Privacy-enabled object tracking in video sequences using compressive sensing

First Author
Institution1
Institution1 address
firstauthor@i1.org

Second Author
Institution2
First line of institution2 address
http://www.author.org/~second

Abstract

In a typical video analysis framework, video sequences are decoded and reconstructed in the pixel domain before being processed for high level tasks such as classification or detection. Nevertheless, in some application scenarios, it might be of interest to complete these analysis tasks without disclosing sensitive data, e.g. *the identity of people captured by surveillance cameras. In this paper we propose a new coding scheme suitable for video surveillance applications that allows tracking of video objects without the need to reconstruct the sequence, thus enabling privacy protection. By taking advantage of recent findings in the compressive sensing literature, we encode a video sequence with a limited number of pseudo-random projections of each frame. At the decoder, we exploit the sparsity that characterizes background subtracted images in order to recover the location of the foreground object. We also leverage the prior knowledge about the estimated location of the object, which is predicted by means of a particle filter, to improve the recovery of the foreground object location. The proposed framework enables privacy, in the sense it is impossible to reconstruct the original video content from the encoded random projections alone, as well as secrecy, since decoding is prevented if the seed used to generate the random projections is not available.*

1. Introduction

Given the consistent reduction of the cost of electronic devices for video acquisition and storage, we have observed the ubiquitous diffusion of video surveillance systems not only in airports, banks or traffic monitoring scenarios, but also within the premises of private houses [12]. These systems enable data gathering about individuals at a level of detail that is far beyond the perceptual capabilities of human observers; furthermore, the constantly improving capacity of digital recording hardware makes the amassed data potentially everlasting and re-usable for tasks different from the original purposes for which it was collected. While the so-obtained enforcement of social security is generally seen as a benefit, the feeling that the privacy of individuals might be seriously violated is getting ahead in the public opinion. For this reason, more and more interest has been recently aroused in the video surveillance research community by systems which enable privacy, though fulfilling at the same time security requirements.

In traditional video surveillance systems, the acquisition process is a preliminary stage for the subsequent *video analysis* step. In order to enable privacy, some systems discard the part of acquired data which is not needed for the analysis but which potentially contains sensitive information (*e.g.* the identity of people). For instance, Chan *et al.* [8] observe that, when the aim of the system is to count people in a crowd, it is not necessary to capture individuals' identity to perform this task. In fact, it could suffice to segment the crowd in order to find the average person dimension, and to track the motion of people in a holistic way regardless of individuals. This analysis-aware approach has been adopted, among others, by Dufaux *et al.* [10], who propose an efficient privacy-enabling technology for Motion JPEG 2000 videos which consists of scrambling the transform-domain coefficients of the regions of interest in a video sequence. Machine learning mechanisms are adopted to identify the regions to protect. Moreover, since the scrambling is done on a pseudo-random basis, it is eventually possible to recover the full scene once the encryption key is made available at the decoder (*e.g.* for use by police authorities).

An alternative approach to privacy protection in video surveillance is to directly acquire video frames in a secure, privacy-enabled format. In this work we propose to leverage the recent findings in Compressive Sensing (CS) theory [1, 4, 9] to sense the scene in a compressed, encrypted form, by computing and transmitting a limited number of random projections of the background subtracted video frames. The most distinctive feature of this work is the fact that the proposed system is able to perform tracking without the need of reconstructing the original video frames. We confine our attention to tracking a single object, using a particle filter applied on a background-subtracted, decimated version of the video frames. However, our system can be extended to handle multiple targets or occlusions by adapting state-of-the-art techniques [19]. We point out that, in the proposed systems, object de-identification is partially achieved by means of both a preliminary decimation stage and the fact that we partially reconstruct only background-subtracted frames. Moreover, on the basis of a recent result by Rachlin and Baron [14] about the secrecy of compressive sensing measurements, we assert that acquiring, transmitting and storing the video sequence in the projection domain is computational *secure*, in the sense that only the knowledge of the random seed used to produce the random projections enables decoding and any kind of further processing. In fact, even if CS cannot achieve a perfect security in the Shannon's sense, it is however secure from a practical point of view, since inverting the system with a wrong key returns a non-sparse solution, and finding a sparse solution by enumeration of all the possible keys is a NP hard problem. Since the simple computation of random projections also embodies the encryption stage (which could be computationally onerous otherwise [17]), the proposed acquisition scheme is suitable for light encoding at each camera. In general, a more practical solution could be to compress video streams using state of the art video encoders, to perform bounding box tracking at a secure, centralized node and then apply strong encryption. While this appears to be very feasible when there are no tight constraints on the resources of the system, in many practical applications one would like to have a light encoder at each camera which is still able to produce compressed and secure data streams. This paradigm is enabled by the proposed system in a natural and intuitive way. Moreover, the system can also be employed with new cameras that directly acquire random projections without sampling the whole image. This is a very attractive feature, since these new devices are deemed by many respectful researches in the CS field as a practical and cheap solution for imaging at wavelengths that are currently impossible to acquire with the more conventional and cheaper CMOS arrays [11, 18]. While preparing this document, we became aware of a very recent work by Cevher *et al*. [7] that shares some similarities with the proposed system. In [7] it is shown that CS can be effectively used to perform background subtraction in the projection domain, provided that the background estimator is linear. In our work we not only propose the idea of applying background subtraction in the projections domain but also develop a well-constructed tracking scheme in which this idea is applied. Besides, we also introduce a weighting term that allows a considerable improvement of reconstruction performance.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the key concepts of CS theory, while Sections 4 and 5 detail, respectively, the background subtraction and the tracking procedures. Finally, Section 6 assesses the tracking performance and Section 7 concludes the paper with some hints for future research directions.

2. Background on Compressive Sensing

Compressive sensing theory asserts that it is possible to perfectly recover a signal from a limited number of incoherent, non-adaptive linear measurements, provided that the signal can be represented by a small number of non-zero coefficients in some basis expansion. Let $\mathbf{s} \in \mathbb{R}^N$ be a k-sparse vector, i.e. just k out of the N elements of $\mathbf{s}$ are nonzero. Suppose we can write the signal to be acquired $\mathbf{x} \in \mathbb{R}^N$ as $\mathbf{x} = \mathbf{\Phi}\mathbf{s}$, i.e. it can be represented by a few basis vector in the orthonormal basis $\mathbf{\Phi}$ using the coefficients $\mathbf{s}$. Let $\mathbf{y} \in \mathbb{R}^n$, $n < N$, a number of linear random projections (measurements) obtained as $\mathbf{y} = \mathbf{A}\mathbf{x}$. If the measurement matrix $\mathbf{A}$ satisfies a *Restricted Isometry Property* (RIP) [4], it can be shown [9] that solving the following optimization problem:

$$\min \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{\Phi}\mathbf{s}. \tag{1}$$

is equivalent to finding the sparsest solution $\mathbf{s}$ to $\mathbf{y} = \mathbf{A}\mathbf{\Phi}\mathbf{s}$, provided that the number of measurements satisfies $n \geq Ck\log(N/k)$. In practice, the RIP is satisfied whenever the columns of matrix $\mathbf{A}$ are incoherent with the basis $\mathbf{\Phi}$ in which the signal is sparse; it turns out that sampling the entries of matrix $\mathbf{A}$ from a Gaussian distribution with zero mean and variance $1/N$ provides a measurement basis which is incoherent with overwhelming probability with any other given basis.

In most practical applications, measurements are affected by noise (*e.g.* quantization noise). Let us consider noisy measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ is a norm-bounded noise, i.e. $\|\mathbf{z}\|_2 \leq \sigma$. An approximation of the original signal $\mathbf{x}$ can be obtained by solving the modified problem:

$$\min \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{\Phi}\mathbf{s}\|_2 \leq \sigma. \tag{2}$$

Problem (2) is an instance of a second order cone program (SOCP) [3] and can be solved in $O(n^3)$ time. Nevertheless, several fast algorithms have been proposed in the literature that attempt to find a solution to (2). In this work, we adopt the SPGL1 algorithm [2], which is specifically designed for large scale sparse reconstruction problems.

A recent work by Candes *et al*. [6] has shown that, by inserting proper weights into the objective function in (2), one can enhance the reconstruction reducing at the same time the number of required measurements. The rationale behind this approach is that, by using some a priori knowledge about the support and the values of the sparse signal, it is possible to direct the reconstruction process towards

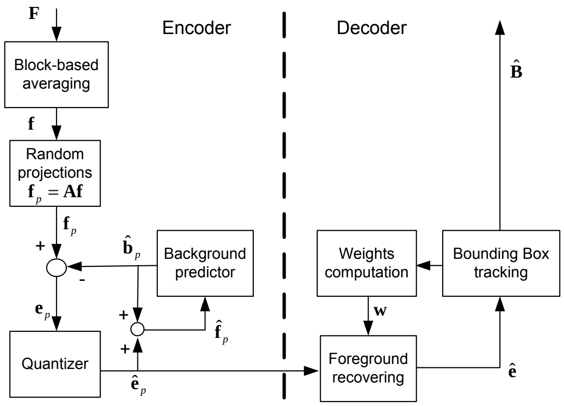


Figure 1. Block diagram of the proposed tracking scheme

the actual nonzero values. Thus, the problem that has to be solved in this case is:

$$\min \|\mathbf{W}\mathbf{s}\|_1 \quad \text{s.t.} \quad \|\mathbf{y} - \mathbf{A}\mathbf{\Phi}\mathbf{s}\|_2 \leq \sigma, \tag{3}$$

where $\mathbf{W}$ is a diagonal matrix with weights $\mathbf{w} = [w_1 \dots w_N]$ on the diagonal chosen to be inversely proportional to the expected signal magnitude. While in the original reweighting approach these weights are updated iteratively, in this work we set them on the basis of the bounding box estimated by the particle filtering, as explained in Section 5.

3. System architecture

The proposed tracking scheme is depicted in Figure 3. The encoder generates the quantized foreground projections $\hat{\mathbf{e}}_p$ as follows:

1. *Block based averaging*: The original frame $\mathbf{F}$ is partitioned into blocks of size $B \times B$. The average of the luminance component of each block is computed and stored in a vector $\mathbf{f} \in \mathbb{R}^N$, where N is the number of blocks in the image.

2. *Random projections*: A number of linear random projections $\mathbf{f}_p \in \mathbb{R}^n$, $n < N$, is produced as $\mathbf{f}_p = \mathbf{A}\mathbf{f}$. The matrix $\mathbf{A} \in \mathbb{R}^{n \times N}$, whose entries are sampled from a Gaussian distribution $\mathcal{N}(0, 1/N)$, is known at the decoder side.

3. *DPCM coding*: The random projections $\mathbf{f}_p$ are coded using a DPCM scheme with an optimal uniform scalar quantizer with step size Δ. The quantization range is adapted depending on the variance of the input signal, so that the distortion is fixed while the rate is variable. The projected background model $\hat{\mathbf{b}}_p$, computed as described in Section 4, is used as a predictor for the random projections $\mathbf{f}_p$ in the DPCM scheme. The resulting quantized foreground projections $\hat{\mathbf{e}}_p$ are finally sent to the decoder.

The decoder works as follows:

1. *Foreground recovering*: a foreground estimate $\hat{\mathbf{e}}$ is obtained by solving the following optimization problem:

$$\hat{\mathbf{s}} = \min \|\mathbf{W}\mathbf{s}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{e}}_p - \mathbf{A}\mathbf{\Phi}\mathbf{s}\|_2 \leq \sigma \tag{4}$$

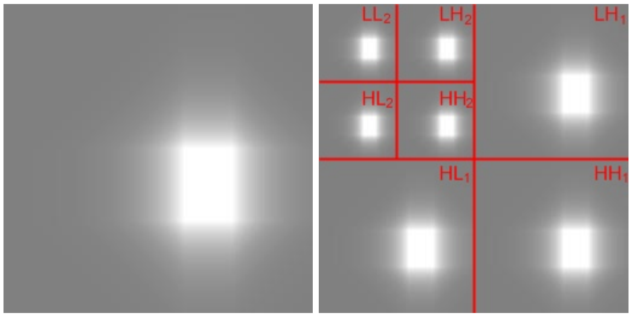
and computing $\hat{\mathbf{e}} = \mathbf{\Phi}\hat{\mathbf{s}}$, where $\mathbf{W}$ is a diagonal matrix with the weights $\mathbf{w} = [w_1 \dots w_N]$ on the diagonal, the parameter σ depends on both the quantization step Δ and the number of projections n as detailed in [5], $\mathbf{\Phi}$ is an othonormal 2D wavelet transformation matrix and $\mathbf{s} = \mathbf{\Phi}^T\mathbf{e}$ is the vector representing $\mathbf{e}$ in the wavelet domain.

2. *Bounding Box tracking*: We use a particle filtering scheme to track the bounding box $\hat{\mathbf{B}}$ enclosing the silhouette of the background-subtracted object. To this end, the foreground estimate $\hat{\mathbf{e}}$ is used as the observed data in order to compute the likelihood of the particles in the update step. In Section 5 we provide further details about the adopted particle filtering approach.

3. *Weights computation*: Besides providing tracking of the moving object, the particle filter prediction of the bounding box $\hat{\mathbf{B}}$ is used as prior information when reconstructing the foreground estimate from random projections. The underlying idea consists in computing the weighting matrix $\mathbf{W}$ in (4) based on the information provided by the bounding box location. Let w_i denote the weighting factor associated with the i-th coefficient of the vector $\mathbf{s}$ in the 2D wavelet domain. The weight value is computed as

$$w_i = \frac{1}{c_i + \epsilon} \tag{5}$$

where the parameter $\epsilon > 0$ has been introduced in order to provide stability and c_i should be (ideally) made proportional to the absolute value of the coefficient s_i [6]. Since s_i is not available, we propose to set the value of c_i depending on the pixel locations corresponding to the i-th wavelet coefficient with respect to the position of the bounding box. If such locations are within the bounding box, we set $c_i = 1$. As for locations outside the bounding box, c_i smoothly decays to zero as they get far from the bounding box. We compute the distances d_x and d_y to the nearest vertical and horizontal bounding box border, respectively. Then the coefficient associated to this pixel is given by $c = e^{-\alpha_W(d_x+d_y)}$, where $\alpha_W > 0$ is a parameter related to the rate of decay of the window. We set the



(a) Window in the frame domain (b) Window in the wavelet domain

Figure 2. Example of window transformation

value of α_W adaptively based on the reliability of the bounding box prediction provided by the particle filter. To this end, we measure the variance of the particles and adapt the value of α_W accordingly. A smaller variance implies a higher confidence in the estimated bounding box, thus the value of α_W can be increased to achieve a sharper decay.

Figure 2 shows the weighting window when a 2D wavelet transform with two decomposition levels is adopted. Figure 2(a) depicts the window coefficients in the pixel domain for a given location of the estimated bounding box. Figure 2(b) shows the values of c_i directly in the 2D wavelet domain. A brighter intensity is assigned to larger values of c_i.

4. Background model

Let $\{\mathbf{f}(1), \mathbf{f}(2) \dots \mathbf{f}(t)\}$ be a sequence of frames from a fixed camera and $\hat{\mathbf{b}}$ an estimate of the scene's static background. Then, at each time instant t, the image of the foreground can be intuitively computed as $\mathbf{e}(t) = \mathbf{f}(t) - \hat{\mathbf{b}}$. The background model cannot be fixed but must adapt to illumination and motion changes, hence it must be continuously updated as new frames are acquired. A very simple and computationally efficient method to do this is the running average method [13]:

$$\hat{\mathbf{b}}(t+1) = \alpha \mathbf{f}(t) + (1-\alpha)\hat{\mathbf{b}}(t) \tag{6}$$

where α is the learning rate. This method can be adopted also when only the frame projections $\mathbf{f}_p(t) = \mathbf{A}\mathbf{f}(t)$ are available, as in the case of cameras that directly acquire random projections without collecting the whole image [11, 18]. In this scenario, let $\hat{\mathbf{b}}_p(t) = \mathbf{A}\hat{\mathbf{b}}(t)$ be the background projections. It comes out that the foreground projections are easily computed as

$$\begin{aligned} \mathbf{e}_p(t) &= \mathbf{A}\mathbf{e}(t) = \mathbf{A}(\mathbf{f}(t) - \hat{\mathbf{b}}(t)) = \mathbf{A}\mathbf{f}(t) - \mathbf{A}\hat{\mathbf{b}}(t) \\ &= \mathbf{f}_p(t) - \hat{\mathbf{b}}_p(t) \end{aligned} \tag{7}$$

while the background projections can still be updated with the running average method [7]:

$$\hat{\mathbf{b}}_p(t+1) = \alpha \mathbf{f}_p(t) + (1-\alpha)\hat{\mathbf{b}}_p(t) \tag{8}$$

5. Bounding Box tracking

A common way to perform tracking consists in formulating it as a stochastic filtering problem. The objective of filtering is to estimate the optimal current state at time t of a random variable $\mathbf{x}(t)$ given the known observations $\{\mathbf{y}(1) \dots \mathbf{y}(t)\}$. The problem can be efficiently solved using a particle filtering approach. In this work we use a Sequential Importance Resampling (SIR) particle filter implementation [15], which requires the definition of a transition model $P(\mathbf{x}(t)|\mathbf{x}(t-1))$ and a likelihood function $P(\mathbf{y}(t)|\mathbf{x}(t))$. In this scenario, $\mathbf{x}(t) = [\mathbf{c}(t), \mathbf{s}(t), \mathbf{u}(t)]^T$ represents the target's state, where the components are the bounding box centroid position, size and velocity, respectively. As already mentioned, the observation $\mathbf{y}(t)$ is set equal to the recovered foreground $\hat{\mathbf{e}}(t)$. In the particle filtering prediction step, we adopt the following transition model [16]:

$$\left\{ \begin{aligned} \hat{\mathbf{c}}_i(t) &= \mathbf{c}_i(t-1) + \mathbf{u}_i(t-1)\Delta_T + \xi_c \\ \hat{\mathbf{s}}_i(t) &= \mathbf{s}_i(t-1) + \xi_s \\ \hat{\mathbf{u}}_i(t) &= \mathbf{u}(t-1) + \xi_u \end{aligned} \right. \tag{9}$$

where $[\mathbf{c}_i(t), \mathbf{s}_i(t), \mathbf{u}_i(t)]^T$ is the state vector associated to the i-th particle, $[\hat{\mathbf{c}}_i(t), \hat{\mathbf{s}}_i(t), \hat{\mathbf{u}}_i(t)]^T$ is the predicted state vector associated to the i-th particle, Δ_T is the time sample interval and ξ_c, ξ_s, ξ_u are random terms which provide the system with a diversity of hypotheses. The bounding box centroid position, size and velocity are estimated in the following way:

$$\left\{ \begin{aligned} \mathbf{c}(t) &= (\mathbf{c}(t-1) + \mathbf{u}(t-1)\Delta_T)(1-\alpha_c) + \\ &\qquad \alpha_c \textstyle\sum \omega_i(t)\hat{\mathbf{c}}_i(t) \\ \mathbf{s}(t) &= \textstyle\sum \omega_i(t)\hat{\mathbf{s}}_i(t) \\ \mathbf{u}(t) &= \mathbf{u}(t-1)(1-\alpha_u) + \alpha_u(\mathbf{c}(t) - \mathbf{c}(t-1)) \end{aligned} \right. \tag{10}$$

where $\omega_i(t)$ is the weight associated to the i-th particle at time t and α_c, α_u are the adaptation rates.

The particles weights are computed according to the following likelihood function:

$$\omega_i(t) = \frac{E_{BB}}{E_{\hat{\mathbf{e}}(t)}} d_{BB}^2 \tag{11}$$

where E_{BB} is the energy of the portion of the recovered foreground $\hat{\mathbf{e}}(t)$ contained in the bounding box associated to the particle, $E_{\hat{\mathbf{e}}(t)}$ is the energy of $\hat{\mathbf{e}}(t)$ and d_{BB} is the density of the bounding box associated to the particle, measured as the number of elements of the bounding box greater than a fixed threshold divided by the total number of elements of the bounding box.

6. Experimental results

We have tested the tracking system with different values of $\delta = n/N$, which represents the fraction of random projections with respect to the original number of pixels. Following the procedure outlined in Section 3, we divide the image into non-overlapping blocks of size 8×8 and assemble the vector $\mathbf{f}$ by computing the average luminance of each block. The resolution of the decimated picture is not high, but for the aim of the system it is adequate to perform tracking. Nevertheless, higher resolutions can be achieved without changing the essence of the proposed system. We have chosen a quantization step Δ such that for $\delta = 0.2$ we have

$$SNR = 10 \log_{10} \frac{\|\mathbf{e}_p\|_2^2}{\|\mathbf{e}_p - \hat{\mathbf{e}}_p\|_2^2} \approx 30dB \qquad (12)$$

corresponding to a mean rate equal to $R = 4.7$ bits per projection. We have verified that a standard PCM coding with the same distortion would have required a mean rate equal to $R = 8.7$ bits per projection, so the DPCM approach allows to save about 4 bits per projection. To evaluate the tracking performance, we threshold the reconstructed foreground $\hat{\mathbf{e}}$ and compare the estimated foreground location with the actual one. By varying the threshold τ it is possible to build a ROC curve for a given number of measurements n and a total rate of $R_T = \delta R$ bits per pixel. We define P_D the probability of detection and P_{FP} the false positive rate and build the ROC curve by plotting P_D against P_{FP} as τ changes.

Figure 3(a) shows the ROC curves for the *hall monitor* sequence, obtained for $\delta = 0.2$. The different curves correspond to different methods used in the computation of the weights vector $\mathbf{w}$:

- *No Prior*: we do not use any tracking system and the following optimization problem is solved instead of Problem (4):

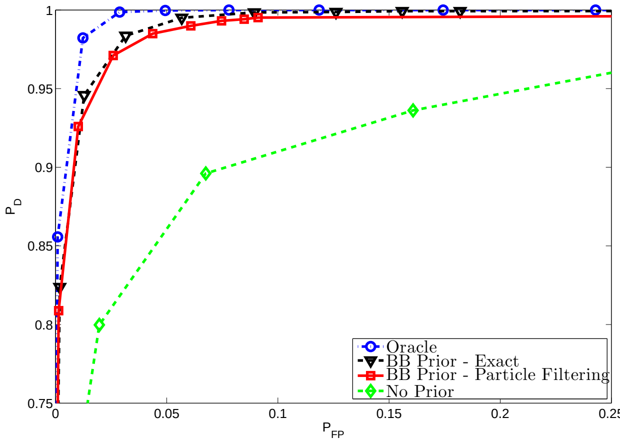
$$\hat{\mathbf{s}} = \min \|\mathbf{s}\|_1 \quad \text{s.t.} \quad \|\hat{\mathbf{e}}_p - \mathbf{A}\mathbf{\Phi}\mathbf{s}\|_2 \leq \sigma \qquad (13)$$

- *BB Prior - Particle Filtering*: we use the proposed tracking system.

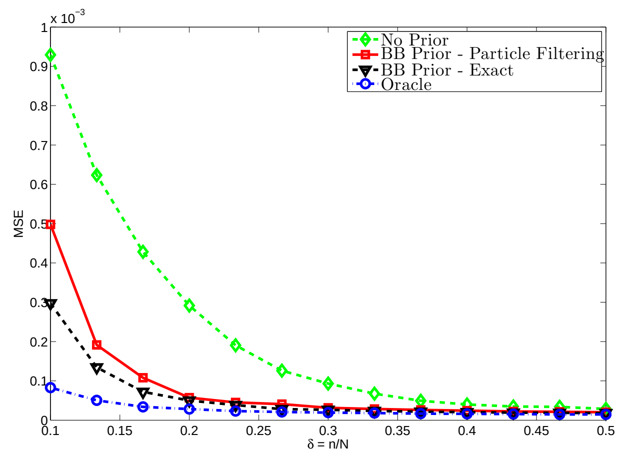
- *BB Prior - Exact*: we do not use any tracking system and compute the weighting window starting from the exact handmade bounding box.

- *Oracle*: we don't use any tracking system and compute the weights starting from the wavelet domain representation of the actual foreground $\mathbf{s} = \mathbf{\Phi}^T\mathbf{e}$. Let s_i be the i-th element of $\mathbf{s}$; the corresponding weight w_i is computed as

$$w_i = \frac{1}{|s_i| + \epsilon} \qquad (14)$$



(a) ROC



(b) MSE

Figure 3. Foreground recovery quality for the *hall monitor* sequence

We observe from the pictures that the presence of prior information allows a significant improvement of performance. Moreover, the curve associated to the proposed tracking system is very close to the one associated to the *BB Prior - Exact* case. This means that the particle filtering scheme described in Section 5 provides a very good estimate of the bounding box identifying the foreground. Nevertheless there is still a small gap between this case and the *Oracle* one, hence it seems reasonable to investigate alternative solutions in order to improve the system and get closer to the optimum.

Similar considerations can be extracted from Figure 3(b), which shows the foreground reconstruction MSE for the *hall monitor* sequence for different values of δ. These figure also points out that, as one could expect, the recovery performance gets better as the number of projections increases.

The MSE plots and the ROC graphs only give an in-direct measure of the system performance. For this reason, we also adopted an evaluation criterion more suitable for the analysis of the tracking capabilities of our application, namely the RMSE between the estimated bounding box centroid position and the actual one. We set $\delta = 0.2$, computed the RMSE for each frame of the sequence and extracted the mean value μ and the standard deviation σ, obtaining a RMSE ($\mu \pm \sigma$) equal to 0.75 ± 0.38 for the *hall monitor* sequence. We can notice that the system is able to identify the bounding box centroid position with an average error lower than one pixel.

We applied the proposed algorithm to other test sequences and obtained very similar results, so that we omit them in order to save space.

7. Conclusions

In this paper we presented a privacy-enabled tracking system that analyses the video frames acquired trough a fixed camera and detects a possible moving object present in the scene. The proposed algorithm enables tracking without the need of reconstructing the original content of the video sequence, thus enabling privacy. In addition, by leveraging compressive sensing, we achieve compression, encoding a limited number of random projections, as well as secrecy.

As a future work, we will extend the system by enabling tracking of multiple targets and occlusions. Moreover, alternative recovering strategies can be studied in order to improve the performance of the decoding stage, thus further reducing the number of random projections needed. For example, we are investigating different ways for exploiting the estimated bounding box position or generating the weighting window based on the object silhouette. Also, the weighting scheme can be adjusted to take into account the decay that characterizes wavelet coefficients in different subbands. We intend to further investigate how to exploit temporal correlations between frames directly in the projection domain. Finally, at the encoding stage, we are studying more sophisticated background subtraction techniques to cope with multimodal background distributions, yet retaining the possibility of operating directly in the projection domain.

References

[1] R. Baraniuk. Compressive Sensing. *Signal Processing Magazine, IEEE*, 24(4):118–121, 2007. 1

[2] E. v. Berg and M. P. Friedlander. In pursuit of a root. Technical Report TR-2007-19, Department of Computer Science, University of British Columbia, June 2007. Preprint available at http://www.optimization-online.org/DB_HTML/2007/06/1708.html. 2

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 2

[4] E. Candes. Compressive sampling. In *International Congress of Mathematicians*, Madrid, Spain, 2006. 1, 2

[5] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207 – 1223, 2006. 3

[6] E. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications, special issue on sparsity*, 2008. 2, 3

[7] V. Cevher, A. Sankaranarayanan, M. Duarte, D. Reddy, R. Baraniuk, and R. Chellappa. Compressive sensing for background subtraction. In *Proc. European Conf. on Computer Vision*, Marseille, France, October 2008. 2, 4

[8] A. Chan, Z. Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, Alaska, 2008. 1

[9] D. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 1, 2

[10] F. Dufaux, M. Ouaret, Y. Abdeljaoued, A. Navarro, F. Vergnenegre, and T. Ebrahimi. Privacy enabling technology for video surveillance. In *Proceedings of SPIE*, volume 6250, pages 205–216, 2006. 1

[11] L. Jacques, P. Vandergheynst, A. Bibet, V. Majidzadeh, A. Schmid, and Y. Leblebici. CMOS Compressed Imaging by Random Convolution. Technical report, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2008. 2, 4

[12] M. T. Moore. Cities opening more video surveillance eyes. *USA Today*, July 18 2005. 1

[13] M. Piccardi. Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 3099 – 3104, 2004. 4

[14] Y. Rachlin and D. Baron. The secrecy of compressed sensing measurements. In *Proc. Allerton Conference on Communication, Control, and Computing*, Monticello, IL, USA, September 2008. 2

[15] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: particle filters for tracking applications*. Artech House, 2004. 4

[16] D. Rowe, I. Rius, J. Gonzles, and J. J. Villanueva. Robust particle filtering for object tracking. *International Conference on Image Analysis and Processing*, 3617:1158–1165, 2005. 4

[17] G. Spanos and T. Maples. Performance study of a selective encryption scheme for the security of networked, real-time video. In *Int. Conf. on Computer Communications and Networks*, pages 2–10, 1995. 2

[18] M. Wakin, J. Laska, M. F. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. F. Kelly, and R. G. Baraniuk. An architecture for compressive imaging. *IEEE International Conference on Image Processing*, pages 1273 – 1276, 2006. 2, 4

[19] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006. 2