

TP - Suivi 3D pour la réalité augmentée

Algorithmes KLT et Homographie

1. Objectifs

Le but de ce TP est d'étudier la chaîne de traitement nécessaire pour réaliser un suivi 3D d'objets dans une séquence d'images afin de mettre en œuvre une application de réalité augmentée. Nous verrons notamment :

- L'algorithme KLT pour le suivi de points 2D
- La combinaison du KLT avec une estimation de l'homographie pour le suivi basé marqueur.

2. Algorithme KLT

La méthode KLT (Kanade-Lucas-Tomasi) permet de suivre des points d'intérêt dans des images successives. Elle suppose que le flot est essentiellement constant dans un voisinage local du point considéré, et résout l'équation du flot optique pour tous les pixels dans ce voisinage par la méthode des moindres carrés.

La fonction Matlab qui met en œuvre l'algorithme KLT est : ***vision.PointTracker***

L'exemple de l'annexe A utilise l'algorithme KLT pour réaliser un suivi du visage dans une séquence vidéo.

Analyse du code de l'annexe A

- Quelle est la fonction Matlab utilisée pour la détection des points caractéristiques dans la région d'intérêt. Expliquer son principe.
- Remplacer cette fonction par les détecteurs suivants : Harris, Surf et Orb. Exécuter à chaque fois le code et comparer les résultats.
- Expliquer les différents étapes d'utilisation de la fonction `vision.PointTracker` (création, initialisation et tracking)
- Adapter le code afin de réaliser un suivi 2D-2D des 4 coins de la boîte de la figure 1, à partir de la séquence « `box_video_data.avi` » disponible dans le package du TD.



Figure 1

3. Suivi 3D : KLT + Homographie

Nous souhaitons réaliser un suivi 3D de la boîte de la figure 1 à partir de la séquence vidéo « `box_video_data.avi` ». Pour cela, nous proposons de combiner la méthode KLT avec une estimation de pose 3D basée sur l'homographie. Le KLT sera utilisé pour le suivi 2D-2D des 4 coins de la cible. Le résultat permettra de générer une séquence en réalité augmentée qui superpose de manière continue le modèle 3D de la boîte sur son image. Le modèle 3D de la boîte est donné en annexe 1.

Le dossier **\TP_KLT_Homographie_Package** contient les données dont vous aurez besoin pour réaliser cette partie du TP :

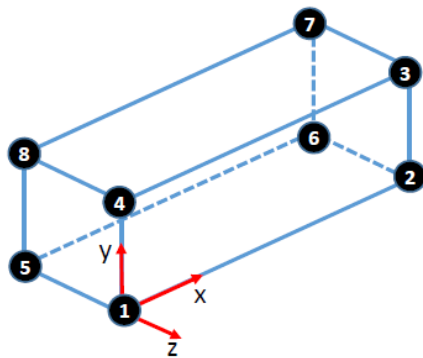
- **`box_video_data.avi`** : une séquence vidéo prises par une caméra en mouvement observant la boîte
- **`ComputeH.m`** : une fonction matlab qui permet d'estimer l'homographie à partir d'un appariement de points 2D-3D

- **Project3DModel** : une fonction matlab qui permet de projeter le modèle 3D de la boîte sur l'image courante.
1. Compléter le script *squelette_Homographie_Sequence.m* qui permet de réaliser la chaîne de traitement complète pour donner en sortie la séquence vidéo augmentée.
 2. Calculer pour chaque frame, l'erreur quadratique moyenne (EQM) entre les quatre points images et les projetés des points 3D qui leur correspondent. Afficher son évolution en fonction des frames, que constatez-vous ?

Rendu du TP :

1. Le code *squelette_Homographie_Sequence.m* complété et commenté
2. Un rapport sur l'analyse et les performance du suivi avec la méthode KLT-Homographie
3. Une vidéo augmentée

Annexe 1 : Modèle 3D de la boîte



```

*****
% Modèle 3D de la boîte. Les coordonnées des
% coins de la boîte sont définis dans un
% repère local de la boîte
*****

cube3d = [0          0          0
          0.125      0          0
          0.125      0.090      0
          0           0.090      0

          0           0         -0.070
          0.125      0         -0.070
          0.125      0.090     -0.070
          0           0.090     -0.070 ];

```

Annexe 2 : Suivi de visage dans une séquence vidéo

```
%% Track a Face in Scene
% Create System objects for reading and displaying video and for drawing a
% bounding box of the object.

videoReader = VideoReader('visionface.avi');
videoPlayer = vision.VideoPlayer('Position',[100,100,680,520]);
%%
% Read the first video frame, which contains the object, define the region.
%%
objectFrame = readFrame(videoReader);
objectRegion = [264,122,93,93];
%%
% As an alternative, you can use the following commands to select the object
% region using a mouse. The object must occupy the majority of the region:
%
% figure; imshow(objectFrame);
% objectRegion=round(getPosition(imrect))
%
% Show initial frame with a red bounding box.
%%
objectImage = insertShape(objectFrame,'Rectangle',objectRegion,'Color','red');
figure;
imshow(objectImage);
title('Red box shows object region');
%%
% Detect interest points in the object region.
%%
points = detectMinEigenFeatures(rgb2gray(objectFrame),'ROI',objectRegion);
%%
% Display the detected points.
%%
pointImage = insertMarker(objectFrame,points.Location,'+','Color','white');
figure;
imshow(pointImage);
title('Detected interest points');
%%
% Create a tracker object.
%%
tracker = vision.PointTracker('MaxBidirectionalError',1);
%%
% Initialize the tracker.
%%
initialize(tracker,points.Location,objectFrame);
%%
% Read, track, display points, and results in each video frame.
%%
while hasFrame(videoReader)
    frame = readFrame(videoReader);
    [points,validity] = tracker(frame);
    out = insertMarker(frame,points(validity,:),'+');
    videoPlayer(out);
end
%%
% Release the video player.
%%
release(videoPlayer);
```