OCCLUSION DETECTION

Helena Cots, Paula Castillo

ABSTRACT

The field of motion in videos leads to another specific field which is the occlusions that originate from this motion. In this lab what we want is to solve the problem of detecting occlusions between consecutive frames of a video. We follow a visual reconstruction method that takes into account the optical flow of the video frames and works at a pixel level. These two criterions work better than others based on intensities of the frames or block-matching methods, for example. The goal is to get the better estimated motion possible using the previous criterions.

Keywords - Occlusion detection, optical flow, Gaussian kernels, image reconstruction, Gaussian Mixture Model.

1. INTRODUCTION

One of the computer vision applications challenge is the occluded areas detection at each frame of a video sequence.

The aim of this challenge is to assess whether or not a pixel of an image can be explained by its spatial neighbours in the same frame and by appropriate pixels in the other. If it is not possible, it is expected to be occluded in the next image.

Occlusion detection is used in several research fields such as action recognition, object tracking in video processing, scene understanding, advanced driving assistance system, pose estimation and more.

One of the main problems of occlusion detection using optical flow is the limitations of this method as a representation of physical motion. When working with 2D images, we might define that one point of one image is occluded if it is not seen by the observer point of view. If the motion vectors of the optical flow are not well-defined, the occlusion method that takes the information from them will not be plausible.

Most of the state-of-the-art approaches understand the problem of occlusion detection strongly dependent of displacement estimation (this is why in this algorithm we will use optical flow). In this field, the occlusion detection methods will be correlated to the movement of pixels on images in order to be applied to the research areas named before.

In order to improve the algorithm, an improvement we can implement is the spatial and temporal image reconstruction, where these more exact formulas will be provided by reasonable motion vectors between the consecutive frames in the video sequence. Instead of trying to guess in some not accurately way, researches came to this method as one of the best on detecting occlusion.

2. PROPOSED ALGORITHM

In this section, we start from the concept that an occluded point is represented as a pixel in an image space as one that is visible in a first frame of the video sequence and not in a second one. This explanation reflects the problem of information loss: we cannot explain this first frame pixel by using the second frame.

Before getting into the details of the proposed algorithm, in Figure 1 we define the images I1 and I2 as two consecutives images of a video sequence.

I1*(above),* I2 *(below)*

Figure 1. **Pair of consecutive images**

We are given the optical flow motion vectors of the sequence of images from where we want to detect occlusions. Optical flow [9] can be defined as the motion understood as the luminance variations at pixel level. In order to explain the following steps, we define the motion vector between I1 and I2 as w.

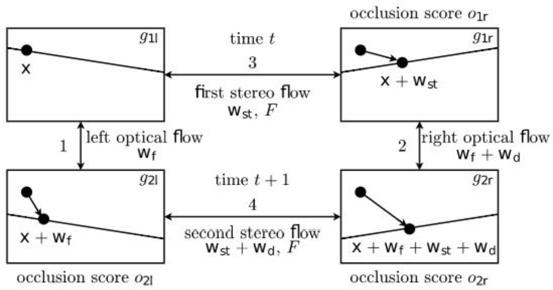

Figure 2. **Explanation of Optical Flow [10]**

As we mentioned before, our aim is to improve and use the best option possible to detect occlusions. In order to do this, we redefine the problem as a spatiotemporal reconstruction problem. Due to this definition, we identify two main issues. The first one is to evaluate the visual reconstruction with a reference information, which we will obtain from the same image I1 (spatial information with bilateral filter) (1). The second one, is that the reconstruction from I2 must be regular with apparent scene changes (we don't want to detect changes of brightness or apparent motion).

The first step we do in our algorithm is warping the image I1. This means getting a new I1 from the intensity values of I2 using the optical flow between the images. This image is going to be referred as I1' (Figure 3).

Figure 3. **Warped Image I1'**

In order to compute the two reconstructions named before, we will define two functions: $\xi(I1)$, which is the intra-image reconstruction, and $\eta(I1' ; I2, w)$, the inter-image reconstruction (reconstruction from I2 having I1').

The first one, $\xi(I1) = \xi_1$, is the inborn appearance of I1 based on each pixel **x** of the image domain. It is obtained by applying a bilateral filter [1][7] to the original image I1:

$$\xi_1(x) = \frac{1}{Z(x,I1)} \sum_{y \varepsilon Nx} \alpha(x,y;I1) I1(y) , \qquad (1)$$

where Nx is a square window (in our algorithm of size 5x5) centered at x, Z(x,I1) is just a normalization factor. The most important part of this filtering is the $\alpha(x,y;I1)$ function:

$$\alpha(x,y;I1) = fa(\|I1(y) - I1(x)\|) \cdot fs(\|y - x\|), \qquad (2)$$

where *f*a and *f*s are Gaussian kernels. The first one, *f*a, is a Gaussian kernel that gives more weight to pixels having similar intensity in I1. The second one, *f*s, is a Gaussian kernel that gives more weight to pixels spatially close in I1.

Figure 4. **Intra-image reconstruction (ξ_1)**

The second function, $\eta(I1';I2, w) =$ eta_12 (Figure 5), transports the appearance of image I1 that is treasured by I2. The function will preserve the structure from I1 but using colors from I2. We compute this reconstruction using a cross bilateral filter of the warped image I1' [1]:

$$eta\_12(x) = \frac{1}{Z(x,I1')} \sum_{y \varepsilon Nx} \alpha(x,y;I1') I2(y + w(y)) \qquad (3)$$

As we see, in both filters we use a window that runs for the image computing a gaussian filter to the region that is inside it. After changing the sizes of this window once we concluded all the code, we have seen that as bigger the window is, the worse is the estimated occlusion.

Figure 5. **Inter-image reconstruction (eta_12)**

Once we computed ξ_1 and eta_12 functions, we segmentate ξ_1 into homogeneous regions, named as *superpixels* (Figure 6). The criterion to do this segmentation is that each region has to be meaningful in order to estimate local color models. In our code, we use SLIC superpixels [8] and a Gaussian Mixture Model of color. In order to have more or less regions, we must modify the *weKeep* value from the code, at this also will affect to the final result of the occlusion (as we can see at the end of this section).

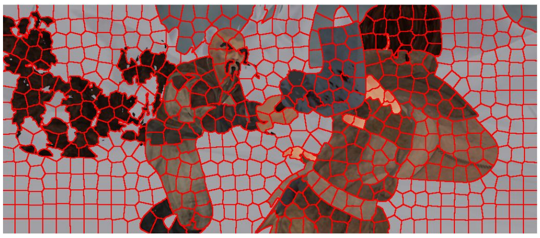

Figure 6. **Over-segmentation of ξ_1**

From this superpixels, we come to compute the Gaussian Mixture Estimation, which for every region we get from the superpixels we apply a the gaussian mixture model to both images ξ_1 and eta_12. This local color model G is adjusted to ξ_1 in the neighborhood of the pixel x which from evaluating on the probability of eta_12 it can decide whether it is occluded or not. In the algorithm implemented we computed this two functions separated. First, computing GMM for the ξ_1 data, which gives us good local information. This also supports on the eta_12 image given that the point we are evaluating is not occluded on I2.

From each GMM(x) we compute the probability of this model to belong to the superpixel we are evaluating. Afterward we get the probability of the previous model of belonging to the superpixel (SP). This process can be understood as:

$$\textbf{\textit{Prob}}\ \varepsilon\ \text{SP} = -\log(\,\textit{pdf}(\text{GMM(n), data})) \qquad (4)$$

where pdf is the probability density function and the data is the pixels of eta_12 belonging to the superpixel number *n*. Due to this probability we obtain the following soft occlusion map:

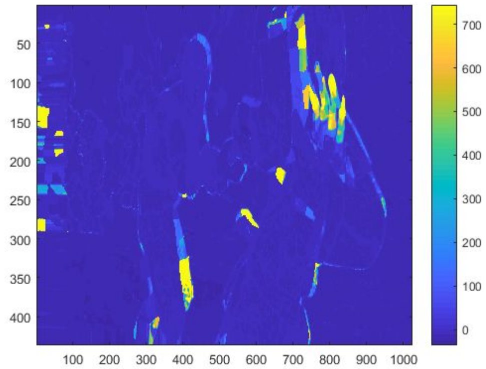

Figure 7. **Soft-Occlusion Map**

which gives the information of the different "coloured" regions of the image.

The last step is computing the estimated occlusion we get from all the process. We can compare our estimated occlusion (Figure 8) with the ground truth (Figure 9) that we are given.

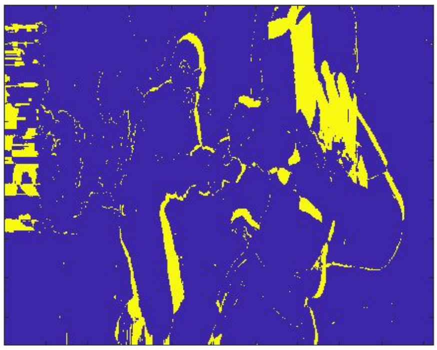

Figure 8. **Estimated Occlusion**

Figure 9. **Ground truth**

3. EVALUATION

When computing this genre of algorithms, there are various parameters we can vary in order to have a better or a worse result. As we said in the section 2, the window size of the bilateral filter changes the ξ_1 and eta_12 images. Another parameter we can modify is the value weKeep from the code that, as mentioned before, changes the over-segmentation of the image ξ_1 . In order to see how it affects to the estimated occlusion we can give a value which highly differ from the original one (0.15) and compare the results.

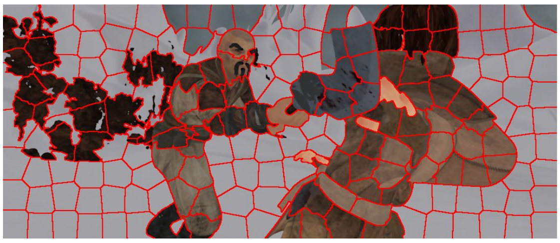

Figure 10. **Segmentation with weKeep=0.05**

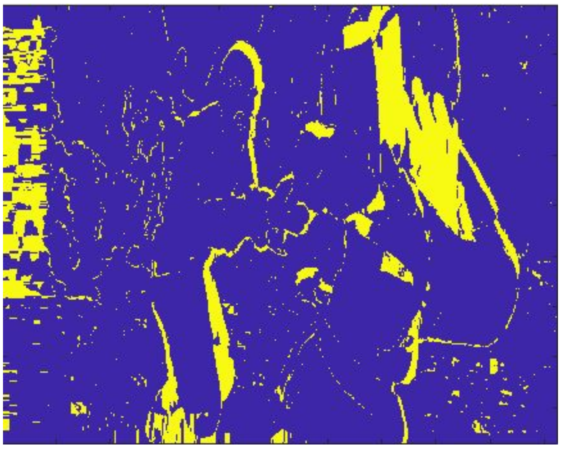

Figure 11. **Estimated Occlusion with weKeep=0.05**

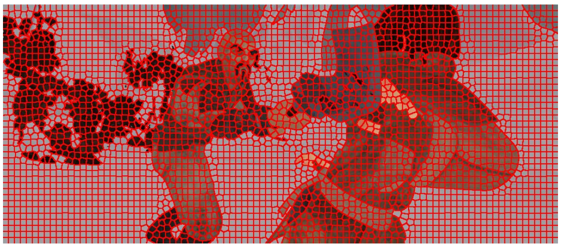

Figure 12. **Segmentation with weKeep=0.9**

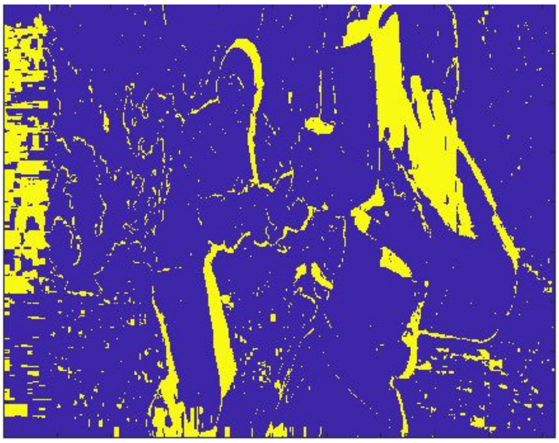

Figure 11. **Estimated Occlusion with weKeep=0.9**

There are much noise with a bigger value than the little one. However, the better way to do it is came to a balance where the estimation is the best.

4. CONCLUSIONS

The outcome of all this variations of parameters gives us an idea of the complexity of getting a plausible occlusion detection method.

There are a lot of constraints due to the complexity of the optical flow motion vectors, the gaussian models, having the best segmentation possible of the image, etc.

Despite the limitations of this method, spatio-temporal reconstruction model only requires the previous cognition of a plausible motion of the image' sequence. This model works at the pixel level of the image, which is more powerful than working with intensity inconsistency through the video flow (that leads to false or missed motion detections).

5. REFERENCES

[1] Pérez-Rúa, Juan-Manuel, et al. "Determining occlusions from space and time image reconstructions. "Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[2] Niu, Xiaoguang & QIAO, Yu. (2017). Context-based Occlusion Detection for Robust Visual Tracking. 10.1109/ICIP.2017.8296964. URL https://www.researchgate.net/publication/321422784_Context-based_Occlusion_Detection_for_Robust_Visual_Tracking

[3] Y. Xu, L. Qin, G. Li and Q. Huang, "An efficient occlusion detection method to improve object trackers," 2013 IEEE International Conference on Image Processing, Melbourne, VIC, 2013, pp. 2445-2449.
doi:10.1109/ICIP.2013.6738504
URL.http://ieeexplore.ieee.org/document/6738504/?reload=true
[4] A.Ayvaci, M.Raptis, S.Soatto, “Occlusion Detection and Motion Estimation with Convex Optimization ”. URL:
http://vision.ucla.edu/papers/ayvaciRS10.pdf
[5] H.Chandel, S.Vatta “Occlusion Detection and Handling: A Review”. URL: https://pdfs.semanticscholar.org/ce76/eca9f5999246964df3d5a9f741b80982c44b.pdf
[6] C. Ballester et al. “A TV-L1 Optical Flow Method with Occlusion Detection”
[7] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In ICCV 1998.
[8] Image and Visual Representation Lab. SLIC Superpixels. 16.07.2017. URL:
https://ivrl.epfl.ch/research/superpixels
[9] David J. Fleet, Yair Weiss. Optical Flow Estimation.
URL.http://www.cs.toronto.edu/~fleet/research/Papers/flowChapter05.pdf
[10] Scene Flow. URL.
http://www.mia.uni-saarland.de/Research/CV_SceneFlow.shtml