

Francisco Hernandez-Lopez and Mariano Rivera, Binary Segmentation of Video Sequences in Real Time, in Advances in Artificial Intelligence and Applications, Proc. MICAI 2010, 163--168. 8-13 Nov. Pachuca Mex, .2010. IEEE Computer Society, Los Alamitos CA.

## Binary Segmentation of Video Sequences in Real Time

Francisco J. Hernandez-Lopez and Mariano Rivera

Centro de Investigacion en Matematicas A.C, Guanajuato GTO 36000, Mexico

**Abstract.** We present a method for foreground-background video segmentation in real time that may be used in applications as Background Substitution, Analysis of Surveillance Cameras, Highway Cars Detection and so on. Our approach implements a probabilistic segmentation based on the binary Quadratic Markov Measure Fields models (QMMFs). That framework regularizes the likelihood of each pixel to belong to each one of the models (foreground and background). Then our proposal consists of a model for the likelihood that takes into account: an estimation of the static background, motion of the foreground, illumination changes and casted shadows. In order to fulfill the real-time requirement we implement a parallel version of our algorithm in CUDA using a NVIDIA GPU.

### 1 Introduction

Images are one of the more richer information sources for humans. Then image processing methods has been extensively developed in order to improve the images quality and to obtain more information about this perceptive source. The variates image processing methods have also developed many new applications: for instance video segmentation. This task consists into partition the video in space-time regions that are homogeneous in their spatial characteristics [1]. The video segmentation requires of selecting appropriate characteristics (intensity, color, texture or movement) and to use proper distance measures for comparing those selected characteristics. Different characteristics and homogeneity criteria usually lead to different segmentations of the same video. On the other hand, there is no guarantee that some of these automatic segmentation results are significantly good in a practical sense. Specific methods for video segmentation need be evaluated in context of the application for which they are developed.

A common problem in the analysis of video sequences is the Foreground/Background segmentation in real time. There are different cases that may occur when we process a video sequence (see table 1). The solution of each case requires of a specific method that rely on assumptions about how to solve the problem. Our research focuses primarily on the problem of replacing the background of a video sequence in real time (see Fig 1d), this problem has been studied for a wide variety of applications in which the objective is to compare an image with a background model and then identifying the regions where changes occur. Many algorithms developed that integrate spatial-temporal information to improve the robustness, some of the techniques in the state of the art are based on Conditional random field (CRF) [2], Temporal Foreground Probability Model (TFPM) [3], Kalman Filter [4], Codebooks [5] (robust to illumination changes), Invariant Color Models [6], Low Pass Filter [7] (robust to shadows) and methods that using GPU [8, 9].

Space	Case I	Case II	Case III	Case IV
Foreground	Mobile	Mobile	Static	Static
Background	Static	Mobile	Mobile	Static

**Table 1:** Cases of the video segmentation problem.

We use a binary segmentation model based on Markov Random Field (MRF) models, which uses a priori knowledge on the two classes we want to segment. Such a knowledge is obtained through a Background Model Likelihood (BML). In order to make robust our approach, our BML uses: a Tonal Transference Operator, a Casted Shadows Detection and Correction of Camouflage Situations. In the following sections we explain our proposal for Foreground-Background (F/B) Segmentation. Our method was implemented in Visual Studio 2008 using CUDA and runs 47 f/s (frames per second) in a NVideo 8800GT GPU. The method performance is demonstrated by experiments in real video and using different situations of F/B segmentation.

## 2 Foreground-Background Segmentation

Similarly to the methods reported by [3], [8], our method for video sequence segmentation requires of an confident estimate of the background model. We estimate that background model as the average of a few initial frames of the scene, assuming that foreground is to yet present (Fig 1a). Then we compute the likelihood of each pixel in consecutive frames to belong to the background (see Fig 1c). Here we note that a simple segmentation based on the maximal likelihood (ML) estimation fails in general because of the following reasons:

1. **Illumination changes.** Lighting changes in the scene product of unstable lamps, changes on external illumination condition (as clouds) may generate large differences between statics images of a single scene.
2. **Objects' shadows.** Objects in the foreground can cast shadow in the background that can erroneously be classified.
3. **Camouflage situation.** Objects in the scene may have regions with similar color than the occluded region in the background.
4. **Video artifacts and noise.** Video compression algorithms as MPEG introduces artifacts at high frequency regions (i.e. at edges) that result in false detected changes with respect the background model; image's noise have a similar effect.

In the following subsections we described the binary segmentation method used in our proposal, then we present a background model and three important approaches in order to obtain a robust method to the problems outlined above.

## 2.1 Binary Probabilistic Segmentation

Let  $f$  be an RGB video sequence such that  $f(x, t) = [f_1(x, t), f_2(x, t), f_3(x, t)]^T$  is the RGB-vector value of the pixel at  $x = [x_1, x_2]^T$  and time (frame)  $t$ . Let  $F(x, t) = \mathcal{T}[f(x, t)]$  be a tonal transference operation that stabilize the pixel values to illumination changes and/or camera's automatic photometric gain control (we detail this transformation in subsection 2.3). Then in order to construct a segmentation method robust to the above listed problem, we use the Quadratic Markov Measure Field Models (QMMFs) framework. In particular our method is based on the two classes variant named Quadratic Markov Probability Field Models (QMPFs). In this framework a smooth (with reduced granularity) binary segmentation is computed from the solution to the quadratic programming problem

$$\arg \min_p U(p(x, t)) \quad \text{s. t. } p \geq 0, \quad (1)$$

where

$$\begin{aligned} U(p(x, t)) = & \frac{1}{2} \sum_x \left( p^2(x, t) d_B(x, t) + [1 - p(x, t)]^2 d_F(x, t) \right. \\ & \left. + \lambda \sum_{\langle x, y \rangle} ([p(x, t) - p(y, t)]^2 W_\gamma(x, y)) \right) \end{aligned} \quad (2)$$

with

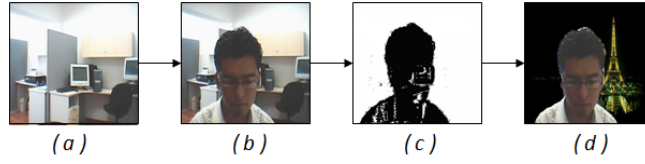
$$\begin{aligned} d_B(x, t) &= -\log[V_M(x, t)], \\ d_F(x, t) &= -\log[1 - V_M(x, t)], \\ W_\gamma(x, y) &= \frac{\gamma}{\gamma + \|F(x, t) - F(y, t)\|^2}, \end{aligned}$$

and  $V_M(x) \in [0, 1]$  is the likelihood of the pixel  $x$  to belong to the background class. This likelihood is defined in subsection 2.2. Finally, the weights  $W_\gamma$  promotes that the probability edges corresponds with gradient edges. Then in the QMMF framework,  $p$  can be computed with the Gauss-Seidel scheme

$$p(x, t) = \frac{d_F(x, t) + \lambda \sum_{\langle x, y \rangle} (p(y, t) W_\gamma(x, y))}{d_B(x, t) + d_F(x, t) + \lambda \sum_{\langle x, y \rangle} W_\gamma(x, y)} \quad (3)$$

Since  $V_M(x, t)$  and  $1 - V_M(x, t) > 0$ , the resulting energy is convex and the iteration of (3) converges to the non-negative global minimum [10]. The Gauss-Seidel iteration produces an improved partial solution at each iteration and converges quickly if a "good" initial point is provided. Following we enlisted our benefits from last properties of the segmentation algorithm:

1. We focus in compute a good background likelihood that correctly classify, as much as possible, the pixels in the two mentioned classes. Then our initial guess is the likelihood itself:  $p^0(x, t) = V_M(x, t)$ ,



**Fig. 1:** Proposal to Foreground-Background segmentation. (a) take  $N$  frames of background and calculate  $\mu(x)$ . (b) Read the following frames  $f(x, t)$ . (c) Calculate the likelihood function  $V_M(x, t)$ . (d) Segmentation with QMPF method.

2. We implement a multigrid algorithm with a fast convergence. Note that the update formula (3) is proper for being implemented in Graphics Processing Units (GPUs), our particular implementation is in CUDA [11].
3. In the implementation of our real time applications we compute a good partial solution by iterating the algorithm just few iterations (we use 10 iterations in our experiments).

## 2.2 Background model likelihood

We estimate the background model from a short images sequence of the static scene (in our experiments we use 10 frames), so we can compare each of the following frames in the video sequence with the background model, and to label the major differences as foreground. Since we are using MPEG codified video, then the noise and artifacts introduced by the compression algorithm cannot be modeled with a Gaussian distribution. We empirically found that a reliable classification of the pixels into the foreground of the background classes can be computed with:

$$V_M(x, t) = \begin{cases} 1 & \|F(x, t) - \mu(x)\| \leq \theta_1 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where we set  $\theta_1 = 0.01 \times \text{dynrange} \times \text{numchannels}$ , where *dynrange* is the dynamic range of the image (255 in our experiments), and *numchannels* is the number of image channels (3 in RGB images). Finally,  $\mu$  is the background model computed with the simple average of the first  $N$  frames:

$$\mu(x) = \frac{1}{N} \sum_{t=0}^N f(x, t). \quad (5)$$

Note that  $\mu(x) = [\mu_1(x), \mu_2(x), \mu_3(x)]^T$  are the corresponding means to the RGB color channels. Now we explain how we adapt the for being robust to lighting changes in the scene, shadows, noise and camouflage situations.



Fig. 2: Frames badly acquired from a WebCam.

### 2.3 Tonal Transference Operator

There may be many factors that cause a change of lighting in the scene, some of them are: unstable lamps, external illumination and passing cloud, or the camera's automatic gain control (see for instance Fig. 3). In our experiments, it is common that after a few seconds appear lighting changes in the acquired video which cause that the new frames are very different from our background model, resulting in a wrong segmentation. For this reason we include a lighting correction process based on the one proposed in Ref. [12]. They implement a robust segmentation method for moving objects using a background update process, the difference is that in our method does not update the background but the current frame. We update the current frame because occasionally the frames acquired from a WebCam are defective, as are shown in figure (2). Follows we present the procedure for controlling the illumination changes at the frame  $f(x, t)$ . Let  $\mu$  be the image average computed with (5),  $f(x, t)$  be the RGB value of the pixel  $x$  at the current frame and  $p(x, t - 1)$  be the probability in the previous frame of the same pixel to belong to the background class. Then, by assuming that the previous frame  $f(x, t - 1)$  was correctly segmented then we do the following:

1. We estimate the photometric gain for each channel (RGB) at each pixel that was classified as background in the previous frame, this is (for  $i = \{1, 2, 3\}$ ):

$$D_i(x, t) = \begin{cases} \mu_i(x)/f_i(x, t) & p(x, t - 1) \geq 1/2 \text{ and} \\ & 0 < f_i(x, t) < \text{dynrange} \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

2. Since the last pixelwise photometric gain is prone to be contaminated by noise, we compute its average for each intensity pixel value in the current frame:

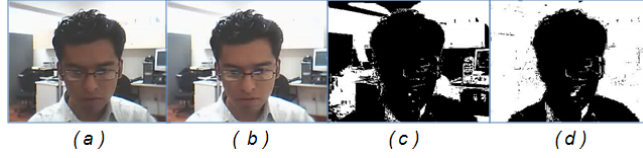
$$\bar{D}[b_{ij}, t] = \frac{1}{\#\mathcal{R}_{ij}} \sum_{x \in \mathcal{R}_{ij}} D_i(x, t) \quad (7)$$

where  $\mathcal{R}_{ij} = \{x \mid f(x, t) = b_{ij}\}$  is the set of the sites (pixels) that have a  $j$ th value in the  $i$ th channel,  $\#\mathcal{R}_{ij}$  is its cardinality and  $\{b_{ij}\}_{i=1,2,3; j=1 \dots n}$  are the (n) values of intensity that can take a pixel for each channel.

3. Now we define the tonal transference operator as

$$\mathcal{T}[f(x, t)] \stackrel{\text{def}}{=} \beta \bar{D}[f(x, t), t] f(x, t) + (1 - \beta) f(x, t), \quad (8)$$

where  $\beta$  is a free parameter (0.75 for our experiments). Then  $F(x, t) = \mathcal{T}[f(x, t)]$ .



**Fig. 3:** Effect on  $V_M$  of the Tonal Transference Operator,  $T$ , under an illumination change. (a) Previous frame. (b) Current frame.  $V_M$  using (c)  $F = f$ . and (d)  $F = T[f]$ .

In figure (3), we can see a best result of the  $V_M(x, t)$  with the modification of intensity under a change of lighting in the scene.

## 2.4 Casted Shadows Detection

Other error source are shadows casted by foreground objects. We note that casted shadows do not affect the foreground but generates that background pixels are misclassified. Then casted shadows need only be detected in the region that is likely foreground,  $V_M(x, t) < 1/2$ . Hence, the next procedure will update the likelihood model,  $V_M$  computed with (4), by setting  $V_M(x, t) = 1$  for these pixels that corresponds to shadows. Recent reported articles have focused on solving the shadow detection problem, see for instance Refs. [6, 7, 13, 12, 14]. Those procedures are based on transforming the original image in the RGB color space to another color space that better separate the luminance and chrominance components. Thus they implement a series of thresholds for classify the pixels as background, object or shadow. Based on the article [13], where the goal is to detect shadows associated with a moving object. In our proposal we use a maximum likelihood approach for detecting casted shadows. Our procedure is only apply at pixels such that  $V_M(x, t) < 1/2$  and is described as follows:

1. First we calculate in each space (Luminance, Chrominance and Gradient magnitude) the likelihoods of a pixel to be in shadow:

- Luminance channel,  $V_L(x, t) = \exp\left(-\|F(x, t) - \mu(x)\|^2\right)$ .

- Chrominance channel,

$$V_C(x, t) = \exp\left(-\left(|\tilde{F}_1(x, t) - \tilde{\mu}_1(x)| + |\tilde{F}_2(x, t) - \tilde{\mu}_2(x)| + |\tilde{F}_3(x, t) - \tilde{\mu}_3(x)|\right)\right), \quad (9)$$

where  $\tilde{F} = [\tilde{F}_1, \tilde{F}_2, \tilde{F}_3]$  and  $\tilde{\mu} = [\tilde{\mu}_1, \tilde{\mu}_2, \tilde{\mu}_3]$  are the corresponding  $F$  and  $\mu$  transformed into the invariant color space  $c1c2c3$ , see [6] for more details.

- Gradient magnitude,  $V_G(x, t) = \exp\left(-\|F^G(x, t) - \mu^G(x)\|^2\right)$  where  $F^G(x, t) = \|\nabla F\|$  and  $\mu^G(x, t) = \|\nabla \mu\|$  are the smoothed gradient magnitude (with a box filter of  $3 \times 3$  pixels) of  $F(x, t)$  and  $\mu(x)$ , respectively.



**Fig. 4:** Likelihoods and function  $S(x, t)$  applying the SD. (a) Background model. (b)  $f(x, t)$ . (c) Likelihood in the domain of luminance  $V_L(x, t)$ . (d) Likelihood in the domain of chrominance  $V_C(x)$ . (e) Likelihood in the domain of smoothing gradient magnitude  $V_G(x)$ . (f) Combination of likelihoods  $S(x, t)$  and threshold applied.

Note that our formulas for calculating these probabilities are parameters-less, differently to the proposed in Ref. [13] that needs four parameters.

- Next, by assuming independence, we combine the three previously calculated likelihoods:

$$S(x, t) = V_L(x, t) \times V_C(x, t) \times V_G(x, t). \quad (10)$$

- Finally, we update  $V_M(x, t)$  by applying a threshold ( $\theta_2 = 0.92$  for our experiments) to the function  $S(x, t)$ :

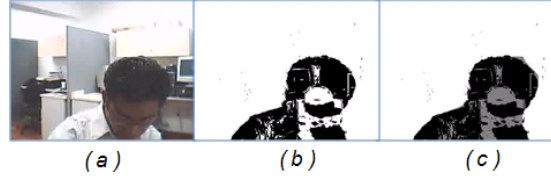
$$V_M(x, t) \leftarrow \begin{cases} 1 & \text{if } S(x, t) > \theta_2 \\ V_M(x, t) & \text{other case.} \end{cases} \quad (11)$$

In Figure 4, we show maximum likelihood estimator on the original  $V_M$  and the updated one with the shadow detection. In spite of the evident improve in the likelihood, there is another possible error source: the similarity between the color of a pixel in the foreground and the color of its corresponding occluded region. In next subsection we focus in to reduce such a problem.

## 2.5 Correction of Camouflage Situation

Static objects (or regions) in the foreground occluding background regions with a similar color can not be detected with the simple model-frame difference. Any visual system (biological or computational) will have similar problems in that camouflage situations. However, if the object is in motion, then the optical flow can provide important clues for detecting camouflaged objects. We propose a camouflage detection procedure to resolve this problem. Our prior is that the probability that each pixel does not undergo a sudden change from one frame to the next one. To introduce this knowledge in our method we do the following:

- We calculate the Optical Flow using the method proposed by [15] in order to detect moving pixels between consecutives frames.



**Fig. 5:** Improvement in the calculation of  $V_M(x, t)$  apply the CCS. (a) Background model. (b)  $f(x, t)$ . (c)  $V_M(x, t)$  without CCS. (d)  $V_M(x, t)$  with CCS.

- Given the displacement field,  $d(x, t)$ , and the the background probability i the previous frame,  $p(x, t - 1)$ , then we modify the likelihood of belonging to the background,  $V_M(x, t)$ , as follows:

$$V_M(x, t) \leftarrow \begin{cases} 1/2 & p(x - d(x), t - 1) < 1/2 \\ & \text{and } V_M(x, t) > 1/2 \\ V_M(x, t) & \text{other case.} \end{cases} \quad (12)$$

This is, using the color based likelihood, we change the value of  $V_M(x, t)$  if the pixel  $x$  in the last frame,  $t - 1$ , was more like to be foreground,  $p(x, t - 1) < 1/2$ , and now is more like to be background. Then we express this uncertainty by updating its likelihood to  $\frac{1}{2}$ . Figure (5) show the  $V_M(x, t)$  using this procedure for correcting the color similarity (CCS). We note that there is an improvement on the inner regions of the object to segment. However, we can see that there are some errors in the contour, those errors depend speed of the foreground in the scene.

### 3 Experiments and Results

To compute the  $V_M$  robust to lighting changes in the scene, shadows and camouflage situations, we have the algorithm 1, which was implemented in Visual C++ 2008, using an architecture GPU (Nvidia GeForce 8800 GT 512 MB, programming language CUDA (Compute Unified Device Architecture)), OpenCV (free software that contains a computer vision library), PC Pentium 4 CPU 3.40 GHz and 512 MB RAM and Windows XP Operating System. This hardware can be considered modest. The experiments were performed with images of 320x240 pixels, acquired in real time from a WebCam and from video files. We used the same set of parameters for all the presented experiments, those values are:

$$[\theta_1, \lambda, \gamma, \text{MaxIter}, \beta, \theta_2] = [0.01, 300, 0.01, 10, 0.75, 0.92].$$

We present the results that we obtain in our scenario (work cubicle), proving that our method is robust to lighting changes, camouflage situations and shadows, the complete method (applying the Tonal Transference Operator (TTO), the Casted Shadows Detection (CSD) and the Correction of Camouflage Situation (CCS). Our system processing



---

**Algorithm 1** Calculate the robust  $V_B$ 

---

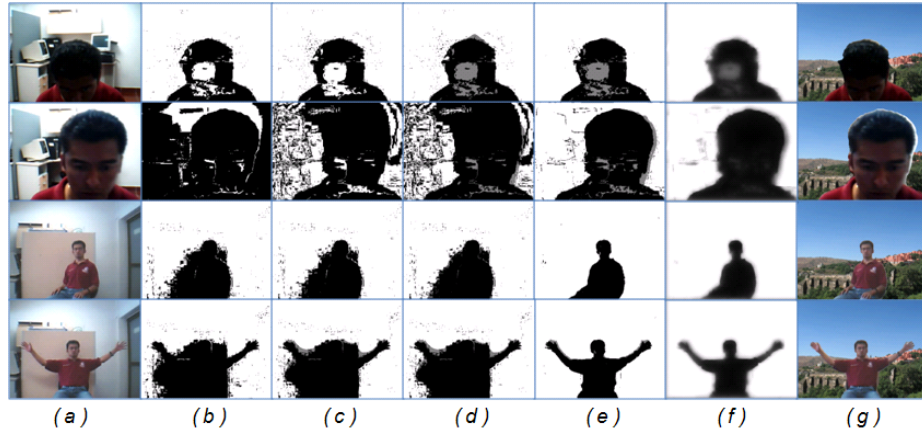
- 1: Calculate the background model (equation 5) with the first N frames.
  - 2: For the next frame, calculate  $V_M$  with the equation (4) and apply the segmentation with QMPF method (3) {simple approach}
  - 3: **while**  $frame <> NULL$  **do**
  - 4:   Apply the Tonal Transference Operator (8) {robust to lighting changes}
  - 5:   Detect casted shadows (11) {robust to shadows}
  - 6:   Correct of Camouflage Situation (12) {robust to color similarity}
  - 7:   Segment with QMPF method (3)
  - 8: **end while**
- 

rate equals 47 frames per second. This time includes the image acquisition, image processing (GPU/CPU) and display of the frame.

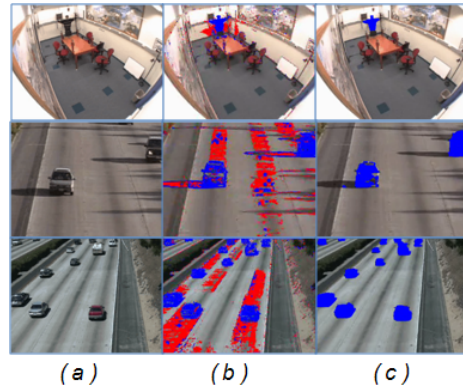
Figure (6) is designed to illustrate the process, this shows the behavior of our method, step by step, under: camouflage situations, lighting changes and shadows. In Panel (a) shows (at time  $t$ ) the frame to be segmented, Panel (b) the simple likelihood between the model and the current frame. Panel (c) shows the  $V_M$  after applying the TTO. Panel (d) presents the  $V_M$  after applying the TTO and the CCS. Next, Panel (e) shows  $V_M$  after applying the TTO, CSD and CCS. Panel (f) shows the final segmentation with our method. Finally, we illustrate the background substitution application in Panel (g).

For the (indoor/out-door) surveillance applications we used the videos of the research team Computer Vision and Robotics UCSD [16], t. In Figure (7) shows some results using the files “intelligentroom\_raw” and “highway\_raw” I and II. Column (b) shows the results of our method before the QMPF binary segmentation (the detected shadows are shown in red). For some applications, the background image can be captured previously, but in other cases it is impractical to wait for a moment in which only show the background of the scene. Note that in highway videos are detected false casted shadows. Those are product of a corrupted background model that averages moving cars. However, our method has not problem in the car detection task. The final segmentation is shown in column (c).

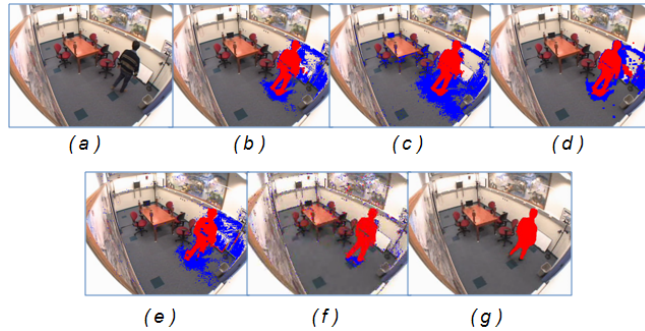
We compare qualitatively our method with other state of the art. In Ref. [17] is presented a comparison of two statistical methods (SNP and SP) and two deterministic (DNM1 and DNM2) for the detection of shadows and objects. They conclude that for a general purpose shadow detection system, the better results was using the method DNM1. For the whole sequence of the file “intelligentroom\_raw” we note that our method detects fewer shadows that the methods mentioned above, however, our method have less mistakes in the segmentation of the person. In Figure (8) we show one of the results reported in [17] and the results that we obtain with our method. Note that compared methods are prone to overestimate the person at the feet region and their solution are not compact.



**Fig. 6:** Background substitution in our scene. (a)  $f(x, t)$ . (b)  $V_M(x)$ . (c)  $V_M(x)$  applying the TTO. (d)  $V_M(x)$  applying the TTO and CCS. (e)  $V_M(x)$  applying the TTO, CSD and CCS. (f) Applying the method QMPF to the robust  $V_M(x)$ . (g) Background substitution.



**Fig. 7:** Surveillance camera and Highway cars detection. (a)  $f(x, t)$ . (b) Method with TTO and CSD, without QMPF. (c) Complete method. Red is Shadow. Blue is Object.



**Fig. 8:** Comparison with others methods using the “intelligentroom\_raw” file. (a)  $f(x, t)$ . (b) SNP method. (c) SP method. (d) DNMI method. (e) DNM2 method. (f) Our method with TTO and CSD, without QMPF. (g) Our Complete method. Red is Object. Blue is Shadow.

## 4 Conclusions

We propose a robust method for binary segmentation of video sequences. The method is designed for foreground-background video segmentation and may be used several application applications as Background Substitution, Analysis of Surveillance Cameras, Highway Cars Detection and so on. Our approach implements a probabilistic segmentation based on the binary Quadratic Markov Measure Fields models (QMMFs). That framework regularizes the likelihood of each pixel to belong to each one of the models (foreground and background). Then our proposal consists of a model for the likelihood that takes into account: an estimation of the static background, motion of the foreground, illumination changes and casted shadows.

Our method address the problems of lighting changes, shadows and camouflage situations. The change lightning produces scene variation respect a background model. Our solution consists of an estimation of the photometric gain for each pixel color value and to adjust the current frame colors. Shadows are detected by adapting the proposal in []. That consist on combine likelihoods of the model and the current frame in a feature vector composed by the luminance, chrominance and gradient magnitude. Different to the original proposal in [] that requires of 4 parameters, our adaptation is parameterless. The camouflage problems (color similarity of the foreground and background) is palliated using optical flow information, this solution assumes the foreground motion is smooth and larger in magnitude than the background motion.

Our implementation runs in real time (47 f/s) using a video card NVIDIA and the programming language CUDA, a modest hardware.

**Acknowledgement.** This work was partially supported by Conacyt, Mexico (grant 61367) and scholarship for F. J. Hernandez–Lopez.

## References

1. Bovik, A.: Handbook of Image and Video Processing. Elsevier Inc. All rights reserved (2005)
2. A. Criminisi, G. Cross, A.B., Kolmogorov, V.: Bilayer segmentation of live video. Microsoft Research Ltd., Cambridge. (2006)
3. Sooyeong, K., Ilkwon, P., Juyong, L., Hyeran, B., Guntae, B.: Automatic background substitution using monocular camera and temporal foreground probability model. *ACM ISBN* (2008) 506–510
4. Stefano M., Carla M. M., N.S., Z., M.: A Kalman filter based background updating algorithm robust to sharp illumination changes. *ICIAP vol* (2005) 163–170
5. Kyungnam K., Thanarat H., D.H., D., L.: Real-time foreground-background segmentation using codebook model. Elsevier Ltd. (2005)
6. Salvador, E., Cavallaro, A., Ebrahimi, T.: Shadow identification and classification using invariant color models. in Proc. Int. Conf. Acoustics, Speech, and Signal Processing. *Vol* (2001) 1545–1548
7. Jianguang, L., Hao, Y., Weiming, H., Tieniu, T.: An illumination invariant change detection algorithm. Asian Conference on Computer Vision (ACCV). (2002)
8. Griesser, A.: Real-time, gpu-based foreground-background segmentation. *VMV Technical Report* (2005)
9. S. FUKUI, Y.I., WOODHAM, R.J.: Gpu based extraction of moving objects without shadows under intensity changes. *IEEE* (2008) 4166–4173
10. Rivera, M., Dalmau, O., Tago, J.: Image segmentation by convex quadratic programming. *Int. Conf. on Pattern Recognition (ICPR08)* (2008)
11. NVIDIA: Cuda zone. Website (2008) [http://www.nvidia.com/object/cuda\\_get.html](http://www.nvidia.com/object/cuda_get.html).
12. Spagnolo, P., Leo, M., D’Orazio, T., Distanto, A.: Robust moving objects segmentation by background subtraction. C.N.R. (2004)
13. Fung, G.F., Yung, N.H., Pang, G.K., Lai, A.H.: Effective moving cast shadow detection for monocular color traffic image sequences. *Society of Photo-Optical Instrumentation Engineers*. **41** (2002) 1425–1440
14. Monteiro, G., Marcos, J., Ribeiro, M., Batista, J.: Robust segmentation process to detect incidents on highways. *ICIAR LNCS* (2008) 110–121
15. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. *IJCAI* (1981) 121–130
16. ATON: Shadow detection. Website (2008) <http://cvrr.ucsd.edu/aton/shadow>.
17. Prati, A., Mikic, I., Trivedi, M.M., Cucchiara, R.: Detecting moving shadows: Algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **25** (2003) 918–923