Sparsity & Surveillance

Rhian Davies (2^{nd} year PhD student) supervised by: Idris Eckley Lyudmila Mihaylova Nicos Pavlidis

STOR-i CDT Lancaster University

Motivation

Surveillance cameras are ubiquitous in many countries, constantly collecting and storing a **huge stream of data**. A typical video camera will need to store $10,368,000$ **pixels every second**. This large data mass leads to the need for **accurate, fast and automated systems** to convert raw video into information.

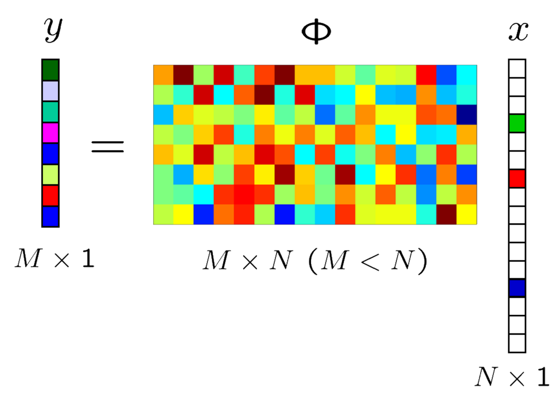
The Problem: Classifying video as foreground & background.

- Traditionally background subtraction is used, modelling the background of video and then subtracting this from the current frame.
- Standard background subtraction requires knowledge of every pixel.
- However **foreground is sparse** both spatially and temporally.
- It is wasteful of resources to store and process every pixel, when we only want to learn about the sparse foreground.
- Compressive sensing can help.

What is Compressive Sensing?

Compressive sensing is a method that **reduces the amount of data collected** from a signal without compromising the ability to later **reconstruct the signal accurately.** This will only work if the signal is **sparse**. The size of the data collected is reduced from N (all the pixels) to M.

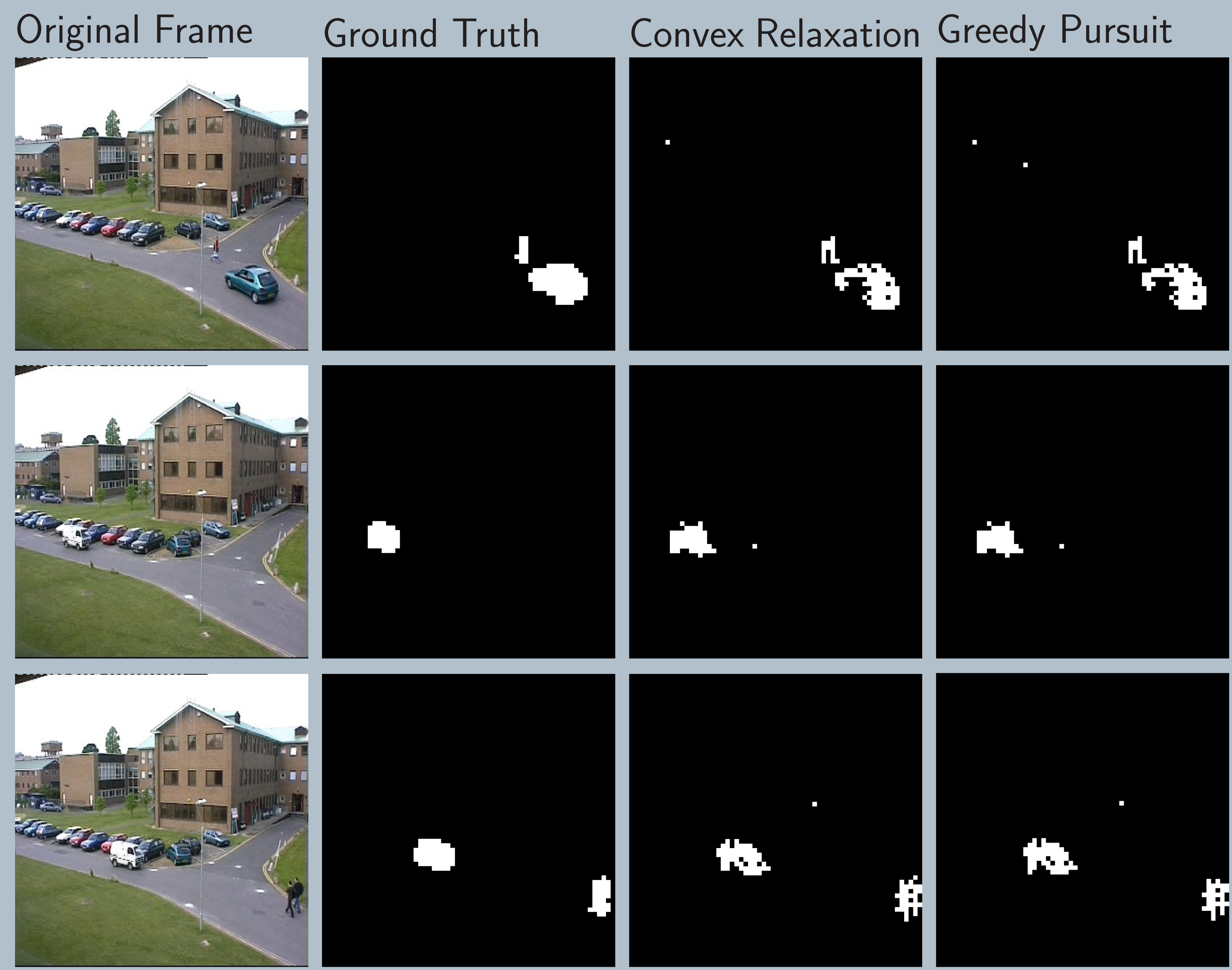
- Select $M \ll N$.
- Encode the signal $\mathbf{x}$ using matrix $\mathbf{\Phi}$.
- Decode the encoded signal $\mathbf{y}$ using recovery algorithm Δ.



Compressive Sensing Background Subtraction

- Compressively sense each frame at time t $\quad \mathbf{y_t} = \mathbf{\Phi x_t}$.
- Reconstruct the foreground mask directly $\quad \mathbf{fg_t} = \Delta(\mathbf{y_t} - \mathbf{y_t^b})$.
- Update the compressed background $\quad \mathbf{y_{t+1}^b} = \alpha \mathbf{y_{t+1}} + (1-\alpha)\mathbf{y_t^b}$.

It is possible estimate the location and shape of foreground activity based **only on the compressed background model and frame** allowing a reduction in storage and computation at the sensor.



Algorithmic Approaches

Reconstruction algorithms Δ must take the M measurements in the vector $\mathbf{y}$, the random measurement matrix $\mathbf{\Phi}$ and reconstruct the size-N signal $\mathbf{x}$.

- Since $M \ll N$ there are **infinitely many solutions** $\hat{x}$ that satisfy $\mathbf{y} = \mathbf{\Phi x}$.
- But we know **$\mathbf{x}$ is sparse** and can use this key fact to select a solution.
- The **ℓ_0 norm** counts the number of non-zero entries in $\hat{\mathbf{x}}$.
- We could minimize this to find a suitable solution;

$$\min_{\mathbf{x}} ||\mathbf{x}||_0 \text{ subject to } \mathbf{y} = \mathbf{\Phi x}.$$

- Unfortunately solving this is **NP-complete**.

There are two main alternative approaches to solving this problem.

- **Convex Relaxation.**
  Replaces the combinatorial problem (ℓ_0 norm) with a convex optimization problem (ℓ_1 norm).
- **Greedy Pursuit.**
  Iteratively refines a sparse solution by successively identifying one or more components that yield the greatest improvement in signal quality.

Our current work investigates the performance of these two methods in the foreground detection application.

Convex Relaxation: ℓ_1 minimization

Originally used in geophysics to aid detection of sparse spike trends in earthquake data, optimisation based on the ℓ_1 norm can approximate sparse signals with high probability. The ℓ_1 norm is defined as,

$$||\mathbf{x}||_1 = \sum_{i=1}^{N} |x_i|.$$

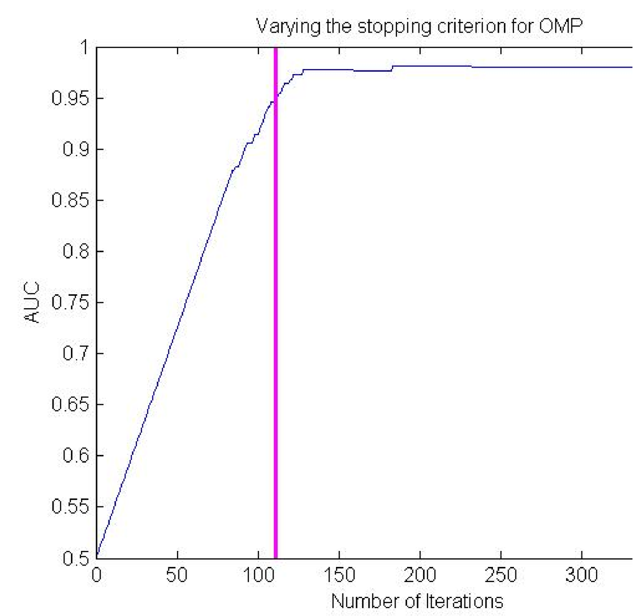
Minimising the ℓ_1 norm **determines the simplest solution** x in terms of the ℓ_1 norm which explains the observations.

$$\min_{\mathbf{x}} ||\mathbf{x}||_1 \text{ subject to } \mathbf{y} = \mathbf{\Phi x}.$$

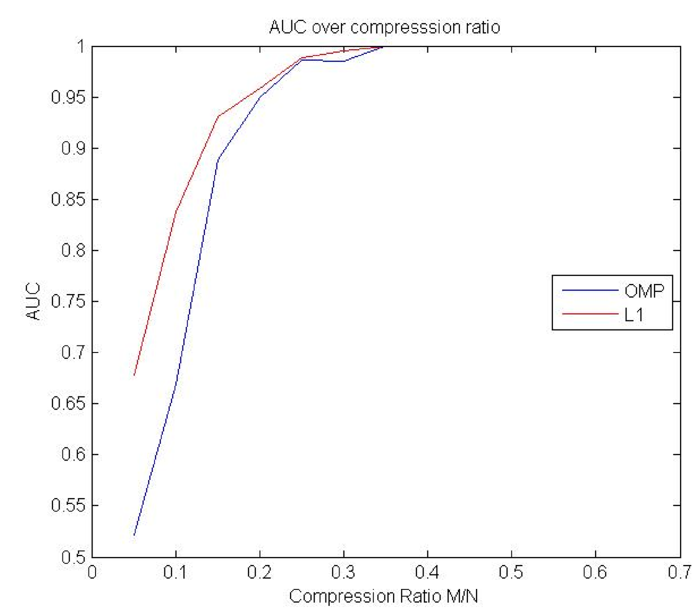
Greedy Pursuit: Orthogonal Matching Pursuit (OMP)

- OMP iteratively computes local optimum solutions in the hope that these will lead to the global optimum solution.
- Each iteration it determines the column of $\mathbf{\Phi}$ which is most correlated with $\mathbf{y}$.
- It uses this column to estimate one component of the solution $\mathbf{x}$.
- This is repeated until it reaches some user-defined stopping criterion.
- The **number of iterations is key** to the performance of the algorithm, both in terms of signal quality and computational time.

Results



(a) Stopping criterion for OMP



(b) AUC for Sparsity Ratios

Conclusion and Future Plans

- ℓ_1 minimisation outperforms OMP, but it's close!
- The stopping criterion is vital for OMP - adaptive methods needed?
- Ideal boundaries for compression ratio of $\frac{M}{N}$ around $25\% - 35\%$
- Can we incorporate spatial information to aid the recovery process?