

Trifocal tensor

Tifaine CAUMONT and Elisa PRANA

January 3, 2013

Contents

- 0.1 List of the elements 2
- 0.2 Description of the elements encoded 2
 - 0.2.1 Tensor 2
 - 0.2.2 Transfert 3
 - 0.2.3 Saving the clicked points 4
 - 0.2.4 Save or load in a file 4

0.1 List of the elements

Display the help in english with -h : element requested, encoded and working.

Launch into the consol with the name of the three pictures to load : element requested, encoded and working.

Launch into the consol with the name of the three lists to load : element requested, encoded and working.

Saving a list of clicked points : element requested, encoded and working.

Calculation of a tensor : element requested, encoded and working.

Transfert the points with the tensor : element requested, encoded and working.

Launch into the consol with the path of the repository in which save the lists : element not requested, encoded and working.

0.2 Description of the elements encoded

0.2.1 Tensor

In order to build the tensor, we found the 27 equations derivated from this one :

$$\sum_{k=1}^3 x_p^k (x_p'^i x_p''^3 T_k^{3l} - x_p'^3 x_p''^3 T_k^{il} - x_p'^i x_p''^l T_k^{33} + x_p'^3 x_p''^l T_k^{i3})$$

Let's call A_k^{il} the coefficient of the T_k^{il} elements in all these equations. We noticed that there were only four coefficients for one equation to find, the others were then equal to zero.

A row of the matrix A is $[A_1^{il}, A_2^{il}, A_3^{il}]$. There is 28 rows since there are four equations for a point, and seven points. We just need to choose carefully the subscripts.

Here is the pseudocode of the fullfilling of the A matrix :
FOR p from 0 to 6

 FOR i from 0 to 1

 FOR l from 0 to 1

 FOR k from 0 to 2

$$\begin{aligned} A[4p+2i+1, 9k+3i+1] &= -x_p^k x_p'^3 x_p''^3 \\ A[4p+2i+1, 9k+3i+2] &= x_p^k x_p'^3 x_p''^l \end{aligned}$$

```

A[4p+2i+1, 9k+6+1]=  $x_p^k x_p'^i x_p''^3$ 
A[4p+2i+1, 9k+8]=  $-x_p^k x_p'^i x_p''^1$ 
    increment k of 1
  increment l of 1 END FOR
  increment i of 1 END FOR
END FOR

increment p of 1
END FOR

```

Then, we use the SVD of Eigen Library to decompose A. t is the last column of the V matrix.

We don't forget to put the elements of t in the real tensor T. And then we have a functional tensor.

0.2.2 Transfert

The transfert was a difficult part of the project. First, we focus on the transfert for the first image.

Transfert on the first picture

The main difficulty was to transform the equations into matrices and vectors. At the beginning, we tried to use the SVD one the same way that for the tensor since we didn't see that it was an equation like $Bx = b$. In order to simplify the calculations, we decided to put all the third coordinates to 1. So the matrix B was a 4x2 matrix. To fulfill the B matrix, we used the same loop than to fulfill A, removing the one with p and the one with k. Thus, in order find the eighth point of the first image, we had this :

```

FOR i from 0 to 1

```

```

  FOR l from 0 to 1

```

$$\begin{aligned}
 B[2i+j, 0] &= list2(7, i)T_2^{j0} - T_i^{j0} - list2(7, i)list3(7, j)T_2^{20} + list3(7, j)T_i^{20} \\
 B[2i+j, 1] &= list2(7, i)T_2^{j1} - T_i^{j1} - list2(7, i)list3(7, j)T_2^{21} + list3(7, j)T_i^{21}
 \end{aligned}$$

```

    increment l of 1
  END FOR

```

```

increment i of 1
END FOR

```

We calculate the b vector in the same loop and apply the method solve of the Eigen Library.

Transfert on the second and third picture

We take time to understand the fact that the column of the matrix B were not depending on k anymore, but on i for the second picture and l for the third. Once understood, the method is the same unless that the k loop is added and so we have to use $+=$ in order to sum the elements.

Once working for the three pictures, we put conditions on the click event to calculate the right transfert and change of 7 on the number of rows - 1 of the list.

0.2.3 Saving the clicked points

At first, the list were lists of 7 points and when clicking, the first point of the list was changed. It was a great idea. Then we add an eighth point, calculated with the transfert in order to keep the list of the seven points needed to build the tensor. The problem was that we did not save all the points and we were writting in the files with `std::ifstream`. So we changed the `saveMatrix` function in order to save as many points as wanted. But it still changed the initial file. We decided to save the point into `tmp/list.list` with a new method : `"j"`. A row is just added into the matrix and the file saved for each click. Finally, we created a function (`updateMatrix`) in order to do the resizing of the matrix and the saving in a file.

0.2.4 Save or load in a file

Sauvegarde/Chargement dans un fichier : - Au dpart, criture dans les fichiers initiaux : brouh pas bien - Sauvegarde dans les fichiers tmp - A faire : l'utilisateur peut choisir le fichier o sont enregistrs les points

Format de sauvegarde : activation du header en true dans `saveMatrix()`