Sparsity Driven People Localization with a Heterogeneous Network of Cameras

Alexandre Alahi[1], Laurent Jacques[1,2], Yannick Boursier[1]and Pierre Vandergheynst[1]

Abstract

This paper addresses the problem of localizing people in low and high density crowds with a network of heterogeneous cameras. The problem is recasted as a linear inverse problem. It relies on deducing the discretized occupancy vector of people on the ground, from the noisy binary silhouettes observed as foreground pixels in each camera. This inverse problem is regularized by imposing a sparse occupancy vector, i.e. made of few non-zero elements, while a particular dictionary of silhouettes linearly maps these non-empty grid locations to the multiple silhouettes viewed by the cameras network. The proposed framework is *(i)* generic to any scene of people, i.e. people are located in low and high density crowds, *(ii)* scalable to any number of cameras and already working with a single camera, *(iii)* unconstraint on the scene surface to be monitored, and *(iv)* versatile with respect to the camera's geometry, e.g. planar or omnidirectional.

Qualitative and quantitative results are presented on the APIDIS and the PETS 2009 Benchmark datasets. The proposed algorithm successfully detects people occluding each other given severely degraded extracted features, while outperforming state-of-the-art people localization techniques.

Index Terms

[1]Institute of Electrical Engineering, Signal Processing Laboratory (LTS2), Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland. E-mail: firstname.lastname@epfl.ch

[2]Communications and Remote Sensing Laboratory, Université catholique de Louvain (UCL), B-1348 Louvain-la-Neuve, Belgium. E-mail: firstname.lastname@uclouvain.be

LJ is a Postdoctoral Researcher of the Belgian National Science Foundation (F.R.S.-FNRS). YB is a Postdoctoral Researcher funded by the APIDIS European Project. This work was supported in part by the EU Framework 7 FET-Open project FP7-ICT-225913-SMALL: Sparse Models, Algorithms and Learning for Large-Scale data

Fig. 1: A scene observed by four planar cameras and one omnidirectional camera (extracted from the APIDIS dataset). The green contours represents the degraded foreground silhouettes extracted, and the bounding boxes correspond to the output of our proposed detection algorithm.

People Localization, Sparse Representation, Dictionary, Multi-view, Omnidirectional Cameras, Convex Optimization.

I. Introduction

Accurate vision-based people detection and tracking has been of interest for the past decades in applications like sport game analysis, video-surveillance (e.g. behavior analysis, automatic pedestrian counting).

Isolated people, in an un-cluttered scene, are successfully detected with a single static or moving camera based on pattern recognition techniques. A set of features such as Haar wavelet coefficients [1], [2], histogram of oriented gradient [3], [4] or covariance matrices of a set of features [5], [6], can be extracted from a large number of training samples to train a classifier with a support vector machine [1], [7], or boosting approaches [8], [5]. Given a fixed camera, a moving object can also be detected by modeling the background and tracking becomes simply an object correspondence across frames. Typically, the work of Stauffer and Grimson [9] can be used to extract the foreground pixels. Each pixel is modeled as a mixture of Gaussians with an on-line approximation for the update. Then, detected people can be tracked using standard approaches [10]. A detailed survey on object tracking is proposed by Yilmaz *et al.* in [11]. Porikli in [10] presents a survey on object detection and tracking methods given a single fixed camera. However, those algorithms fail to detect a group of people due to their mutual occlusions. For instance, in sport games such as basketball, players can strongly occlude each other and have abrupt changes of behavior. In order to handle the occlusion problem, several cameras should be fused to correctly detect and track all the people present in the scene.

In this work, a novel framework is proposed to robustly detect moving people occluding each other given severely degraded foreground silhouettes from a set of calibrated pseudo-synchronized cameras (see Figure 1). The only features extracted from the cameras are indeed the binary masks, or *foreground silhouettes*, representing the connected pixels belonging to the foreground of the scene. A single silhouette can correspond to several people due to their dense spatial distribution. It is usually made of many false positives pixels (e.g. shadows, reflections) and false negatives ones (i.e. missing foreground pixels).

Our approach relies on an inverse problem formulation regularized by the assumed "sparsity" of people's location points on the ground floor. Reconstruction methods based on the Basis Pursuit DeNoise (BPDN) [12] and the Lasso algorithms [13] are evaluated. The sparsity measure is reinforced by iteratively re-weighting the ℓ_1-norm of the occupancy vector for better approximating its ℓ_0 "norm" (referred to RW-BPDN and RW-Lasso in the paper). A new kind of "repulsive" sparsity is used to adapt further the Lasso procedure to the occupancy reconstruction (referred to O-Lasso) outperforming other methods. A dictionary made of atoms representing the silhouettes viewed by the cameras network is used within the formulation. Finally, we propose an adaptive process to sample the ground plane in function of both the cameras' topology and the scene activity. We locate people's location points on the ground and propagate the detection results in each camera view.

The proposed approach is *(i)* generic to any scene of people and sensing modality, *(ii)* versatile with respect to heterogeneous cameras network, i.e. able to merge specific camera geometries such as planar and omnidirectional cameras, *(iii)* scalable to any number of cameras and already working with a single camera, *(iv)* robust to people having similar appearance and to abrupt change of behavior (as for sport players), and *(v)* this method does not impose any constraint on the scene surface to be monitored.

To achieve a complete detection system, we provide also a simple graph-driven tracking procedure suited to the particular temporal dynamics of people occupancy vectors. This tracking is based on an iterative method coined Dijkstra Pursuit. It identifies the people tracks by recording the longest geodesics in a graph connecting the non-zero locations of the occupancy vectors across time.

The rest of the paper is structured as follows. First, we recall some important previous works about people detection given multiple cameras. In Section III, our approach is formulated as the

inverse problem of deducing an *occupancy vector* from the noisy binary silhouettes observed as foreground pixels in each camera. We show how this problem can be solved theoretically by regularization, i.e. by using a sparsity prior on the occupancy grid. In Section IV, the dictionary involved in the corresponding forward (or generative) model, i.e. the generation of the observed silhouettes from the occupancy vector, is detailed. Section V explains the particular simplifications we bring to the theoretical methods of Section III to achieve a solvable people localization. First, a re-weighting ℓ_1-norm is presented. Then, a repulsive spatial sparsity constraint is considered with a dynamic update. In parallel, in order to reduce the complexity of the problem, Section VI presents the process used to reduce the dimensionality of the problem, i.e. in the number of observation and in dimension of the search space. The graph-driven tracking procedure is detailed in Section VII. Finally, the performance of our approach is evaluated quantitatively and qualitatively in Section VIII, on synthetic and real data, in comparison with the state-of-the art techniques.

II. Previous Work

In order to deal with a dense spatial distribution of people, and their mutual occlusions, the output of several cameras are used to detect the objects of interest. Robustness with respect to the appearance variability between views is achieved by estimating the object coordinates in a common reference (e.g. ground plane). The unique 'world' coordinates, i.e. the coordinates of the object on the ground plane, is linked to the view coordinates by a planar homography. The planar homography is a 3×3 matrix transformation obtained by matching at least four points from two different coordinate systems. Most systems compute the homographies at an initial calibration step [14]. Stauffer and Tieu's method in [15] rely on tracking data to estimate homography from one camera to another one (correspondence between trajectories). Note that instead of projecting each view on a reference ground plane, some works compute planar homographies between camera views [16], [15], [14]. However, those approaches suffer to solve the occlusion problem.

After projecting all detected objects into a common reference, Mueller *et al.* in [14] mark with the same label the nearest object with the same size and center of gravity. Orwell *et al.* in [17] and Caspi *et al.* in [18] match objects by fusing the estimated trajectories obtained by each camera. However, special care should be applied when using such methods. A point

from the object region in the image coordinate is selected to be projected in other coordinates. Ideally, the foot region should be used. However, some works consider that the center of gravity of the detected object is a reliable approximation. If objects are very far from the cameras, then the approximation is correct. Otherwise, such approximation will lead to poor matching performance. In addition, object segmentation should be perfect. If a person is extracted with its shadow, again, the matching procedure will be affected.

Kim and Davis in [19] take special care to extract the feet region of the foreground people by computing the center vertical axes of the people across views. The axes are mapped to the top-view plane by homography and their intersection point is estimated as the ground point. However, such approaches do not take full advantage of the multi-view infrastructure, as each camera detects the objects independently without helping each other.

Relevant works have decided to neither detect and track objects from each camera, but preferred to gather evidences from all the views and locate in a reference plane. The problem is reformulated as determining the occupied point in the occupancy grid defined by Elfes in [20]. The occupancy grid can be 2-D [21], or even 3-D [22]. It is usually the ground plane or planes parallel to the ground. Yang *et al.* in [23] compute the occupancy grid with a standard visual hull procedure given an upper and lower bound constraint.

Some works locate people's head positions instead of their ground plane locations. Zhao and Nevatia in [24] locate the head locations given a single camera calibration and a head detector. Eshel and Moses in [25] use a set of cameras to better handle occlusions. Those approaches require a good observation of the heads or a good foreground extraction at the head level.

Khan and Shah in [26] pay attention to extract the feet region of the foreground people. Each point of the foreground likelihood (foreground silhouettes) from all views is mapped to the ground plane given a planar homography. Multiplying the mapped points segments the pixel corresponding to the feet of the people. Their approach can not be applied to an object viewed by one camera. In addition, a poor foreground segmentation - people detected with their shadow or missing foreground pixels - affects the performance of their system. To handle such noisy segmentation, they apply their approach to multiple planes parallel to the reference plane in [27]. Delannay *et al.* in [28] use the same approach, i.e. they also project the foreground likelihood on multiple planes parallel to the ground. They combine such process with a heuristic step to handle the non-linearity induced by occlusion. However, wrapping the foreground silhouettes on

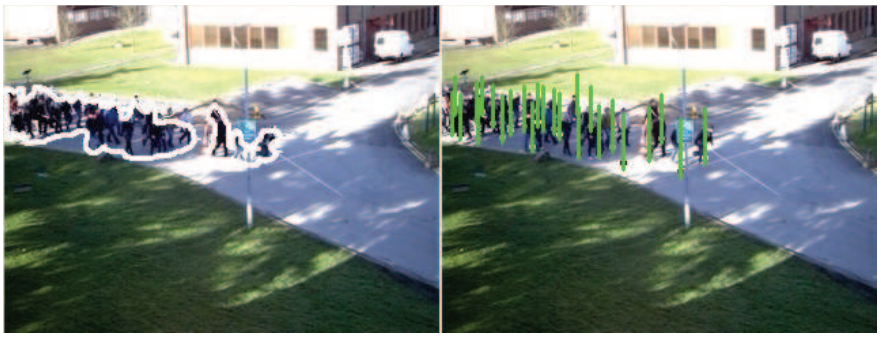


Fig. 2: People localization with a single camera. Left side: contour (in white) of the foreground silhouette extracted by the camera. Right hand-side: Located people by our proposed algorithm given the silhouette extracted.

reference planes do not allow to locate grouped of people specially when a single camera is used such as in Figure 2.

More recently, Reddy *et al.* in [29] use compressed sensing to detect and track people in a multi-view setup. They use the sparsity of the observations, i.e. the foreground silhouettes extracted from the cameras. However, their sparsity constraint depends on the distance of the objects to the cameras. Objects close to the cameras will unfortunately generate large foreground silhouettes with poor sparsity. To accurately estimate the position of the objects on the ground plane multiple cameras are needed. No dictionary is used to model the presence of a person. Also, the complexity cost of their algorithm depends on the number of ground plane points, the grid size, to be evaluated.

Fleuret *et al.* in [21] take advantage of the multi-view infrastructure to accurately track people across multiple cameras given degraded foreground silhouettes. They develop a mathematical framework to estimate the probabilities of occupancy of the ground plane at each time frame with dynamic programming to track people over time. They approximate the occupancy probabilities as the marginals of a product law minimizing the Kullback-Leibler divergence from the true conditional posterior distribution (referred to as Fixed Point Probability Field algorithm). They are able to detect people occluding each other given noisy observation. We will consider their work as the state-of-the-art and compare it with our proposed algorithm in Section VIII since both approaches have a generative model and try to minimize the difference between a synthetic image and the observed image. However, their mathematical framework does not explicitly consider

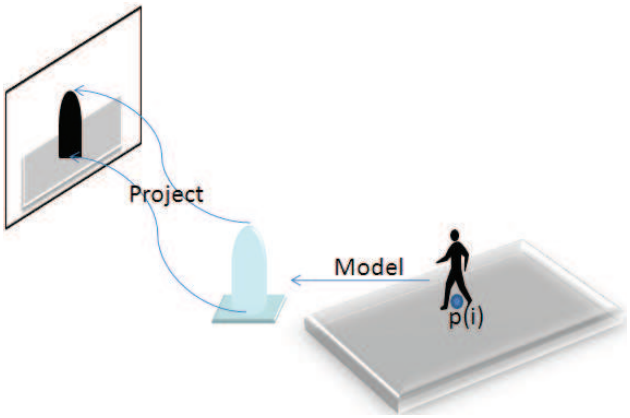


Fig. 3: To each point $p(i)$ corresponds a silhouette modeling the presence of a person in a camera view

the sparsity of the desired solution leading to potentially high false positives rate. In addition, the computation cost of their algorithm depends on the number of ground plane points to be evaluated, leading to a limited area to be monitored.

We propose a framework to cope with the limitations of previous works. It scales to any number of cameras. A single camera can also be used whereas previous multi-view approaches could not be applied to group of people viewed by a single camera. We do not have any constraint on the surface to monitor. Omnidirectional cameras can also be integrated to the system. We used severely degraded foreground silhouettes representing realistic scenarios. Foreground silhouettes are made of many false negative and positive pixels. Finally, we explicitly consider the sparsity present in the desired solution during the detection process similar to other sparsity-based algorithms used for localization [30], [31], [32]. The strength of the proposed approach is quantitatively and qualitatively presented in Section VIII.

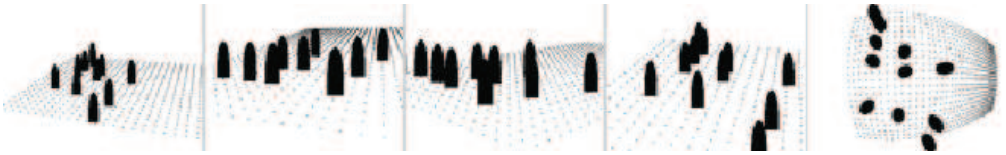
III. Conventions and Problem Formulation

The objective of this paper is to deduce the ground plane points occupied by the people present in the scene given the foreground silhouettes provided by a set of C calibrated cameras (planar or omnidirectional).

To simplify notations, we will often refer to two-dimensional objects, e.g. the grid of occupancy or a given camera view, as 1-D vectors, i.e. the vectors obtained for instance by the concatenation of the columns of these 2-D objects. This will allow us to model easily the construction and the

(a) Camera Views

(b) Degraded foreground silhouettes y



(c) $\hat{y} = Q(Dx)$

Fig. 4: Illustration of the atoms modeling the given foreground silhouettes. The grid is only for visual purposes.

action of some important linear operators such as the Multi-Silhouettes Dictionary described in Section IV.

Up to the selection of an appropriate background subtracting method, we assume that at a given time, each camera is the source of a binary silhouette image $y_c \in \{0,1\}^{M_c}$, where $M_c \in \mathbb{N}$ is the number of pixels (resolution) of each camera indexed by $1 \leq c \leq C$. Stacking all these vectors gives the Multi-Silhouette Vector (MSV)

$$y \;=\; (y_1^T, \cdots, y_C^T)^T \;\in\; \{0,1\}^M,$$

with $M = \sum_{c=1}^{C} M_i$.

The continuous ground plane is discretized in a 2-D grid of N sub-areas (or *cells*). The presence (or *occupancy*) of people on the ground is therefore represented by the binary vector $x \in \{0,1\}^N$, coined *occupancy vector*, with $x_i = 1$ meaning that the i^{th} cell is occupied by somebody. The index i of each component x_i of x is actually linked to a particular position $\mathbf{p}(i) \in \mathbb{R}^2$ on the ground plane in the center of one cell. For simplicity, we assume that one and only one observed person is exactly supported by one subarea of this grid.

Notice that, as explained in Section VI, the 2-D grid underlying the occupancy vector is actually not regular. It is adaptively built in function of the cameras' topology and the scene activity. This adaptive sampling process is described in Section VI-B.

Assuming that a person is represented by an invariant volume, it is clear that any configuration of x will correspond to a particular configuration of silhouettes in y. For instance, if x contains only one non-zero component, all y_c observing the object will contain one silhouette (i.e. a connected area of non-zero pixels) with size and location related to the particular projective geometry combining the scene and the cameras (see Figure 3).

Our inverse problem is thus to find x from y assuming that x is a sparse vector, i.e. it is composed of few non-zero components compared to N. However, the difficulty in the resolution of this problem arises from its non-linearity, i.e. the vector y is binary and it does not contain any information about possible occlusion between persons. In addition, the background subtracting methods leading to the silhouette definition are severely degraded (e.g. light reflection, shadows, and noise).

To bypass these two difficulties, we propose to handle them both as a noise on some linear observation obtained by a generative (forward) model described hereafter.

IV. Forward Model and Multi-Silhouette Dictionary

Our forward model that associates to the occupancy vector $x \in \mathbb{R}^N$ a certain configuration of occluding silhouettes in the cameras is the quantization of a linear operator. More precisely, we obtain it from the one bit quantization of a *dictionary* $D \in \mathbb{R}^{M \times N}$ multiplied by x.

The Multi-Silhouette (MS) dictionary D is one of the key ingredient of our approach. It is made of atoms modeling the presence of a single person at a given location. By construction, it maps non-empty locations of the occupancy vector to a linear approximation of the multiple silhouettes viewed by the cameras network. In other words, each atom approximates the silhouette generated by a single person in all the camera views. The columns of D, i.e. the atoms, live thus in the same space as the observed Multi-Silhouette Vector (MSV), i.e. in a space of $M = \sum_{c=1}^{C} M_c$ dimensions.

Mathematically, the *Forward Model* generating silhouettes is thus the application of D on the occupancy vector x, i.e. $D\,x$. Of course, by linearity, the components of $D\,x$ are not binary. They are higher than one each time two or more silhouettes occlude. A more faithful, but non-linear,

forward model, is therefore achieved by applying a quantization operator $Q : \mathbb{R}^N \to \{0,1\}^N$ on $D\,x$, with $(Q[v])_i = 1$ if $v_i \neq 0$ and 0 else. We will develop further the use of these two forward models in Section V.

The dictionary $D \in \{0,1\}^{M \times N}$ can also be seen as the merging of all the sub-dictionaries $D_c \in \{0,1\}^{M_c \times N}$ made of the index restriction of the atoms of D to the pixel range of each camera c for $1 \leq c \leq C$. Therefore,

$$D = (D_1^T, D_2^T, \ldots, D_C^T)^T \tag{1}$$

meaning implicitly that there is no theoretical constraint on the number or on the type of camera used, e.g. planar or omnidirectional.

Practically, the atoms of each D_c are generated *(i)* thanks to the homographies mapping points in the 3-D scene to their 2-D coordinates in the planar view, and *(ii)* thanks to the approximation of the silhouettes by simple shapes (e.g. rectangular or elliptical shapes). Indeed, to cope with the various poses and shapes a person can generate in a camera view, a half-cylinder-half-spherical shape is used to approximate the silhouette of a person in the views (see Figure 3). Figure 4 illustrates an example of severely degraded foreground silhouettes (made of shadows, people's reflection, missed regions) and the silhouettes used to model their presence in the set of planar and omnidirectional cameras.

Note that the shape used to generate the atoms does not affect the computation complexity of the approach since the dictionary is computed off-line. We do not need to use rectangular shape as in [21] to take advantage of integral images.

V. Occupancy Reconstructions

A. Ideal formulations

The ill-posed problem of reconstructing the occupancy vector x from the observed data y can be regularized by the a priori sparsity of x.

A first approach is to use the following theoretical optimization problem, i.e. ℓ_0-Regularized problem:

$$\underset{x \in \{0,1\}^N}{\arg\min} \ \|x\|_0 \ \ \text{s.t.} \ \ \|y - Q(Dx)\|_2^2 < \varepsilon \tag{2}$$

where $\|x\|_0 = \#\{i : x_i \neq 0\}$, ε is the desired residual error and the quantization Q is defined in Section IV.

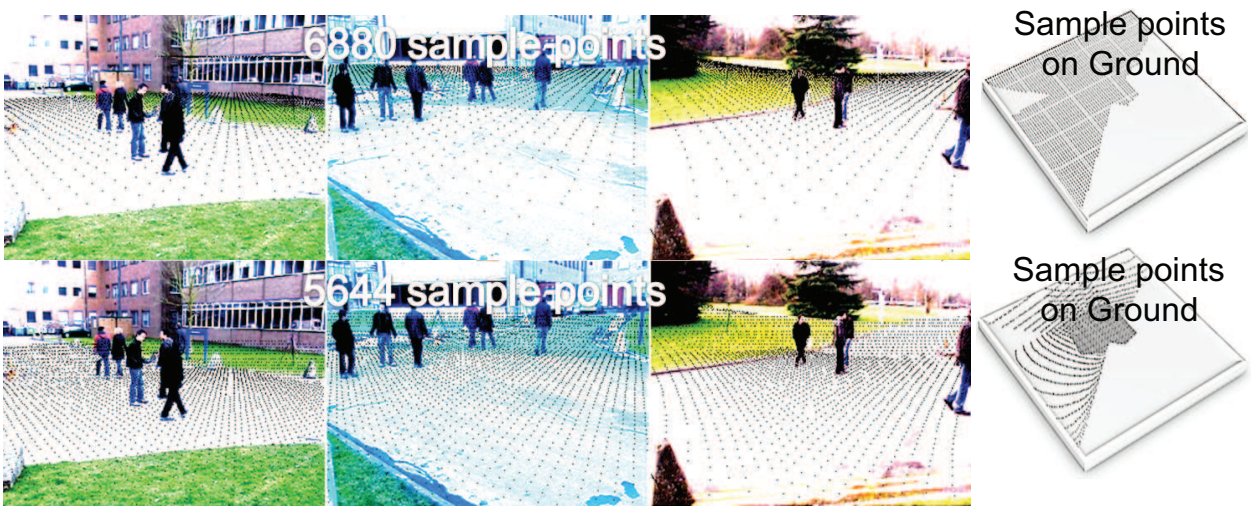


Fig. 5: Illustration of the adaptive sampling process. Top row: sample points given a regularly spaced grid. Bottom row: proposed non-regular grid

Such formulation does not require to know the sparsity of the vector x but needs an upper bound on the residual error for the fidelity term. Since the degradations on the foreground likelihood are not predictable, another alternative is to bound the sparsity, i.e. reformulate the reconstruction method as the ℓ_0-Regularized following problem that can be compared to the Lasso problem [13]:

$$\underset{x\in\{0,1\}^N}{\arg\min} \ \|y - Q(Dx)\|_2^2 \ \ \text{s.t.} \ \ \|x\|_0 < \varepsilon_p \tag{3}$$

where ε_p is the maximum number of people to be detected.

In Section VIII, we quantitatively measure the strength of both reconstruction methods.

These optimizations are non-convex and also NP-hard [33], i.e. the numerical complexity is combinatorial in the dimension of the space. This is due to the use of the ℓ_0 sparsity term and to the discrete nature of the binary space $\{0,1\}^N$. We need therefore to simplify these formulations, which we do in the next sections.

B. Linearized and Re-Weighted Optimizations

To linearize (2) and (3), we first remove the quantization operator Q from their fidelity terms. This amounts to consider possible object occlusions has an additional noise on the measured silhouettes, increasing therefore the value of ε and ε_p. Second, the vector x is now considered

in $\mathbb{R}_+^N$ and not in the binary space $\{0,1\}^N$, while the reconstructed vector will be subsequently one-bit quantized to form the binary occupancy vector x. An adaptive threshold T is used driven by the maximum value of x. Typically, one-fourth of its value is used as a threshold ($T = 0.25$).

Interestingly, in the minimizations (2) and (3), the ℓ_0-norm can be approximated with an iterative re-weighted ℓ_1-norm. The weights used for the next iteration are computed from the value of the current solution as introduced by Candès *et al.* in [34].

Explicitly, (2) leads to the Re-Weighted Basis Pursuit DeNoise (RW-BPDN) program, i.e

$$x^{(l+1)} \;=\; \arg\min_{u\in\mathbb{R}_+^N} \|W^{(l)}u\|_1 \text{ s.t. } \|y - Du\|_2 < \varepsilon, \tag{4}$$

while (3) provides the Re-Weighted Lasso (RW-Lasso), i.e.

$$x^{(l+1)} \;=\; \arg\min_{u\in\mathbb{R}_+^N} \|y - Du\|_2^2 \text{ s.t. } \|W^{(l)}u\|_1 < \varepsilon_p, \tag{5}$$

where, for both equations, the diagonal weighting matrix is defined at each iteration $l > 0$ by $W_{ii}^{(l)} = (|x_i^{(l)}| + \eta)^{-1}$, for $1 \leq i \leq N$, with $W^0 = \mathrm{Id}$ and the corresponding previous solution $x^{(l)}$. The parameter η is added to assure stability and guarantees that a zero-valued component in x does not strictly prohibit a nonzero estimate at the next iteration. We set $\eta = 10^{-7}$.

Practically, as explained in Appendices A and B, at each iteration of the re-weighted process, (4) and (5) are solved by monotone operator splitting and proximal methods [35], [36].

C. Occupancy Lasso (O-Lasso)

In this section, we specialize further the Re-Weighted Lasso procedure (5) to the particularities of our occupancy reconstruction. As explained below, this involves the addition of two processings in the reweighting loop.

1) Repulsive Spatial Sparsity (RSS): Although the re-weighted ℓ_1-norm provides a sparse solution close to the one that would have been obtained with the true ℓ_0 "norm", it does not enforce a certain form of spatial sparsity desired in our application. Indeed, taking a simple example, the linearity of our formulation allows two (or more) neighboring points in x to have non-zero values so that the generated silhouette Dx fits a single person with a shape slightly larger than what is prescribed in the dictionary model.

We want however to avoid such situation and allow the occupancy reconstruction to spend more effort on the reconstruction of other isolated persons in x. We impose therefore that two

TABLE I: Greedy RSS Projection

Input: A sparse vector z.

Output: An approximation of $\mathrm{Proj}_{\mathcal{R}_\tau} z$.

Program:

1) Initialize: $r \leftarrow z$, $p \leftarrow 0 \in \mathbb{R}^N$.

2) Pick the index $i^* = \arg\max_i |r_i|$.

3) Set $p_{i^*} \leftarrow r_{i^*}$, and then $r_{i^*} \leftarrow 0$.

4) For all $j \in \mathrm{supp}\{r\}$, if $\Delta_{i^* j} < \tau$, set $r_j \leftarrow 0$.

5) If $\mathrm{supp}\{r\} = \emptyset$, return p and stop; else, return to Step 2.

detected points, i.e. with non-zero components in x, are never closer than a minimum spatial distance related to the minimum surface occupied by a person on the ground[1]. This is what we call the concept of Repulsive Spatial Sparsity (RSS).

Mathematically, if $j, k \in \mathrm{supp}\{x\} = \{i : x_i \neq 0\}$ with $j \neq k$, we should have

$$\Delta_{jk} \triangleq \|\mathbf{p}(j) - \mathbf{p}(k)\|_2 > \tau \tag{6}$$

where $\mathbf{p}(k)$ is the location of the k^{th} cell in the discrete ground plane. We choose a typical value of $70\,\mathrm{cm}$, i.e. the average width of a standing person, for the minimal spatial distance τ between two occupied ground points.

We achieve this result by inserting in the iterative algorithm, the non-convex projection

$$p = \mathrm{Proj}_{\mathcal{R}_\tau} z \triangleq \underset{x \in \mathcal{R}_\tau}{\arg\min} \|x - z\|_2,$$

of the current solution z on the Repulsive Spatial Sparsity (RSS) set

$$\mathcal{R}_\tau = \{x \in \mathbb{R}^N : \forall\, j, k \in \mathrm{supp}\{x\} \text{ s.t. } j \neq k, \Delta_{jk} > \tau\}.$$

Practically, we approximate $p = \mathrm{Proj}_{\mathcal{R}_\tau} z$ by the suboptimal greedy method detailed in Table I.

[1]Notice this approach applies to other objects detection, e.g. cars or traffic, with non-zero ground-surface.

This method converges in at most $n = \|z\|_0$ iterations, i.e. when we have already $z \in \mathcal{R}_\tau$, and by construction the output belongs to $\mathcal{R}_\tau$. In addition, this greedy method implicitly preserves the highest non-zero component of z amongst two or more non-zero components within a distance τ of the highest. This guarantees a sub-optimal minimization of the distance $\|z - p\|_2$. Notice that the Step 2 can be efficiently realized by sorting the non-zero elements of z by decreasing magnitudes, a process that takes at most $O(N \log N)$ operations. Indeed, Step 3 in Table I inserts just some zeros in the sorted z and the selection of the maximum in Step 2 can be realized sequentially, taking each time the next non-zero element in the sequence.

2) Adaptive Sparsity Level: The Lasso formulation requires the knowledge of the number of people present in the scene. In order to make the algorithm generic enough, we propose an adaptive sparsity constraint selection in Equation (5). Since the sparsity constraint, ε_p, bounds the potential number of detected people, an iterative approach is proposed to choose the constraint independently of the number of people present in the scene. First, ε_p is high enough in order to have a first approximated solution. Typically, ε_p is initialized to the dimension of x (see Section VI). Then, ε_p is set to the number of non-zeros values obtained after each iteration. Experiments show that the algorithm converges towards the right number of people present in the scene when the Repulsive Spatial Sparsity constraint is used.

The final iterative algorithm, including both the Repulsive Spatial Sparsity and the Adaptive Sparsity level, is summarized in Table II. It is coined the Occupancy Lasso (O-Lasso). We will see in Section VIII that it outperforms RW-BPDN, RW-Lasso and the state-of-the-art method.

VI. Dimensionality Reduction

If many cameras are used, the dimensions of y and x become an issue in Equations (4) and (5). These sizes define indeed both the dimensionality of D, which requires a large memory storage, and the total computational time of the algorithms. There exist however some possibilities of dimensionality reductions that we detail below.

A. Dimensionality reduction on the observations

The dimension of the observation vector y is by default equal to the sum of each camera resolution. To reduce the computation cost, all images are first down scaled to a QVGA resolution

TABLE II: Occupancy Lasso

Inputs:

- The Multi-Silouettes Vector $y = (y_1^T, \cdots, y_C^T)^T \in \mathbb{R}^M$.
- A set of ground point locations (cell):
$$\{\mathbf{p}(j) : 1 \leq j \leq N\}.$$
- A stopping tolerance Tol (e.g. $Tol = 10^{-4}$).

Output: The occupancy vector $x \in \mathbb{R}^N$.

Program:

1) Initialize: $l = 0,\ x^{(0)} = 0,\ W^{(0)} = \mathrm{Id},\ \varepsilon^{(0)} = \|x\|_0$

2) Solve:
$$z^{(l+1)} \ = \ \underset{u \in \mathbb{R}_+^N}{\arg\min} \ \|y - Du\|_2^2 \text{ s.t. } \|W^{(l)}u\|_1 < \varepsilon^{(l)},$$

3) RSS Projection:
$$x^{(l+1)} \ = \ \mathrm{Proj}_{\mathcal{R}_\tau} z^{(l+1)}$$

4) Re-weight: Define the diagonal matrix $W^{(l+1)}$ by
$$W_{ii}^{(l+1)} = \left(|x_i^{(l+1)}| + \eta\right)^{-1},\ 1 \leq i \leq N$$

5) Dynamic constraint: $\varepsilon^{(l+1)} = \|x^{(l+1)}\|_0$

6) Stop if
$$\frac{\|x^{(l+1)} - x^{(l)}\|_2}{\|x^{(l+1)}\|_2} < Tol,$$
else, set $l \leftarrow l + 1$ and return to Step 2.

(320×240). A background subtraction algorithm extracts foreground silhouettes on the QVGA resolution. Then the image plane of each camera view is cropped to the region where people can occur. Finally, all images, y_c, are normalized to the same size (107×80).

B. Dimensionality reduction in the search space

The complexity cost depends on the number N of ground plane points to locate as occupied or not. Fleuret *et al.* in [21] discretize the visible part of the ground into a fixed number of points regularly spaced. They do not consider the resolution of the cameras and the sparsity of the people present in the scene to discretize the ground. In this work, we address these considerations.

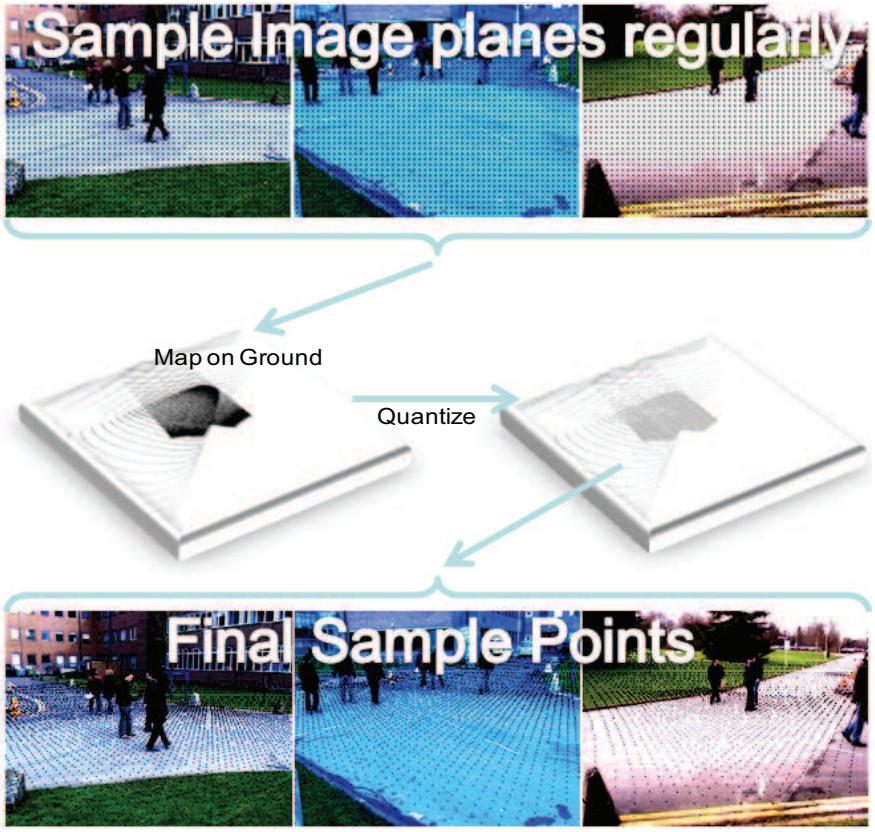


Fig. 6: Overview of the adaptive sampling process

Two different ground plane points can correspond to the same pixel in the image plane of a camera. A translation of one pixel in the image plane can be equivalent to a translation of a few meters on the ground plane for far away regions, just as a translation of a few centimeters on the ground plane can correspond to a shift of several pixels for closer regions, depending on the resolution of the camera and the distance of the objects to the camera. Therefore, we propose a non-regularly spaced sampling process to discretize the ground (see Figure 6). Points regularly

spaced in the image plane of all cameras are mapped to the ground to form the *sample points*. The mapped location points are quantized to avoid points spaced with less than few centimeters. Figure 5 compares a regularly spaced grid with our proposed non-regular grid. Although our proposed grid has less number of points, regions of interest have higher density of points, i.e. higher spatial resolution. In order to have the same spatial resolution in the region of interest with a regular grid, 42777 are needed compared to 5644 with our proposed non-regular grid. A further reduction in the search space can be achieved by measuring the activity of a *sample point* according to three possible assumptions.

Assumption 1 (Foreground pixels only): *Sample points are ground plane points belonging to the foreground pixels of at least one camera.*

In this assumption, each foreground pixel represents the potential feet location of the people. Each image plane is downscaled to reduce the sample points as explained in Section VI-A. Given the calibration data, each point of each camera is mapped to a ground plane point sampling x. In order to be certain to not miss a potential ground point, each foreground pixel is also considered as the upper limit (the head) of a person. Therefore, missing the feet region in the foreground will not affect the sampling process.

Assumption 2 (Intersecting foreground pixels): *Sample points are ground plane points belonging to the foreground pixels of all the cameras observing the corresponding points.*

Assumption 2 is similar to the work of Khan and Shah in [27]. However, such step may be affected by degraded foreground silhouettes extraction, e.g. including shadows (see Figure 7). Missing foreground pixels in some views can lead to missing people whereas high false positive foreground pixels induce high false positive rates.

Assumption 3 (Least significant silhouette): *Sample points are ground plane points corresponding to a significant foreground silhouette in all the cameras observing the corresponding points.*

In this last assumption, the sample points x are kept if

$$\forall c \in C : \frac{y_c^T Q(D_c x)}{\|D_c x\|_0} > \delta. \tag{7}$$

Typically, δ is set to 20%. The operator Q is the one bit quantizer defined in Section IV.

The constraint δ represents the minimum amount of foreground pixels needed to keep a sample point.

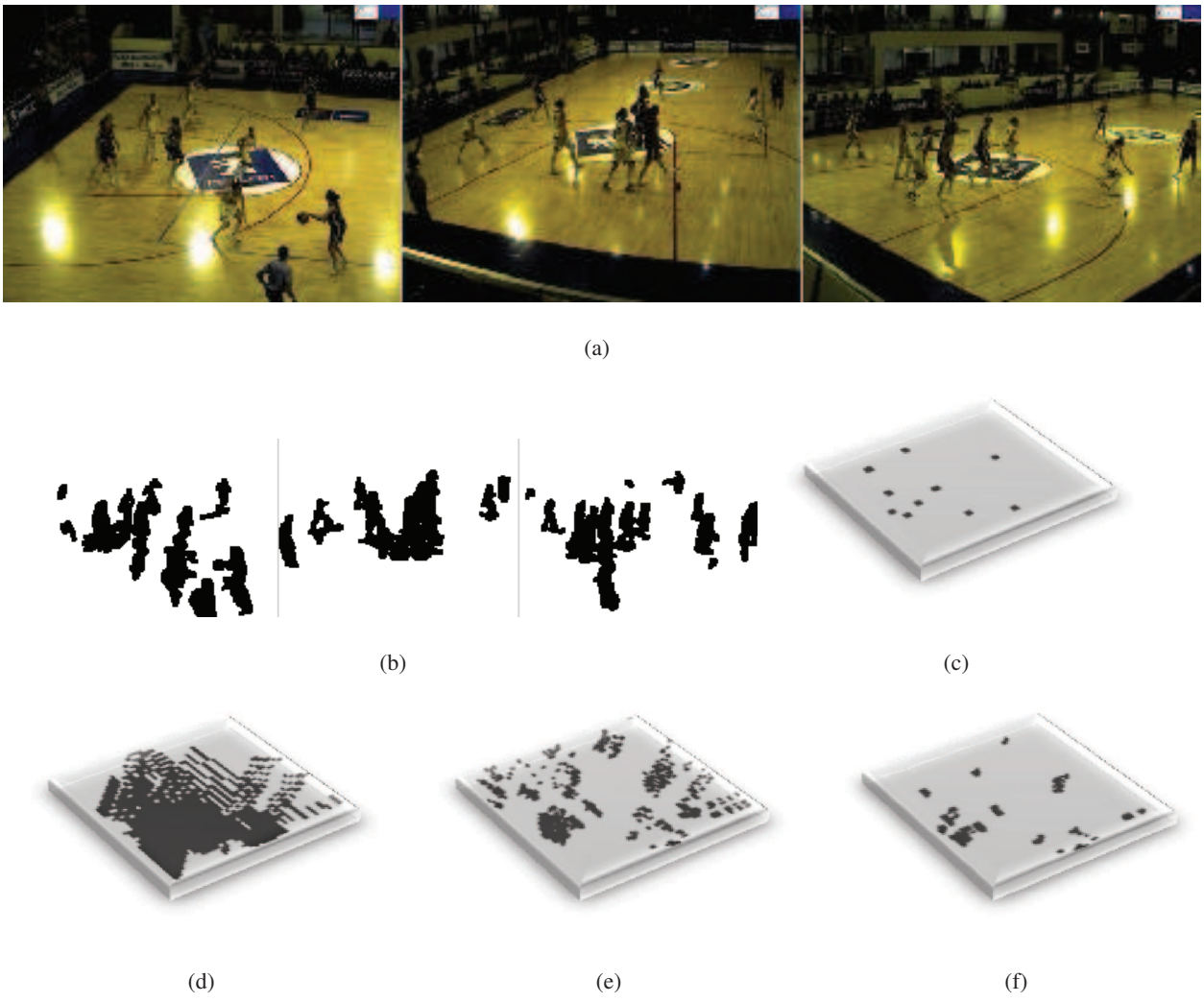


Fig. 7: Illustration of the sample points used given the three strategies. (a) Camera view examples. (b) Corresponding foreground silhouettes. (c) People exact locations (top view). Sampled points are given in (d) for "Foreground pixels only" assumption (top view), in (e) for "Intersecting foreground pixels" and in (f) for "Least significant silhouette".

Figure 7 presents the three strategies used to reduce the search space. As expected from their definition, we observe that these assumptions have different impact on the dimensionality reduction, i.e. we have approximately $\text{Assumption } 3 \subset \text{Assumption } 2 \subset \text{Assumption } 1$. When an assumption further reduces the search space, it may have the counter part of potentially removing correct locations. However, reducing the search space increases the likelihood to better

localize people. In the next section, we evaluate the influence of all 3 assumptions on the performance of the system.

VII. Tracking People

In this Section, we present a simple tracking algorithm that suits the temporal evolution of the occupancy vectors as computed above. We do not aim at presenting here the best tracking method. Our objective is simply to prove that the non-empty locations of the occupancy vectors detected at each time of a video, i.e. the positions of the spatio-temporal occupancy vector, can be tracked across time according to a simple spatio-temporal connectivity criterion. The output of this procedure is also a sorting of people trajectories by decreasing tracking-period.

A. Spatio-Temporal Graph

Our tracking method relies on the definition of a directed graph on the spatio-temporal occupancy vector $x(t) \in \mathbb{R}^N$, where t is taken in the discrete time interval $\{t_1, \cdots, t_{N_t}\}$ composed of N_t instants $t_j < t_{j+1}$.

The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, d_{\mathcal{G}})$ of interest corresponds to:

(i) a set of spatio-temporal vertices

$$\mathcal{V} = \big\{ \big(\mathbf{q},\, t_j\big) \in \mathbb{R}^3 : 1 \leq j \leq N_t,\, \mathbf{q} \in \mathbf{p}\big(x(t_j)\big) \big\},$$

with $\mathbf{p}(u) = \{\mathbf{p}(i) : 1 \leq i \leq N_t, u_i \neq 0\}$ and $\mathbf{p}(i) \in \mathbb{R}^2$ is as before the location of the i^{th} cell in the discrete ground plane,

(ii) an *edge* set $\mathcal{E} = \mathcal{V} \times \mathcal{V}$ defining the connectivity between vertices in $\mathcal{V}$,

(iii) and a *distance* $d_{\mathcal{G}} : \mathcal{E} \to \mathbb{R}_+$ weighting these edges.

In this graph $\mathcal{G}$, the *length* $|\mathcal{P}|$ of a given connected path $\mathcal{P} = \overline{\mathbf{v}_1 \mathbf{v}_2 \cdots \mathbf{v}_K}$ of $K-1$ "hops" between K distinct vertices $\mathbf{v}_j \in \mathcal{V}$ following $K-1$ edges $(\mathbf{v}_j, \mathbf{v}_{j+1}) \in \mathcal{E}$ is simply defined as the sum $|\mathcal{P}| = \sum_{j=1}^{K-1} d_{\mathcal{G}}(\mathbf{v}_j, \mathbf{v}_{j+1})$.

The *vertex* set $\mathcal{V}$ contains at most NN_t elements. However, in our case the graph is essentially empty compared to its potential dimensionality. Indeed, as a result of our methods, for each time t_j the occupancy vector $x(t_j)$ is spatially sparse inducing a small coordinate set $\mathbf{p}\big(x(t_j)\big)$.

Moreover, as described hereafter, the proposed connectivity follows a particular *causal* geometry that prevents too long or unrealistic connections. First, we consider a directed connectivity

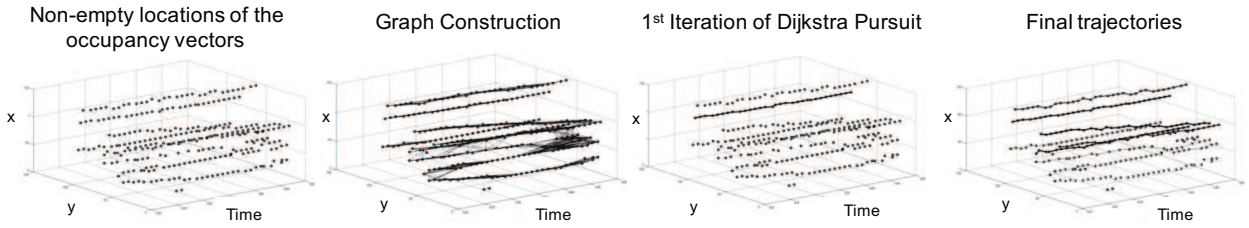


Fig. 8: Spatio-temporal graph-based tracking.

where the vertex $\mathbf{v} = (\mathbf{q}, t)$ is connected to $\mathbf{v}' = (\mathbf{q}', t')$ only if $t' > t$. Second, given a prior maximal speed $V_{\max} > 0$ for the people motion, two vertices $\mathbf{v}$ and $\mathbf{v}'$ in $\mathcal{V}$ cannot be connected if $\|\mathbf{q}-\mathbf{q}'\|_2 > V_{\max}\,(t'-t) > 0$. This induces a *causality* in the connectivity preventing connections between events that cannot result of a valid people motion. Third, for vertices respecting this *causality*, the corresponding edge $(\mathbf{v}, \mathbf{v}')$ is weighted by

$$d_{\mathcal{G}}(\mathbf{v}, \mathbf{v}') \;=\; \|\mathbf{q} - \mathbf{q}'\|_2 + \gamma\,\varphi(t' - t), \tag{8}$$

for a certain factor $\gamma > 0$ balancing the influence of the spatial and time domains, and given a particular increasing function φ.

The role of this function φ is to allow us a certain flexibility in the selection of paths of minimal length in $\mathcal{G}$, i.e. what will define our tracking procedure described in Section VII-B. For instance, for $\varphi(t) = t$, a direct path $\overline{\mathbf{v}_1\mathbf{v}_3}$ joining $\mathbf{v}_1 = (\mathbf{q}, t_j)$ to $\mathbf{v}_3 = (\mathbf{q}, t_{j+2})$, with both vertices sharing the same spatial coordinate $\mathbf{q}$, will always have a smaller length, i.e. $|\overline{\mathbf{v}_1\mathbf{v}_3}| = \gamma\,(t_{j+2} - t_j)$, than an indirect path $\overline{\mathbf{v}_1\mathbf{v}_2\mathbf{v}_3}$ with $\mathrm{v}_2 = (\mathbf{q}', t_{j+1})$ of length $|\overline{\mathbf{v}_1\mathrm{v}_2\mathbf{v}_3}| = \gamma\,(t_{j+2} - t_j) + 2\|\mathbf{q} - \mathbf{q}'\|_2$. We consider however that the indirect path is valid in our tracking if the *causality* between $\mathbf{v}_1$ and $\mathbf{v}_2$ is respected.

It is easy to prove that taking a φ that increases quicker than the linear function prevents such a case. We took $\varphi(t) = \exp t$, a choice that also penalizes temporally too long "1-hop" path. The factor γ is set in function of $V_{\max}$ so that the indirect path in the example above is selected against the direct one as soon as the causality between vertices is respected.

B. Dijkstra Pursuit

Given the directed graph defined above, our tracking method uses iteratively the well known fast Dijkstra algorithm computing the shortest paths in a graph, or *geodesics*, between one source

TABLE III: Dijkstra Pursuit

Input: A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, d_{\mathcal{G}})$.

Output: A set of tracks $\mathcal{T}$ of decreasing length.

Program:

1) Initialize: $\mathcal{R} \leftarrow \mathcal{V}$, $\mathcal{T} \leftarrow \emptyset$.

2) Pick $\mathbf{v}^* = \arg\max_{\mathbf{v} \in \mathcal{R}} |\mathcal{P}(\mathbf{v})|$.

3) Store: $\mathcal{T} \leftarrow \mathcal{T} \cup \mathcal{P}(\mathbf{v}^*)$.

4) Update: $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{P}(\mathbf{v}^*)$, and recompute connectivity.

5) If $\#\mathcal{R} = 0$, return $\mathcal{T}$ and stop; else, return to Step 2.

$\mathbf{v} \in \mathcal{V}$ and all the other vertices of $\mathcal{V}$ [37].

More precisely, let us first define $\mathcal{P}(\mathbf{v}) \subset \mathcal{V}$ as the longest geodesic initiated from $\mathbf{v} \in \mathcal{V}$ in $\mathcal{G}$, i.e. the longest of all the shortest paths between $\mathbf{v}$ and any other point $\mathbf{v}' \in \mathcal{V}$.

At the first iteration, our method removes from the initial graph $\mathcal{G}$ the vertices of the path $\mathcal{P}(\mathbf{v}^*)$, with $\mathbf{v}^*$ the vertex providing the longest $\mathcal{P}$, i.e. $\mathbf{v}^* = \arg\max_{\mathbf{v}} |\mathcal{P}(\mathbf{v})|$. The next iterations are then defined by applying the same process on the residual graphs until reaching an empty one. This iterative procedure, coined *Dijkstra Pursuit*, is summarized in Table III and Figure 8.

VIII. Performance Evaluation

A. Experiments

Synthetic and real challenging data are used to evaluate the proposed framework.

Real data have been obtained from the APIDIS dataset[2] and from the PETS 2009 Benchmark dataset[3].

The APIDIS dataset consists in seven cameras monitoring a basketball game, including one omnidirectional camera. The dataset has the following challenges:

[2]The dataset is publicly available at http://www.apidis.org/Dataset/

[3]http://winterpets09.net/

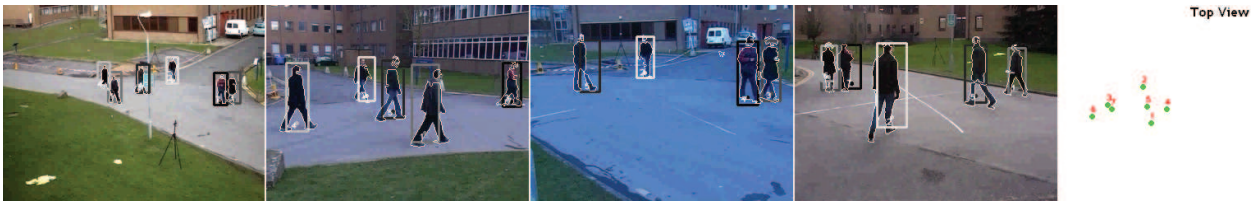
- Basketball players have abrupt changes of behavior, e.g. they run, jump, crouch, change suddenly their motion path, etc.
- Players on the same team have the same appearance.
- In some camera views, players greatly occlude each other.
- Some cameras have very similar viewpoints, affecting the resolution of the ambiguities arising with the occlusion problem.
- The reflection of the players on the floor and their strong shadows lead to severely degraded foreground silhouettes. Many false positives silhouettes are extracted with a standard background subtraction algorithm (e.g. the work of Stauffer and Grimson [9]).
- Players interact strongly with each other and their spatial distribution on the ground can be very dense and compact or spatially scattered.

All videos are scaled to a QVGA resolution with approximately 25 fps. Performance over the left-half of the basketball court is measured since it is the side where the most number of cameras are monitoring the game, i.e. camera's id 1, 2, 4, 5, and 7.

The PETS 2009 Benchmark datasets are multisensor sequences containing different crowd activities filmed from multiple cameras and involve up to approximately forty actors. We evaluate our algorithm on sparse crowd, as well as medium and high density crowd (Figure 9). Our detection scheme is able to count people in high density crowds given multiple or even single camera (Figure 10). Videos are available at the following website: http://lts2www.epfl.ch/~alahi/pets.htm.

Synthetic data are constructed given the same scene geometry as the APIDIS dataset. Random vectors x are created given a spatial sparsity constraint, e.g. location points have a minimum spatial distance with respect to each other (> 70 cm). Five to fifteen people are randomly triggered for each frame (a few hundred frames are generated). The synthetic data will allow us to evaluate the performance of our algorithms with controlled foreground silhouettes, hence measuring how well the noise or occlusion problem is solved.

The performance of the detection process is quantitatively evaluated by computing the *precision* and the *recall* measures given by the ratios $TP/(TP + FP)$ and $TP/(TP + FN)$ respectively, where TP, FP and and FN are the number of True Positive, False Positive and False Negative. A true A true positive is when a person is correctly located on the ground plane.



(a) Sparse crowd

(b) Medium crowd

(c) High density crowd

Fig. 9: Detecting and tracking people given the PETS dataset. White contours represents the degraded foreground silhouettes used.

The foreground silhouettes are extracted using the work of Stauffer and Grimson [9]. The outcome of the background subtraction algorithm is noisy. The silhouettes are severely degraded. Only part of the people are extracted, their shadow and reflections are considered, and random false positives are generated due to lighting conditions, camera noise.

B. Results

All three reconstruction methods, i.e. RW-BPDN, RW-Lasso, and O-Lasso are compared with the state-of-the-art approach proposed by Fleuret *et al.* in [21]. They propose a detection stage referred to as the Probability Occupancy Map (POM) to locate people on the ground given the degraded foreground silhouettes. Their algorithm depends on two parameters: the maximum number of iterations and a constant σ accounting for the quality of the background subtraction. We set the maximum number of iterations to 1500, and measure the performance of their

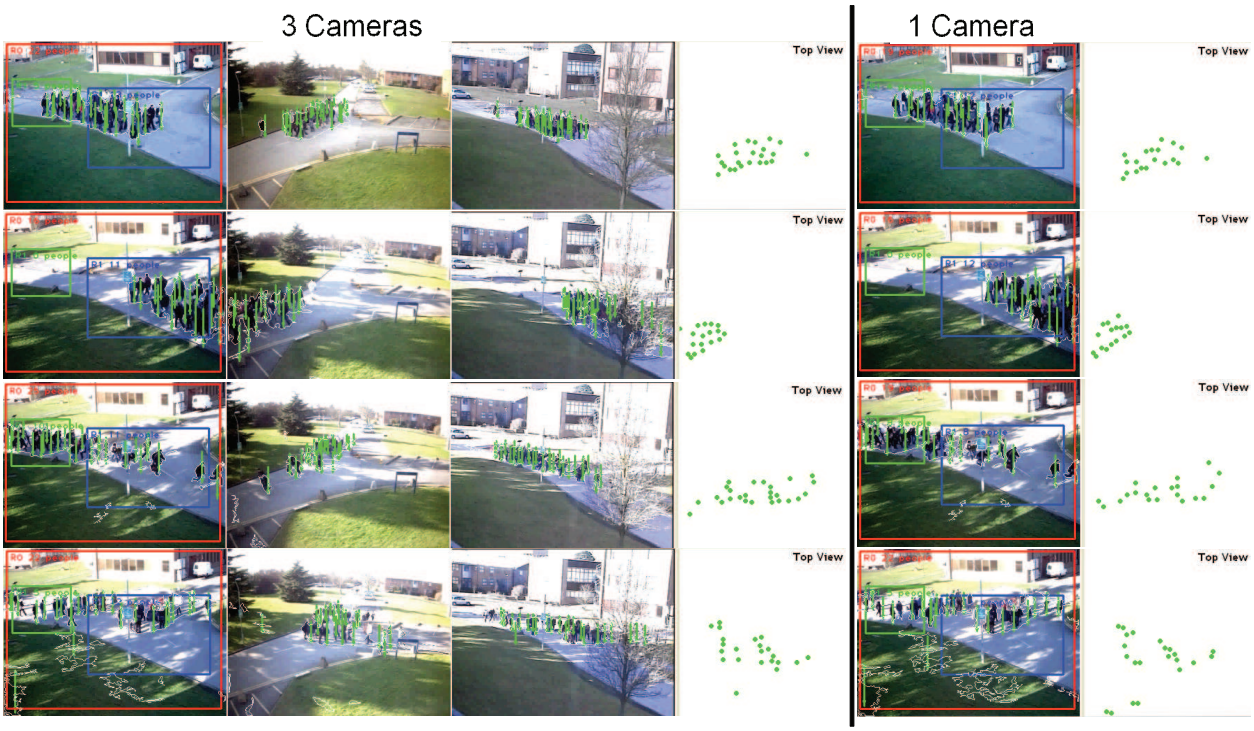


Fig. 10: Locating people with either 3 cameras (left hand-side) or a single camera (right hand-side) given the PETS dataset. White contours represents the degraded foreground silhouettes used.

algorithm for various σ.

Figure 11 illustrates all performances given four cameras monitoring the APIDIS dataset (camera's id 2, 4, 5, and 7). The proposed Occupancy Lasso (O-Lasso) clearly outperforms other approaches in term of both recall and precision rate. The Re-Weighted Lasso (RW-Lasso) with a fixed sparsity bound outperforms the RW-BPDN with various residual error ε. The performance of the BPDN formulation is affected by the difficulty to estimate the residual error, i.e. the degradation occurring in the foreground silhouettes. Finally, the state-of-the-art approach (POM) has a much lower precision rate for a given recall rate than the sparsity driven methods. Considering the sparsity of the desired solution allow us to reduce the false positive rate consequently.

Figure 12 presents all performances given four planar cameras on the synthetic data. It is interesting to notice that both formulation BPDN with $\varepsilon = 0$ and Lasso with $\varepsilon_p = \texttt{ppl}$, i.e. the number of people present in the scene, leads to the same performance. Since they are equivalent

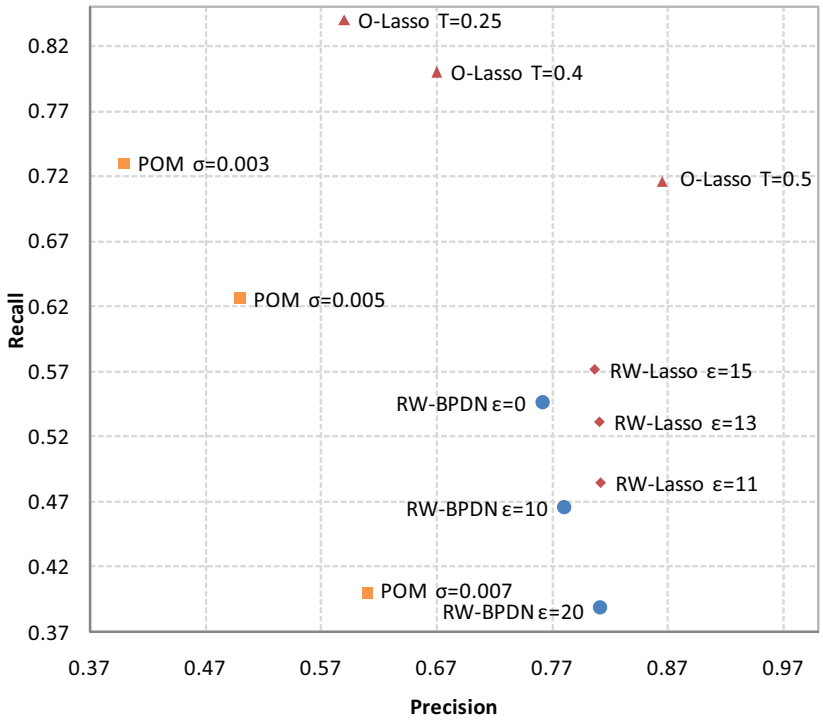


Fig. 11: Precision and recall rate on the APIDIS dataset given four cameras monitoring the scene (camera's id 2, 4, 7, and 1). Our proposed approaches (RW-BPDN, RW-Lasso, and O-Lasso) are compared with the state-of-the-art probability of occupancy (POM) by Fleuret *et al.* in [21].

problem and the foreground silhouettes are not noisy, it coincides with our expectations to obtain the same performance. However, relaxing the formulations influences the performances accordingly. Relaxing the fidelity term with the BPDN increases the precision and reduce its recall rate. High fidelity constraint allows to miss some people hence reduces the recall rate. With the Lasso formulation, increasing ε_p increases the number of false positives hence reduces the precision rate. Interestingly, the state-of-the-art outperforms the RW-BPDN and RW-Lasso given the synthetic data. Since noise is not present in the data, better performances are achieved with POM. However, using our proposed O-Lasso outperforms again other methods. The recall rate is above 90% with roughly perfect precision rate ($> 98\%$).

The strength of our proposed formulation is emphasized with real data, i.e. when noise is present on the data. BPDN is useful when we can bound the noise. However, in our application, the noise can severely degrade the observations. Hence, the Lasso formulation suits best our

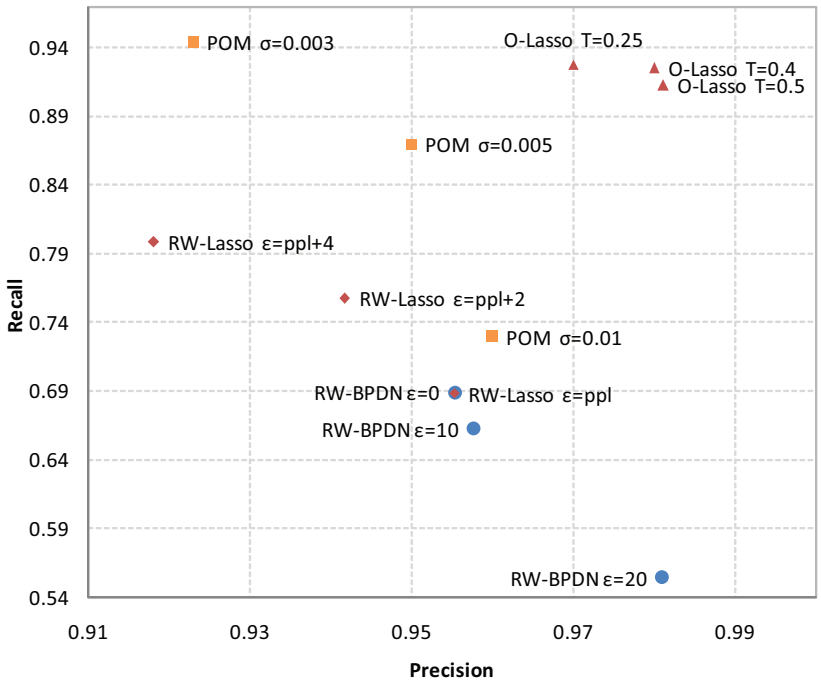


Fig. 12: Precision and recall rate with the synthetic data given four cameras. Our proposed approaches (RW-BPDN, RW-Lasso, and O-Lasso) are compared with the state-of-the-art probability of occupancy (POM) by Fleuret *et al.* in [21].

problem. Note that the adaptive formulation (O-Lasso) does not need any prior on the number of people present in the scene. It correctly updates the constraint to reach the right number of people.

Interestingly, reducing the search space according to the three assumptions of Section VI-B not only increases the processing speed since fewer number of points are evaluated, but it also increases the performance of the detection. Figure 13 shows that for each proposed reduction step, the recall and precision rate increases with O-Lasso (given camera's id 2,5, 7 in the APIDIS dataset).

Finally, to reach a full detection picture, the proposed graph-driven tracking increases the performance of the system. Table IV illustrates its impact. First, the located points obtained with the severely degraded foreground silhouettes are used. Then, the located points obtained with the synthetic foreground silhouettes (i.e. perfect silhouettes) are tested. With both scenarios, the recall rate is increased without strongly degrading the precision rate.

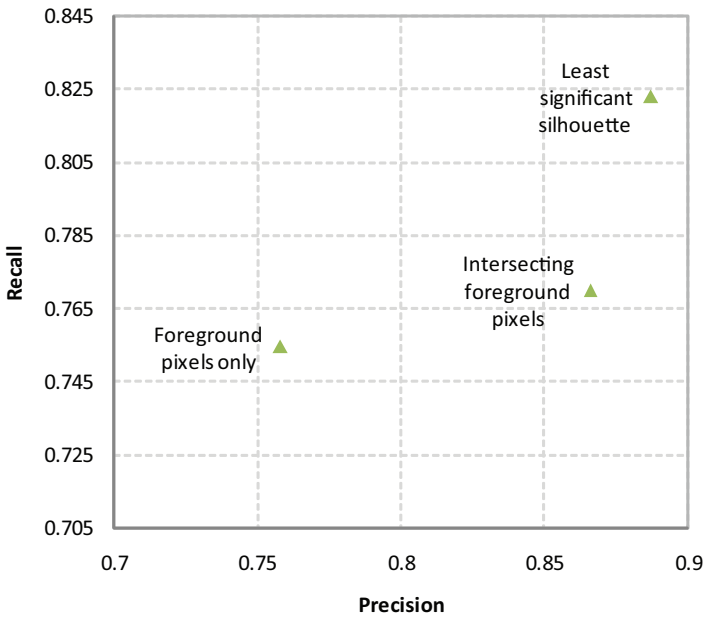


Fig. 13: Precision and recall rate of our proposed algorithm (O-Lasso) given four cameras monitoring the scene (camera's id 2, 4, 5, and 7 in the APIDIS dataset) and various search space reduction assumptions

Located points given	No tracking		Tracking	
	Recall	Precision	Recall	Precision
Severely degraded FS	0.82	0.92	0.88	0.91
Synthetic FS	0.93	0.965	0.96	0.967

TABLE IV: Impact of the proposed graph-based tracking

C. Validation

Given the proposed approach based on the O-Lasso, we analyze its performance when *(i)* the number of cameras is increased, *(ii)* when people are occluding each other, and *(iii)* when the silhouettes extracted are degraded.

One of the advantages of our framework is that it scales to any number of cameras. Therefore, the performance of the system with various number of cameras is compared in Figure 14. Note that a precise temporal window is selected having all players in the field of view of all cameras. It is interesting to see that even when a single camera is used, we can locate as many people as using multiple cameras. Nevertheless, adding cameras reduces the number of false positives due

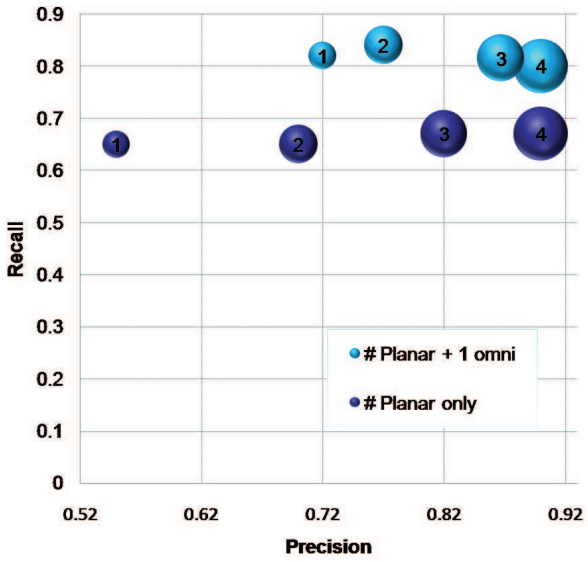


Fig. 14: Precision and recall rate of camera id = 2 in the APIDIS dataset when a set cameras are monitoring the scene. The number in each bubble represents the number of cameras used. (The sequence of cameras id 7,2,4,1 is used for planar cameras, and camera id 5 for the omnidirectional one)

to a degraded foreground silhouettes. Shadows and reflected players on the ground have a strong impact on the precision rate. In addition, merging an omnidirectional camera with other planar cameras have the best performance. Surprisingly, if the omnidirectional camera is monitoring the scene alone, a poor detection is achieved due to the severely degraded foreground silhouette: $R = 0.47$ and $P = 0.55$ (not shown in Figure 14). Most of the time, the people's shadow is much bigger than its silhouette affecting considerably the performance. In addition, in some areas, people are almost missed by the background subtraction algorithm since they occupy only few pixels (small surface). Finally, due to the small bounding box of the people, a small offset in the detection considerably affects the performance and leads to a missed person. Figure 17 presents the foreground silhouettes extracted and the detected people given various number of cameras.

One of the main challenges we want to handle is the mutual occlusions generated by the people. The proposed relaxation step (Section V-B) wrongly considers the occlusion problem as a linear operator. Indeed, according to our model, overlapping silhouette are not quantized but summed (Dx). Hence, whenever people are occluding each other, their silhouettes are overlapped

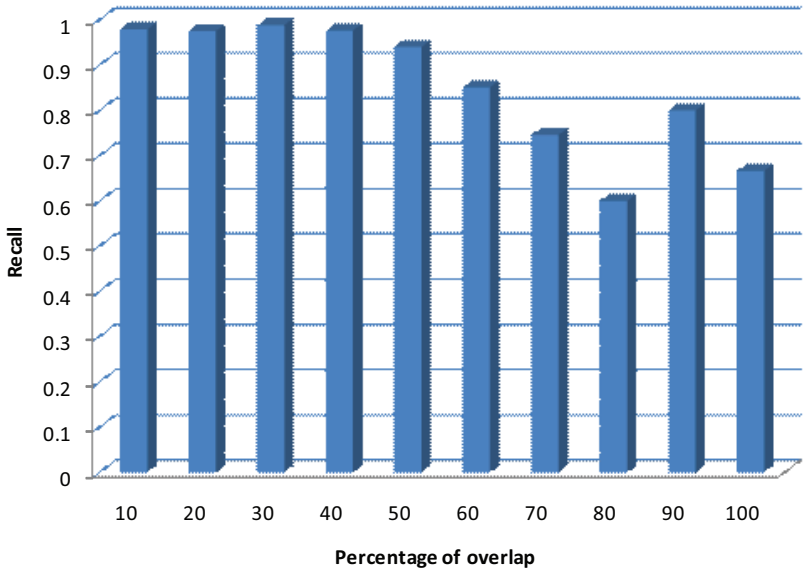


Fig. 15: Recall rate with respect to the PO present for the people given the O-Lasso approach

and generate noise in our fidelity term. We evaluate the performance of our algorithm with respect to the Percentage of Overlap (or PO) present in each MSV:

$$\mathrm{PO} = \frac{\text{Number Of Pixels Overlapping}}{\text{Number Of Pixels Visible}} \tag{9}$$

We cluster each person present in our synthetic dataset given their PO with other people and measure how well we have detected them. Figure 15 illustrates the recall rate with respect to the PO present in the people[4]. When half of of the MSV is overlapping with other MSVs, the recall rate is still higher than 90%. Then, it decreases to reach in the worst case (when overlap is more than roughly 90%) a recall of 65% which is very satisfying. Therefore, we have a generic measurement about the performance of the algorithm to detect occluded people regardless of the scene geometry and people's density.

Finally, we measure the performance of our approach with respect to the percentage of degradation present in the foreground silhouettes. The degraded foreground silhouette is made of false positive pixels or false negative ones. Setting to zero a percentage of the MSV is equivalent to degrading the foreground silhouettes with false negative pixels. Figure 16 illustrates the recall

[4]Remark that the PO cannot be defined for false-positives. This prevents a plot of the precision in function of the PO.

Shape	Recall	Precision
Rectangular	0.64	0.87
Ellipsoid	0.7156	0.85
Half rectangular and half ellipsoid	0.6938	0.9001

TABLE V: Precision and Recall rate using various shape in the *Forward Model*given the O-Lasso method and 4 planar cameras

rate when we degrade the silhouettes accordingly. Note that it also informs the percentage of foreground silhouettes needed to trigger a positive detection although the silhouettes are only made of false positive pixels. The recall rate remains higher than 90% although 30% of the silhouettes are removed. Then, the performance decreases considerably with respect of the degradation applying to the silhouettes. Such information informs us about the sensitivity of our *Forward Model* generating silhouettes: 30% of the generated silhouette can be discarded. In other words, if our silhouettes are fitting a person with a height of 1m70cm, we can still detect people in the range of 1m20 till 2m20 with the same performance. Moreover, we can also say that only 70% of the silhouette model is relevant. Hence, the proposed approximated shape, i.e. half rectangular and half ellipsoid can handle 30% of a 'shape' approximation error. Note that other shapes can be used. Table V illustrates the performance of using other shape. The rectangular shape is used by Fleuret *et al.* in [21] to efficiently compare their generative model with the foreground silhouettes based on integral images. However, in our framework, the *Forward Model* is used to create the atoms of the dictionary. Such atoms are computed off-line hence allow the use of any shape. Therefore, using an ellipsoid shape for instance has the same computational cost as using a rectangular one.

IX. Conclusions

We propose a framework to efficiently deal with simple and very noisy features to locate people in a well defined mathematical formulation. The strength of our approach is quantitatively and qualitatively illustrated on challenging real world scenarios as well as on synthetic data outperforming the state-of-the-art. We show the advantage afforded by the sparsity driven framework. The approach is generic enough to be used with any calibrated camera. Planar and omnidirectional cameras are naturally merged. Any number of cameras can be used. The multi-

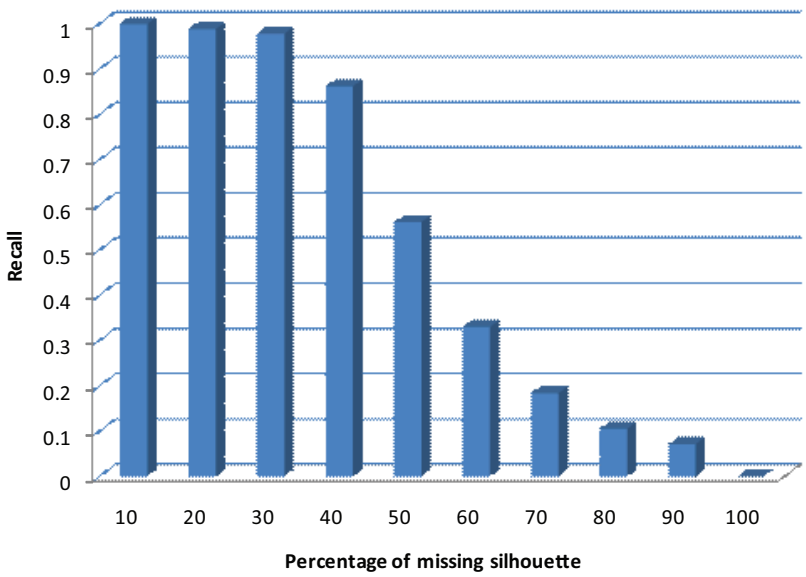


Fig. 16: Recall rate with respect to percentage of missing silhouette region extracted for each people given the adaptive O-Lasso approach

view infrastructure is fully taken into consideration during the detection process and does not impose any constraint on the scene surface to be monitored. Furthermore, detected people are perfectly matched across cameras so that their reconstruction from all the views can be performed. Since the coordinates of the people are computed in the ground floor, each person can have a flag informing if a clear visualization is available in a view, i.e. other people are not occluding. Therefore, further processing such as identification can be performed. In that perspective, a simple tracking module to match people across frames is explained in Section VII. It detects people tracks by finding the longest geodesics in the graph connecting non-zero occupancy location across time. In future work, we plan to improve this graph-driven tracking by attaching more information (e.g. silhouette intensity histograms) on the nodes of the corresponding graph and by redefining its connectivity in function of these features.

On the theoretical side, the proposed Repulsive Spatial Sparsity can be compared to a recent trend in the field of sparse representation of signals, and in particular in these new developments surrounding the recovery of structurally-sparse signals in linear inverse problems. In [38], the concept of *block*-sparsity is for instance successfully introduced to improve the recovery of sparse

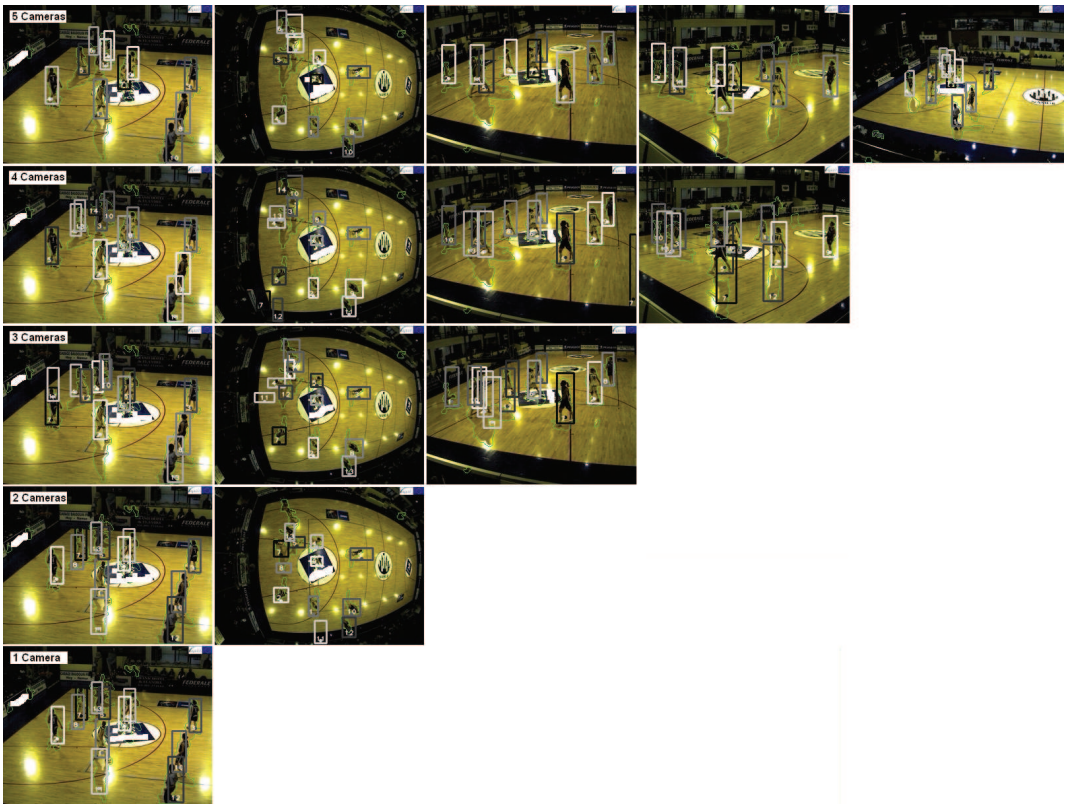


Fig. 17: Illustration of the detected people with various number of cameras given the APIDIS dataset. The green contours represents the degraded foreground silhouettes used

signal from random measurements, i.e. in a Compressed Sensing scenario. Another approach followed in [39] plays on the sparsity measure replacing the common ℓ_1-norm of Lasso or BPDN programs by other "mixed-norms" on the vector components organized in group and elements. In the future, we plan to explore the connections between the RSS and these alternative sparsity variations.

APPENDIX A
PROXIMAL METHODS

In order to solve our different minimization problems, we use a powerful proximal operator-based iterations and monotone operator splitting theory introduced by Moreau in 1962 [40] and brought to light in the signal processing community by Combettes [35], [41].

Thanks to this theory, very efficient methods can be designed to solve general convex opti-

mization problems of the form

$$\arg\min_{\mathbf{x}} \; f(\mathbf{x}) \; + \; g(\mathbf{x}).$$

In the case of the Re-Weighted BPDN (4), $f = \|W \cdot\|_1$ is convex, and $g = \iota_C = \iota_{B^2_\varepsilon} \circ A(\cdot)$ where ι is the indicator function and A is the affine function such that $A = D \cdot -y$. The convex sets B^2_ε and C are such that $B^2_\varepsilon = \{x \in \mathbb{R}^N \; : \; \|x\|_2 \leq \varepsilon\}$ and $C = \{x \in \mathbb{R}^N \; : \; \|Ax\|_2 = \|y - Dx\|_2 \leq \varepsilon\}$. As f and g are both non-differentiable, the Douglas-Rachford splitting method [41], [42] is used. The Douglas-Rachford recursion to solve the reweighted ℓ_1-BPDN can be written in the compact form

$$x^{(t+1)} = x^{(t)} + \mu_t[S_W \circ (2P_C - \mathrm{Id}) - P_C](x^{(t)}), \tag{10}$$

where S_W is the component-wise soft-thresholding operator with threshold vector W, and P_C is the orthogonal projection onto the closed convex set C. When D is a bounded linear operator with a bound $0 \leq c < \infty$ such that $0 \leq DD^* \leq c\,\mathrm{Id}$, the numerical implementation of this projection is defined as described in [42]. Let $\{\beta_t\}_{t\in\mathbb{N}}$ be a sequence with $0 < \inf_t \beta_t \leq \sup_t \beta_t < 2/c$, and define the two sequences $\{u_t\}_{t\in\mathbb{N}}$ and $\{p_t\}_{t\in\mathbb{N}}$ by

$$\begin{aligned} u^{(t+1)} &= \beta_t(\mathrm{Id} - P_{B^2_\varepsilon})(\beta_t^{-1}u^{(t)} + D(x - D^*u^{(t)}) - y) \\ p^{(t+1)} &= x - D^*u^{(t+1)} \end{aligned} \tag{11}$$

Then from [42] we get that $u^{(t)} \to \overline{u}$ and $p^{(t)} \to P_C(x)$

In the case of the Re-Weighted Lasso problem (5), $f = \|y - D\cdot\|_2^2$ is convex and differentiable with a β-Lipschitz gradient, and $g = \iota_{B^1_{W,\varepsilon_p}}$ with $B^1_{W,\varepsilon_p} = \{x \in \mathbb{R}^N \; : \; \|Wx\|_1 \leq \varepsilon_p\}$. More precisely, as the ℓ_1-norm is non-differentiable, the Forward-Backward splitting is used [41], [36]. Forward-backward (FB) splitting is essentially a generalization of the classical gradient projection method for constrained convex optimization. It can be written in the compact form

$$x^{(t+1)} = P_{B^1_{W,\varepsilon_p}} \circ (\mathrm{Id} - \mu_t \nabla f)\,(x^{(t)}), \tag{12}$$

where $0 < \inf_t \mu_t \leq \sup_t \mu_t < 2/\beta$ for the iteration to converge (weakly in general), ∇ is the gradient operator, and $P_{B^1_{W,\varepsilon_p}}$ is the orthogonal projection onto the convex set B^1_{W,ε_p}. This projection can be efficiently computed thanks to the developments of Appendix B. From [35], one can show that in the both cases presented above, the sequence $(x^{(t)})_{t\in N}$ converges to some point x^*, which is the solution of the problem.

APPENDIX B
PROJECTION ONTO A ℓ_1 WEIGHTED BALL

We present in this section an algorithm and its numerical implementation that solves the problem

$$y \;=\; \underset{u\in\mathbb{R}^n}{\arg\min}\; \|u-x\|_2^2 \;\text{s.t.}\; \|Wu\|_1 \leq \varepsilon, \tag{13}$$

for a non-negative diagonal matrix $W \in \mathbb{R}^{n\times n}$. This problem can be seen as the projection of the point $x \in \mathbb{R}^n$ on the weighted ℓ_1 ball $B^1_{W,\varepsilon} = \{u : \|Wu\|_1 \leq \varepsilon\}$.

If $\|Wx\|_1 \leq \varepsilon$, there is nothing to do and $y = x$. In the other case, the solution is clearly on the surface of $B^1_{W,\varepsilon}$, so that we must solve

$$y \;=\; \underset{u\in\mathbb{R}^n}{\arg\min}\; \|u-x\|_2^2 \;\text{s.t.}\; \|Wu\|_1 = \varepsilon. \tag{14}$$

In addition, since this ball is convex and centered on the origin, it is clear that $\operatorname{sign} x_i = \operatorname{sign} y_i$, therefore, up to the appropriate flipping of some coordinate axis in $\mathbb{R}^n$, we can assume $x_i, y_i \geq 0$.

The Lagrangian form of problem (14) is

$$\mathcal{L}(u,\theta) = \frac{1}{2}\|x-u\|_2^2 + \theta(\sum_{i=1}^n w_i u_i - \varepsilon), \tag{15}$$

where $\theta \in \mathbb{R}^n$ is a Lagrange multiplier. For a given θ, the minimum of $\mathcal{L}$ is reached when

$$u_i = x_i - w_i\theta + \zeta_i = x_i - w_i\theta = w_i\left(\tfrac{x_i}{w_i} - \theta\right), \tag{16}$$

if $\frac{x_i}{w_i} > \theta$, and $u_i = 0$ otherwise. In other words,

$$u_i \;=\; S_{w_i\theta}(x_i),$$

where $S_\lambda(v) \;= \operatorname{sign} v\;(|v|-\lambda)_+$, is the soft-thresholding of $v \in \mathbb{R}$ by the threshold $\lambda > 0$, with $(v)_+ = v$ if $v \geq 0$ and 0 else. The minimum of $\mathcal{L}$ with respect to θ is thus reached when $\|WS_{w_i\theta}(x_i)\|_1 = \varepsilon$, i.e. when

$$\sum_i w_i^2(|\tfrac{x_i}{w_i}| - \theta)_+ = \varepsilon. \tag{17}$$

This can be computed very efficiently by the following simple method.

Let the vector $z \in \mathbb{R}^n$ be the vector obtained by sorting the values $|\frac{x_i}{w_i}|$ by decreasing order, a process that can be realized in $O(n\log n)$ operations.

If we set $\theta = z_{n-k+1}$ for some index k, we get $\sum_{i=1}^{n} w_i^2 (z_i - \theta)_+ = \sum_{i=1}^{n-k} w_i^2 (z_i - z_{n-k+1})$. Let the index i^* be such that

$$i^* = \max\{1 \leq k \leq n : \sum_{i=1}^{n-k} w_i^2 (z_i - z_{n-k+1}) \geq \varepsilon\}.$$

Therefore, by construction the θ satisfying (17) belongs to the interval $I = [z_{n-i^*+1}, z_{n-i^*}] \subset \mathbb{R}$. In that range, (17) becomes

$$\sum_{i=1}^{n-i^*} w_i^2 (z_i - \theta) = \varepsilon,$$

so that finally,

$$\theta \;=\; \theta(i^*) \;=\; \frac{(\sum_{i=1}^{n-i^*} w_i^2 z_i) - \varepsilon}{\sum_{i=1}^{n-i^*} w_i^2}.$$

ACKNOWLEDGMENT

The authors would like to thank the CVLAB [21] and M.J. Fadili [36] for their source codes.

REFERENCES

[1] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," *Proc. IEEE Int'l Conference on Computer Vision and Pattern Recognition*, vol. 97, pp. 193–199, 1997.

[2] C. Papageorgiou and T. Poggio, "Trainable pedestrian detection," in *Proc. IEEE Int'l Conference on Image Processing*, vol. 4, 1999.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int'l Conference on Computer Vision and Pattern Recognition*, 2005, pp. I: 886–893.

[4] F. Suard, A. Rakotomamonjy, A. Bensrhair, and A. Broggi, "Pedestrian detection using infrared images and histograms of oriented gradients," in *Proc. IEEE Symposium on Intelligent Vehicles*, Tokyo, Japan, Jun. 2006, pp. 206–212.

[5] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1713–1727, 2008.

[6] A. Alahi, M. Bierlaire, and M. Kunt, "Cascade of descriptors to detect and track objects across any network of cameras," *Submitted to Computer Vision and Image Understanding*, 2009.

[7] H. Cheng, N. Zheng, and J. Qin, "Pedestrian detection using sparse gabor filter and support vector machine," *Proc. IEEE Symposium on Intelligent Vehicles*, pp. 583– 587, june 2005.

[8] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[9] C. Stauffer and W. Grimson, "Adaptive background mixture models for real-time tracking," *Proc. IEEE Int'l Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, 1999.

[10] F. Porikli, "Achieving real-time object detection and tracking under extreme conditions," *Journal of Real-Time Image Processing*, vol. 1, no. 1, pp. 33–40, 2006.

[11] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, 2006.

[12] S. S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[13] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[14] K. Mueller, A. Smolic, M. Droese, P. Voigt, and T. Wienand, "Multi-texture modeling of 3d traffic scenes," *Proc. of the 2003 Int'l Conference on Multimedia*, vol. 2, pp. 657–660, 2003.

[15] C. Stauffer and K. Tieu, "Automated multi-camera planar tracking correspondence modeling," in *Proc. IEEE Int'l Conference on Computer Vision and Pattern Recognition*, 2003, pp. I: 259–266.

[16] J. Black, T. Ellis, and P. Rosin, "Multi view image surveillance and tracking," *Proc. IEEE Workshop on Motion and Video Computing*, vol. 00, p. 169, 2002.

[17] J. Orwell, S. Massey, P. Remagnino, D. Greenhill, and G. A. Jones, "A multi-agent framework for visual surveillance," in *Proc. IEEE Int'l Conference on Image Analysis and Processing*. Washington, DC, USA: IEEE Computer Society, 1999, p. 1104.

[18] Y. Caspi, D. Simakov, and M. Irani, "Feature-based sequence-to-sequence matching," *International Journal of Computer Vision*, vol. 68, no. 1, pp. 53–64, June 2006.

[19] K. Kim and L. Davis, "Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering," in *Proc. European Conference on Computer Vision*, 2006, pp. III: 98–109.

[20] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[21] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua, "Multicamera people tracking with a probabilistic occupancy map," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 267–282, 2008.

[22] J. Franco and E. Boyer, "Fusion of multiview silhouette cues using a space occupancy grid," in *Tenth Proc. IEEE Int'l Conference on Computer Vision, 2005. ICCV 2005*, vol. 2, 2005.

[23] D. Yang, H. Gonzalez-Banos, and L. Guibas, "Counting people in crowds with a real-time network of simple image sensors," in *Proc. IEEE Int'l Conference on Computer Vision*, 2003, p. 122.

[24] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," in *2003 Proc. IEEE Int'l Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, vol. 2, 2003.

[25] R. Eshel and Y. Moses, "Homography based multiple camera detection and tracking of people in a dense crowd," in *Proc. IEEE Int'l Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[26] S. Khan and M. Shah, "A multiview approach to tracking people in crowded scenes using a planar homography constraint," in *Proc. European Conference on Computer Vision*, 2006, pp. IV: 133–146.

[27] S. M. Khan and M. Shah, "Tracking multiple occluding people by localizing on multiple scene planes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 3, pp. 505–519, 2009.

[28] D. Delannay, N. Danhier, and C. D. Vleeschouwer, "Detection and recognition of sports(wo)man from multiple views," in *Proc. ACM/IEEE Int'l Conference on Distributed Smart Cameras*, Como, Italy, 30 Aug. - 2 Sep. 2009, accepted.

[29] D. Reddy, A. Sankaranarayanan, V. Cevher, and R. Chellappa, "Compressed sensing for multi-view tracking and 3-D voxel reconstruction," in *Proc. IEEE Int'l Conference on Image Processing*, 2008, pp. 221–224.

[30] I. Gorodnitsky and B. Rao, "Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm," *IEEE Trans. on Signal Processing*, vol. 45, no. 3, pp. 600–616, 1997.

[31] D. Malioutov, M. Cetin, and A. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Trans. on signal processing*, vol. 53, no. 8 Part 2, pp. 3010–3022, 2005.

[32] V. Cevher, M. Duarte, and R. Baraniuk, "Distributed Target Localization via Spatial Sparsity," in *Proc. European Signal Processing Conference*, 2008.

[33] B. Natarajan, "Sparse Approximate Solutions to Linear Systems," *SIAM Journal on Computing*, vol. 24, p. 227, 1995.

[34] E. J. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, 2007, (to appear).

[35] P. Combettes, "Solving monotone inclusions via compositions of nonexpansive averaged operators," *Optimization*, vol. 53, no. 5, pp. 475–504, 2004.

[36] M. Fadili and J.-L. Starck, "Monotone operator splitting for fast sparse solutions of inverse problems," *SIAM Journal on Imaging Sciences*, 2009, submitted.

[37] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[38] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Trans. on Information Theory*, 2008, submitted.

[39] M. Kowalski and B. Torresani, "Sparsity and persistence: mixed norms provide simple signals models with dependent coefficients," *Signal, Image and Video Processing*, 2008, accepted, doi:10.1007/s11760-008-0076-1.

[40] J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *CR Acad. Sci. Paris Ser. A Math*, vol. 255, pp. 2897–2899, 1962.

[41] P. Combettes and V. Wajs, "Signal Recovery by Proximal Forward-Backward Splitting," *Multiscale Modeling and Simulation*, vol. 4, no. 4, p. 1168, 2006.

[42] M. Fadili and J.-L. Starck, "Monotone operator splitting for fast sparse solutions of inverse problems," *SIAM Journal on Imaging Sciences*, 2009, submitted.

Alexandre Alahi received the M.S. degree in Communication Systems from the Ecole Polytechnique Fédérale de Lausanne (Swiss Federal Institute of Technology), in 2006. During his studies, he earned a one-year exchange fellowship to study at Carnegie Mellon University, Pittsburgh , PA , USA. In the past few years, he has worked with various companies in the field of image processing and computer vision. Among them are Logitech (Silicon Valley-CA, USA) and Mitsubishi Electric Research Laboratories (Cambridge - MA, USA). He is now working towards his Ph.D. degree in the Signal Processing Institutes (LTS2) of EPFL, under the supervisions of Prof. M. Kunt, Prof. M. Bierlaire, and Prof. P. Vandergheynst.

Laurent Jacques received the B.Sc. in Physics, the M.Sc. in Mathematical Physics and the PhD in Mathematical Physics from the Université catholique de Louvain (UCL), Belgium. He was a Postdoctoral Researcher with the Communications and Remote Sensing Laboratory of UCL in 2005-2006. He obtained in Oct. 2006 a four-year (3+1) Postdoctoral funding from the Belgian FRS-FNRS in the same lab. He was a visiting Postdoctoral Researcher, in spring 2007, at Rice University (DSP/ECE, Houston, TX, USA), and from Sep. 2007 to Jul. 2009, at the Swiss Federal Institute of Technology (LTS2/EPFL, Switzerland). His research focuses on Sparse Representations of signals (1-D, 2-D, sphere), Compressed Sensing, Inverse Problems, and Computer Vision.

Yannick Boursier Yannick Boursier graduated from the Institut Supérieur de l'Aéronautique et de l'Espace (ISAE) and received the M.S degree of Applied Mathematics from the Université Paul Sabatier and the ISAE, Toulouse, France, in 2004. In 2007, he received the PhD degree in Optics and Image processing from the Université Aix-Marseille 3, France. He was a postdoctoral researcher with the Laboratoire d'Astrophysique de Marseille in 2007-2008, and joined the Signal Processing Laboratories at the Swiss Federal Institute of Technology (LTS2 - LTS4 / EPFL), Lausanne, Switzerland as a Postdoctoral Researcher in 2008, under the joint supervision of Prof. P. Vandergheynst and Prof. P. Frossard. He has been an Assistant Professor in Signal and Image processing since September 2009 at Université Aix-Marseille 2, France. His current research interests focus on Sparse Representations of signals, Inverse Problems and Computer Vision. His areas of applications include biomedical imaging, astronomy (Radio-interferometry and Solar physics) and omnidirectional vision.

Pierre Vandergheynst received the M.S. degree in physics and the Ph.D. degree in mathematical physics from the Université catholique de Louvain, Louvain-la-Neuve, Belgium, in 1995 and 1998, respectively. From 1998 to 2001, he was a Postdoctoral Researcher with the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He was Assistant Professor at EPFL (2002-2007), where he is now an Associate Professor. His research focuses on harmonic analysis, sparse approximations and mathematical image processing with applications to higher dimensional, complex data processing. He was co-Editor-in-Chief of Signal Processing (2002-2006) and is Associate Editor of the IEEE Transactions on Signal Processing (2007-present). He has been on the Technical Committee of various conferences and was Co-General Chairman of the EUSIPCO 2008 conference. Pierre Vandergheynst is the author or co- author of more than 50 journal papers, one monograph and several book chapters. He's a senior member of the IEEE, a laureate of the Apple ARTS award and holds seven patents.