AN ONLINE BACKGROUND SUBTRACTION ALGORITHM USING A CONTIGUOUSLY WEIGHTED LINEAR REGRESSION MODEL

*Y. Hu**, *K. Sirlantzis**, *G. Howells**, *N. Ragot*†, *P. Rodríguez*‡

* University of Kent, UK
† ESIGELEC, France
‡ Pontificia Universidad Católica del Perú, Peru

ABSTRACT

In this paper, we propose a fast online background subtraction algorithm detecting a contiguous foreground. The proposed algorithm consists of a background model and a foreground model. The background model is a regression based low rank model. It seeks a low rank background subspace and represents the background as the linear combination of the basis spanning the subspace. The foreground model promotes the contiguity in the foreground detection. It encourages the foreground to be detected as whole regions rather than separated pixels. We formulate the background and foreground model into a contiguously weighted linear regression problem. This problem can be solved efficiently and it achieves an online scheme. The experimental comparison with most recent algorithms on the benchmark dataset demonstrates the high effectiveness of the proposed algorithm.

Index Terms— online background subtraction, contiguity

1. INTRODUCTION

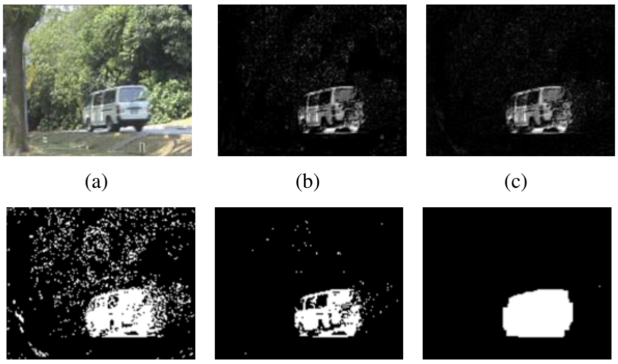
Background subtraction aims to separate the foreground (moving objects) from background in video sequences. It is one of the key steps for many video based applications such as surveillance and navigation. The general framework of background subtraction consists of two components: a background model and a foreground model. The background model estimates the potential background in videos, while the foreground model finds foreground regions by comparing between video frames and the estimated background. Based on this framework, a significant number of algorithms have been proposed [1–11], achieving an impressive performance.

One main challenge of background subtraction is how to accurately model the dynamic background and distinguish between the background changing and foreground motion. Recently, low rank model has shown its power for this challenge. Representative algorithms are Principal Component Pursuit (PCP) [10] and PCP-like online algorithms [12–16]. These algorithms usually assume the potential background images of a video lie in a low rank subspace, and the foreground region is spatially sparse. It decomposes the video into a low rank component as the background and a sparse component as the foreground. However, although the background can be well modeled using a low rank subspace, the foreground is sometimes not only pixel-level sparse but also contiguous regions. Therefore, based on PCP, some algorithms further promote a contiguous foreground detection using models like Markov random field [11] and group sparsity [17]. These models have shown an improved performance, but heavy computational burden is induced with the contiguous foreground model, even using an online scheme [17], restricting the usability of such models in real applications.

In this paper, we propose a fast online background subtraction algorithm detecting a contiguous foreground. The algorithm includes a background model and a foreground model. For background modeling, we use a regression based low rank model. We assume that the potential background images lie in a low rank subspace, so we expect that the background of one frame can be represented as the linear combination of the basis spanning the subspace. Therefore, we represent the background of the current frame as the linear combination of the estimated background of previous frames. In foreground model, we use a contiguity constraint encouraging the foreground to be detected as contiguous regions rather than separated pixels. The proposed algorithm can be solved cheaply in terms of computational load.

An illustration of the proposed algorithm is shown in Fig. 1. Fig. 1(a) shows a frame of a video sequence with a tree waving in the background. We show the deviation between the frame and the background model computed by a recent incremental PCP algorithm [15, 16] in Fig. 1(b), and the deviation of our algorithm in Fig. 1(c). It can be seen that in the background region, the proposed model has a lower deviation. It means that the proposed background model represents better the changing background. In Fig. 1(d) and Fig. 1(e), we show the foreground detection result by applying general hard thresholding to Fig. 1(c). It can be seen that

This research is financed by the European cross-border cooperation program INTERRG IV A France (Channel) - England, the European Regional Development Fund (ERDF), and Cognitive Assisted Living Ambient System (COALAS) Project (Nr. 4194), URL: http://coalas-project.eu.



(a) (b) (c) (d) (e) (f)

Fig. 1: An illustration of the proposed algorithm. (a) A frame of a video sequence with dynamic background (tree waving); (b) the deviation between the frame and the background model in [15, 16]; (c) the deviation between the frame and our background model; (d) the detected foreground with a lower threshold to (c); (e) the detected foreground with a higher threshold to (c); (f) the detected foreground of our algorithm.

a lower threshold (Fig. 1(d)) cannot suppress the changing background, but a higher threshold (Fig. 1(e)) eliminates the foreground pixels similar to the background. In contrast, as shown in Fig. 1(f), our algorithm obtains a more complete foreground due to the promoted foreground contiguity.

Contribution. The contribution of this paper is that we propose an online algorithm for background subtraction detecting a contiguous foreground. We propose a formulation including a background model and a foreground model. In the background model, we represent the background of video frames using linear regression in a low rank background subspace to better model the background changing. In the foreground model, we explicitly formulate the contiguity of foreground, encouraging the foreground to be detected as contiguous regions rather than separated pixels. As this formulation can be solved efficiently, it is able to run in an online scheme.

The rest of this paper is organised as follows. In section 2, we describe the details of the proposed algorithm. In section 3, we report the experimental results. Finally, in section 4, we conclude the paper.

2. THE PROPOSED ONLINE BACKGROUND SUBTRACTION ALGORITHM

2.1. Problem formulation

The problem of online background subtraction can be posed as follows. In a video sequence, given a current input frame $\boldsymbol{y} = [y_1, y_2, ..., y_n]^{\mathrm{T}} \in \mathbb{R}^n$ and the background model estimated using the previous frames, we aim to compute a foreground mask $\boldsymbol{s} = [s_1, s_2, ..., s_n]^{\mathrm{T}} \in \{0, 1\}^n$ for $\boldsymbol{y}$, where $s_i = 1$ if y_i is detected as foreground, otherwise $s_i = 0$.

In background model, we assume the potential background images of a sequence lie in a low rank subspace. Therefore, we represent the background of current frame using the basis spanning the subspace. Specifically, denote the estimated background of the nearest k frames prior to the current frame by $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_k] \in \mathbb{R}^{n\times k}$ ($\boldsymbol{b}_i \in R^n$ is a column vector of the estimated background of one of the frames), we represent the background of the current frame by $\boldsymbol{Bx}$, where $\boldsymbol{x} \in \mathbb{R}^k$ is a column vector of coefficients.

In foreground model, we use two priors for $\boldsymbol{s}$. The first one is the sparsity prior, as generally used in many low rank based background subtraction algorithms [10, 12–17]. The sparsity prior assumes that $\boldsymbol{s}$ is sparse (most of the elements in $\boldsymbol{s}$ are 0). It means the foreground region is small comparing to the background. The other prior is the contiguity prior. It restricts that in $\boldsymbol{s}$, the elements with the same value should be distributed as groups. It promotes the contiguity of the detected foreground regions.

We formulate the above background and foreground model into three terms: a background fidelity term, a foreground sparsity term and a foreground contiguity term.

Background fidelity term. The background fidelity term is defined as follows:

$$f_1(\boldsymbol{x}, \boldsymbol{s}) = \frac{1}{2}\left\|(\mathbf{1} - \boldsymbol{s}) \otimes (\boldsymbol{y} - \boldsymbol{Bx})\right\|_2^2 \tag{1}$$

where $\mathbf{1} \in R^n$ is a column vector of ones and $\otimes$ is element-wise multiplication operator. Minimizing $f_1(\boldsymbol{x}, \boldsymbol{s})$ restricts that all background pixels are represented as the linear combination of the basis in low rank background subspace ($\boldsymbol{B}$).

Foreground sparsity term. The foreground sparsity term is defined as follows:

$$f_2(\boldsymbol{s}) = \|\boldsymbol{s}\|_1 \tag{2}$$

As shown in literature [10, 11], ℓ_1-norm is able to induce sparsity. Hence, minimizing $f_2(\boldsymbol{s})$ leads to a sparse $\boldsymbol{s}$.

Foreground contiguity term. The foreground contiguity term is defined as follows:

$$f_3(\boldsymbol{s}) = \sum_i \sum_{j \in N^i} |s_i - s_j| = \|\boldsymbol{Gs}\|_1 \tag{3}$$

where N^i denotes the neighboring pixels of pixel i, and $\boldsymbol{G}$ is a matrix indicating the neighborship of all pixels. Minimizing $f_3(\boldsymbol{s})$ encourages neighboring pixels in $\boldsymbol{s}$ to have the same value. Thus, the detected foreground will be distributed as groups of pixels.

Combining the three terms, our formulation has the following form:

$$\arg\min_{\boldsymbol{s},\boldsymbol{x}} \frac{1}{2}\left\|(\mathbf{1} - \boldsymbol{s}) \otimes (\boldsymbol{y} - \boldsymbol{Bx})\right\|_2^2 + \alpha\|\boldsymbol{s}\|_1 + \beta\|\boldsymbol{Gs}\|_1 \tag{4}$$

where α and β are penalty parameters. We call it a contiguously weighted linear regression, because the first term is a weighted linear regression, and the third term constrains the weights to be spatially contiguous.

2.2. Algorithm

Equ. 4 has discrete variables in $\boldsymbol{s}$ and it is not convex in both $\boldsymbol{s}$ and $\boldsymbol{x}$. As a result, it is difficult to solve $\boldsymbol{s}$ and $\boldsymbol{x}$ jointly. Instead, we seek a solution by optimizing over $\boldsymbol{s}$ and $\boldsymbol{x}$ alternatively. It leads to a $\boldsymbol{x}$-subproblem and a $\boldsymbol{s}$-subproblem.

$\boldsymbol{x}$-subproblem. Let $\bar{\boldsymbol{s}} = \mathbf{1} - \boldsymbol{s} \in \{0,1\}^n$. With $\boldsymbol{s}$ fixed, minimizing Equ. 4 leads to the following problem:

$$\arg\min_{\boldsymbol{x}} \|\bar{\boldsymbol{s}} \otimes (\boldsymbol{y} - \boldsymbol{B}\boldsymbol{x})\|_2^2 \tag{5}$$

$\bar{\boldsymbol{s}}$ is binary, so we remove from $\boldsymbol{y}$ and $\boldsymbol{B}$ the rows corresponding to the zero elements of $\bar{\boldsymbol{s}}$, and solve a linear system to seek $\boldsymbol{x}$.

$\boldsymbol{s}$-subproblem. Let $\boldsymbol{e} = [e_1, e_2, ..., e_n]^{\mathrm{T}} = \boldsymbol{y} - \boldsymbol{B}\boldsymbol{x} \in \mathbb{R}^n$. With $\boldsymbol{x}$ fixed, we rewrite the objective function in Equ. 4:

$$\begin{aligned} &\tfrac{1}{2}\,\|(\mathbf{1}-\boldsymbol{s}) \otimes (\boldsymbol{y} - \boldsymbol{B}\boldsymbol{x})\|_2^2 + \alpha\|\boldsymbol{s}\|_1 + \beta\|\boldsymbol{G}\boldsymbol{s}\|_1 \\ &= \tfrac{1}{2}\sum_i (\alpha - e_i^2)s_i + \beta\sum_i \sum_{j\in N^i} |s_i - s_j| + \tfrac{1}{2}\sum_i e_i^2 \end{aligned} \tag{6}$$

where the third term can be ignored since it is a constant with $\boldsymbol{x}$ fixed. Equ. 6 is a first order Markov random field with binary labels [18]. It can be solved using graph cuts [19].

With the parameters α and β fixed, alternating between $\boldsymbol{x}$-subproblem and $\boldsymbol{s}$-subproblem leads to a sequence of monotonically decreasing objective function values. Therefore, the algorithm will converge to a local minimum. In our experiment, the algorithm usually converges in 5-10 iterations.

Parameter setting. There are 3 parameters in the proposed algorithm: k which is the basis number in $\boldsymbol{B}$, α and β in Equ. 4. We set k to 10 empirically (see Section 3.2).

As for α and β, borrowing an idea from [11], we update them as follows. In the first iteration, α is set to a large value, $\alpha = 0.5\sigma^2$ where σ is the standard deviation of the current frame. The reason is that $\boldsymbol{B}\boldsymbol{x}$ is an inaccurate background estimation at the beginning of the algorithm (foreground is not fully masked). It will lead to an inaccurate estimation of $\boldsymbol{s}$. Therefore, we apply a large penalty resulting a conservative estimation of $\boldsymbol{s}$. In each iteration, α is reduced by a factor of 0.5, since, along with more foreground region is found, $\boldsymbol{B}\boldsymbol{x}$ becomes more accurate and we relax the penalty to encourage more foreground detection. β is set to 5α in each iteration.

2.3. Background updating

After the foreground of the current frame $\boldsymbol{y}$ is detected, a key problem to achieve an online scheme is how to update the low rank background subspace $\boldsymbol{B}$ for the detections in the following frames. In this paper, we update $\boldsymbol{B}$ based on an incremental PCP algorithm. Let $\boldsymbol{Y}^{'} \in \mathbb{R}^{n\times k}$ be a matrix including the nearest k frames to $\boldsymbol{y}$ with each of its column being a frame. Assuming we have the low rank approximation of $\boldsymbol{Y}^{'}$, we seek the low rank approximation of $[\boldsymbol{Y}^{'}, \boldsymbol{y}]$ by the incremental PCP algorithm in [15, 16]. After the low rank approximation of $\boldsymbol{y}$ is computed, we estimate the background of $\boldsymbol{y}$ as a new basis for $\boldsymbol{B}$ by preserving the detected background pixels in $\boldsymbol{y}$ and replacing the detected foreground pixels by their low rank approximations. Then, we use this new basis to replace the basis corresponding to the oldest frame in $\boldsymbol{B}$.

Now, the problem becomes how to find an initial low rank background subspace so that we can update it frame by frame using the above described method. We employ the incremental initialization algorithm in [15, 16] to seek this subspace as well as the low rank approximation of some initial frames (*i.e.* initial $\boldsymbol{Y}^{'}$). This algorithm works in an incremental scheme so it is much faster than the batch initialization.

We note that a more efficient way to update the background is to use $\boldsymbol{B}\boldsymbol{x}$. We do not choose this way because it exploits little new information additional to the current background subspace. In our experiment, we find that updating $\boldsymbol{B}$ via the former method leads to a better performance.

3. EXPERIMENT

3.1. Dataset and evaluation

We use the I2R dataset[1] [20] which is a benchmark dataset. It includes 9 challenging videos: Bootstrap ($120 \times 160 \times 3057$ frames, crowd scene), Campus ($128 \times 160 \times 1439$ frames, waving trees), Curtain ($128 \times 160 \times 2964$ frames, waving curtain), Escalator ($130 \times 160 \times 3417$ frames, moving escalator), Fountain ($128 \times 160 \times 523$ frames, fountain water), hall ($144 \times 176 \times 3548$ frames, crowd scene), Lobby ($128 \times 160 \times 1546$ frames, switching light), ShoppingMall ($256 \times 320 \times 1286$ frames, crowd scene), WaterSurface ($128 \times 160 \times 633$ frames, water surface). Ground truth of some frames is provided in the dataset. On each sequence, we use the first 200 frames for initial background subspace learning if needed, and we perform background subtraction on the remaining frames.

We evaluate the algorithms by accuracy and speed. The accuracy is evaluated by F-measurement defined as follows:

$$\mathrm{F} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

where $\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$ and $\text{recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$; TP, FN and FP are the number of true positive, false negative and false positive pixels, respectively. We use frames per second (FPS) to evaluate the speed.

3.2. Comparison methods

We report comparisons with the following algorithms: mixture of Gaussian (MoG) [2] as the baseline; Grasta[2] [14], Gosus[3] [17] and incPCP[4] [15, 16] as recent online low rank al-

[1] http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

[2] http://sites.google.com/site/hejunzz/grasta

[3] http://pages.cs.wisc.edu/~jiaxu/projects/gosus/

[4] https://sites.google.com/a/istec.net/prodrig/Home/en/pubs/incpcp

gorithms. Moreover, we compare with two batch low rank methods: PCP[5] [10] as a traditional low rank batch algorithm; Decolor[6] [11] as an improved PCP with the foreground contiguity prior.

Most of the comparison methods need a threshold to produce foreground mask. For these algorithms, we use the first image with ground truth in each video as the training image; we choose the threshold to maximize the F-measurement on the training image, and fix this threshold for the other images. For other parameters in different algorithms (including k in ours), we find an optimal option using the training images and fix it for all the videos. We use fixed parameters for all the videos since it is closer to the scenario of real applications.

3.3. Results and discussions

We report the mean F-measurement and FPS of all the methods on each video in Tab. 1, and we show some example results in Fig. 2. Among the online methods, the one with the highest F-measurement is marked red in Tab. 1, and the second highest is marked blue. It can be seen that our algorithm achieves the highest overall F-measurement among all the online methods. We note that Gosus reports a promising performance with tuned parameters for each video as in [17], but when uniform parameter setting (closer to real scenario) is used, its performance varies on each video and the overall performance drops.

Comparing the accuracy between the online and batch methods, we find that incPCP achieves a comparable F-measurement to the original PCP, and Grasta, Gosus and our algorithm outperform the original PCP. Since Grasta, Gosus and our algorithm adopt a regression based low rank background model, it is reasonable to suggest that this model is preferable comparing to the traditional low rank background model. On the other hand, Decolor outperforms all other methods. The reason is the combination of a contiguity foreground prior and the batch scheme.

As for speed, incPCP is the fastest algorithm. Our algorithm is an online algorithm but it is not real time, because additional computational cost is induced by the foreground contiguity prior modeling a pixel-wise neighboring information. However, comparing to the algorithms using this prior, our algorithm is approximately 7.5 times faster than batch Decolor algorithm and 6 times faster than online Gosus algorithm.

4. CONCLUSION

In this paper, we propose an online algorithm for background subtraction including a regression based low rank background model and a foreground model promoting the foreground contiguity. We formulate the background and foreground model as a contiguously weighted linear regression problem. Experimental results show an improved accuracy comparing to most recent low rank based online algorithms. Future work may consider a region based foreground contiguity model to achieve a better speed.

[5] http://perception.csl.illinois.edu/matrix-rank/sample_code.html

[6] http://fling.seas.upenn.edu/~xiaowz/dynamic/wordpress/?p=144

REFERENCES

[1] N. Friedman and S. Russell, "Image segmentation in video sequences: a probabilistic approach," *Proceedings of the 13th conference on Uncertainty in Artificial Intelligence*, 1997.

[2] G. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[3] A.M. Elgammal, D. Harwood, and L.S. Davids, "Non-parametric model for background subtraction," *6th European Conference on Computer Vision (ECCV)*, 2000.

[4] A. Monnet, A. Mittal, N. Paragios, and V. Ramesh, "Background modeling and subtraction of dynamic scenes," *International Conference on Computer Vision (ICCV)*, 2003.

[5] N.M. Oliver, B. Rosario, and A.P. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 831–843, 2000.

[6] E. Lopez-Rubio, R. Luque-Barna, and E. Dominguez, "Foreground detection in video sequences with probabilistic self-organizing maps," *IEEE International Workshop on Haptic Audio Visual Environment and their Applications*, 2011.

[7] E. Lopez-Rubio and R. Luque-Barna, "Stochastic approximation for background modeling," *Computer Vision and Image Understanding*, vol. 115, no. 6, pp. 735–749, 2011.

[8] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surnelliance applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168–1177, 2008.

[9] C. Zhao, X. Wang, and W. Cham, "Background subtraction via robust dictionary learning," *EURASIP Journal on Image and Video Processing*, 2011.

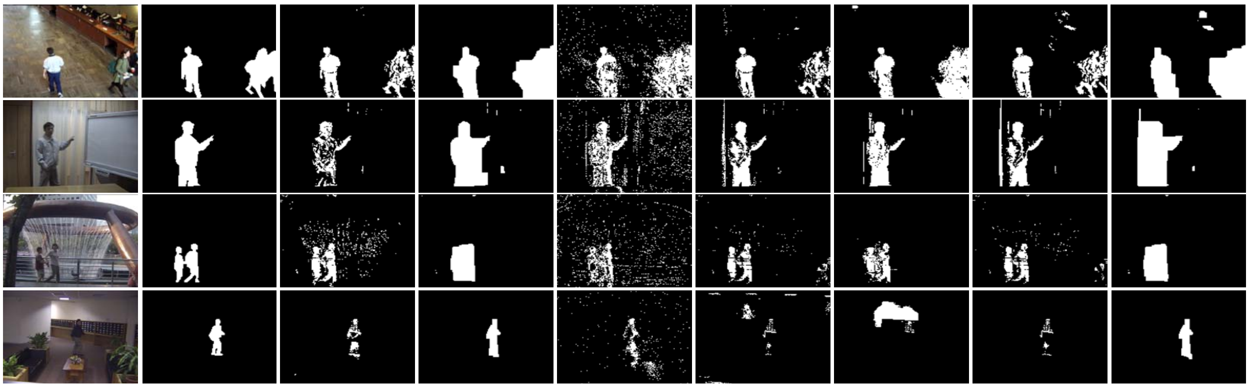
[10] J. Wright, A. Ganesh, S. Gao, and Y. Ma, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 11, 2011.

[11] X. Zhou, Can Yang, and Weichuan Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

Video	PCP* [10]	Decolor* [11]	MoG# [1]	Grasta# [14]	Gosus# [17]	incPCP# [15, 16]	Proposed#
Bootstrap	0.5964 (1.36)	0.6248 (0.17)	0.4240 (2.98)	0.6017 (21.4)	0.6599 (0.81)	0.5289 (27.8)	0.5443 (5.30)
Campus	0.3405 (3.37)	0.7652 (2.55)	0.2115 (2.77)	0.2152 (21.8)	0.1669 (0.97)	0.2092 (26.9)	0.7911 (7.41)
Curtain	0.4927 (1.29)	0.8342 (0.27)	0.4186 (2.65)	0.7816 (21.8)	0.8700 (1.24)	0.6182 (28.1)	0.7615 (5.74)
Escalator	0.6039 (1.12)	0.7183 (0.26)	0.2744 (2.83)	0.4265 (27.0)	0.4058 (0.83)	0.3747 (36.2)	0.6137 (6.00)
Fountain	0.5226 (4.20)	0.8618 (1.63)	0.3875 (2.84)	0.6620 (18.5)	0.6778 (1.31)	0.6502 (22.0)	0.7958 (7.35)
Hall	0.4840 (0.93)	0.5597 (0.14)	0.3603 (2.19)	0.5355 (17.2)	0.4644 (0.67)	0.4744 (26.7)	0.4807 (4.49)
Lobby	0.5833 (1.69)	0.5654 (0.79)	0.3441 (2.64)	0.4059 (19.1)	0.1856 (1.13)	0.4460 (26.2)	0.6375 (6.03)
ShoppingMall	0.5739 (0.73)	0.6800 (0.14)	0.5407 (0.71)	0.6724 (11.8)	0.7152 (0.18)	0.6809 (14.4)	0.6279 (1.65)
WaterSurface	0.3392 (4.65)	0.8873 (0.72)	0.2280 (2.83)	0.7725 (21.3)	0.7873 (1.02)	0.5258 (26.5)	0.5612 (6.40)
Mean	0.5041 (2.15)	0.7219 (0.74)	0.3543 (2.49)	0.5635 (20.0)	0.5481 (0.91)	0.5009 (26.1)	0.6460 (5.60)

*batch (off-line) method   #online method

Table 1: Performance of all methods compared on I2R dataset. Format: F-measurement (FPS).



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 2: Example results of comparison methods. (a) The original video frame (from top to bottom: Bootstrap, Curtain, Fountain, Lobby); (b) the ground truth; (c) PCP; (d) Decolor; (e) MoG; (f) Grasta; (g) Gosus; (h) incPCP; (i) Proposed.

[12] C. Qiu and N. Vaswani, "Support predicted modified-cs for recursive robust principal components pursuit," in *IEEE International Symposium on Information Theory (ISIT)*, 2011.

[13] F. Seidel, C. Hage, and M. Kleinsteuber, "A smoothed lp-norm robust online subspace tracking method for background subtraction in video," *Machine Vision and Applications*, vol. 25, no. 5, 2014.

[14] J. He, L. Balzano, and L. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] Paul Rodríguez and Brendt Wohlberg, "A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2014.

[16] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *submitted to Journal of Mathematica Imaging and Vision*, 2015.

[17] J. Xu, V. K. Ithapu, J.M. Rehg L. Mukherjee, and V. Singh, "Grassmannian online subspace updates with structured-sparsity," *IEEE International Conference on Computer Vision (ICCV)*, 2013.

[18] S.Z. Li, *Markov random field modeling in computer vision*, Springer, 1995.

[19] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximation energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[20] L. Li, W. Huang, I.Y.H. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," *ACM International Conference on Multimedia*, 2003.