Detecting Half-Occlusion with a Fast Region-Based Fusion Procedure

Pierre-Marc Jodoin† Christophe Rosenberger†† Max Mignotte†

† D.I.R.O.
Université de Montréal
P.O. Box 6128, Stn. Centre-Ville
Montréal, Qc, Canada

†† Laboratoire de Vision et Robotique (L.V.R.)
ENSI de Bourges - Université d'Orléan
10 bd Lahitolle
18020 Bourges Cedex, France

Abstract

This paper presents a novel region-based approach for detecting occlusion between two consecutive frames. Based on a generalization of Marr and Poggio's uniqueness assumption, the explicit goal of our method is to reduce the number of false positives while optimizing the hit rate. To do so, our method relies on a fusion procedure that blends together two segmentation maps: one pre-estimated occlusion binary map and one color segmentation map. While the occlusion map is obtained after a simple thresholding procedure, the color segmentation map is obtained with an unsupervised Markovian approach. Assuming that the color segmentation regions exhibit more precise edges, the occlusion areas are iteratively modified to fit the color-region shapes. Since our method has been entirely implemented on a parallel architecture (a Graphics Processor Unit), its processing times are remarkably low. Our method is compared with other occlusion approaches both quantitatively and qualitatively on scenes that represent different challenges.

1 Introduction

The goal of most optical flow and stereovision algorithms is to estimate a matching function (be it a disparity map [7] or an optical flow field [10]) between the pixels of two (or more) images. Due to motion or to a parallax effect between a *left* and a *right* image, most scenes contains areas that are visible in only one frame. Generally speaking, these half-occluded areas are either *newly exposed* or *newly occluded* [18, 19]. Since these areas have no direct correspondence in the second image, they are a classical source of error for most motion or depth estimation algorithm.

While many authors have considered occlusion as a source of noise that is to be fought with spatial smoothing [7], others have explicitly included an occlusion criterion to the energy function to be minimized [2, 4, 6, 8, 12, 13]. During the past few years, a variety of occlusion metrics have been proposed among which the one Egnal and Wildes [9] call the *left-right-check* (LRC) has drew a lot of attention. This approach stipulates that the matching function between the left and the right image shall differ only by a sign with the right-left matching function. In this context, every pixel for which the difference between the left-right match and the righ-left match is above a given threshold are considered as being occluded. Although the LRC can be useful within a global energy function [5, 13, 14], many have pinpointed that the LRC is error-prone in noisy areas [19] and in areas having little or no texture [8, 9]. Others have also argued that estimating the forward *and* the backward matching functions can be prohibitive time wise.

BMVC 2006 doi:10.5244/C.20.43

Another idea that enjoys a great deal of popularity is Marr-Poggio's [16] uniqueness assumption. This assumption stipulates that there shall always be a one-to-one correspondence between the pixels of the two frames. Kolmogorov and Zabih [20] incorporated that assumption to their graph-cut algorithm and stipulated that each pixel in one image shall correspond to *at most* one pixel in the other image. A pixel with no match would then be considered as being occluded. A variation of that approach has been proposed by Sun *et al.* [12] for which a non-occluded pixel must have *at least* one match. Although the difference between the two approaches is conceptually thin, Sun *et al.* [12] demonstrated that their method performs better in scenes containing slanted surfaces. The uniqueness assumption has also been used by Zitnick and Kanade [15] who proposed a cooperative algorithm that iteratively enforce the uniqueness constraint within a local 3D array. In their method, occlusion is identified by thresholding the left-right correspondence error map obtained after their optimizer has converged. More recently, Ince and Konrad [19] proposed a generalization of the uniqueness constraint : instead of counting the number of matches for each pixel independently, they count the number of matches within a given local neighborhood. As mentioned by the authors, this simple but decisive modification makes the metric significantly more robust to noise. The reader shall notice that Ince-Konrad's idea can be seen as a generalization of Egnal-Wildes' [9] Occlusion Constraint (OCC).

Let us also mention that some authors use the so-called *Ordering constraint* [6, 9, 12] which stipulates that a point P laying to the right of a point Q in one image shall also lay to the right of Q in the other image. Although this assumption is often true, it can be easily violated by narrow front-ground objects (what Sun *et al.* [12] call the "double nail illusion").

In this contribution, we propose a novel occlusion detection framework based on Marr-Poggio's [16] uniqueness assumption. Given a matching function between two frames, we propose a simple occlusion-detection method that works without having to iteratively reestimate the matching function. The main objective being to provide a simple, fast and accurate algorithm. Our method is built over a *fusion* procedure that blends together two label fields. The first label field is a rough occlusion map estimate obtained after a simple thresholding procedure. Although this occlusion map is typically noisy, it gives a good estimate of where the main occlusion areas are located. The second label field is a region map obtained after segmenting the two input frames. Since the occlusions are assumed to lie along objects' silhouette, the fusion procedure *encourages* occlusion areas to fit the color regions. In this way, isolated false positives are eliminated and blobby occlusion areas are warped to fit the objects' silhouette.

The rest of the paper is organized as follows. In Section 2, our method is presented. The Section includes details on how the two label fields are estimated and blended together. Several experimental results are presented in Section 3 to illustrate how good qualitatively and quantitatively our method is as compared to similar approaches. Section 4 then briefly concludes.

2 Proposed Method

The proposed method is initially fed with two frames (that we call I^{ref} and I^{mat}, the *reference* and the *matching* frames) and a matching function $\mathscr{M}$ linking the pixels of I^{ref} to those of I^{mat}. The frames can be either a stereo pair or two consecutive images taken from a video sequence. Based on $\mathscr{M}$, an occlusion label field $\mathscr{O}$ is first estimated with a simple thresholding procedure. Although the occlusion map $\mathscr{O}$ is typically noisy involving numerous false positives and false negatives (see Figure 1), it nevertheless gives a good indication of where the major occlusion areas are located (see Section 2.1 for more details on how $\mathscr{O}$ is estimated).

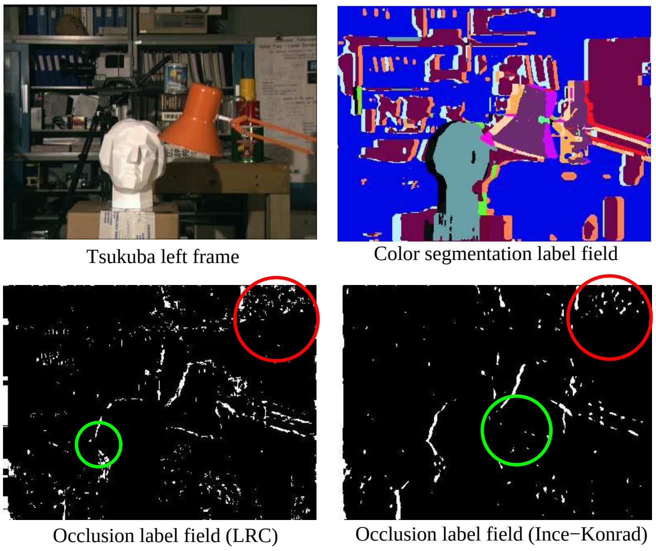
Tsukuba left frame
Color segmentation label field
Occlusion label field (LRC)
Occlusion label field (Ince−Konrad)

Figure 1: Tsukuba left image with the color segmentation field r^c and two occlusion maps (obtained with LRC and Ince-Konrad). Red circles identifies false positives and greens circles false negatives.

Once a rough occlusion map has been estimated, a second label field is computed. This second field (that we call r^c) is obtained after segmenting the two input frames based on their color distribution. The two resulting fields (called r^{ref} and r^{mat}) can be seen as a series of regions made of pixels whose color follows a uniform distribution. The two label fields are then merged together (r^c = r^{ref} merged with r^{mat}) in such a way that each region of r^c is uniform in the sense of I^{ref} and I^{mat}. As shown in Figure 1, r^c fit the main silhouettes of the scene and gives a good indication of where occlusion is likely to occur.

Once $\mathscr{O}$ and r^c have been estimated, they are fused together in order to reduce the number of false positive/negative and make the regions of $\mathscr{O}$ better fit the regions of r^c.

2.1 Occlusion Detection

After thorough evaluations of many occlusion detection approaches, we came to the conclusion that the ones based on Marr-Poggio's uniqueness assumption are the most accurate, at least in the context of our method (in their review paper, Egnal and Wildes [9] came to a similar conclusion). More specifically, since our implementation of Ince-Konrad's [19] metric outperformed every other ones we have implemented, their method was retained to compute $\mathscr{O}$, the "rough" occlusion map estimate.

The way Ince-Konrad's approach works is simple. Lets consider $\Lambda = \{s|s \in S\}$ the set of pixels in the reference image I^{ref} and $\Delta = \{u|u = s + \mathscr{M}_s\}$, the set of *matching* pixels in

I^{mat}. Based on Δ, an accumulation function M is computed

$$M_t = \sum_{i \in \Delta} \zeta_{i,t} \tag{1}$$

where $\zeta_{i,t} = 1$ if the euclidean distance between pixels $t \in S$ and $i \in \Delta$ is lower of equal to D, and zero otherwise. The occlusion map $\mathscr{O}$ is obtained by thresholding M_s :

$$\mathscr{O}_s = \left\{ \begin{array}{ll} 1 & \text{if } M_s < \tau \\ 0 & \text{otherwise} \end{array} \right. \tag{2}$$

As suggested by the authors [19], D is set to 2.

2.2 Color Segmentation

Although the color label field r can be estimated by any valid segmentation approach, we resorted to an unsupervised statistical Markovian segmentation which classifies the color pixels into a predetermined number of classes. The reason for this choice is twofold. First, the segmentation method we have implemented is *unsupervised* since it requires no parameter adjustment during runtime. To our opinion, this property appears as a major advantage. Second, this segmentation method can be parallelized and implemented on a parallel architecture such as a graphics processor unit (GPU) [17]. In this way, the segmentation map r^c can be computed in interactive time.

Let $Z = \{R, I\}$ be a pair of random fields where $R = \{r_s | s \in S\}$ is the label field to be estimated and $Y = \{\vec{Y}_s | s \in S\}$ is an input color image (I^{ref} or I^{mat}). Here, R and Y are defined on a 2D finite lattice $S = \{s = (i, j) | i \in [0, N[, j \in [0, M[\}$ and can be seen as random variables for which r and y are specific realizations. Notice that each pixel of R are called *labels* and takes a value in $\Gamma = \{1, \ldots, m\}$, where m is the number of classes.

Here, the goal is to associate the *best* label $r_s \in \Gamma$ to each pixel given the observed color image y. According to the *Maximum a posteriori* criteria, the *optimal* label field r can be formulated as : $r = \arg\max_r P(r|y)$ or, with the Bayes rule [3], $r = \arg\max_r P(y|r)P(r)$. With the assumption that $P(y|r)$ and $P(r)$ follow a Gibbs distribution of the form $P(Y|R) \propto \exp -U_1(Y, R)$ and $P(R) \propto \exp -U_2(R)$ and that each random variable $\vec{Y}_s$ given R_s are independent, the actual posterior PDF may be maximized by minimizing the following functional

$$U(r, y) \quad = \quad \sum_{s \in S} U_1(r_s, \vec{y}_s) + \beta U_2(r_s, \eta_s) \tag{3}$$

where U_1 is the likelihood energy function, U_2 the prior energy function, β a constant, and η_s a neighborhood centered on pixel s. In this paper, U_1 is modeled with a log-Gaussian law of the form

$$U_1(r_s, \vec{y}_s) = -\ln((2\pi)^{d/2} |\Sigma_{r_s}|^{1/2}) + \frac{(\vec{y}_s - \vec{\mu}_{r_s}) \Sigma_{r_s}^{-1} (\vec{y}_s - \vec{\mu}_{r_s})}{2} \tag{4}$$

where $\vec{\mu}_{r_s}$ and Σ_{r_s} are the mean and the variance-covariance matrix of class $r_s \in \Gamma$. In this way, each class is modeled with a Normal law defined by two parameters $(\vec{\mu}_{r_s}, \Sigma_{r_s})$, which means a grand total of $2m$ parameters $\Phi = [(\mu_1, \sigma_1), \ldots, (\mu_m, \sigma_m)]$ for the entire model. Since none of these parameters are known a priori, they need to be estimated. To this end, we resort to an iterative method called Iterated Conditional Estimation (ICE) [21] which is a stochastic and Markovian version of the well known EM algorithm.

As for U_2, we use the isotropic Potts model : $U_2(r_s, \eta_s) = \sum_{t \in \eta_s}(1 - \delta_{r_s,r_t})$ where δ_{r_s,r_t} is the Kronecker function (returns 1 if $r_s = x_t$ and 0 otherwise) and η_s a second-order neighborhood. Notice that the Potts model is a n-class generalization of the well known two-classes Ising model.

As mentioned before, the segmentation is preformed by computing $r = \arg\max_r P(r|y)$ or, equivalently, $r = \arg\min_r U(r,y)$. Since there is no analytical solution to those equations, we implemented the deterministic downhill-search ICM algorithm [11].

For the sake of our method, the input frames I^{ref} and I^{mat} are respectively segmented into two label fields, namely r^{ref} and r^{mat} that are then linearly combined together : $r^c = r^{\text{ref}} + m \times r^{\text{mat}}$. This last operation results in a label field r^c whose regions are uniform in the sense of both input images. For more details on how ICE and ICM have been implemented, please refer to [17].

2.3 Fusion Procedure

Once $\mathscr{O}$ and r^c have been estimated, they are fed to an iterative fusion procedure. This procedure aims for an occlusion map $\hat{\mathscr{O}}$ that would be locally uniform both in the sense of color (r^c) *and* occlusion ($\mathscr{O}$). To this end, the fusion procedure works as an optimizer, looking for a solution $\hat{\mathscr{O}}$ whose corresponding energy E is minimum :

$$\begin{aligned} \hat{\mathscr{O}} &= \arg\min_{\mathscr{O}} E(r^c, \mathscr{O}) && (5) \\ &= \arg\min_{\mathscr{O}} \sum_{s \in S} V_{\psi_s}(r^c_s, \mathscr{O}_s). && (6) \end{aligned}$$

where $V_{\psi_s}(r^c_s, \mathscr{O}_s)$ is a *local* energy function. This energy term returns a low value when the neighborhood (here ψ_s) surrounding s is uniform both in the sense of r^c and $\mathscr{O}$ and a large value otherwise. To measure this *degree of uniformity* inside the given neighborhood ψ_s, two potential δ-functions are being used

$$V_{\psi_s}(r^c_s, \mathscr{O}_s) = -\sum_{t \in \psi_s} \delta_{r^c_t, r^c_s} \delta_{\mathscr{O}_t, \mathscr{O}_s}. \tag{7}$$

where ψ_s is a $L \times L$ window centered on pixel s and δ is the Kronecker delta function. Thus, for a given pixel s, $V_{\psi_s}(r^c_s, \mathscr{O}_s)$ counts the number of pixels $t \in \psi_s$ that are simultaneously in spatial region r^c_s and part of occlusion class $\mathscr{O}_s \in [0,1]$. In this way, the occlusion label $\mathscr{O}_s \in [0,1]$ that occurs the most frequently within the neighborhood belonging to the color class r^c_s in ψ_s has the smallest energy. This procedure is illustrated in Figure 2. As can be seen in the middle image, both pixels a and b are initially classified as being occluded. However, when looking at every pixel t within ψ_a that are *part of the deep-blue uniform background in* r^c, we see that there is a majority of non-occluded pixels in $\mathscr{O}$. In other words, among the neighbors around pixel a that are part of class r^c_s, there is a majority of *non-occluded* pixels and thus $V_{\psi_a}(r^c_a, \text{occluded}) > V_{\psi_a}(r^c_a, \text{non-occluded})$. For this reason, pixel a is assigned the *non-occluded* label in the resulting occlusion map $\hat{\mathscr{O}}$. As for pixel b, since most of its neighbors $t \in \psi_b$ part of the black region in r^c are *occluded* pixels, b is kept occluded in $\hat{\mathscr{O}}$.

Since there are no analytical solutions to $\hat{\mathscr{O}} = \arg\min_{\mathscr{O}} E(r^c, \mathscr{O})$, we again resort to the ICM [11] algorithm whose mode (the minimum local energy for each site at each

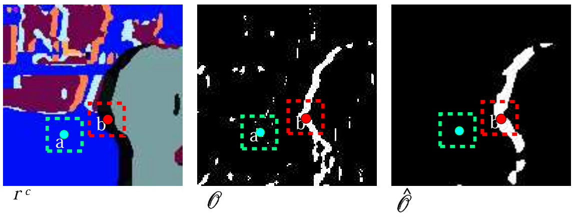

Figure 2: Zoom on the *Tsukuba* reference image. Left is label field r^c obtained after combining r^{ref} and r^{mat}. Middle is the initial occlusion map $\mathcal{O}$ obtained with Ince-konrad's [19] method. Right is the occlusion map obtained after merging r^c and $\mathcal{O}$.

```
Proposed Algorithm

| $r^c$                        Color segmentation label field
| $I_{\text{ref}}, I_{\text{mat}}$    Two input frames
| $\mathcal{M}$                Matching function between $I_{\text{ref}}$ and $I_{\text{mat}}$
| $k$                          The iteration step
| $\mathcal{O}^{[k]}$          Occlusion map after the $k^{\text{th}}$ iteration.
1. Occlusion Estimation
$\mathcal{O}^{[0]} \leftarrow$ Occlusion estimation with Ince-Konrad's [19] method.
2. Color Segmentation
Learning the $2m$ Gaussian parameters with ICE
$r^{\text{ref}}, r^{\text{mat}} \leftarrow$ segmentation of $I_{\text{ref}}$ and $I_{\text{mat}}$ with ICM
$r^c \leftarrow r^{\text{ref}} + m \times r^{\text{mat}}$
3. Fusion
$k \leftarrow 1$
while !Convergence do
    for each pixel $s \in S$ do
        $V \leftarrow 0$
        for each pixel $t \in \psi_s$ do
            $V(0)$ -= $\delta_{r_s,r_t}\delta_{0,\mathcal{O}_t}$
            $V(1)$ -= $\delta_{r_s,r_t}\delta_{1,\mathcal{O}_t}$
        $\mathcal{O}_s^{[k]} \leftarrow \arg\min_{i\in[0,1]} V(i)$
    $k \leftarrow k+1$
```

Algorithm 1: Proposed algorithm. Here δ is the Kronecker delta and m the number of classes (that we set to 4).

iteration) is defined by the local energy function $V_{\psi_s}(r_s, \mathcal{O}_s)$. Notice that when minimizing $E(r^c, \mathcal{O})$, each ICM iteration works in a similar way the well known K-nearest neighbor algorithm does [3]. In our case, though, the variable K is defined as : $K = \text{Card}(\{t | r_t^c = r_s^c \text{ and } t \in \psi_s\})$. The complete algorithm of our method is presented in Algo. 1.

3 Experimental Results

To validate our method, we detected occlusion on various data sets representing different challenges. The goal of these tests is to demonstrate how stable and robust our framework

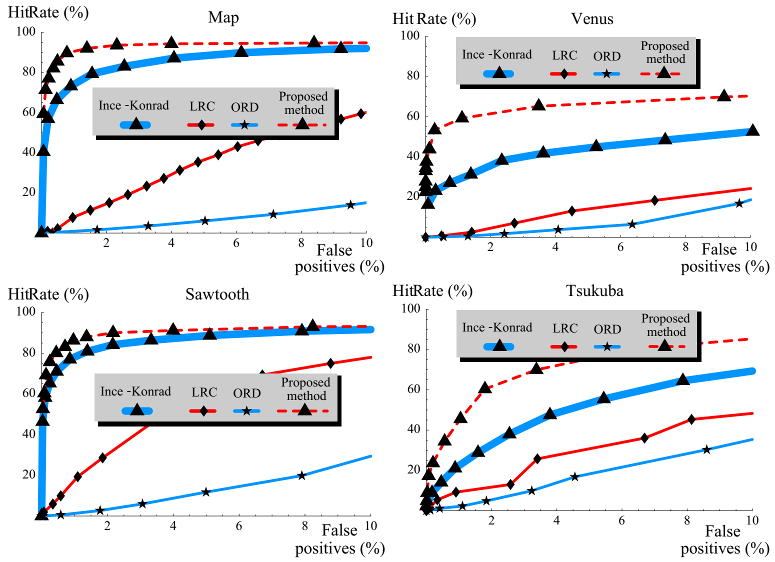

Figure 3: Hit rate versus false positive rates obtained with four different data sets.

is with respect to other frequently-used approaches. Among the methods we have compared our method with, is the *left-right check* (LRC) [9], the *ordering constraint* (ORD) [9] and Ince-Konrad's [19] uniqueness-based approach.

For each example presented in this section, we used a 5×5 neighborhood ψ_s, a number of $m = 4$ segmentation classes, and a smoothing constant β=2. Four sequences with ground truth taken from Middlebury web page [1] have been used to test the methods. The disparity map of each data set has been computed with a pixel-based matching strategy implemented together with a 3×3 *shiftable* aggregation filter [7]. The *data* energy function [7] was minimized with the deterministic *winner-take-all* optimization algorithm.

Following Egnal and Wildes' [9] methodology, we have plotted the hit rate / false positive rate curve of every method by varying their threshold (see Figure 3). On every graphics of Figure 3, our method appears to be more precise than the other ones we have implemented. This is especially true for those sequences containing large textureless areas such as Venus and Tsukuba. This can be explained by the fact that, as mentioned by Egnal and Wildes [9], most common occlusion detection methods are error-prone in textureless areas. In this context, using a region-based approach to eliminate isolated false positives brings a clear advantage.

A qualitative comparison have also been made in Figure 5. To make the results objectively comparable, each method have been tuned to return an occlusion map with a specific hit rate. In this way, every results in the second and third column of Figure 5 have respectively a hit rate of 60%, 90%, 45%, and 90%. Although the hit rate is the same for both approaches, the false positive rate is clearly to our method's advantage.

As for the *flowergarden* sequence of Figure 6, our method produced again a significantly lower amount of false positives. Notice that for this sequence, the matching function was computed with a pixel-based window-matching strategy [10].

Since our method depends mostly on one variable, namely ψ_s (the 2D neighborhood),

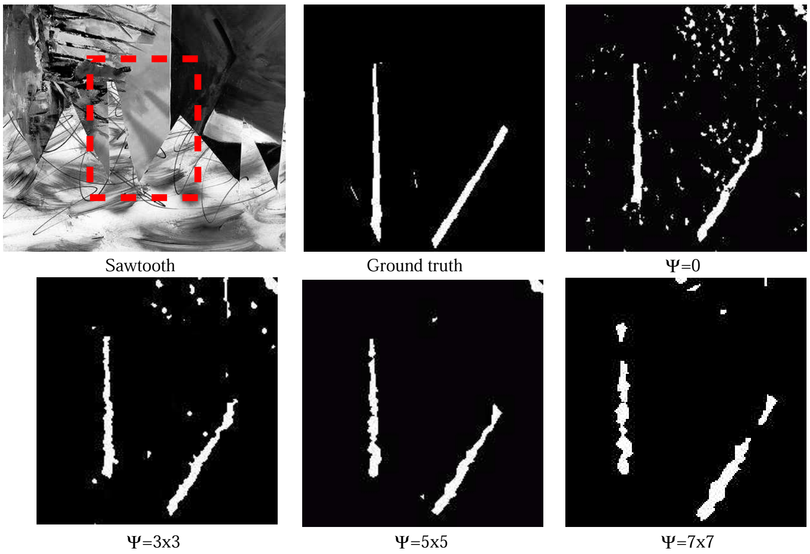

Figure 4: This figure illustrates the influence the neighborhood size ψ_s.

we have illustrated its influence in Figure 4. As can be seen, a modification of this variables brings a smooth and predictive variation in the resulting image. As for the other variables on which our methods depends (such as the number of classes m and the smoothing parameter β) we noticed that a variation of their value has little or no influence on the resulting occlusion map.

As for the implementation, since every pixel of the scene are independently processed, we have implemented our method on a parallel architecture, namely a Graphics Processor Unit (GPU) [17]. A GPU is a processor embedded on most graphics card nowadays available on the market which, among other things, can load, compile and execute programs implemented with a C-like language. The key feature of GPUs is their fundamental ability to process *in parallel* each pixel of the scene, making all kinds of applications much more efficient than on traditional sequential CPUs. For example, the fusion procedure (with $\psi_s = 5 \times 5$) can process at a rate of 25 fps a scene of size 384×288 such as $Tsukuba$ [1]. Also, the same color scene can be segmented in approximately 1 second or, if the Gaussian parameters are reused from a previous calculation, in 0.05 second. These processing rates outperformed by a factor of almost 100 what we obtained with a traditional CPU implementation.

4 Conclusion

In this paper, an occlusion detection method based on the uniqueness assumption has been proposed. The core of our method is a fusion procedure that blends together two label fields: a pre-estimated occlusion map $\mathcal{O}$ and a color segmentation map r^c. With the assumption that the color regions' silhouette are more precise than the pre-estimated

[1] Since there are no efficient way to access the framebuffer content to verify if the ICM algorithm has converged (see part 3 of Algo 1), a predefined number of 5 ICM iterations has been used to produce the results here presented.

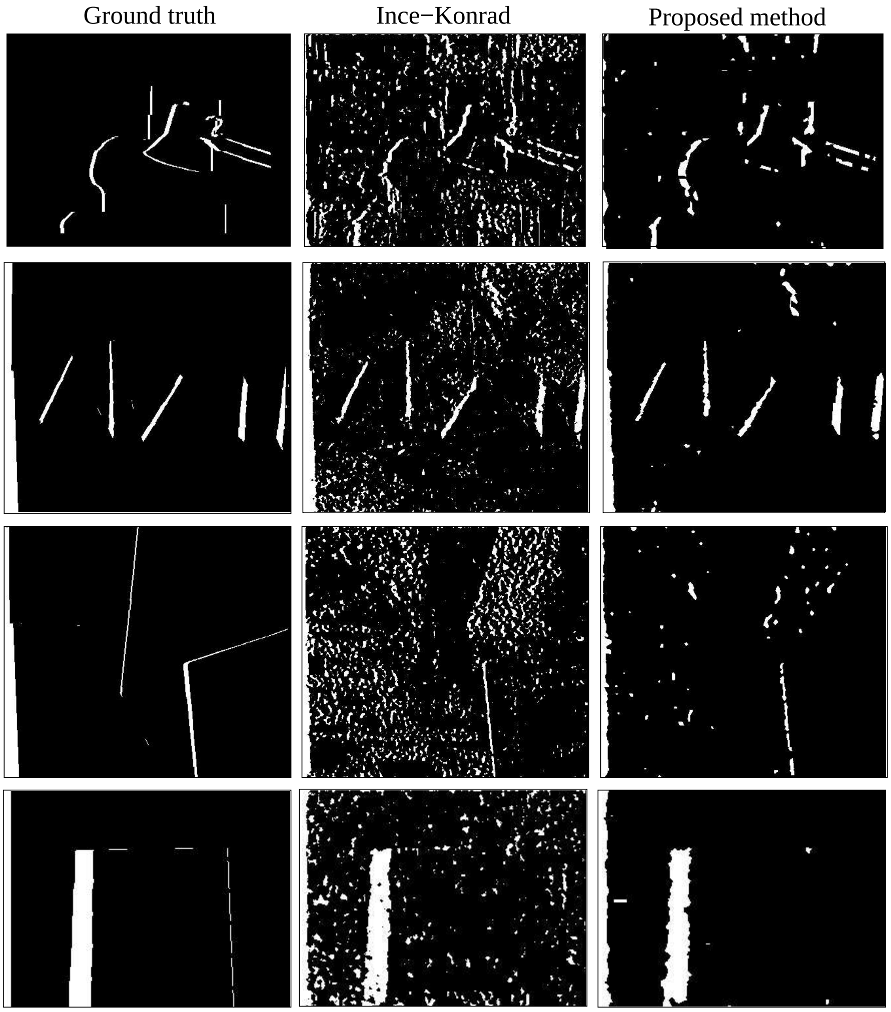

Figure 5: From top to bottom, ground truth and results obtained for *Tsukuba*, *Sawtooth*, *Venus*, and *Map* data set. Hit rate for every results is respectively 60%, 90%, 45%, and 90%.

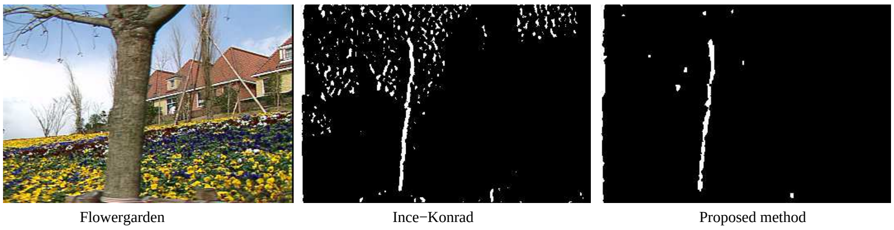

Figure 6: *Flowergarder* sequence.

occlusion areas, the occlusion map is iteratively modified to fit the color regions. In this way, isolated false positives/negatives are filtered out resulting in better hit rate versus false positive rate ratios. The fusion procedure is an ICM-based optimization method that minimizes a local energy function V_{ψ_s}. Since our method processes every pixel independently, it can be implemented on a parallel architecture. A direct implementation on a mid-end GPU have shown that our method can work in interactive time.

In the future, we intend to adapt our method to other applications that could benefit from the blending of two label fields whose content is complementary. Among the applications that appears to us as promising is motion detection, stereovision and optical flow.

References

[1] www.middlebury.edu/stereo.

[2] Luo A. and Burkhardt H. An intensity-based cooperative bidirectional stereo matching with simultaneous detection of discontinuities and occlusions. *Int. J. Comput. Vision*, 15(3):171–188, 1995.

[3] Bishop C. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

[4] Strecha C., Fransens R., and Van Gool L. A probabilistic approach to large displacement optical flow and occlusion detection. In *proc of ECCV Workshop SMVP*, pages 71–82, 2004.

[5] C. Chang, S. Chatterjee, and P.R. Kube. On an analysis of static occlusion in stereo vision. In *Proc. of CVPR*, pages 722–723, 1991.

[6] Geiger D., Ladendorf B., and Yuille A. Occlusions and binocular stereo. *Int. J. Comput. Vision*, 14(3):211–226, 1995.

[7] Scharstein D., Szeliski R., and Zabih R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proc. of the IEEE Workshop on Stereo and Multi-Baseline Vision*, 2001.

[8] P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6:35–49, 1993.

[9] Egnal G and Wildes R.P. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(8):1127–1133, 2002.

[10] Barron J., Fleet D., and Beauchemin S. Performance of optical flow techniques. *Int. J. Comput. Vision*, 12(1):43–77, 1994.

[11] Besag J. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc.*, 48(3):259–302, 1986.

[12] Sun J., Li Y., and Kang S.B. Symmetric stereo matching for occlusion handling. In *proc. of CVPR (2)*, pages 399–406, 2005.

[13] Lim K., Das A., and Chong M. Estimation of occlusion and dense motion fields in a bidirectional bayesian framework. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):712–718, 2002.

[14] Alvarez L., Deriche R., Papadopoulo R., and S&#225;nchez J. Symmetrical dense optical flow estimation with occlusions detection. In *proc of ECCV*, pages 721–735, 2002.

[15] Zitnick L. and Kanade T. A cooperative algorithm for stereo matching and occlusion detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(7):675–684, 2000.

[16] D. Marr and T.A. Poggio. Cooperative computation of stereo disparity. 194(4262):283–287, 1976.

[17] Jodoin P-M, St-Amour J-F, and Mignotte M. Unsupervised markovian segmentation on graphics hardware. In *proc of ICAPR (2)*, pages 444–454, 2005.

[18] Depommier R. and Dubois E. Motion estimation with detection of occlusion areas. In *Proc. of ICASSP*, pages 269–272, 1992.

[19] Ince S. and Konrad J. Geometry-based estimation of occlusions from video frame pairs. In *Proc. of ICASSP*, volume 2, pages 933–936, 2005.

[20] Kolmogorov V. and Zabih R. Computing visual correspondence with occlusions via graph cuts. In *Proc. of ICCV*, pages 508–515, 1999.

[21] Pieczynski W. Statistical image segmentation. *Machine Graphics and Vision*, 1(1):261–268, 1992.