

Direct Tracking from Compressive Imagers: A Proof of Concept

Henry Braun*, Pavan Turaga*[†], and Andreas Spanias*

*SenSIP Center, School of Electrical, Computer, and Energy Engineering

[†]School of Arts, Media, and Engineering

Arizona State University

Tempe, Arizona 85281

Email: {hcbraun, pturaga, spanias}@asu.edu

Abstract—The compressive sensing paradigm holds promise for more cost-effective imaging outside of the visible range, particularly in infrared wavelengths. However, the process of reconstructing compressively sensed images remains computationally expensive. The proof-of-concept tracker described here uses a particle filter with a likelihood update based on a “smashed filter” which estimates correlation directly, avoiding the reconstruction step. This approach leads to increased noise in correlation estimates, but by implementing the track-before-detect concept in the particle filter, tracker convergence may still be achieved with reasonable sensing rates. The tracker has been successfully tested on sequences of moving cars in the PETS2000 dataset.

I. INTRODUCTION

If commercialized, compressive sensing (CS) infrared cameras could dramatically lower the cost of sensors in a wide variety of applications. However, the output of any compressive sensor must be processed, or reconstructed, before it is readable by a human operator. In the case of compressively sensed video, real-time reconstruction is extremely computationally costly. If automated systems are developed which perform their tasks without reconstructing data, this problem may be avoided.

A proof-of-concept automated tracking system for surveillance applications is presented in this paper. A particle filter based implementation of the track-before-detect sensing paradigm is used to estimate target location. The likelihood calculation step of the particle filter is handled by an implementation of the smashed filter [1], described in Section II-B.

This work focuses on automated surveillance applications of CS cameras, a constrained problem with desirable properties for a first attempt at a direct CS processing system. Most notably, many surveillance cameras are stationary. Many surveillance cameras also have a wide field of view and are intended to detect targets which are small relative to the total image. These characteristics may be exploited to simplify the tracking problem, for instance by allowing fast and simple background subtraction.

The tracker was evaluated using Monte Carlo trials on a short video sequence from the PETS2000 dataset. On this sequence, the tracker was able to converge to the correct target location at a sensing rate of 0.3, a level comparable to that required to generate high quality image reconstructions.

The paper is organized as follows: Section I-B describes existing work related to compressive sensing, target tracking, and detection. Section II gives an overview of the tracking algorithm, including the particle filter motion model and the smashed filter based likelihood function. In Section III, the test procedures and results used to verify operation of the tracker are described. Finally, Section IV discusses the implications of the simulation results and suggests directions for further research.

A. Overview of compressive sensing

The problem of compressive sensing and reconstruction in image and video applications is described here. In a typical compressive sensing implementation, a vectorized video frame \mathbf{x} is known to be sparse in some basis with inverse transform operator \mathbf{B} , e.g. a wavelet basis in the case of natural images. Each image is compressively sensed by multiplying with a measurement matrix \mathbf{M} :

$$\mathbf{y} = \mathbf{M}\mathbf{x} = \mathbf{M}\mathbf{B}\boldsymbol{\theta} \quad (1)$$

where $\boldsymbol{\theta}$ is a k -sparse vector in the transformed (e.g. wavelet) domain.

In a typical compressive sensing camera, the measurement step is accomplished by reflecting light off a digital micromirror device (DMD) on which a series of pseudorandom masks are displayed [2]. (1) is an underdetermined system; that is, \mathbf{M} is a fat matrix and the compressively sensed vector \mathbf{y} has fewer elements than the original image \mathbf{x} . However, it is well-known that $\boldsymbol{\theta}$ can be recovered with high probability from \mathbf{y} by solving the convex optimization problem

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \|\mathbf{M}\mathbf{B}\boldsymbol{\theta} - \mathbf{y}\|_2 + \tau \|\boldsymbol{\theta}\|_1. \quad (2)$$

B. Previous Work

Several solvers exist for equations of the form in (2), including GPSR [3], and SPG-11 [4], [5]. Other solvers [6], [7] work to approximately solve a non-convex ℓ_0 minimization form of the problem.

In [8]–[11] video reconstruction is improved by incorporating temporal information between frames in the reconstruction problem. [8] and [9] give methods for incorporating optical flow into video reconstruction. In [11], a low rank background

with small moving targets is assumed, allowing a dramatic improvement in reconstruction quality. The background subtraction approach of [10], in which quasi-static background with small targets is assumed, is implicitly used in this work.

Although extensive work has been performed in compressive sensing reconstruction, very little work exists in directly processing compressively sensed data. Davenport [1], [12] describes a method of performing classification directly in the compressed domain and coins the term “smashed filter” to describe it. This method is used here to perform the likelihood estimation step of a particle filter. In addition, we show in Section II-B that the smashed filter output, when used to estimate cross-correlations, can be calculated efficiently using the FFT.

The track-before-detect paradigm used herein is also extensively researched. Particle filter implementations have been published [13], [14]; the work of [13] is adapted here for use in the direct CS tracker.

II. TRACKING ALGORITHM

This section describes the algorithm used to perform vehicle tracking in the proof-of-concept implementation. A particle filter with a simple constant motion model was used, with likelihood estimate given by a simple MACH filter implemented directly in the compressive domain using the smashed filter paradigm of [1]. Because the smashed filter suffers from dramatically reduced SNR relative to its non-compressive equivalent, a track-before-detect approach was used to improve tracker convergence. This was implemented through a particle filter using the approach described in [13].

A. Track-Before-Detect Using Particle Filter

The tracker was developed using the work in [13], which gives an example of a particle filter for implementation of track-before-detect. The state vector for the particles was chosen to be position and velocity in the image plane, with the addition of a binary “alive” state for track-before-detect:

$$\mathbf{x}[n] = [p_x[n] \ p_y[n] \ v_x[n] \ v_y[n] \ a[n]]^T. \quad (3)$$

$a[n]$, the “alive” state variable, takes on a value of 1 if the target is estimated to be present by that particle, and 0 otherwise.

Particle state updates of position are given by a constant motion model with added Gaussian noise. The “alive” state of the filter is updated with a $\lambda = 0.05$ probability of switching states between 0 (dead) to 1 (alive).

As with all particle filters, the implementation discussed above requires a likelihood estimate for the importance sampling step of the filter. Dead ($a[n] = 0$) particles may be assigned a constant likelihood based on the probability of a target’s presence, while alive ($a[n] = 1$) particles require some estimate of target probability. This step is particularly important and difficult for the case of a compressive sensing tracker, since only the compressed vector \mathbf{y} is available as input to the likelihood function. The compressive matched filter, discussed below, was used for this purpose.

B. The Fast Smashed Filter

The smashed filter [1] is a method of performing distance-based classification directly in the compressed domain, without performing ℓ_1 reconstruction. A small modification allows the smashed filter to be employed as an estimator of cross-correlation, analogous to the matched filter [12]. This special case of the smashed filter is able to be quickly computed using the FFT, just as with conventional cross-correlation.

The compressive matched filter adapts the well-known matched filter to the case where only a compressively sensed vector \mathbf{y} is available for classification. A template $\mathbf{h}_{u,v}$ corresponding to a target at location (u, v) is measured with sensing matrix \mathbf{M} , generating output \mathbf{y} . In this case, the log likelihood of receiving \mathbf{y} given the transmission of $\mathbf{h}_{u,v}$ in Gaussian noise is a function of the test statistic [12]

$$t = \mathbf{y}^T (\mathbf{M}\mathbf{M}^T)^{-1} \mathbf{M}\mathbf{h}_{u,v} = \mathbf{y}^T \mathbf{M}^\dagger \mathbf{h}_{u,v}. \quad (4)$$

This is typically interpreted as projecting the shifted template $\mathbf{h}_{u,v}$ into the compressive domain by multiplying with \mathbf{M}^\dagger . However, by defining $\hat{\mathbf{x}} = \mathbf{M}^\dagger \mathbf{y}$, the minimum norm solution of the underdetermined problem $\mathbf{y} = \mathbf{M}\mathbf{x}$, (4) may be stated as a simple inner product:

$$t = \hat{\mathbf{x}}^T \mathbf{h}_{u,v}. \quad (5)$$

Since each $\mathbf{h}_{u,v}$ is a shifted version of the target template, this is a cross-correlation. Like all cross-correlations, this value may be efficiently computed using the FFT.

III. ALGORITHM PERFORMANCE SIMULATIONS

The tracking algorithm described in section II was tested on the PETS2000 dataset [15], which contains several sequences of cars moving against a static background. This is a relatively easy dataset by the standards of modern computer vision. However, our purpose is only to show the feasibility of target tracking without frame reconstruction, so this is an ideal starting point. This section describes the simulations performed to evaluate the algorithm and summarizes the results of testing.

A. Simulation Set-up

The test used resized and cropped frames from the PETS2000 dataset. Frames 2700-2910 were chosen; in this series, a parked car backs out of its parking space, stops briefly, and reverses direction, driving forward out of the frame. Frames were processed by first resizing the image with a scale factor of 0.5, then cropping the result to a size of 128×128 pixels. Fig. 1 shows an example of a frame from this sequence. A compressive sensing camera was simulated by multiplication of an image vector with a measurement matrix \mathbf{M} consisting of pseudorandom i.i.d. Gaussian elements. A template was then generated for this target based on image-domain difference frames taken from the testing dataset. Figure 3 shows the image-domain representation of the generated filter.



Fig. 1. Example frame from testing video sequence.



Fig. 2. Example difference frame from testing video sequence.

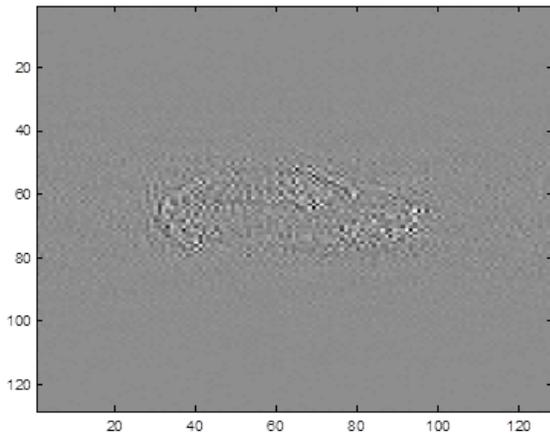


Fig. 3. image domain template for cross-correlation.

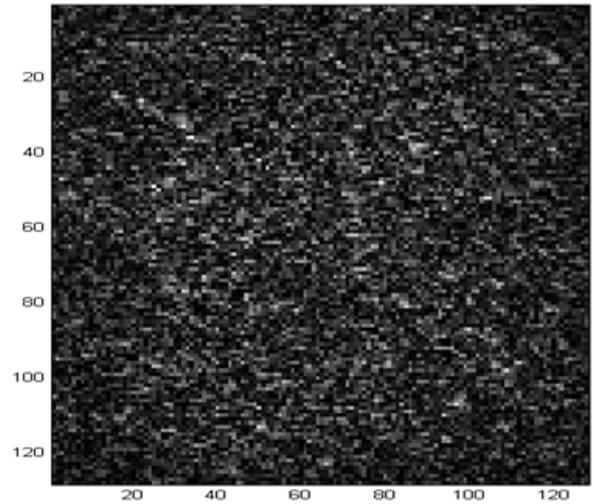


Fig. 4. Correlation estimate using smashed filter. Note that no correlation peak is clearly visible, leading to the use of the track-before-detect particle filter.

1) *Template Creation*: Because the goal of this research was to determine the feasibility of tracking directly in the CS domain, restrictions on the use training vs. testing data were relaxed. The target template was generated using examples from throughout the video sequence of interest, rather than restricting training data to the beginning of the sequence. This enabled a highly specific template for cross-correlation to be quickly developed. By eliminating problems of template generation, overfitting, etc. in this way, the effect of compressive sensing on detector performance may be studied more or less in isolation.

The template was generated using difference frames from the test video sequence. In the case of a stationary surveillance camera, this quickly eliminates any motionless background objects from the video. Unfortunately, a target which is still or temporarily stopped will also be undetectable. This is a problem which must be overcome if a system is to be commercialized.

The template was generated using a maximum average correlation height (MACH) filter. Positive (target) examples were selected from the test video sequence difference frames as described above, while negative (non-target) examples for filter training were selected from segments of the source video where no targets were present. Figure 3 shows the generated template; it can be seen that the difference image based template acts similarly to an edge detector.

B. Results

The particle filter successfully converged on the target in the test sequence at a sensing rate of 0.3 (4915 sensors). Figure 5 summarizes the results of the Monte Carlo trials at each sensing rate. Figure 4 shows the output of the smashed filter for one frame of the test sequence. No clear peak is visible in

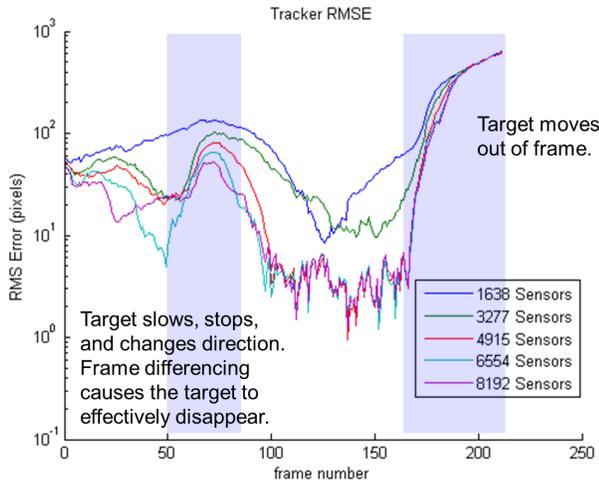


Fig. 5. Simulation Results.

the correlation estimate, but the tracker is able to converge successfully despite this by implementing the track-before-detect concept.

IV. CONCLUSION

The automated tracker described above is adequate as a proof of concept showing the feasibility of performing tracking without reconstruction of compressively sensed images. As expected, performance is slightly reduced relative to tracking in the image domain. However, tracker convergence was achieved at a sensing rate of 0.3. This is comparable to sensing rates used in other work on compressive sensing image and video.

Several challenges must be overcome before direct tracking becomes commercially feasible. Most importantly, more sophisticated likelihood estimates will be necessary. The authors intend to explore the possibility of adapting common feature extraction methods such as the histogram of oriented gradients (HoG) [16] or the Haar-like features of Viola & Jones [17].

ACKNOWLEDGMENT

This work was performed under the SenSIP site of the NSF NetCentric I/UCRC with support from Raytheon Missile Systems.

REFERENCES

- [1] M. A. Davenport, M. F. Duarte, M. B. Wakin, J. N. Laska, D. Takhar, K. F. Kelly, and R. G. Baraniuk, "The smashed filter for compressive classification and target recognition," *Proc. SPIE*, vol. 6498, pp. 64980H–64980H–12, 2007. [Online]. Available: + <http://dx.doi.org/10.1117/12.714460>
- [2] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, "A new compressive imaging camera architecture using optical-domain compression," in *Electronic Imaging 2006*. International Society for Optics and Photonics, 2006, pp. 606 509–606 509.
- [3] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 586–597, 2007.

- [4] E. van den Berg and M. P. Friedlander, "SPGL1: A solver for large-scale sparse reconstruction," June 2007, <http://www.cs.ubc.ca/labs/scl/spgl1>.
- [5] —, "Probing the pareto frontier for basis pursuit solutions," *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008. [Online]. Available: <http://link.aip.org/link/?SCE/31/890>
- [6] Y. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993, pp. 40–44.
- [7] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.
- [8] H. Braun, P. Turaga, C. Tepedelenioglu, and A. Spanias, "Optical flow for compressive sensing video reconstruction," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013, pp. 2267–2271.
- [9] A. Sankaranarayanan, C. Studer, and R. Baraniuk, "CS-MUVI: Video compressive sensing for spatial-multiplexing cameras," in *Computational Photography (ICCP), 2012 IEEE International Conference on*, april 2012, pp. 1–10.
- [10] V. Cevher, A. Sankaranarayanan, M. Duarte, D. Reddy, R. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," *Computer Vision—ECCV 2008*, pp. 155–168, 2008.
- [11] A. Waters, A. Sankaranarayanan, and R. Baraniuk, "SpaRCS: Recovering low-rank and sparse matrices from compressive measurements," *Proc. Adv. in Neural Processing Systems (NIPS), Granada, Spain*, 2011.
- [12] M. A. Davenport, M. B. Wakin, and R. G. Baraniuk, "The compressive matched filter," Rice University, Tech. Rep., 2006.
- [13] D. Salmond and H. Birch, "A particle filter for track-before-detect," in *Proceedings of the American Control Conference*, Jun 2001.
- [14] L. Fan and X. Zhang, "A gaussian particle filter for track-before-detect," in *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, vol. 2, july 2010, pp. V2–309–V2–312.
- [15] (2007) Pets: Performance evaluation of tracking and surveillance. Vision@Reading. [Online]. Available: <http://www.cvg.rdg.ac.uk/slides/pets.html>
- [16] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *International Conference on Computer Vision & Pattern Recognition*, C. Schmid, S. Soatto, and C. Tomasi, Eds., vol. 2, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005, pp. 886–893. [Online]. Available: <http://lear.inrialpes.fr/pubs/2005/DT05>
- [17] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Conference on Computer Vision and Pattern Recognition*, 2001.