T1. Create a denoised version of the 16th frame (i.e. noisy_ frames [15]) by naive temporal averaging of the pixels (you can use the function np.mean). Change the value of m (use 2,4 and 6) and *comment its impact on the results, identify the benefits and the pitfalls.*

Figure 1: m = 2; m=4; m=6

Observaciones

(a) El "temporal averaging of the pixels" es computacionalmente muy barato

(b) El "temporal averaging of the pixels" puede producir efectos indeseados en los objetos en movimiento, efecto fantasma. Mirar Figura 1.

(c) Notar como el efecto fantasma se incremento a medida que m aumenta. Seria necesario una compensacion del movimiento, (por ejemplo, calculando el optical flow) para reducir "artefactos" fantasma.

(d) PSNR: 27.88, 26.70, 25.68

(e) Otra desventaja es que la ventana de tiempo limita la primera parte vs la ultima parte del video perdiendo informacion.

(f) Si asumimos el ruido como un proceso estocastico gaussiano, el "temporal averaging of the pixels" reduciria la varianza del ruido en 2m+1. En situaciones ideales (objetos estaticos) y camara fija, podria ser util. Aunque preferiblemente usariamos otros metodos.

OPTIONAL. Denoise the whole video by temporal averaging.

T2. As a first step towards this second naive method you will write the code that takes the 16 th frame as target frame and computes the 2 previous and 2 posterior motion-compensated frames. Write the 5 frames as images and visualize them one after the other, you can also compute the difference of each frame with respect to the central (target) frame. Comment the results.

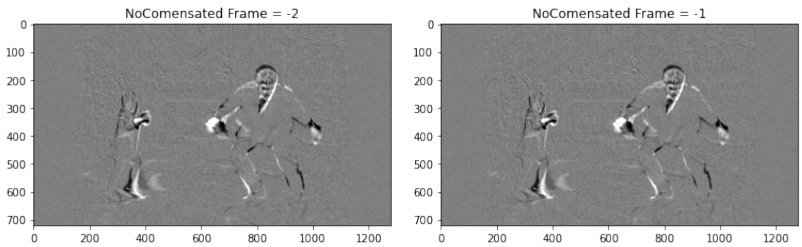


Figure 2: No compensated

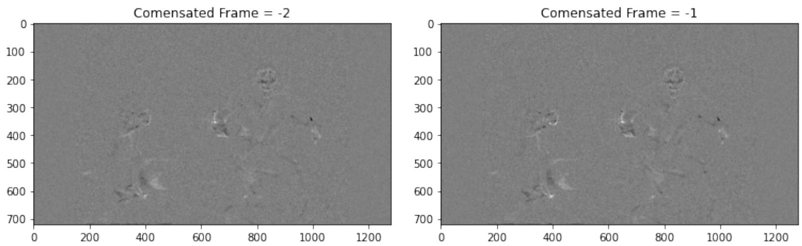


Figure 3: compensated

(a) La compensacion de las imagenes de los frames anteriores y posteriores se respecto a la imagen de referencia son ”mas compactas” respecto al movmiento de los objetos. Esto puede observarse comparando las regiones ocluidas.

(b) (Relacion apartado anterior)Las regiones ocluidas pueden resultar en la aparicion de artefactos fantasmas no deseados.

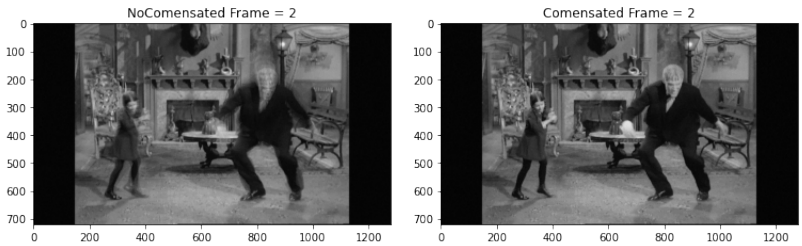
(c) (Relacion apartado anterior, apartado siguiente) Aplicar una compensacion en el movimiento (usando OF + warped) podemos reducir las regiones ocluidas y reduciendo las apariciones de artefactos fantasmas.

T3. Extend the previous code so as to denoise the frames 15th, 16th and 17th by temporal averaging in a temporal window of $2m + 1$ motion-compensated frames. Try different values of m and compare the result with respect to the first naive method.

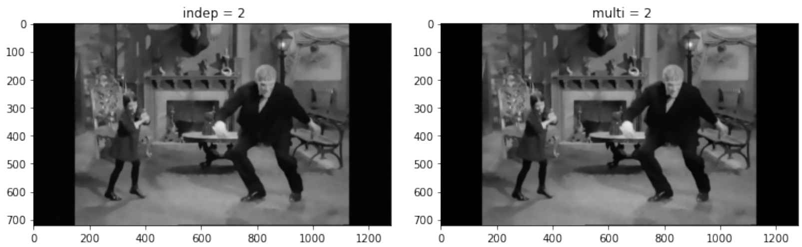
Observaciones + Propuesta

(a) Una tabla que resuma la información, calculando el psnr! + figura ejemplo

Estimated FRAME	Psnr (m = 2)	Psnr (m = 4)
14	28.71	VOS
14 Compensated	31.03	VOS
15	27.90	VOS
15 Compensated	30.93	VOS



1. T4. Compute the PSNR of the four denoised versions of frame 16th (obtained with the four different denoising methods). Compare the results and comment.



Observaciones + Propuestas

(a) Tabla + Figuras

(b) NL no necesita calcular optical flows o compensar la imagen. Self-similarity.

(c) Mejor que los casos anteriores.

(d) Resaltar puntos de las diapos

i. Intuitively, h_u is proportional to σ.

ii. Searching for similar patches is often constrained to a small neighborhood: $W \times W$.

iii. Averaging N observations of the noisy pixel reduces its noise variance by N.

T5. Study of the parameter h Finally, you will study the effect of the parameter h, i.e. the value in the denominator of the exponent/power of the exponential. In this link you can check the meaning of the different input variables to the video version of the Non-Local means algorithm (cv2. fastNlMeansDenoisingMulti). The goal is to focus only on denoising the frame 16 th. For that you will create a small video (subvideo) around frame 16 th, i.e. from frame $15 - 4$ to frame $15 + 4$.

Try and explore different values of the parameter $h(h = 20, 10, 8)$. Apply the algorithm to both the noisy subvideo and the warped noisy subvideo. Comment and compare the results, both qualitatively and quantitatively.

Observaciones + Propuestas

(a) Parámetro que regula la fuerza del filtro. Un valor h más grande elimina perfectamente el ruido pero también elimina los detalles de la imagen, un valor h más pequeño conserva los detalles pero también conserva algo de ruido

(b) Ejemplificarlo con psnr + tabla y/o figura

T6. Analyse the code above. Which type of 2D transformation are we applying to the frames? (Check the slides from lecture 9).

ver codigo

T7. Complete the function below. The part that needs to be completed is the one that computes h, the stabilizing homography for every frame. In order to do the Gaussian smoothing you can use the function above.

ver codigo

T8. Complete the function below that displays the non-stabilized and the stabilized trajectories of the central pixel (exercise 6 of seminar 4 may be helpful for that).

ver codigo

T9. Adapt the previous code to stabilize a given video. Stabilize the two videos provided (walking.mp4 and dancing.mp4) and submit them together with the code. You may reuse the functions used in the first part that read and write a video. Display the original and stabilized trajectories of the middle pixel in both cases (these plots should be displayed also in the report that you submit).

ver codigo

OPTIONAL. Write a new function crop_and_zoom that automatically estimates the parameters of the similarity function to apply to every frame (exercise 9 of seminar 4 and section 2.3 in paper [3]) .