SCOOP: A Real-Time Sparsity Driven People Localization Method

Mohammad Golbabaee, Alexandre Alahi and Pierre Vandergheynst

***Abstract*—Detecting and tracking people in scenes monitored by cameras is an important step in many application scenarios such as surveillance, urban planning or behavioral studies to name a few. The amount of data produced by camera feeds is so large that it is also vital that these steps be performed with the utmost computational efficiency and often even real-time. We propose SCOOP, a novel algorithm that reliably detects pedestrians in camera feeds, using only the output of a simple background removal technique. SCOOP can handle a single or many video feeds. At the heart of our technique is a sparse model for binary motion detection maps that we solve with a novel greedy algorithm based on set covering. We study the convergence and performance of the algorithm under various degradation models and provide mathematical and experimental evidence of both its efficiency and robustness using standard datasets. This clearly shows that SCOOP is a viable alternative to existing state-of-the-art people detection algorithms, with the marked advantage of real-time computations.**

***Index Terms*—People Localization, Sparse Representation, Dictionary, Multi-view, Greedy, Matching Pursuit, SCOOP, group testing**

I. Introduction

The present paper deals with a simple but very important problem in computer vision: given a set of cameras observing a scene (there can be only one camera as extreme example), we want to automatically detect pedestrians and locate them in the scene. The detection output would generally be used in a second step for tracking people, but we focus here on the detection/localization problem. This problem has numerous applications, surveillance being the most obvious, and has been the subject of intense research over the past decade. However, there remain two important challenges to most existing solutions:

- robustness: due to occlusions and variable lighting conditions, existing algorithms tend to produce false or miss detections. Often, robustness is achieved at the expense of computationally complex scene modeling
- computational complexity: cameras operating at 25 frames per second or more generate tremendous amount of data. In order to achieve real-time performance, existing algorithms have to sacrifice on robustness.

This trade-off between robustness and computational efficiency brings unbearable constrains on real-world applications where both are desirable. The objective of this paper is thus to propose a solution to the people detection problem that would be at the same time robust and computationally efficient.

In a previous paper, we have proposed a model of motion detection maps based on the assumption that the number of people in the scene is much smaller than the total possible ground locations [**?**]. The model was relaxed into a LASSO-like problem and solved with a re-weighted ℓ_1 algorithm. We showed that the resulting technique, deemed O-LASSO, reached state-of-the-art performances in terms of robustness. Unfortunately, O-LASSO is a computationally complex algorithm and, despite various optimizations, cannot reach real-time operation. Acknowledging that the excellent robustness properties reported in [**?**] were due to the sparsity hypothesis, we conserve that part of the model but we propose a completely different way of exploiting it. First, where O-LASSO was based on complex floating point calculations, we derive a new regression model that involves only boolean arithmetics and takes full advantage of the binary output of basic motion detection algorithms. Second, instead of solving a difficult convex optimization problem with iterative shrinkage, we derive a novel greedy algorithm inspired by the set cover problem. This algorithm operates with only binary operations and is therefore extremely efficient.

The relevance and performance of our model and algorithm are analyzed at two different levels. First we draw a connection with group testing that allows us to study the mathematical properties of the model and sate the existence and uniqueness of solutions. We also show that these solutions can be recovered by a simple thresholding algorithm. We then extend these findings and propose a greedy heuristics that also incorporates physical constrains on the localization of detected people, the resulting algorithm is called Set Covering Object Occupancy Pursuit or SCOOP. We then study experimentally its performances: SCOOP matches O-LASSO in terms of robustness but at a fraction of the computational cost, easily reaching real-time implementation.

II. Related work

A. People localization in camera networks

As hinted at above, the problem of detecting and localizing people in networks of camera has been the subject of an intense research activity. Let us review the main approaches leading to our own model. Detection can occur independently in each camera then fused across cameras [22], [23], or they can be detected concurrently in a unique referential [24], [25] since cameras are calibrated to match 3D points across image planes [26]. These approaches suffer to detect people occluding each other and a good alternative is to fuse features extracted from all cameras in a unique referential and make the decision once all features are combined. The most commonly used features are the silhouettes extracted from all cameras using a motion detection algorithm and a reference background

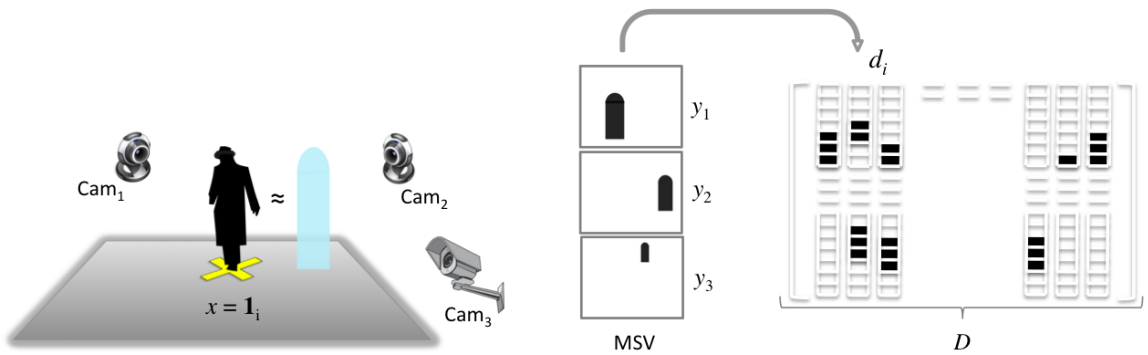
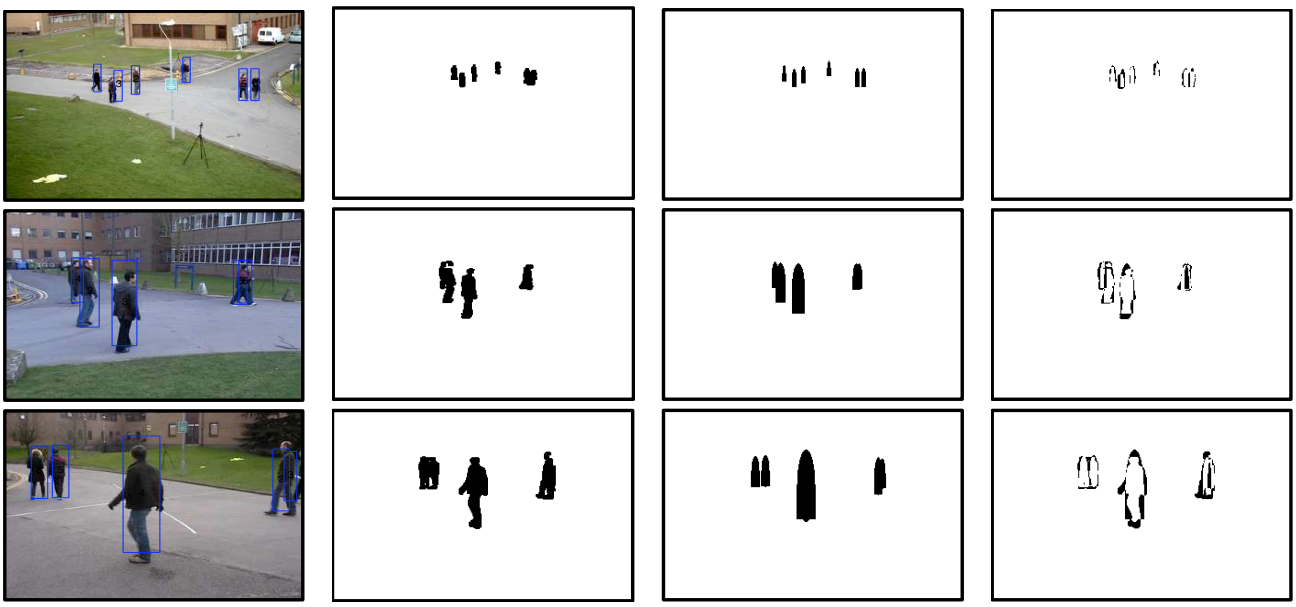


Fig. 1: Dictionary reconstruction: an atom/column d_i corresponds to the MSV of a half-rectangular-half-elliptical object approximating a person standing at position Id $= i$.



(a) Camera views (b) Extracted MSV (y) (c) Dictionary approximation ($D.x$) (d) Noise (z)

Fig. 2: Illustration of the boolean regression model described by equation (2) for three camera views of a single frame of the PETS 2009 dataset, http://winterpets09.net.

image. In [27] Khan and Shah project foreground silhouettes a reference ground plane given a global homography. By stacking and normalizing the obtained re-projected silouhettes, they construct a probability map. Alternatively probability maps over several planes can estimated [28], [29], which provides more robustness. Eshel and Moses in [30] use a probability map at head level, but grouped people with few number of cameras are hardly segmented. Closer to our model, Fleuret *et al.* in [31], [32] use a dictionary of ideal silhouettes that is used to detect people in crowded environment. Rectangular shape prototypes are used to approximate the ideal human foreground silhouettes observed by the cameras. They then estimate the probability of occupancy map (POM) of the ground plane at each time. Recently, Alahi *et al.* in [33] have also proposed a dictionary based framework with a generative model to approximate foreground silhouettes and their model is the starting point of our investigations. The localization of people in the monitored scene arises as the solution to an inverse problem referred to as O-Lasso. It outperforms previous approaches in terms of detection rate but it is computationally costly for real-time application. As a result, we propose in Section III a greedy approach that achieves the same detection rate but real-time performance.

III. Dictionary-based boolean regression model

Alahi *et al.* in [33] propose a sparsity driven framework that performs well with respect to the state-of-the-art. A key feature of this scheme is to cast the multi-view localization as a sparse linear inverse problem that is followed by a quantization step. A huge collection of silhouettes of an individual standing at various positions, that is called dictionary, is used for this purpose. Later, inspired by the recent massive developments in sparse linear approximation tools, an algorithm was proposed (O-Lasso) to approximate the foreground silhouettes by a few number of individuals. The main drawback of these schemes is their numerical complexity: they are based on iterative algorithms that converge slowly.

Our proposed approach is similar to the dictionary-based framework, however, we use a boolean (non-linear) regressive formulation to model the localization problem. Using boolean arithmetics, we design much faster and memory efficient approximation algorithms that are mainly rooted in the old literature of group testing and set cover. The following steps precisely describe our problem formulation:

1) Discretization of the scene: The ground plane is discretized into 2-D grid of N cells (sub-regions). We assume each cell can be occupied by only one person at each time instance. To simplify notations, the 2-D grid is concatenated into a 1-D vector $x \in \{0,1\}^N$, whose elements are indicating the presence of a person in the corresponding cell (with Id$= i$) if $x_i = 1$. Typically, we refer to this vector as the *occupancy vector*. An adaptive sampling process can be used to discretize the ground plane into non-regularly spaced grid points to take advantage of the cameras' topology and the scene activity [33].

2) Arranging the Foreground Silhouettes: A 2-D binary mask representing the foreground pixels observed by each camera c, is extracted given a background subtraction algorithm [3]. We used the well known mixture of Gaussians to classify each pixel as foreground [4]. The foreground images are also rearranged by concatenation of the 2-D masks into binary vectors $y_c \in \{0,1\}^{M_c}$. Each of these vectors contain the extracted foreground silhouettes from the corresponding camera c. A foreground silhouette is a connected region of foreground pixels. As a result, M_c denotes the resolution (number of pixels) of the c^{th} camera. Further, we concatenate all these vectors into the Multi-Silhouette Vector (MSV):

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

where, $M = \sum_{c=1}^{C} M_c$.

3) Dictionary construction: Imagine a person with a given volume walking in a scene. The shape observed by each camera can vary depending on the view-points and the behavior of that person. We approximate the shape of people with a half-rectangular-half-elliptical shape as in [33] (see Figure 1). We consider an average person with a height of 1m70 cm. A 3-D model is projected into all the camera views given the calibration data [34]. Such approximation is used to construct atoms of a dictionary $D \in \{0,1\}^{M \times N}$. Each column of the dictionary, i.e. each atom, represents the approximated multi-view silhouette observed at a given ground plane point. A typical dictionary contains a huge collection of atoms i.e., there is as many columns as the discretized ground plane points in the scene. Figure 1 demonstrates reconstruction of an atom of the dictionary used in people detection applications. Note that in urban scenes, dictionary atoms can be modified so that they can also represent approximate MSVs of various urban objects e.g., pedestrians, cars, buses, trucks, etc.

A. Boolean regression model

Suppose a single person occupies the scene. This refers to an occupancy vector x with only one nonzero element whose index depends on the location of that person on the scene. Moreover, each of the cameras will capture only one silhouette (if they have a view over the position) whose size, the position and possibly the shape depends on the location. In more general cases, for a given configuration of x with more nonzero elements (depending on the number of people and their positions), the resulting MSV may not be necessary unique. This non-uniqueness comes from the occlusions in the camera view which are highly dependent on the density of the crowd and the positions of the cameras.

Let us define the following *regression model* which describes the underlying correspondence between each occupancy vector and its resulting MSV:

$$y = D \cdot x \oplus z. \tag{2}$$

Note that here the operations are boolean i.e., sums and products correspond to AND and OR, and $\oplus$ denotes the bitwise XOR operation between two boolean vectors. The dictionary $D \in \{0,1\}^{M \times N}$, as previously defined, is a very huge matrix of silhouettes. Each column of D, say d_i, indicates the corresponding MSV of an average person who is standing at position i in the scene. Finally, $z \in \{0,1\}^M$ denotes the noise vector that corrupts the MSV by both missing and extra foreground pixels. This may occur due to several reasons e.g., non ideal silhouette extraction, non ideal modeling of the dictionary atoms, shadows, reflections, etc. Figures 2(a)-2(d) provide an illustration of the regression model described by the Equation (2).

Assume an occupancy vector x representing k individuals in a scene. The support of x is the set $\mathcal{S}$ that contains indices of the k nonzero elements, $\mathcal{S} = \text{supp}(x) := \{i : x_i = 1\}$. Equation (2) formulates the observed MSV as the boolean superposition of k atoms (bitwise OR among the columns) of the dictionary, indexed by $\mathcal{S}$ and possibly corrupted by some noise,

$$y = \sum_{i \in \mathcal{S}} d_i \oplus z. \tag{3}$$

Recalling that each of the atoms represents the silhouette of a single individual at a certain location, the use of the boolean operators in Equations (2) and (3), explicitly demonstrates the nonlinearity of the MSV model caused by occlusions in scenes with more than a single person. We use later this model frequently, specially as one of the most important priors in order to infer the locations of the individuals given their foreground MSV.

IV. Problem statement and connections with group testing

Considering descriptions in the previous section, the problem of detecting and localizing objects in a scene is equivalent to recover an occupancy vector from an inaccurate noisy MSV, provided with the knowledge of the dictionary D that links them through Equation (2). Accordingly, we formulate people detection and localization as a non-linear inverse problem, in which one need to identify the support set $\mathcal{S}$ that approximately leads to the observed MSV, even in presence of noise (see Equation (3)).

As one can observe, noise together with the non-linearity of the formulation can impose many different possible solutions

to Equation (2). Hence, different occupancy vectors may result in the same MSV. This fact severely challenges the performance of any decoding algorithm to reliably detect and localize the objects e.g., a decoder may mistakenly add or neglect some individuals.

In the next part, we determine necessary and sufficient conditions in order to preserve the uniqueness of the solutions. Our theoretical analysis finds interesting and intuitive implications in multi-view people detection and localization problem. Moreover, a simple algorithmic approach called Thresholding, is introduced to recover the occupancy vector from the MSV and we show it performs optimally i.e., if there exists a unique solution, Thresholding will recover it. We develop our results based on some popular tools existing in the well established group testing literature and therefore we show how these two problems are related to each other.

The classical group testing which was introduced by Dorfman [35] problem finds its historical roots in World War II, when blood samples of many U.S. soldiers were examined to detect few cases of syphilis. The main idea is to pool the blood samples into certain groups and test the groups instead of one by one testing. The original problem can be formulated as Equation (2) in the noiseless case, where, x contains N blood samples with sparse nonzero elements indicating the infected cases and D (called the *contact* matrix) determines the way of collecting $M < N$ group tests into a vector y. The question is, how to design a contact matrix and a recovery algorithm that can efficiently identify as many defective cases as possible? For more details see [36].

In our localization application, the design of the dictionary is however fixed by the number of the cameras, their relative positions, silhouette model and the density of the points on the scene. Thus, contrary to the group testing, our application deals mainly with the recovery problem rather than compression. Another difference is that the cameras are typically providing multi-view images with much higher resolution than the number of grid points on the scene i.e., $M \gg N$, which may help us compensate for the rather non-optimal design of the dictionary .

A. Uniqueness of the representation

Given the regression model (2), there can be many realizations of the occupancy vector leading to the same MSV y, which makes any decoding scheme hopeless to recover the original x. There are two main reasons for this non-uniqueness: the presence of noise and non-linearity of the formulation. Particularly, in people detection applications, occlusions often occur due the relative placement of the cameras and the people in the scene. Therefore, any decoder fails to decide correctly whether there are some individuals present at those positions. This section defines precisely a set of conditions that avoids such uncertainties and guarantees the uniqueness of the solution to (2). These constraints provide an upperbound on the performance of any recovery scheme and in addition, they measure how efficiently they do perform. In the following, we define the notion of disjunct matrices that is often used in group testing literature [36] and it appears to be the key element of the theoretical framework that we establish in this paper.

Definition 1: A boolean matrix D with N columns $d_1, ..., d_N$ is (k, e)-disjunct if for every subset $\mathcal{S} \subseteq \{1, ..., N\}$ with cardinality $|\mathcal{S}| \leq k$, and every $i \notin \mathcal{S}$, we have:

$$\Big|\mathrm{supp}(d_i) \backslash \bigcup_{j \in \mathcal{S}} \mathrm{supp}(d_j)\Big| > e,$$

where $|.|$ represents cardinality of a set, and $.\backslash.$ means set difference.

Assuming an occupancy vector that is k-sparse (i.e., x has less than k nonzero elements) and a noise that flips at most e bits of MSV (i.e., $|\mathrm{supp}(z)| \leq e$), the following proposition guarantees the uniqueness of the solution to (2):

Proposition 1: For any $(k, 2e)$-disjunct dictionary Equation (2) implies a one-to-one mapping between all k-sparse x and their corresponding y. Conversely, if every k-sparse x is mapped to a distinct y then D must be $(k\text{-}1, 2e)$-disjunct.

The proof of Proposition 1 is presented in the Appendix. In our people detection application, disjunction is a measure of robustness against occlusions and noise. Intuitively, Proposition 1 implies that any person at any position must have a silhouette with enough distinguishable pixels to be robust against the noise and not be submerged (occluded) into the silhouettes of other people.

By increasing the number of people on the scene, the occlusions become more probable, thus, Proposition 1 can also be interpreted as an upperbound for the number of individuals that can be reliably localized by a fixed camera setup. Moreover, by changing the camera setup e.g., increasing the number of the cameras and selecting well their positions with respect to the scene, one can design optimal dictionaries that are disjunct for larger number of people.

Note that in general, verifying wether a matrix is (k, e)-disjunct is a hard problem, which makes Proposition 1 impractical for large size setups. Moreover, Proposition 1 provides a *worst case* analysis for (2), since it guarantees the uniqueness for *all* occupancy vectors and it is robust against any *adversarial* noise setup. In practice, simulation results indicate a reliable recovery under much milder conditions than in Proposition 1, because the worst case situations are not very likely.

B. Decoding by Thresholding

In this part we introduce a simple algorithm for recovering the occupancy vector from the corresponding MSV. Note that a similar approach has been considered in [37], and in [38] for real-valued x. This algorithm works based on selecting atoms of D whose supports are approximately included in $\mathrm{supp}(y)$, that is the number of elements of $\mathrm{supp}(d_i)$ not included in $\mathrm{supp}(y)$ should not exceed a threshold.

Thresholding: *Select the columns of D that satisfy the following equation:*

Defining theorems environments & numbering

check if NPhard

$$\Big|\mathrm{supp}(d_i)\backslash\mathrm{supp}(y)\Big| \leq e, \tag{4}$$

and indicate their corresponding indices as the support of x.

The following theorem characterizes the performance of Thresholding and highlights its optimality for solving the regression model (2) when $|\mathrm{supp}(z)| \leq e$:

Theorem 2: *Thresholding successfully recovers any* k*-sparse occupancy vector, if* D *is* $(k, 2e)$*-disjunct.*

Theorem 2 shows that Thresholding achieves the optimal bound of Proposition 1 as long as there is an unique solution to Equation (2) (the proof is presented in the Appendix). In addition, note that if the dictionary is not $(k, 2e)$-disjunct then there may exist column indices $i \notin \mathrm{supp}(x)$ that also satisfy (4), and therefore the recovery is not exact but contains the support of the original occupancy vector i.e., $\mathrm{supp}(x) \subset \mathrm{supp}(\widehat{x})$, where $\widehat{x}$ denotes the recovered occupancy vector.

In our application, the highest value k to have a $(k, 2e)$-disjunct dictionary is much smaller than the number of potential individuals in the scene. As a result, for typical populated scenes, the occlusions become more probable and therefore many different realizations of the occupancy vector solve (2) (no unique solution). In this case, by setting correctly the threshold value in (4), the algorithm does not miss any individual, but its performance is dramatically affected by many false positives.

In the next section, we consider an additional prior to better recover the exact solution. We assume that among all possible solutions, the one of interest is the sparsest one, and we design a real-time algorithm in order to recover it. Nevertheless, we keep taking advantage of the output of Thresholding as a very fast preprocessing step which efficiently refines the search space, and thus, accelerates the main step of the recovery algorithm.

V. Real-time sparsity driven people localization

In practice, the configuration of the cameras and the density of the people on the scene are such that full occlusions are inevitable and therefore, there is no unique solution to the localization problem. As an example, there might be many positions on scene, entirely covered by the people near to the cameras so that no decoder would be able to decide wether there are some individuals hidden there or not. The Thresholding algorithm defined in the previous section outputs a conservative solution that considers all those points as if they are occupied by people and thus it results in too many false positives that are not desired.

In this section we address this problem by selecting the *sparsest* solution to (2), according to our hypothesis regarding the distribution of the individuals on the scene. Among all possible solutions, we set our problem to the recovery of the sparsest occupancy vector x. In the noiseless case, we thus solve

$$\begin{aligned} \widehat{x} \quad &= \arg\min_{x\in\{0,1\}^N} |\mathrm{supp}(x)| \\ s.t. \quad &\mathrm{supp}(y) = \mathrm{supp}(D\cdot x) \end{aligned} \tag{5}$$

and, in the noisy setup,

$$\begin{aligned} \widehat{x} \quad &= \arg\min_{x\in\{0,1\}^N} |\mathrm{supp}(x)| \\ s.t. \quad &|\mathrm{supp}(y \oplus D\cdot x)| \leq e. \end{aligned} \tag{6}$$

Note that, if disjunction holds, the solution of both problems coincide with the unique solution of (2) which can be simply identified by Thresholding. However, if D is not $(k, 2e)$-disjunct, this new approach neglects objects that might be fully occluded, and approximates the MSV by very few number of atoms of D corresponding to the large-size silhouettes (i.e., people who are mainly in front of the scene).

Problem (5) is equivalent to the well known set cover problem [39], which recovers the set of atoms with minimal cardinality, so that the union of its elements covers the support of y. In the noisy case (6), however, we relax the constraint since we are not interested in approximating the noise. Set cover is known as one of Karp's 21 NP-complete problems, thus designing feasible algorithms that can approximate the solution with polynomial time complexity is of high importance.

It has been shown in [39] that the simple greedy approach is indeed an effective way to approximate the solution of the set cover problem. This approach follows the heuristic of making the locally optimal choice at each iteration with the hope of finding the global optimum. For example, the greedy method proposed in [39] works iteratively and recovers one element of the support set (i.e., $\mathrm{supp}(x)$) per iteration. More precisely, at each iteration, the algorithm selects the index of the atom i of the dictionary which contributes the most in energy of the MSV y (i.e. the atom whose support shares the most common elements with $\mathrm{supp}(y)$). It then subtracts its contribution to update the remainder (initially MSV). This procedure continues until meeting the stopping criteria. This criteria can be either an a priori knowledge on the sparsity level, or a threshold on the energy level of the remainder. For the first criteria the algorithm performs k iterations and for the latter it runs until the remainder energy falls below a limit e.

It is noteworthy to mention that the same approach is extensively used in the compressed sensing and sparse approximation research literature because of the simplicity of the analysis and low computational cost [19], [20]. Among those, the Matching Pursuit (MP) algorithm [40] works quite similarly to above mentioned greedy algorithm, and its selection criteria is rephrased as choosing the atom having the highest coherence (i.e., inner product) with the remainder.

In the following part we introduce a novel method that extends the greedy approach described in [39] to approximate the noisy covering problem (5). We apply this approach to our localization problem and experimentally show that this approach outperforms the state-of-the-art algorithms, with a computational complexity amenable to real-time applications.

A. Set Covering Object Occupancy Pursuit: SCOOP

The proposed localization problem by its construction consists of non-normalized dictionaries i.e., the columns corresponding to the objects on the far back of the scene have much less energy than the ones in front because the corresponding

targets appear smaller. As a result, the selection criteria based on the maximal common elements in the support (in the original set cover problem) or maximal coherence (like in MP) often chooses high energy columns that cover highly the MSV as well as many pixels out of the MSV support. For example, a person at the back of the scene with small silhouette is mistakenly approximated by a person in front with a much larger, but well covering silhouette. This indicates that some high energy columns of D, despite their good covering, are not fitting the silhouettes well enough. It is thus necessary to modify the algorithm to avoid such mistakes. We address this problem so that, at each iteration, the selecting criteria searches for the column i that has the minimum difference with the remainder r (i.e., MSV at initial step). Thus, the selected column, in addition to a good covering must fit well the MSV i.e., not contain many extra pixels out of the MSV support. In summary, an iteration of SCOOP selects a column index i based on the following criteria:

$$i \Leftarrow \arg\min_i \Big\{ w \overbrace{\frac{\big|\mathrm{supp}(r)\backslash\mathrm{supp}(d_i)\big|}{\big|\mathrm{supp}(r)\big|}}^{\text{Covering factor}} + (1-w) \underbrace{\frac{\big|\mathrm{supp}(d_i)\backslash\mathrm{supp}(r)\big|}{\big|\mathrm{supp}(d_i)\big|}}_{\text{Fitting factor}} \Big\}, \tag{7}$$

where $0 \leq w \leq 1$ is a regularization factor to penalize the uncovered pixels in the remainder support and the extra covered pixels out the remainder support. This brings a degree of freedom to the algorithm that balances between the covering and fitting factors of the columns.

Compared to simple Thresholding this criteria better respects the sparsity constraint as we now argue. Typically, the superposition of several atoms with poor coherence can have the same energy contribution in the MSV as a single atom with high covering factor. Practically this means that atoms approximating faraway people can be covered by few atoms corresponding to close-by people or atoms corresponding to cars or trucks can cover several atoms corresponding to people standing at the same location. The proposed selection criteria however promotes the selection of the largest atom in case of ambiguity. Typically, it prefers to select a single atom of a truck than several people in the scene. Likewise, it prefers to select one single close-by person instead of several faraway people. As a result, we can say that it promotes a sparse solution. Nevertheless, as mentioned above, a fitting factor is used to avoid selecting atoms with too many out-support pixels.

The full algorithm is presented in table **??** and coined as 'Set Covering Object Occupancy Pursuit' (SCOOP). As we can see, Thresholding is used as a preprocessing step in order to reduce the dimension of the search space $\mathcal{U}$ of all possible locations. When $|\mathcal{U}| \ll N$, this step can massively accelerate the main greedy pursuit.

Note that the stopping criteria can adopt three forms: either one knows a priori how many individuals are present in the scene (e.g., team sports like basketball, soccer,...) and runs k iterations to detect them or a good estimation of the noise power e is available to the decoder, which leads to the same criteria used in table **??**. Finally, if the decoder does not have access to any of those priors, it continues the iterations until by adding the next index the outcome error E (see table **??**) starts to increase.

Repulsive Spatial Sparsity (RSS): In many application (including people localization) there exists another sort of sparsity: *spatial sparsity*. Two individuals are separated by a minimum spatial distance related to the minimum surface occupied by a person on the ground e.g., here we choose 70 cm to be the average width of a standing person. This is what we refer to as the concept of *Repulsive Spatial Sparsity* (RSS) introduced in [33]. More precisely, if $i, j \in \mathrm{supp}(x)$ and $i \neq j$ then we must have,

$$\Delta_{i,j} := \|\mathbf{P}(i) - \mathbf{P}(j)\|_2 > \tau, \tag{8}$$

where $\mathbf{P}(i)$ is the position of a point i on the ground plane, and τ cm is the minimum spatial distance. Lets denote by $\mathcal{N}_\tau(i)$ the set of indices corresponding to positions that are τ-close to $\mathbf{P}(i)$ i.e.,

$$\mathcal{N}_\tau(i) := \{j : \Delta_{i,j} \leq \tau\}. \tag{9}$$

Finding a sparse occupancy vector does not necessarily impose the constraint above. For this purpose, at each iteration, SCOOP excludes all the neighboring points $\mathcal{N}_\tau(.)$ of the selected atom from the search space and modifies $\mathcal{U}$.

B. Complexity of SCOOP

All atoms selected by Thresholding satisfy inequality (4). This criteria is directly related to the boolean inner product, which counts number of elements that two m-dimensional boolean vectors share in their supports. We define the inner product of y and ith column of D as,

$$\begin{aligned} \langle d_i, y\rangle &:= \sum_j d_{ji}.y_j \\ &= |\mathrm{supp}(d_i) \cap \mathrm{supp}(y)|, \end{aligned}$$

with the predefined boolean arithmetic notations. Computing this inner product for all N atoms of the dictionary leads the preprocessing step to complete with complexity $\mathcal{O}(MN)$. The main greedy pursuit performs iteratively. Thanks to the preprocessing step, each iteration consists of searching over $u = |\mathcal{U}| \ll N$ atoms of the dictionary for finding the maximizer of (7). By simple computation we can rewrite (7) as,

$$i \Leftarrow \arg\max_{i\in\mathcal{U}} \Big\{ w \frac{\langle r, d_i\rangle}{|\mathrm{supp}(r)|} + (1-w) \frac{\langle r, d_i\rangle}{|\mathrm{supp}(d_i)|} \Big\}. \tag{10}$$

Therefore, each iteration roughly consists of computing u inner products between m-dimensional vectors, finding their maximum and modifying $\mathcal{U}$ by finding (and excluding) the neighbors of the selected atom i.e., complexity of $\mathcal{O}(Mu + u\log u) \approx \mathcal{O}(Mu)$. Now, if we consider a typical scene with k individuals (sparsity is known a priori e.g., in team sport

Algorithm 1: Set Covering Object Occupancy Pursuit' (SCOOP)

Input: MSV signal y, Dictionary D, Error parameter e, RSS parameter τ.

Output: Support set $\widehat{\mathcal{S}}$ (equivalently the occupancy vector $\widehat{x}$).

Initiation:

$\widehat{\mathcal{S}} \Leftarrow \{\}$, $\mathcal{U} \Leftarrow \{\}$, $r \Leftarrow y$, $\widehat{y} \Leftarrow \mathbf{0}$

Preprocess:

for $(i = 1 : n)$ **do**

  if $\Big(\big|\mathrm{supp}(d_i)\backslash\mathrm{supp}(y)\big| \leq e\Big)$

  $\mathcal{U} \Leftarrow \mathcal{U} \cup \{i\}$.

  end

end

Greedy Process:

while $(E > e)$ **do**

$$j \Leftarrow \arg\min_{j' \in \mathcal{U}} \Big\{ w \, \frac{\big|\mathrm{supp}(r)\backslash\mathrm{supp}(d_{j'})\big|}{\big|\mathrm{supp}(r)\big|} + (1-w) \frac{\big|\mathrm{supp}(d_{j'})\backslash\mathrm{supp}(r)\big|}{\big|\mathrm{supp}(d_{j'})\big|} \Big\}$$

  Updates:

  Recovered support: $\widehat{\mathcal{S}} \Leftarrow \widehat{\mathcal{S}} \cup \{j\}$

  Recovered MSV: $\mathrm{supp}(\hat{y}) \Leftarrow \mathrm{supp}(\hat{y}) \cup \mathrm{supp}(d_j)$

  Remainder: $\mathrm{supp}(r) \Leftarrow \mathrm{supp}(r)\backslash\mathrm{supp}(d_j)$

  Search space: $\mathcal{U} \Leftarrow \mathcal{U} \setminus \mathcal{N}_\tau(j)$

  Error: $E \Leftarrow \Big|\mathrm{supp}(y \oplus \hat{y})\Big|$

end

activities), the whole complexity of SCOOP in table 1 scales as $\mathcal{O}(kMu)$.

Compared to O-Lasso, our method performs enormously faster so that, in typical problem sizes (i.e., typical M, N, k), it is able to detect and localize the objects of a frame in real-time. Mainly two reasons are behind the success of SCOOP: First, a novel formulation of the problem based on boolean regression model. Second, using the greedy approach to solve the localization problem. Unlike O-Lasso, our approach does not need to solve a sparsity-inducing convex optimization (reweighed ℓ_1-minimization) with heavy computations. As a consequence, instead of performing many iterations to converge in a solution (like in O-Lasso), our approach identifies one individual per iteration. In addition, each iteration of SCOOP performs only basic boolean arithmetic operations that are cheap in terms of computational complexity and memory usage. In the next section we demonstrate by several simulations that this low complexity does not impair robustness.

VI. Experimental results and comparisons

In this section we present results of a few experiments on the APIDS dataset[1] that consists of seven pseudo-synchronized cameras monitoring a basketball game (including one omni-directional camera). We evaluate the performance over the left-half of the basketball court wherein the most number of cameras are monitoring the game i.e., cameras id 1, 2, 4, 5, and 7. For this purpose, a dictionary D has been constructed using camera calibration data (similar to the one in [33]). This dictionary maps the grid points that densely sample the basketball court (distance between each two adjacent gird points $\approx$ 10cm) into their approximate MSV. The performance of the detection process is quantitatively measured by computing the *Precision* and the *Recall* measures given by the following ratios:

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN}, \qquad (11)$$

where TP, FP and and FN are the number of True Positive, False Positive and False Negative. A true positive is when a person is correctly located on the ground plane.

Mainly, two classes of experiments are performed as following:

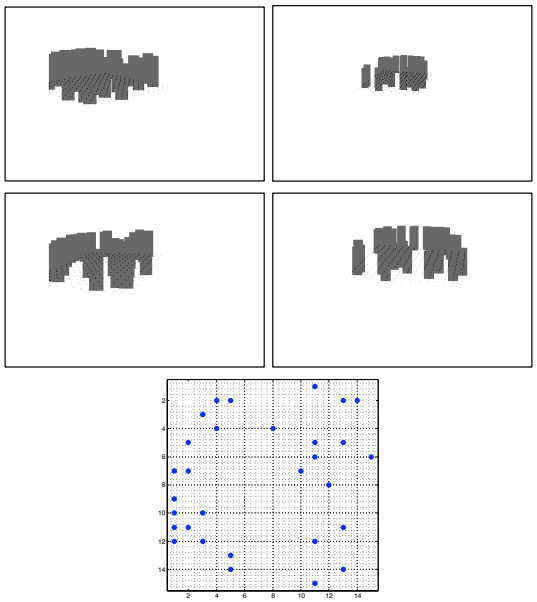


Fig. 3: Demonstration of a densely populated synthetic scene: $k = 30$ synthetic rectangular objects are randomly distributed on $N = 15\times15$ grid points. Silhouettes of the objects viewed by four cameras as well as their location on the grid are shown.

A. Synthetic-Noiseless Setup

Once the dictionary is constructed, we are able to synthesize foreground silhouettes with the same scene geometry as in APIDIS dataset. We generate random occupancy vectors x and the corresponding noiseless MSVs are computed by $y = D.x$. Our main goal here is to analyze the performance of SCOOP as the problem size scales i.e., number of objects or size of the scene.

First we choose a submatrix of the original dictionary which corresponds to a square subregion of the basketball court including 15×15 grid points observed by four cameras. MSVs are synthesized from occupancy vectors corresponding to k individuals/objects randomly distributed on the scene, and the results are averaged over a hundred independent realizations.

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

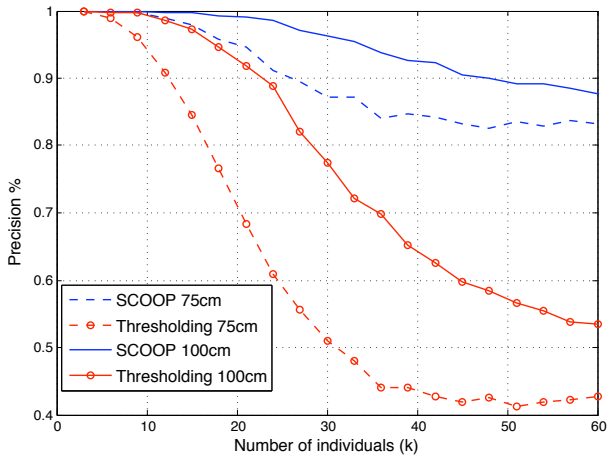


Fig. 4: Comparing the precision of SCOOP and Thresholding on synthetic data. Solid and dashed-line curves correspond to setups wherein any two adjacent points on the grid have 100 cm and 75 cm distance, respectively.

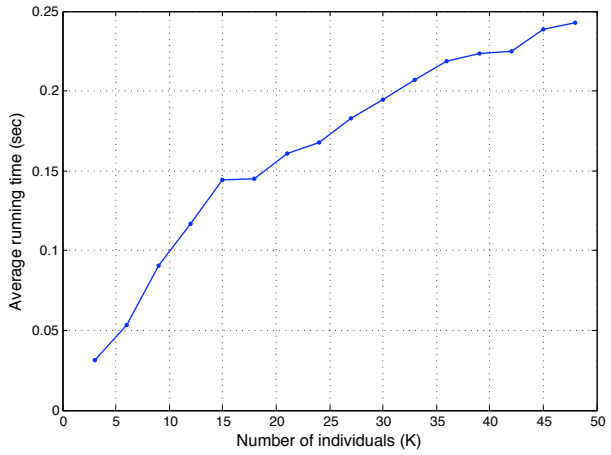


Fig. 5: Average running time (per frame) of SCOOP for localizing various number of individuals on a scene with $N = 225$ grid points, using four cameras (synthetic data).

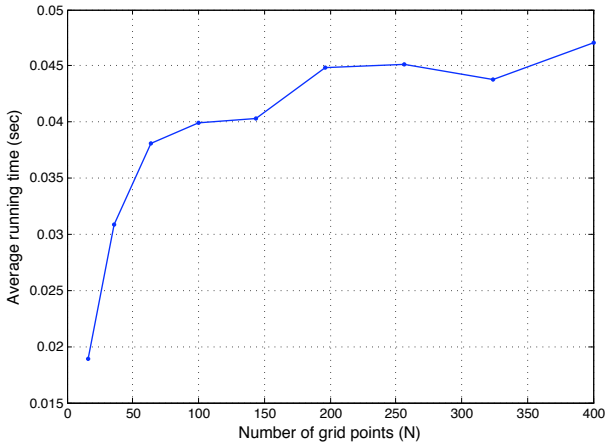


Fig. 6: Average running time (per frame) of SCOOP for localizing $k = 5$ individuals who are randomly distributed on scenes with different number of grid points N, using four cameras (synthetic data).

Figure 3 illustrates a realization of such densely populated scene (for $k = 30$) and the rectangular-shaped silhouettes observed by four cameras.

In figure 4 we compare Thresholding and SCOOP methods. Solid and dashed-line curves correspond to setups wherein any two adjacent points on the grid have 100 cm and 75 cm distance, respectively. For both methods the average performance decreases as the scene becomes more dense i.e., more people/objects or less relative distance between them. We can observe that by increasing the number of individuals, Thresholding reports many false positives (due to the ambiguity raising by many positions hidden from the camera views) resulting in a huge decrease in the method precision. In contrast, SCOOP discard many of those false positives by selecting a sparse occupancy vector, which let the method to be more robust against densely populated scenes.

Figures 5 and 6 demonstrate running time of scoop for different problem sizes (again, averaged over hundred synthetic frames). We can observe that running time of SCOOP almost linearly increases by adding more individuals. This comes from the fact that SCOOP wont result in too many false positive and thus, it manages to localize the people within few number of iterations proportional to the number of the individuals. In figure 6, for $k = 5$ individuals, we vary the size of the scene i.e., $N = i \times i$ for $i \in \{4, ..., 20\}$. The running time of SCOOP scales sub-linearly with respect to the scene size. This highlights the advantage of the preprocessing step (also RSS step) in SCOOP which reduces the search space at each iteration of the greedy pursuit i.e., in our experiments as the dimension N grows, the search space $\mathcal{U}$ grows sub-linearly for a fixed k.

B. Sequence of a Basketball Match

As previously mentioned, we consider left-half of the basketball court monitored by cameras id=1, 2, 3, 5 and 7 (see Figure 7). All videos are scaled to a QVGA resolution with approximately 25 fps and the foreground silhouettes are extracted using the work of Stauffer and Grimson [4]. The dataset has several challenges: Basketball players may have unexpected changes of behavior, e.g., running, jumping, crouching, sudden changes in the motion path, etc. Players can be either strongly grouped together or spatially scattered. Shadows and reflection of the players on the ground floor mislead many typical silhouette extraction techniques, and they output severely degraded MSVs often corrupted by many false positive pixels (i.e., noisy data).

We run several experiments on this dataset and we measure the performance of SCOOP together with several state-of-the-art methods of localization namely, the sparsity-driven convex-approach in [33] (RW-Lasso, RW-BPDN, O-Lasso) and the work of Fleuret *et al.* in [32] (referred to as POM). Results are reported in Figure 8 and providing a clear comparison between performance of SCOOP and its counterparts; SCOOP outperforms RW-Lasso, RW-BPDN, POM, and record a similar performance as for O-Lasso.

In addition to its precision, our method offers a huge acceleration in detection time. We measure the computation

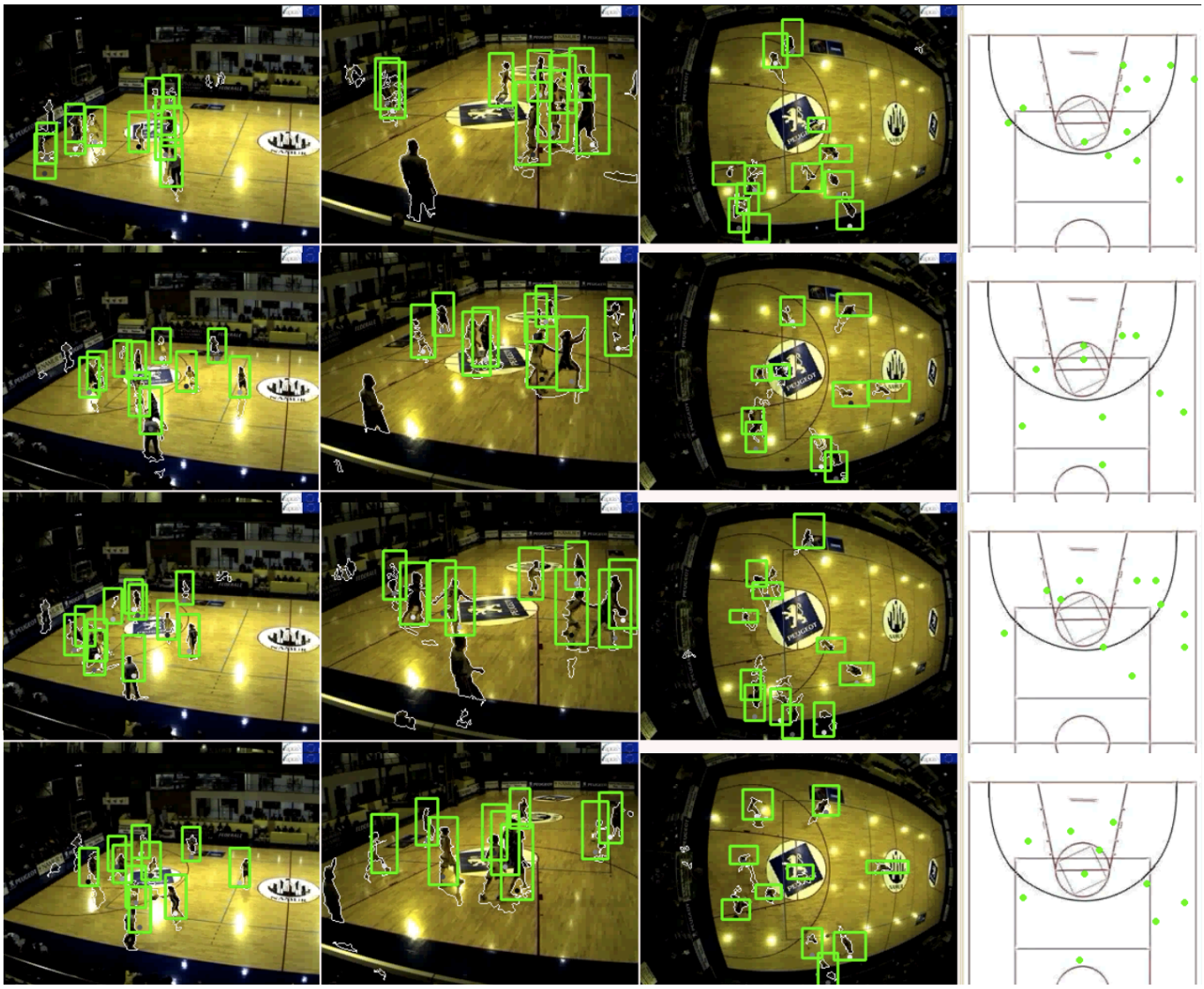


Fig. 7: Detecting and localizing players in the APIDIS dataset using SCOOP: demonstration for four frames and three camera views per frame. players' positions are marked for the left-half of the basketball court.

time of SCOOP as opposed to O-Lasso. Given the setup above with five cameras and a non-optimized matlab implementation for both algorithms, SCOOP locates the basketball players in 0.1 sec/frame on average, whereas O-lasso takes 10 sec/frame. Videos corresponding to a real-time implementation of SCOOP on C++, using Kineckt depth cameras for precise foreground silhouettes extraction, are available at **www.lts2.???**.

Finally, to evaluate the influence of noisy observations, we also evaluate the performance of various methods over noise-free foreground silhouettes. Synthetic foreground silhouettes are constructed as explained above with a spatial sparsity constraint; location points have a minimum spatial distance with respect to each other (> 70 cm). Five to fifteen people are randomly triggered for each frame (few hundred frames are generated). Given the synthetic data, we also obtain similar performance (see Figure 9). SCOOP outperforms other sparsity driven formulations such as RW-BPDN and RW-Lasso presented in [33] as well as POM [32].

VII. Conclusions

A sparsity driven system has been presented to capture people's motion behavior given a network of fixed cameras and analyze it in real-time. The proposed data collection algorithm is robust to noisy observation present in real-world conditions, outperforming previous works. It is optimized to monitor large scenes with minimal computational constraint. To analyze the captured data, the behavior of people is studied given the POI/AP framework where POIs are automatically identified and ranked given their mutual flows. The proposed system could be further used to quantitatively study human psychology over large scale data given specific events that inflence their behavior such as color, or layout.

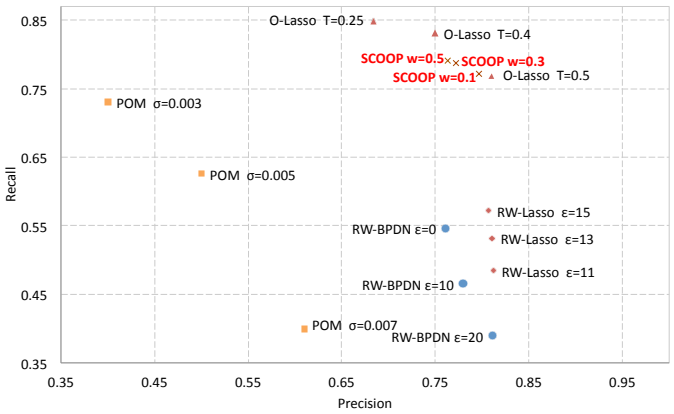


Fig. 8: Precision and recall rate with the Apidis dataset given four cameras (with severely degraded foreground silhouettes). Our proposed approach SCOOP is compared with other sparsity driven formulation and the probability of occupancy (POM) approach presented by Fleuret *et al.* in [32].

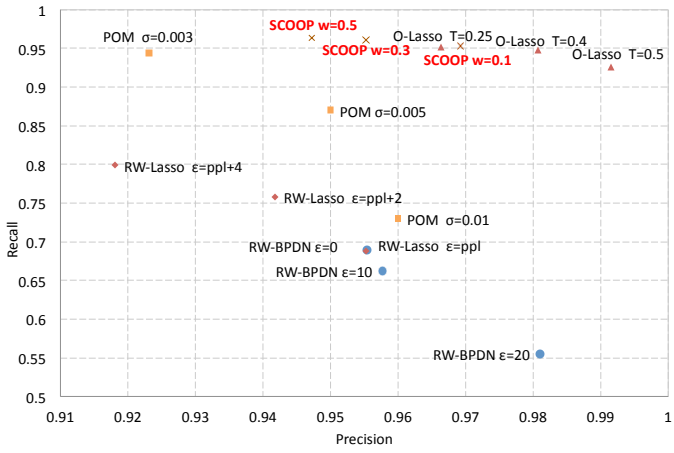


Fig. 9: Precision and recall rate with the synthetic data given four cameras (Noiseless foreground silhouettes, however with possible occlusions). Our proposed approach SCOOP is compared with other sparsity driven formulation and the probability of occupancy (POM) approach presented by Fleuret *et al.* in [32].

APPENDIX

A. Proof of Proposition 1

Having a $(k, 2e)$-disjunct dictionary implies a one-to-one map between all k-sparse x and their corresponding y. Conversely, if every k-sparse x leads to a distinguishable y then, D must be $(k\text{-}1,\ 2e)$-disjunct.

Proof: Assume two different k-sparse boolean vectors x and x' that are supported on sets $\mathcal{S}$ and $\mathcal{S}'$. Let $\overline{y}$ and $\overline{y}'$ be correspondingly the bitwise OR of the columns of D indexed by $\mathcal{S}$ and $\mathcal{S}'$ i.e., $\overline{y} = D.x$ and $\overline{y}' = D.x'$ with boolean arithmetic. Choose a column of the dictionary d_i so that, $i \in \mathcal{S}'$ but not in $\mathcal{S}$. Since D is $(k, 2e)$-disjunct, it implies the support of d_i has $2e + 1$ elements that are not included in $\text{supp}(\overline{y})$. Therefore, assuming noises that flip e bits of d_i and e bits of $\overline{y}$, there will be still at least one element of $\text{supp}(d_i)$ that is not included in the support of y (recall $y = \overline{y} \oplus z$), which makes x and x' distinguishable from their noisy MSVs.

For the converse, suppose D is not a $(k\text{-}1,\ 2e)$-disjunct matrix and pick a pair of a set $\mathcal{S} \subseteq [n]$ with $|\mathcal{S}| \leq k-1$, and an index $i \notin \mathcal{S}$ that is a counterexample to the $(k\text{-}1,\ 2e)$-disjunctness. Assume vectors x and x' that are supported on $\mathcal{S}$ and $\mathcal{S} \cup \{i\}$ correspondingly. The noise can configure in an adversarial way, so that, by flipping e zero bits of $\overline{y} = D.x$ to one, and e one bits of d_i to zero, the MSV outcome of x and x' becomes indistinguishable. ■

B. Proof of Theorem 2

Thresholding successfully recovers any k-sparse occupancy vector, if D is $(k, 2e)$-disjunct.

Proof: Assume a boolean vector x, supported on the set $\mathcal{S} \subseteq [n]$ with $|\mathcal{S}| \leq k$. Since the noise has flipped at most e bits of y, obviously every $i \in \mathcal{S}$ satisfies (4).

In addition, define $\overline{y}$ to be the bitwise OR of the columns of D indexed by $\mathcal{S}$ i.e., the noiseless version of the MSV. Again by the assumption on the noise power, for any column of D we have,

$$|\text{supp}(d_i) \backslash \text{supp}(\overline{y})| \leq |\text{supp}(d_i) \backslash \text{supp}(y)| + e.$$

Now, if any $i \notin \mathcal{S}$ satisfies (4), it implies $|\text{supp}(d_i) \backslash \text{supp}(\overline{y})| \leq 2e$, which violates the assumption that the dictionary is $(k, 2e)$-disjunct. Therefore, thresholding recovers exactly the support of x. ■

ACKNOWLEDGMENT

Part of this work was funded by the EU under research projects FET-OPEN 225913 (SMALL), and ICT-216023 (APIDIS).

REFERENCES

[1] "By herb sorensen, ph.d., scientific advisor, tns retail and shopper," 2010.
[2] "From keeneo website: http://www.keeneo.com/vs1.html," 2010.
[3] 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.
[4] 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.
[5] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," 1998.
[6] 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.
[7] 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.
[8] V. Cevher, M. Duarte, and R. Baraniuk, "Distributed Target Localization via Spatial Sparsity," in *Proc. European Signal Processing Conference*, 2008.
[9] G. Bretti, M. Fornasier, and F. Pitolli, "Electric current density imaging via an accelerated iterative algorithm with joint sparsity constraints," 2009.
[10] I. Loris, G. Nolet, I. Daubechies, and F. Dahlen, "Tomographic inversion using ?1-norm regularization of wavelet coefficients," *Geophysical Journal International*, vol. 170, no. 1, pp. 359–370, 2007.
[11] I. Daubechies and G. Teschke, "Variational image restoration by means of wavelets: Simultaneous decomposition, deblurring, and denoising," *Applied and Computational Harmonic Analysis*, vol. 19, no. 1, pp. 1–16, 2005.
[12] M. Elad, J. Starck, P. Querre, and D. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Applied and Computational Harmonic Analysis*, vol. 19, no. 3, pp. 340–358, 2005.

[13] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of statistics*, vol. 32, no. 2, pp. 407–451, 2004.

[14] D. Donoho, "Superresolution via sparsity constraints," *SIAM Journal on Mathematical Analysis*, vol. 23, p. 1309, 1992.

[15] D. Donoho *et al.*, "Nonlinear solution of linear inverse problems by wavelet-vaguelette decomposition," *Applied and Computational Harmonic Analysis*, vol. 2, no. 2, pp. 101–126, 1995.

[16] J. Starck, M. Nguyen, and F. Murtagh, "Wavelets and curvelets for image deconvolution: a combined approach," *Signal Processing*, vol. 83, no. 10, pp. 2279–2283, 2003.

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

[18] M. Fadili and J.-L. Starck, "Monotone operator splitting for fast sparse solutions of inverse problems," *SIAM Journal on Imaging Sciences*, pp. 2005–2006, 2009.

[19] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 629–654, 2008.

[20] M. Fornasier and H. Rauhut, "Iterative thresholding algorithms," *Applied and Computational Harmonic Analysis*, vol. 25, no. 2, pp. 187–208, 2008.

[21] B. Elahi, *Spirituality is a science: foundations of natural spirituality*. Associated University Presses, 1999.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] K. Mueller, A. Smolic, M. Droese, P. Voigt, and T. Wienand, "Multi-texture modeling of 3d traffic scenes," in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol. 1, 2003, pp. I – 657–60 vol.1.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] J. Berclaz, F. Fleuret, and P. Fua, "Robust people tracking with global trajectory optimization," in *Conference on Computer Vision and Pattern Recognition*, 2006.

[32] 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.

[33] A. Alahi, L. Jacques, Y. Boursier, and P. Vandergheynst, "Sparsity driven people localization with a heterogeneous network of cameras," *Journal of Mathematical Imaging and Vision*, pp. 1–20, 2011, 10.1007/s10851-010-0258-7. [Online]. Available: http://dx.doi.org/10.1007/s10851-010-0258-7

[34] J. Kannala and S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE Trans. on pattern analysis and machine intelligence*, vol. 28, no. 8, p. 1335, 2006.

[35] R. Dorfman, "The detection of defective members of large populations," *Annals of Mathematical Statistics*, vol. 14, pp. 436–440, 1943.

[36] D. Z. Du and F. Hwang, *Combinatorial Group Testing and its Applications.* World Scientific Series on Applied Mathematics, 1999.

[37] A. K. M. Cheraghchi, A. Hormati and M. Vetterli, "Group testing with probabilistic tests: Theory, design and application," *IEEE Transactions on Information Theory*, 2010.

[38] K. Schnass and P. Vandergheynst, "Average performance analysis for thresholding," *Signal Processing Letters, IEEE*, vol. 14, no. 11, pp. 828–831, 2007.

[39] V. Vazirani, *Approximation algorithms*. Springer Verlag, 2001.

[40] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993.