important

- Get the context in a canvas
- Server is needed for using some functions like `getImageData`
- `AddEventListener` for each interactions
- A function for each interactions

Example 1 : Video compression with auto-refresh

Function for reading the threshold given by the users

```
// Action associated to any change of threshold
function handlethreshold() {
        threshold = thresholdElement.value;
}
```

Example 1 : Video compression with auto-refresh

Functions associated to the video

(1) For the frequency of updating

(2) For choosing the blocks to update

Example 1 : Video compression with auto-refresh

Functions associated to the video: timer

```
// Function for the frequency of the updating
function timerCallback() {
        if (video.paused || video.ended) {
                return;
        }
        // Refreshing video
        computeFrame();
        // timerCallback each 40 milliseconds
        setTimeout(timerCallback, 40);
 }
```

Example 1 : Video compression with auto-refresh

Functions associated to the video: the compression !

```
function computeFrame() {
        // For validation : display of the original video
        origCtx.drawImage(video,0,0,videoWidth,videoHeight);
        // For the number of modified blocks
        var bm = 0;
        // For each block : To decide which one has to be refreshed
        for (ligne=0;ligne<Wblock;ligne++) {
                for (colonne=0;colonne<Hblock;colonne++) {
                        // Block of the non-compressed video
                        var frame1 = origCtx.getImageData(blockSize*ligne,blockSize*colonne,blockSize,blockSize);
                        // Block of the compressed video
                        var frame2 = compCtx.getImageData(blockSize*ligne,blockSize*colonne,blockSize,blockSize);
                        // Size of the bloc
                        var l = frame1.data.length;
                        // Mean of differences between the two blocks
                        sum = 0;
                        for (i=0;i<l;i++) {
                                sum += Math.abs(frame1.data[i]-frame2.data[i]);
                        }
                        sum = sum/(l*255);
                        if (sum>=threshold) {
                                        // Updating only if necessary
                                        compCtx.putImageData(frame1,blockSize*ligne,blockSize*colonne);
                                        bm++;
                                }
                }
        }
```

Example 1 : Video compression with auto-refresh

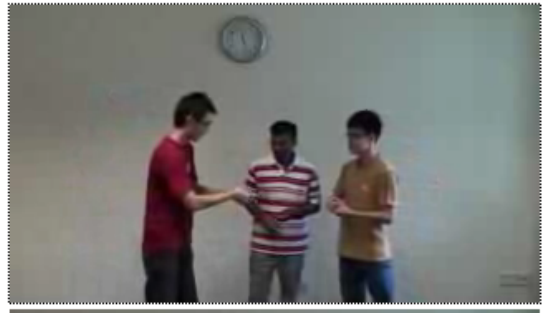
Functions associated to the video: update of the rate compression

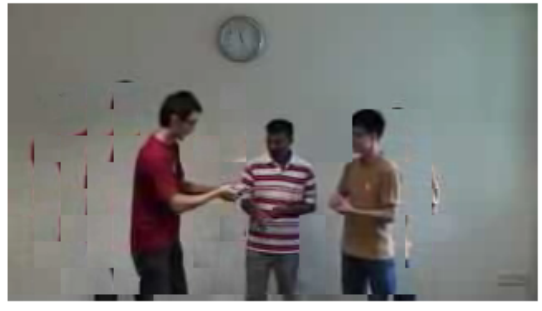
```
// Updating the number of modified blocs
nbModified.innerHTML = bm;
totalModified = totalModified+bm;
totalFrame++;

// Updating the compression rate
compression = 100*totalModified/(totalFrame*Wblock*Hblock);
compressionElement.innerHTML = compression;
```

Example 1 : Video compression with auto-refresh

Behavior

Compression by ROI coding





Number of modified blocks 68

Compression rate 13.293226381461675 %

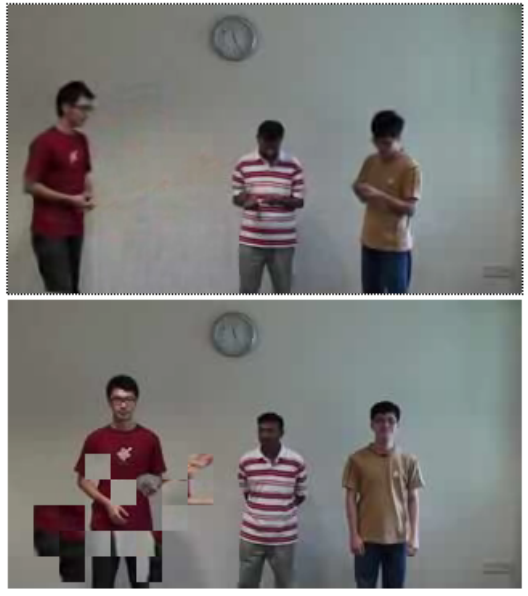
Threshold (with or without ROI) : 0.03

with reasonable threshold

Example 1 : Video compression with auto-refresh

Behavior

Compression by ROI coding



Number of modified blocks 3

Compression rate 0.19592476489028213 **%**

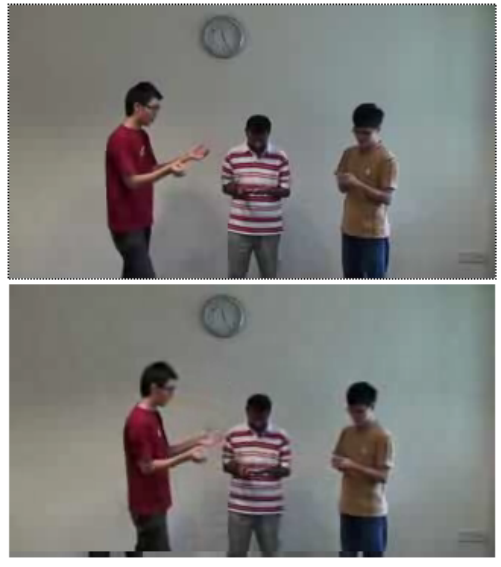
Threshold (with or without ROI) : 0.3

with permissive threshold

Example 1 : Video compression with auto-refresh

Behavior

Compression by ROI coding



Number of modified blocks 122

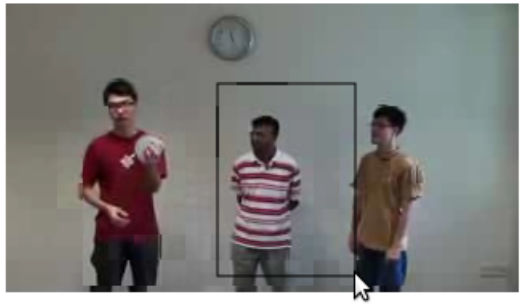
Compression rate 43.3 **%**

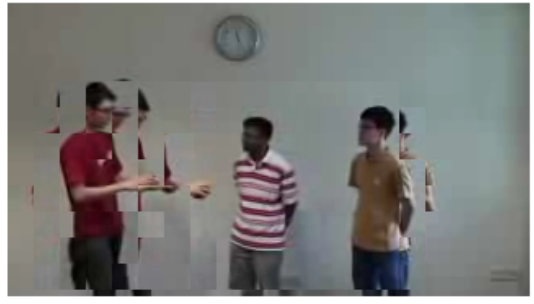
Threshold (with or without ROI) : 0.001

with strict threshold

Example 2: Video compression with Region Of Interest

- **Hypothesis:** Some parts of the video are more interesting then others
- **Selection of these parts:** By interaction/by users





Example 2: Video compression with Region Of Interest

Which elements/parts have to be adapted

- **HTML:** nothing !
- **Javascript**
  - **To add**
    (i) Elements manipulated: positions of the mouse, size of the ROI
    (ii) Functions about the interactions with the mouse
  - **To change/to complete**
    (i) `Doload` function: `addeventlistener` on the mouse
    (ii) `computeFrame` function: Now, refreshing is also conditioned by the position of the ROI

Example 2: Video compression with Region Of Interest

Elements of the script

```
// Events and thresholds for the ROI
var selecting = false;
var seuilROI = 0;
var seuilOutROI = 0.08;

// Position of the mouse
var tlpX=0;
var tlpY=0;

// Size of the ROI
var width=0;
var height=0;

// Boolean to indicate if ROI is selected
var ROIselected = false;
// Coordinates of ROI
var bXmin;
var bYmin;
var bXmax;
var bYmax;
```

Example 2: Video compression with Region Of Interest

Changes in `doload`

```
// Action associated to the video
video.addEventListener("play", timerCallback, false);
// New :
// Actions associated to the mouse events
comp.addEventListener("mousedown", dragStart, false);
comp.addEventListener("mouseup", dragStop, false);
comp.addEventListener("mousemove",dragging, false);
comp.addEventListener("click",stopROI,false);
```

Example 2: Video compression with Region Of Interest

Functions for the selection of the ROI

```
// Stop the selection : click
function stopROI() {
        // Nothing to do !
}
// Starting the selection : mousedown
function dragStart(event) {
        var tab = new Array(event.clientX,event.clientY);
        var mousePosition = getMousePosition(tab,comp);
        // Coordinates of the mouse
        tlpX = mousePosition[0];
        tlpY = mousePosition[1];
        // Boolean to indicate the start of the selection
        selecting = true;
}
// End of the selction : mouseup
function dragStop(event) {
        //  Boolean to indicate the end of the selection
        selecting = false;
        // Computing the coordinates of the ROI
        bXmin = Math.floor(tlpX/blockSize);
        bYmin = Math.floor(tlpY/blockSize);
        bXmax = Math.ceil((tlpX+width)/blockSize);
        bYmax = Math.ceil((tlpY+height)/blockSize);
        // To indicate that the ROI is selected
        ROIselected = true;
        // Re-Initialisation of the ROI
        tlpX = 0;       tlpY = 0;       width = 0;      height = 0;
}
```

Example 2: Video compression with Region Of Interest

Functions for the selection of the ROI

```
// Action associated to the movement of the mouse: mousemove
function dragging(event) {
        if (selecting==true) {
                var tab = new Array(event.clientX,event.clientY);
                var mousePosition = getMousePosition(tab,comp);

                // Calcul de la largeur et la hauteur de la ROI
                width = mousePosition[0]-tlpX;
                height = mousePosition[1]-tlpY;
        }
}
```

Example 2: Video compression with Region Of Interest

Changes in `computeFrame`

Inside the double loop, after the computation of the mean of the differences

```
// NEW : Ajusting the refreshing if ROI is selected
if (ROIselected) { // ROI has been selected
        if (ligne>=bXmin && ligne<=bXmax && colonne>=bYmin && colonne<=bYmax) {
                // inside ROI
                if (sum>=seuilROI) {
                        // Update (if necessary)
                        compCtx.putImageData(frame1,blockSize*ligne,blockSize*colonne);
                        bm++;
                }
                // else no updating
        }
        else {
                // outside ROI
                if (sum>=seuilOutROI) {
                        // Update (if necessary)
                        compCtx.putImageData(frame1,blockSize*ligne,blockSize*colonne);
                        bm++;
                }
                // else no updating
        }
}
else {  // No ROI has been selected (initial case)
        if (sum>=seuil) {
                // Update (if necessary)
                compCtx.putImageData(frame1,blockSize*ligne,blockSize*colonne);
                bm++;
        }
}
```

Example 2: Video compression with Region Of Interest

Changes in `computeFrame`

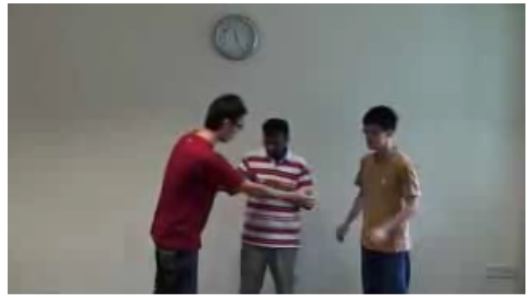
After the double loop, in order to display the current ROI

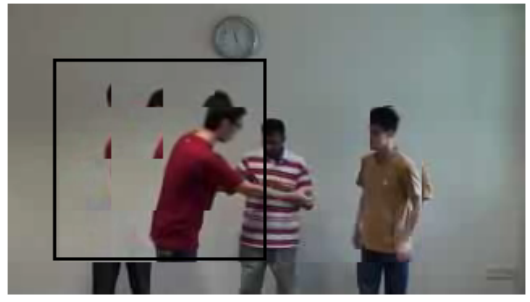
```
// Draw ROI
compCtx.fillStyle = "black";
compCtx.strokeRect(tlpX,tlpY,width,height);

// Updating the number of modified blocs
nbModified.innerHTML = bm;
totalModified = totalModified+bm;
totalFrame++;

// Updating the compression rate
compression = 100*totalModified/(totalFrame*Wblock*Hblock);
compressionElement.innerHTML = compression;
```

Example 2: Video compression with Region Of Interest

Behavior

Compression by ROI coding





Number of modified blocks 11

Compression rate 10.150537634408602 **%**

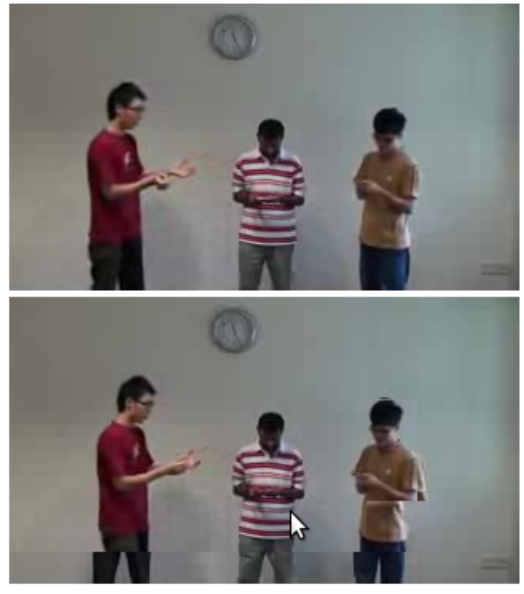
Threshold (with or without ROI) : 0.03

Selection of ROI

Example 2: Video compression with Region Of Interest

Behavior

Compression by ROI coding



Number of modified blocks 36

Compression rate 18.46728971962617 **%**

Threshold (with or without ROI) : 0.03

Refreshing

Thank you for your attention

Questions ?

Appendix

Functions for position of the mouse in canvas

```
////// Two functions for computing the position of the mouse in a canvas
// Return the position of the mouse in the element "obj"
function getMousePosition(coord,obj) {
        var scroll = new Array((document.documentElement && document.documentElement.scrollLeft) || window.pageXOffset ||
self.pageXOffset || document.body.scrollLeft,
                                                (document.documentElement && document.documentElement.scrollTop) ||
window.pageYOffset || self.pageYOffset || document.body.scrollTop);;
        var offset = findPosition(obj);
        var mouseVals= Array(coord[0] + scroll[0] - document.body.clientLeft -offset[0],coord[1] + scroll[1] -
document.body.clientTop-offset[1]);
        return mouseVals;
}

function findPosition(obj) {
        var curleft = curtop = 0;
        if (obj.offsetParent) {
                do {
                        curleft += obj.offsetLeft;
                        curtop += obj.offsetTop;
                } while (obj = obj.offsetParent);
        }
        return [curleft,curtop];
}
```