IFT 6145 - Vision tridimensionnelle

TP1 - Panoramas

étudiant: Zhibin Lu

Partie 4: Programmation

In[ ]:=

```
cornersHomogene[im_, homo_] := Module[{w, h, c}, (
                                 |module
    {w, h} = ImageDimensions[im];
             |dimensions d'image
    c = {{0, 0}, {w, 0}, {w, h}, {0, h}};
    homo /@ c
   )];
e2p[v_] := Append[v, 1];
           |appose
p2e[v_] := v[[1 ;; -2]] / v[[-1]];

(* trouver Geometrique Transforme matrix
 de image1 à image2 par 4 points correspondants,
identique à FindGeometricTransform du Mathematica*)
             |trouve transformée géométrique
trouverGeometriqueTransforme[points2_, points1_] :=
 Module[{aa, bb, cc, dd, ee, ff, gg, hh, ii, k, h12, i2x, i2y, i1x, i1y,
 |module
   equations, coef, getCoefMatrix, coefMatrix, u, w, v, h12solve},
  h12 = {{aa, bb, cc}, {dd, ee, ff}, {gg, hh, ii}};
  (*homogene matrix de image1 à image2 *)
  (* equations=Cross[{i2x,i2y,1},h12.{i1x,i1y,1}] ;
                  |produit vectoriel
  i1 x h.i2=0 entre image i1 et i2, ca donne 3 eqations*)
  (* coef=Table[Coefficient[equations[[k]],Flatten[h12]],{k,1,3}];
           |table  |coefficient                    |aplatis
    obtenir les coefficient de h{a...i} vient des 3 eqations*)
  (* getCoefMatrix[{i2x_,i2y_},{i1x_,i1y_}]=coef[[1;;2]];
  on a besion just des coefficients de 2 equations*)
```

```
image1 =  ;
```

```
points1 =
  {{238.6625`, 362.903125`}, {505.10625000000005`, 278.69374999999997`},
   {399.21562500000005`, 216.03437499999995`}, {143.125`, 336.21875`}};
image2 =
```

;

```
points2 = {{497.2068655643418`, 427.35662293314175`},
   {494.41872978073286`, 24.682872034508023`}, {322.8682602444282`,
    27.37842604241598`}, {347.89382638389617`, 420.36313803019425`}};

(*obtenir homogene projective transforme matrix en utilisant mon fonction*)
h12iphone = trouverGeometriqueTransforme[points2, points1];
showImageAndPoints[image1, points1]
showImageAndPoints[image2, points2]
Print[" Compare H12 by myself and H12 by Mathematica, It's the same:"];
                                                    imprime
h12iphone / h12iphone[[3, 3]] // MatrixForm
                                                    apparence matricielle
{err, h12} = FindGeometricTransform[points21, points12]
                                    trouve transformée géométrique
ImagePerspectiveTransformation[
                                    transformée d'image inverse
  image1, h12iphone, DataRange → All, PlotRange → All]
                                    plage de donn··· tout  zone de tracé  tout
```

Out[•]=

Out[•]=

Compare H12 by myself and H12 by Mathematica, It's the same:

Out[•]//MatrixForm=

$$\begin{pmatrix} 0.999257 & -0.157585 & -294.527 \\ -0.322545 & 0.840966 & 106.762 \\ -0.000535732 & 5.68857\times10^{-6} & 1. \end{pmatrix}$$

Out[•]= $\{1.36279\times10^{-10}, \text{TransformationFunction}\left[\left(\begin{array}{cc|c} 0.999257 & -0.157585 & -294.527 \\ -0.322545 & 0.840966 & 106.762 \\ \hline -0.000535732 & 5.68857\times10^{-6} & 1. \end{array}\right)\right]\}$

Out[ ]=

Utiliser my function sur partie 2

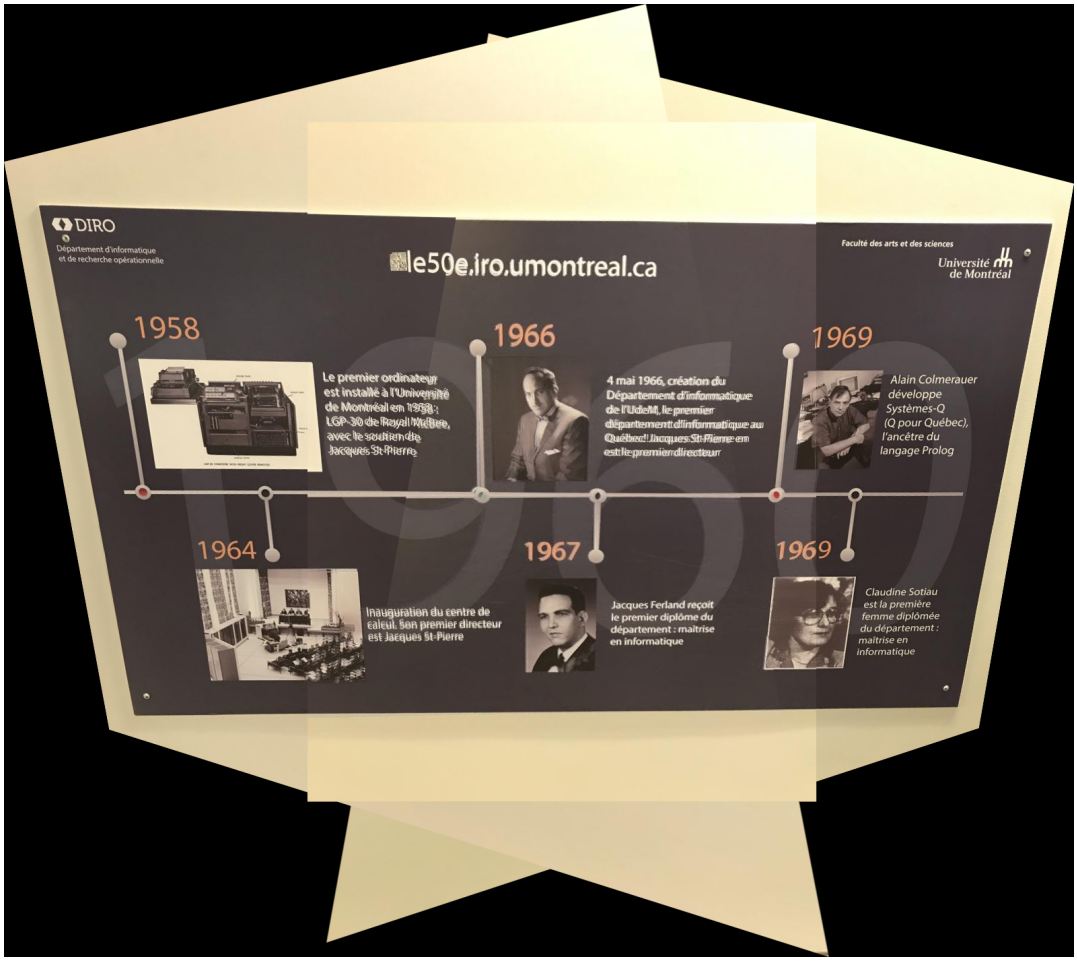
In[ ]:=

```
image1 = Import["/Users/louis/Google\ Drive/M.Sc-DIRO-UdeM/IFT6145-Vision\
         importe
    tridimensionnelle/TP1/part2-1.jpeg"];
image2 = Import["/Users/louis/Google\ Drive/M.Sc-DIRO-UdeM/IFT6145-Vision\
         importe
    tridimensionnelle/TP1/part2-2.jpeg"];
image3 = Import["/Users/louis/Google\ Drive/M.Sc-DIRO-UdeM/IFT6145-Vision\
         importe
     tridimensionnelle/TP1/part2-3.jpeg"];
points12 = {{718.7601753287408`, 336.77582968065155`},
   {714.8389480275512`, 606.7803381340015`}, {513.0338134001249`,
    766.9431433938637`}, {392.93537883531604`, 193.80488415779632`}};
points21 = {{600.8372399868053`, 256.37535859580396`},
   {522.4283957898101`, 622.8550923103066`}, {133.33130753775873`,
    803.6276800975581`}, {85.47440600541762`, 180.8834201293446`}};
```

```
points23 = {{679.661705466649`, 197.28156892536242`},
    {716.2212648561115`, 668.2409363973493`}, {248.01255460152123`,
     672.1628899573186`}, {227.8587007586737`, 244.95710844336963`}};
points32 = {{385.50721025458455`, 204.36959230906132`},
    {345.06996617411454`, 674.116076197258`}, {31.965461990386416`,
     659.3783158269537`}, {62.12141712658013`, 324.305501157201`}};

(*obtenir homogene projective transforme matrix en utilisant mon fonction*)
h12 = TransformationFunction[trouverGeometriqueTransforme[points21, points12]];
      fonction de transformation
h22 = TransformationFunction[trouverGeometriqueTransforme[points21, points21]];
      fonction de transformation
h32 = TransformationFunction[trouverGeometriqueTransforme[points23, points32]];
      fonction de transformation


allCorners =
  MapThread[cornersHomogene, {{image1, image2, image3}, {h12, h22, h32}}];
  applique en enfilade
allCorners = Flatten[allCorners, {{3}, {1, 2}}];
                aplatis
range = MinMax /@ allCorners;
        minimum et maximum


image12 = ImagePerspectiveTransformation[
          transformée d'image inverse
    image1, h12, DataRange → All, PlotRange → range];
                 plage de donn··· tout  zone de tracé
image22 = ImagePerspectiveTransformation[image2, h22,
          transformée d'image inverse
    DataRange → All, PlotRange → range];
    plage de donn··· tout  zone de tracé
image32 = ImagePerspectiveTransformation[image3, h32,
          transformée d'image inverse
    DataRange → All, PlotRange → range];
    plage de donn··· tout  zone de tracé
imageData = Flatten[ImageData /@ {image12, image22, image32},
            aplatis      données d'image
    {{2}, {3}, {4}, {1}}];
Print["L'image composé entre 1,2,3 de la murale:"]
imprime
Image[Map[Max, imageData, {3}]]
image  app··maximum
```

```
L'image composé entre 1,2,3 de la murale:
```

Out[ ]=



Utiliser my function sur partie 3

In[ ]:=

```
image2 = Import["/Users/louis/Google\ Drive/M.Sc-DIRO-UdeM/IFT6145-Vision\
         importe
      tridimensionnelle/TP1/part3-2.jpeg"];
image3 = Import["/Users/louis/Google\ Drive/M.Sc-DIRO-UdeM/IFT6145-Vision\
         importe
     tridimensionnelle/TP1/part3-3.jpeg"];
image4 = Import["/Users/louis/Google\ Drive/M.Sc-DIRO-UdeM/IFT6145-Vision\
         importe
     tridimensionnelle/TP1/part3-4.jpeg"];

points23 = {{969.5255474452555`, 564.5786699107866`},
    {816.670214922952`, 68.59276155717748`}, {582.6414233576642`,
     111.9027777777776`}, {527.8634428223844`, 632.2190794809408`}};
points32 = {{471.0239253852392`, 530.9341038118409`},
    {330.70813057583126`, 77.13341443633408`}, {83.58272506082724`,
     81.34874290348728`}, {32.899432278994325`, 673.4036901865368`}};
points34 = {{987.4477899432278`, 658.8274533657744`},
```

```
    {946.1496350364963`, 206.33252230332505`}, {526.1744728304947`,
     211.72942011354405`}, {480.8064679643146`, 703.8335360908353`}};
points43 = {{573.1746755879967`, 643.0590024330907`},
    {547.2404703974047`, 248.8903081914034`}, {149.24320762368205`,
     207.21603811841078`}, {76.73104217356041`, 752.4091646390923`}};


(*get the transform matrix of homographie,
from image1 to image2, form image3 to image1, etc*)
(*image3 is the reference*)
{i1w, i1h} = ImageDimensions[image2];
              dimensions d'image

(*obtenir homogene projective transforme matrix en utilisant mon fonction*)
h23 = TransformationFunction[trouverGeometriqueTransforme[points32, points23]];
      fonction de transformation
h33 = TransformationFunction[trouverGeometriqueTransforme[points32, points32]];
      fonction de transformation
h43 = TransformationFunction[trouverGeometriqueTransforme[points34, points43]];
      fonction de transformation
allCorners =
  MapThread[cornersHomogene, {{image2, image3, image4}, {h23, h33, h43}}];
  applique en enfilade
allCorners = Flatten[allCorners, {{3}, {1, 2}}];
              aplatis
range = MinMax /@ allCorners;
        minimum et maximum


image23 = ImagePerspectiveTransformation[
           transformée d'image inverse
    image2, h23, DataRange → All, PlotRange → range];
                  plage de donn··· tout  zone de tracé
image33 = ImagePerspectiveTransformation[image3, h33,
           transformée d'image inverse
    DataRange → All, PlotRange → range];
    plage de donn··· tout  zone de tracé
image43 = ImagePerspectiveTransformation[image4, h43,
           transformée d'image inverse
    DataRange → All, PlotRange → range];
    plage de donn··· tout  zone de tracé
imageData = Flatten[ImageData /@ {image23, image33, image43},
             aplatis      données d'image
    {{2}, {3}, {4}, {1}}];
Print["L'image composé entre 1,2,3 de la murale:"]
imprime
Image[Map[Max, imageData, {3}]]
image  app··  maximum


L'image composé entre 1,2,3 de la murale:
```

Out[ ]=