Estimation de mouvement

IMA 208

6 april 2020

1 Introduction

The goal of these practical works (PW) is to test several motion estimation (ME) technques, and to understand the trade-offs related to them.

We will use block-based ME, optical flow, direct and inderect parametric estimation.

Before starting the PW, you need the following :

- Install Matlab or Octave if not yet done
- Download the files from eCampus
- Create a working directory
- extract *all files* from the compressed files to the working directory
- launch Matlab or Octave and go into the created working directory

The report of this PW consists in the answers to the questions in **bold**. Please, upload the report in eCampus not later than 2 hours later than the end of the session

The script files can be executed in Matlab or Octave. Start by opening the script in the editor. Then you can execute the script. However, executing all the script could be annoying (long). Therefore the scripts are organized in *sections* starting by `%%` . Each section can be executed separately from the others, even though it can require that previous sections have already been executed.

Notice that **in Matlab** you can execute the current section (i.e., the section where the cursor is currently positioned) by typing `Ctrl + Enter`, while by typing `Shift + Ctrl + Enter`, the section is executed and the cursor is moved at the beginning of next section.

In Octave you cannot execute sections by `Ctrl + Enter`; you have to select the section in the editor and then press `F9`. Also notice that the graphical engine of Octave is a bit rougher than the one of Matlab. If sometimes the figure is created but nothing appear (in particular with the `displayMVC` function), try to "move" the content of the figure (by clicking on the icon of the arrowed cross, clicking on the figure and moving the mouse without releasing. the button). This should fix the bug.

2 Motion estimation techniques

2.1 Block matching

1. Open the `block_matching.m` file in the editor by typing `edit block_matching.m`. Load images from one of the videos `akiyo` or `flower`, using the the function `readFrame`
2. Compute the motion vector field using the SSD criterion with the function `me_ssd`.
3. **Is motion estimation able to find the real scene motion ? Are there errors ? If yes, where and why ?**

4. **Change the block size and the search area radius. What is the impact on the previous question ?**

5. Use the `fracMc` function to carry out motion compensation and show the resulting image.
6. PSNR (peak signal-to-noise ratio) allows to evaluate the quality of the prediction. The definition is

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\mathrm{MSE}}$$

where $\mathrm{MSE} = \frac{1}{MN} \sum_{m=1}^{M} \sum_{n=1}^{N} \left[f_k(n,m) - f_{\mathrm{MC}}(n,m) \right]^2$ and f_{MC} is the motion-compensated image. The hight the PSNR, the better the quality of the prediction. **Compute the PSNR of the MC-ed prediction with respect to the original image. Change the block size and comment the result.**

7. Re-do questions from 2 by using the SAD criterion now.
8. **Comment the difference observed between SSD and SAD in terms of regularity of MVF, PSNR of the prediction, impact of the block size.**
9. Load two images from `flower`.
10. Compute and display the MVF with the criterion SSD, without and with regularization ($\lambda = 20$).
11. **Comment the difference observed between SSD and regularized SSD in terms of regularity of MVF, PSNR of the prediction, impact of the block size.**

2.2 Optical flow

1. In Matlab or Octave, launch `edit optic_flow.m`
2. The optical flow (OF) is implemented by the iterative Horn and Schunk's algorithm :

$$u^{n+1} = \bar{u}^n - f_x \frac{\bar{u}^n f_x + \bar{v}^n f_y + f_t}{\alpha^2 + \|\nabla f\|^2}$$
$$v^{n+1} = \bar{v}^n - f_y \frac{\bar{u}^n f_x + \bar{v}^n f_y + f_t}{\alpha^2 + \|\nabla f\|^2}$$

where u^n is the value of the u component at n-th iteration, and $\bar{u}$ is a local average of u. This algorithm is implemented in `HS.m`

3. Use this algorithm to estimate the MVF in `akiyo` and `flower`. Block matching is used to initialize OF. Type `help HS.m` to understand the input parameters of the OF algorithm.
4. **Comment the result in terms of regularity of MVF, PSNR of the prediction.**

3 Parametric estimation

3.1 Direct parametric estimation

1. Let $a : (n,m) \in \mathbb{Z}^2 \to \mathbb{R}$ be a bi-dimensionnel signal and $b(n,m)$ a signal of the same type.
2. Remind the 2D Discrete Time Fourier Transform (DTFT) :

$$\hat{a}(\nu_x, \nu_y) = \sum_{n,m \in \mathbb{Z}^2} a(n,m) e^{-i2\pi(\nu_x n + \nu_y m)}$$

3. We suppose that
$$\exists c, d \in \mathbb{Z}^2 : \forall n, m \in \mathbb{Z}^2, b(n,m) = a(n+c, m+d)$$
that is, b is a translation of a **Compute the DTFT of b ias a function of the one of a and of the du displacement (c,d).**

4. **Show that the ratio between the DTFTs is :**
$$\frac{\hat{b}(\nu_x, \nu_y)}{\hat{a}(\nu_x, \nu_y)} = \exp\left[i2\pi(c\nu_x + d\nu_y)\right]$$

5. **Finite signals** Now, $a : (n,m) \in \{0, 1, \dots, N-1\}^2 \to \mathbb{R}$ and $b(n,m)$ are both finite signals. In that case we have the Discrete Fourier Transform (DFT) rather than the DTFT :
$$\hat{A}(k_x, k_y) = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} a(n,m) e^{-i2\pi \frac{k_x n + k_y m}{N^2}}$$
In which hypotheses the ratio between a's and b's DFT is a linear phase signal ?
$$\frac{\hat{A}(k_x, k_y)}{\hat{B}(k_x, k_y)} = \exp\left[\frac{i2\pi(ck_x + dk_y)}{N^2}\right]$$

6. Open `me_parametric.m`. Load the image `ball.bmp` and apply a pure translational motion with integer values.
7. Compute the DFT of the two images. Compute the phase of the ratio between the DFTs by using the function `angle2D`. Display the phase ϕ. **What is the behavior of ϕ ? Is it coherent with the theoretical result ? Why ?**
8. Method 1 for estimating the translation. The median slope of ϕ is used as estimation of the translation vector. Compute the gradient of ϕ and the medians of its components **What is the theoretical value of the gradiant ? Is it what you observe ? Comment**
9. Method 2. Least square approximation. We look for the LS plane fitting ϕ. The equation of the plane is $z = b_x f_x + b_y f_y$ (f_x and f_y are the spatial frequencies). We look for b_x and b_y minimizing the squared error between z and ϕ. We set also $\phi(0,0) = 0$, and we smooth ϕ with a median filter to remove some noise. The LS problem is solved by using the \ operator.
10. **Compare the results of the 2 methods and comment.**
11. **What happens if you add some noise to the image before translation ? You can do this by setting a non-nul value of `sigma` in line 6, for example, `sigma=2`.**

3.2 Indirect parametric estimation

1. Use block-matching estimation to find the motion vector field between the two images **without noise**, i.e., `sigma=0`.
2. Show the MVF and comment the result.
3. Use the median of MVF components to estimate the translation. **Comment the result** .

4. Redo the previous questions when there is some noise, `sigma=2`. **in the case of noise**. **Commentez and justifiez.**
5. Finally, compare all the parametric estimation methods when the translation values are **not integer**.
6. Optional. Redo the questions of this part with HS instead of block matching.
7. Use a more complex image, such as `lena.bmp`. Perform the translation, the direct and inderect estimation. Is it reasonable to have a circulatr translation in this case ? You can have a different translation if you remove the last parameter of `applyAffineMotion` at line 15. Comment and conclude.