

Static compressive tracking

D. J. Townsend,¹ P. K. Poon,² S. Wehrwein,¹ T. Osman,³ A. V. Mariano,¹ E. M. Vera,³ M. D. Stenner,^{1,*} and M. E. Gehm^{2,3}

¹MITRE Corp., 7525 Colshire Dr., McLean, VA 22102, USA

²College of Optical Sciences, University of Arizona, 1630 E University Blvd., Tucson, AZ 85721, USA

³Department of Electrical and Computer Engineering, University of Arizona, 1230 E Speedway Blvd., Tucson, AZ 85721, USA

[*mstenner@mitre.org](mailto:mstenner@mitre.org)

Abstract: This paper presents the Static Computational Optical Undersampled Tracker (SCOUT), an architecture for compressive motion tracking systems. The architecture uses compressive sensing techniques to track moving targets at significantly higher resolution than the detector array, allowing for low cost, low weight design and a significant reduction in data storage and bandwidth requirements. Using two amplitude masks and a standard focal plane array, the system captures many projections simultaneously, avoiding the need for time-sequential measurements of a single scene. Scenes with few moving targets on static backgrounds have frame differences that can be reconstructed using sparse signal reconstruction techniques in order to track moving targets. Simulations demonstrate theoretical performance and help to inform the choice of design parameters. We use the coherence parameter of the system matrix as an efficient predictor of reconstruction error to avoid performing computationally intensive reconstructions over the entire design space. An experimental SCOUT system demonstrates excellent reconstruction performance with 16X compression tracking movers on scenes with zero and nonzero backgrounds.

© 2012 Optical Society of America

OCIS codes: (110.1758) Computational imaging; (100.4999) Pattern recognition, target tracking; (100.3190) Inverse problems.

References and links

1. M. Wakin, J. Laska, M. Duarte, D. Baron, S. Sarvotham, D. Takhar, K. Kelly, and R. Baraniuk, "An architecture for compressive imaging," in *Proceedings of IEEE Intl. Conference on Image Processing*, (IEEE, 2006), pp. 1273–1276.
2. M. Lustig, D. Donoho, and J. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.* **58**, 1182–1195 (2007).
3. R. Willett, R. Marcia, and J. Nichols, "Compressed sensing for practical optical imaging systems: a tutorial," *Opt. Eng.* **50**, 072601 (2011).
4. M. Neifeld and J. Ke, "Optical architectures for compressive imaging," *Appl. Opt.* **46**, 5293, (2007).
5. M. Stenner, D. Townsend, and M. Gehm, "Static architecture for compressive motion detection in persistent, pervasive surveillance applications," in *Imaging Systems*, OSA Technical Digest Series (Optical Society of America, 2010), paper IMB2.
6. Y. Rivenson, A. Stern, and B. Javidi, "Single exposure super-resolution compressive imaging by double phase encoding," *Opt. Express* **18**, 15094–15103 (2010).
7. Y. Kashter, O. Levi, and A. Stern, "Optical compressive change and motion detection," *Appl. Opt.* **51**, 2491–2496 (2012).

8. W. Bajwa, J. Haupt, G. Raz, S. Wright, and R. Nowak, "Toeplitz-structured compressed sensing matrices," in *Proceedings of IEEE Workshop on Statistical Signal Processing*, (IEEE, 2007), pp. 294–298.
9. H. Rauhut, "Circulant and Toeplitz matrices in compressed sensing," <http://arxiv.org/abs/0902.4394>.
10. J. Romberg, "Compressive sensing by random convolution," *SIAM J. Imaging Sci.* **2**, 1098–1128 (2009).
11. F. Seibert, Y. Zou, and L. Ying, "Toeplitz block matrices in compressed sensing and their applications in imaging," in *Proceedings of IEEE International Conference on Information Technology and Applications in Biomedicine*, (IEEE, 2008), pp. 47–50.
12. B. Liu, F. Seibert, Y. Zou, and L. Ying, "SparseSENSE: randomly-sampled parallel imaging using compressed sensing," in *Proceedings of the 16th Annual Meeting of ISMRM* **3154** (2008).
13. S. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE J. Sel. Top. Sig. Proc.* **1**, 606–617 (2007).
14. J. Tropp, "Just relax: Convex programming methods for identifying sparse signals in noise," *IEEE Trans. Inf. Theory* **52**, 1030–1051 (2006).

1. Introduction

Compressive sensing (CS) is a potentially powerful new approach to imaging and sensing. Some of its demonstrated benefits include the ability to form images when only a single detector element is available [1], accelerating magnetic resonance imaging (MRI) by reducing the number of required measurements [2], and in general use less data by sampling in a compressed basis. One application where reduced data volume is particularly important is large-area persistent surveillance. Systems designed for this purpose traditionally acquire and store large volumes of data by imaging large areas at a high resolution and framerate. However, one of the most important capabilities of these devices is tracking moving targets, which requires comparatively little information. These sensors are often on airborne platforms so the data must either be processed on the platform at high cost in size, weight, and power (SWAP), or the data must be stored until the platform lands. We present an optical tracking architecture based on compressive sensing techniques, which we call the Static Computational Optical Undersampled Tracker (SCOUT), to address these challenges.

Much of the existing CS literature focuses on sampling strategies that are appealing either for their mathematical tractability, flexibility, or optimality for certain classes of scenes. Unfortunately, such sampling strategies are often difficult or impractical to implement. As a result, experimental demonstrations of CS are often very general, but exhibit substantial practical challenges, as described in Sec. 2. For example, the single-pixel camera uses a micro-mirror array to measure in any arbitrary basis, but must do so over many time-sequential measurements [1]. By contrast, SCOUT was developed under the constraint that measurements must be acquired "single-shot" using a conventional focal plane array.

The SCOUT architecture, described in detail in Sec. 3, uses a defocused imaging system with two binary amplitude masks. The focal plane array then samples at a much lower resolution than the final reconstruction. Whereas previous compressive imaging systems have attempted to reconstruct entire images, the current version of SCOUT only reconstructs frame-to-frame differences. As a result, the system can acquire optically compressed tracking data and store or transmit that data with no processing located at the sensor and minimal bandwidth requirements. The data can then be reconstructed offline where computation is comparatively cheap. Section 4 describes simulations used to predict reconstruction performance and optimize the design of such a system. Section 5 presents experimental results from a laboratory system that demonstrates the feasibility of this approach.

We believe that the SCOUT system represents an important step toward practical optical CS system design. The system enables parallel "single-shot" acquisition of compressed, task-oriented data, using a fully-static architecture.

2. Challenges in traditional optical CS architectures

We now discuss why the canonical CS imaging architecture, depicted in Fig. 1, is impractical for motion tracking applications. A well known example of this architecture is the single pixel camera [1], which employs a micromirror array to form the projections. Each measurement is an integrated point-by-point multiplication of the scene locations with the micromirror array values. A single programmable micromirror array allows the use of an arbitrary projection basis but only allows one projection to be recorded at a single point in time.

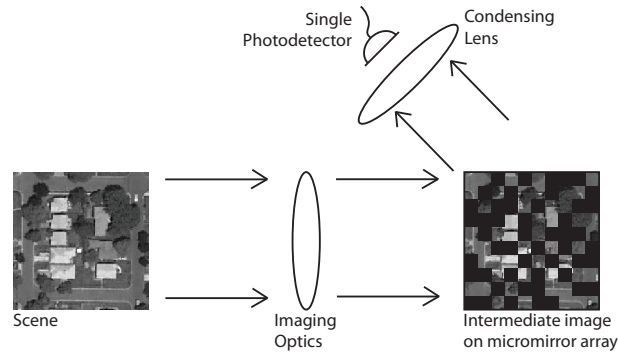


Fig. 1. A canonical CS imaging architecture. An object is imaged onto micromirror pattern, which represents a projection vector, then condensed onto a photodetector for a single measurement. Each set of projections, created by different mirror configurations, is captured sequentially in time.

For static scenes this architecture allows an arbitrarily long exposure time (within the limit of detector saturation) to increase the signal to noise for each measurement. However, such a camera tends to be less practical with dynamic scenes: it needs to record all the projections for each frame before the object moves. Increasing the rate at which projections are made is possible, but this reduces exposure time and signal to noise ratio [3].

An alternative is the parallel architecture shown in Fig. 2. This imager uses multiple spatial light modulators and detectors, but requires additional optics to split the light from the object [4]. The complexity of the optical design scales with the number of simultaneous measurements—capturing all M of the projections simultaneously requires using M spatial light modulators and M detector elements.

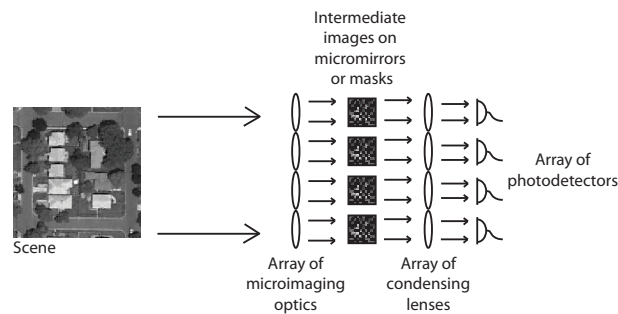


Fig. 2. An example of a typical parallel optical CS architecture. Capturing M simultaneous projections requires using M spatial light modulators or masks and M detector elements.

3. The SCOUT architecture

This section motivates and introduces the SCOUT architecture, then formally defines it and describes its design parameters. In contrast to the parallel architecture in Fig. 2, SCOUT uses a volumetric optical component to form all of the required projections simultaneously on a single traditional sensor array. Early examples of this architecture can be found in [5, 6]. A related approach that uses a small number of Radon projections onto linear arrays was recently published [7]. The SCOUT approach allows for capture of dynamic scenes while avoiding the scaling issues of earlier parallel CS architectures. The cost of this parallelism is in the loss of flexibility to implement arbitrary projections. Rather than fully designing the projections themselves, we describe a process for optimizing a parameterized system and analyzing its performance.

The performance of any compressive sensing architecture depends on the design of the system matrix, denoted as $\mathbf{H}$. The typical mathematical formulation for a compressive sensing system is shown in Eq. (1) and represented graphically in Fig. 3. In order to adhere to this model in an imaging context, the 2D scene and measurement arrays are lexicographically reordered into the column vectors $\mathbf{f}$ and $\mathbf{g}$, respectively. Thus, any $M \times 1$ compressed measurement $\mathbf{g}$ is obtained as:

$$\mathbf{g} = \mathbf{H}\mathbf{f}, \quad (1)$$

where $\mathbf{f}$ represents the $N \times 1$ scene, and $\mathbf{H}$ is the $M \times N$ system matrix, being $M < N$ in order to the system to be considered compressive. The i th column of $\mathbf{H}$ describes the PSF of the i th image element, whereas the j th row of $\mathbf{H}$ describes the weights of each scene location's contribution to the j th measurement. In this way, the resulting system matrix $\mathbf{H}$ typically acts as a space-variant optical system and presents a *block* structure, as seen in the example shown in Fig. 4.

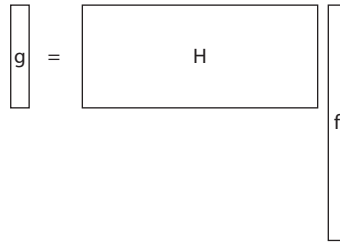


Fig. 3. A matrix formulation describing the SCOUT system.

An effective compressive sensing system must take measurements that are both compressive and multiplexed: the number of measurements must be fewer than the number of scene locations, and each measurement must contain information about many scene locations. In other words, the system matrix must exhibit a many-to-few mapping from scene locations to sensor elements. Because we are not considering temporal multiplexing, the signal must be spatially multiplexed; this is done using structured blur. The blur is achieved by defocusing the lens so that the PSF is broad, spanning many pixels, leading to a many-to-many mapping from scene locations to detector pixels. Because the pixels are large relative to the ultimate reconstruction resolution, this process is compressive. Blur alone suppresses high spatial frequencies and is poorly conditioned for reconstruction. Therefore, high-frequency structure is imposed atop the blur by coding the optical path with two pseudo-random binary occlusion masks that are placed at different positions between the lens and the sensor. The separation between masks results in strong space-variance. Collectively, these design elements result in a broad shift-variant PSF, which is then undersampled.

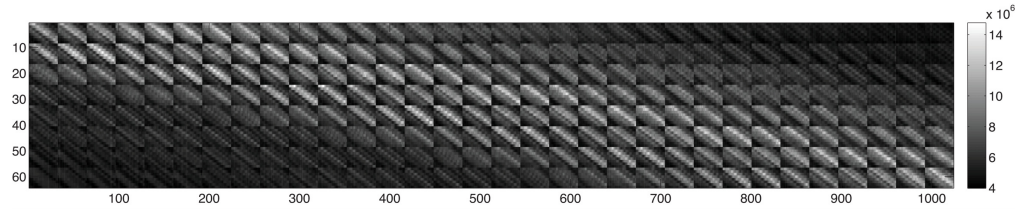


Fig. 4. Experimentally measured system response matrix for the SCOUT system described in Sec. 5. The approximate block-Toeplitz structure is clearly evident, as is the deviation from the Bernoulli or Gaussian ensembles typically considered in CS treatments.

Finally, a reconstruction algorithm chooses the most sparse solution to the under-determined system. The reconstruction basis must be sparse for this choice to be correct. Frame differences are a naturally sparse basis for video surveillance applications, where relatively few targets of interest move on an otherwise-static background. The architecture is modeled as an entirely linear system, so frame differences can be calculated in the measurement basis and subsequently reconstructed in the spatial domain to show dots indicating where an object moved to and from.

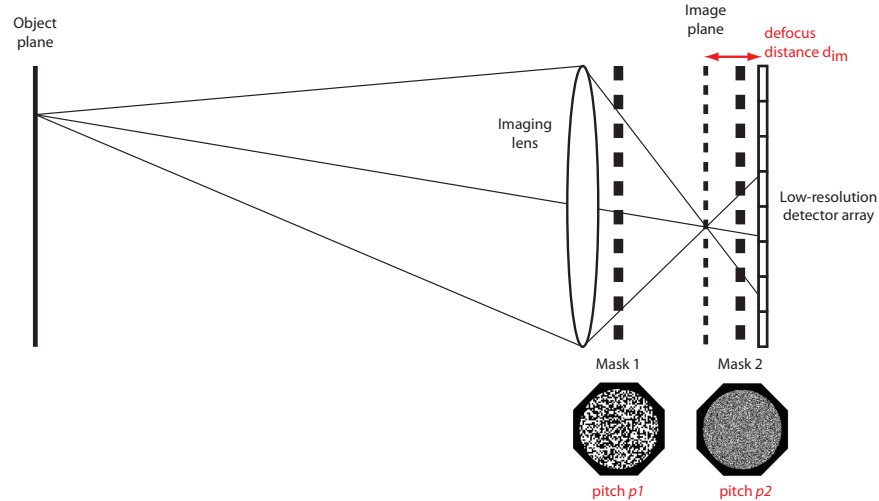


Fig. 5. A diagram of the SCOUT architecture. A defocused lens projects light through a pair of binary occlusion masks onto a low-resolution sensor to capture compressive, multiplexed measurements with a shift-variant PSF.

The shift variant PSF leads to an approximate block-Toeplitz structure for the system matrix, with approximate Toeplitz structure within individual blocks due to the shifting PSF. This circulant structure is modified by random variations corresponding to the differing projections created by the two masks. There has been some work in the CS community to investigate system matrices with Toeplitz and circulant structure [8–10], but there has been relatively little investigation of the approximately block-Toeplitz structure that naturally arises in optical systems such as SCOUT. Two notable exceptions are [11, 12], which provide both theoretical and practical evidence for the viability of CS system matrices with block-Toeplitz structure.

A diagram showing the complete SCOUT architecture is shown in Fig. 5. A lens of focal length f is focused at infinity and placed some distance $f + d_{im}$ from the sensor. Two binary

occlusion masks, m_1 and m_2 , are placed at distances d_{m_1} and d_{m_2} from the sensor. Masks m_1 and m_2 achieve fill factors f_1 and f_2 using randomly-patterned occluders of pitch p_1 and p_2 , respectively. The sensor captures images at resolution $r_x \times r_y$; adjacent frames are subtracted, and frame differences are reconstructed at some higher resolution $R_x \times R_y$ using a sparsity-favoring reconstruction algorithm. Such algorithms require knowledge of the $r_x r_y \times R_x R_y$ system matrix $\mathbf{H}$, which is determined experimentally.

While the SCOUT architecture is well-suited for tracking applications, it does have limitations which make it less useful for general imaging applications. Without sparse scene motion, the priors used in reconstruction will lead to incorrect results. Reconstructions only show the locations of moving objects, and the sensing platform must be stationary relative to the scene so that frame differences are sparse. However, more sophisticated techniques could potentially estimate platform motion and use the additional information to reconstruct the entire scene. Despite its limitations, the architecture is well-suited for applications such as fixed-camera wide-area surveillance where bandwidth and data volume are key concerns.

4. Simulation performance

A simulation of the SCOUT architecture was used to evaluate theoretical performance and find values for design parameters, such as defocus distance or mask fill factor, in order to minimize reconstruction errors. In this section we discuss the optical model and the reconstruction technique used, define our tracking-specific reconstruction error metric, then explain how the simulation informed our choice of design parameters.

4.1. Simulating a SCOUT System

The simulation uses a paraxial ray-based approach to model the light from a scene as it travels through a lens and two masks onto the detector plane. The scene has a native resolution of $R_x \times R_y$; the lens is modeled as a single thin lens with transmittance function $\mathbf{t}_f$ and the two masks have transmittance functions $\mathbf{t}_1$ and $\mathbf{t}_2$. The mask transmittances are 0 where the mask is black and 0.88 where the mask is clear based on measurements. The lateral magnification of the scene and the two masks is calculated using similar triangles.

Using the optical model, the simulation records the $r_x \times r_y$ PSF from each scene location in order to obtain the system matrix $\mathbf{H}$. Once $\mathbf{H}$ is known, we use it to simulate the low resolution measurements, $\mathbf{g}$, of the higher resolution scenes, $\mathbf{f}$. Subsequent simulated measurements are subtracted to find $\Delta\mathbf{g}$.

Given $\Delta\mathbf{g}$ and $\mathbf{H}$, the reconstruction algorithm finds an estimate, denoted $\hat{\Delta\mathbf{f}}$, of the original scene's sparse difference frames. In our simulations, we use the ℓ_1 -norm minimization software package [13], which has been shown to be an effective method for recovering sparse signals from undersampled measurements [14].

Several assumptions are made in the simulation. The distance from the scene to the lens is much larger than the focal length of the lens, allowing us to approximate the image plane as the rear focal plane of the lens. We also assume that we are using a single thin lens. For now, we treat the system as noise-free for simplicity, but readily acknowledge the important real-world impact of noise in CS applications.

4.2. Quantifying reconstruction error

The mean squared error (MSE) is a typical choice for error metrics, however it is not suitable for our application because it weights all errors equally. For motion tracking purposes, we classify errors into three categories. A false positive occurs when the reconstruction shows an object where there is none; a false negative occurs when the reconstruction fails to show an object where one exists. When an object is being tracked but appears in a neighboring pixel from its

true location, we call this a shift error. False negatives and false positives indicate a serious failure in the motion tracking task, while shift errors may be less problematic.

We created a custom metric that heavily penalizes false positives and false negatives, while assigning a lower penalty for shift errors. We define reconstruction error as

$$P = \frac{|\mathbf{a} \otimes \mathbf{e}|}{2n} \quad \text{where} \quad \mathbf{a} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad \text{and} \quad \mathbf{e} = \Delta \mathbf{f} - \hat{\Delta \mathbf{f}} \quad (2)$$

The error frame $\mathbf{e}$ is the difference between the true and reconstructed difference frames, and n is the number of movers in the scene. The error frame is convolved with a three pixel averaging kernel $\mathbf{a}$, which reduces the penalty for one-pixel shift errors. The absolute value is taken in order to count positive and negative errors equally, and the error is divided by $2n$ to make the metric independent of the number of movers. From now on, we will refer to P from Eq. (2) as the reconstruction error.

4.3. Identifying optimal design parameters

We can use the previously described simulation and error metric to optimize the design for performance. Both simulation and experimental study demonstrates a relationship between mask position and pitch: performance depends most sensitively on the projected mask pitch on the sensor. Furthermore, improved performance occurs when the masks are well-separated and generate a highly space-variant PSF. With these observations in mind, we focus our study on the mask pitches (p_1, p_2) as well as the defocus distance (d_{im}). Even after constraining the parameter space, an exhaustive search is not feasible because reconstruction is computationally intensive. Therefore, we also use a simpler metric calculated directly from the simulated system matrix $\mathbf{H}$ that is correlated with a lower reconstruction error. This metric is based in the coherence parameter [14], well-known in the compressive sensing community. In this case, our coherence parameter μ is defined as the maximum absolute value of inner products between unique columns of the measurement matrix:

$$\mu = \max |\langle \phi_i, \phi_j \rangle|; \quad i \neq j \quad (3)$$

where ϕ_i and ϕ_j are unique columns of $\mathbf{H}$. The columns are unnormalized because their relative magnitude is related to the physical light throughput on the given sensor pixel. This coherence parameter provides a measure of maximum similarity between any two columns of the system matrix. Ideally, the columns, which represent the system response to each location, are orthogonal, but that is not possible in an undersampled system. We chose coherence as a predictor of system matrix performance because it describes the extent to which the system matrix deviates from this ideal.

Notice that although system matrices with nearly pairwise orthogonal columns will result in small coherence values, system matrices with numerically small entries can accomplish the same. Optimizing $\mathbf{H}$ for minimum coherence would encourage small PSF magnitudes and drive total system throughput down. To eliminate this effect we normalize $\mathbf{H}$ by the sum of the basis vector magnitudes:

$$\mathbf{H}_{normed} = \frac{\mathbf{H}}{\sum_{j=1}^M \sum_{i=1}^N h_{i,j}} \quad (4)$$

where M and N are the total number of rows and columns in the system matrix. Physically, this normalization represents division by the sum of each PSF's light throughput. The coherence of a system matrix normalized in this way cannot be biased by reducing throughput. An unfortunate consequence of this normalization step is that mask fill factor—one of the SCOUT

architecture's parameters—cannot be optimized based on the coherence parameter because the effect of different mask fill factors are normalized out during the calculation.

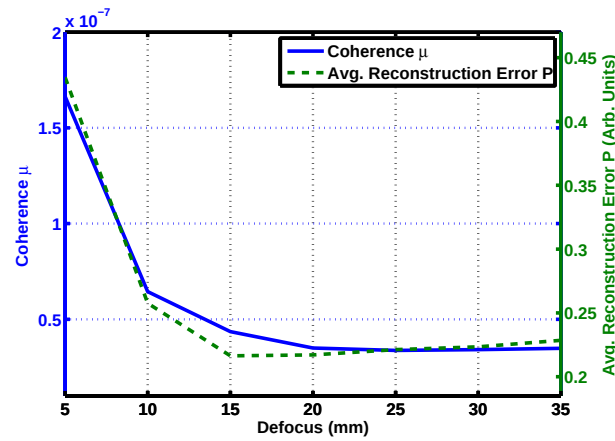


Fig. 6. The coherence μ (left vertical axis - solid blue) and reconstruction error P (right vertical axis - dashed green) plotted as a function of defocus distance d_{im} .

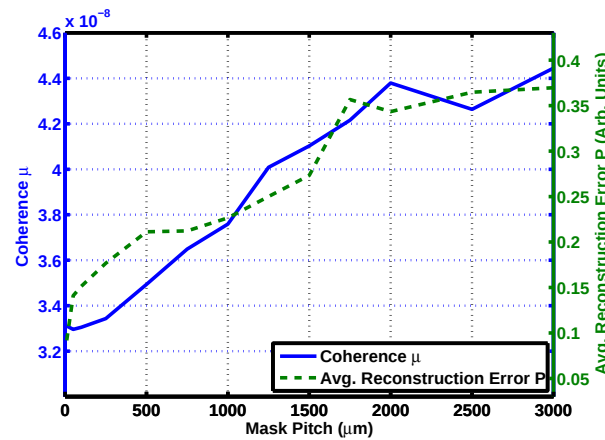


Fig. 7. The coherence μ (left vertical axis - solid blue) and the reconstruction error P (right vertical axis - dashed green) is plotted as a function of the pitch of mask 2.

To demonstrate the effectiveness of the coherence parameter as a predictor of reconstruction error trends, we ran several simulations with complete reconstructions in order to compare reconstruction error to coherence. Figure 6 shows that reconstruction error and the coherence parameter follow similar trends for different defocus distances when all other variables are held constant. Figure 7 shows similar results for varying values of mask pitch p_2 .

The compressive sensing coherence parameter is also a good predictor of reconstruction error when multiple parameters are optimized simultaneously, see Fig. 8. In this case, we consider d_{im} , p_1 , and p_2 . We compare a jointly optimized set of parameters based on the coherence parameter to a set of parameters found by varying each parameter individually. Reconstructions were performed in order to compare reconstruction error for scenes with one through six

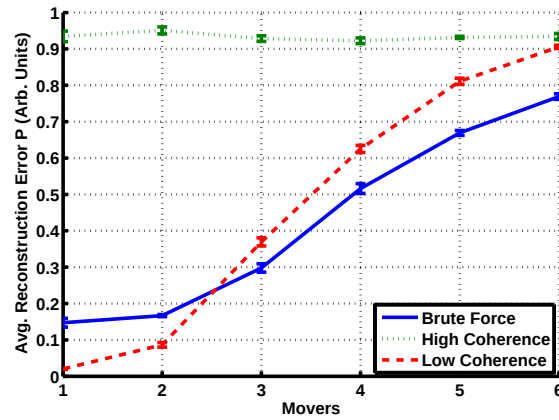


Fig. 8. The reconstruction error, P , for three different choices of d_{im} , p_1 , and p_2 are plotted versus number of movers. The performance obtained from a set of parameters that led to a minimum μ in our parameter space is shown in dashed red. The solid blue line shows performance of parameters chosen individually using actual reconstruction performance. A set of parameters that leads to a maximum μ is shown in dotted green. The error bars on each line plot represent the standard deviation of the mean.

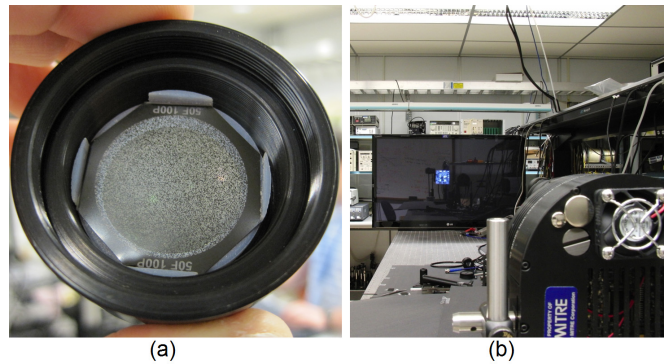


Fig. 9. Photographs of a SCOUT implementation. (a) An optical tube contains mask m_2 and the lens; mask m_1 (not shown) is attached near the sensor. (b) The camera captures images of scenes displayed on a plasma television approximately 2 meters away.

movers. Note also that reconstruction error is much higher for a set of parameters that result in high coherence.

Our simulations demonstrate the viability of the architecture and provide an efficient way to optimize most architecture parameter values using the simulated system matrix coherence. Mask throughput cannot currently be optimized because coherence is normalized by full-system throughput. The problem of finding optimal mask throughput warrants further investigation.

5. Experimental results

This section presents an experimental demonstration of the SCOUT. We used a camera with modified optics to capture scenes with sparse frame differences displayed on a plasma monitor. Figure 9 shows photographs of the experimental setup. The camera's optical path contains a 35mm lens and two random amplitude binary masks. Each mask was printed on transparencies using a high resolution laser printer. Mask m_1 has pitch $p_1 = 30$ microns and fill factor $f_1 = 0.4$ and is located at a distance $d_{m_1} = 14$ mm from the sensor. Mask m_2 has pitch $p_2 = 500$ microns and fill factor $f_2 = 0.2$ and is located at a distance $d_{m_2} = 57$ mm from the sensor. These parameters were chosen based on simulated and experimental results. The plasma monitor was used because it provides a higher contrast compared to the traditional liquid crystal displays (LCD). Note that even though the plasma monitor was chosen for its high contrast compared to LCDs, the black background still produces a small amount of irradiance, which is a source of noise in our experiment.

For our experiment, the captured image resolution $r_x \times r_y$ is 8×8 , while the ground-truth and reconstructed frame differences have a resolution $R_x \times R_y$ of 32×32 . To simulate a low-resolution detector, the camera captures the scenes at 128×128 sensor pixels and the images are binned down to 8×8 before being used in reconstruction. The system response matrix is determined experimentally by measuring the point spread function of each scene location one at a time, as discussed in detail in Sec. 3.

A video showing experimental results for the reconstruction of scenes that contains two dots (movers) changing position on a black background is presented in [Media 1](#). Frame 1 of this video is shown in Fig. 10. The top row shows two consecutive frames of the scene and the ground-truth difference frame, all at 32×32 resolution. The bottom row shows the difference of corresponding 8×8 measurement frames. Finally, the 32×32 reconstructed difference frame is shown at the bottom right. The reconstructed video clearly shows the two movers in the majority of the scenes. Strong quantitative agreement with the ground-truth is achieved when the system response matrix used in the reconstruction is scaled according to:

$$\mathbf{H}_{\text{recon}} = \frac{t_{\text{exp}}}{t_{\text{cal}}} \mathbf{H}_{\text{cal}}, \quad (5)$$

where t_{exp} and t_{cal} are the experiment and calibration exposure times, respectively. This scaling accounts for the physical effect of increased photon collection (and hence photodetector counts) as a function of increased exposure time. The resulting peaks are easily identified against background noise.

By inspecting the experimental reconstruction of the 9th difference frame (occurring at 8 seconds in [Media 1](#), and shown in Fig. 11), we note that SCOUT reconstructs the amplitude of one or more of the mover locations at a much lower amplitude when compared to the amplitudes in the other reconstructed difference frames. This phenomenon has been correlated to when two locations are adjacent in the ground-truth difference scene. This is an issue that can be traced to the system response matrix $\mathbf{H}$, which is a result of either inaccurate calibrations or through a combination of system design parameters. We are currently working on non-isomorphic cal-

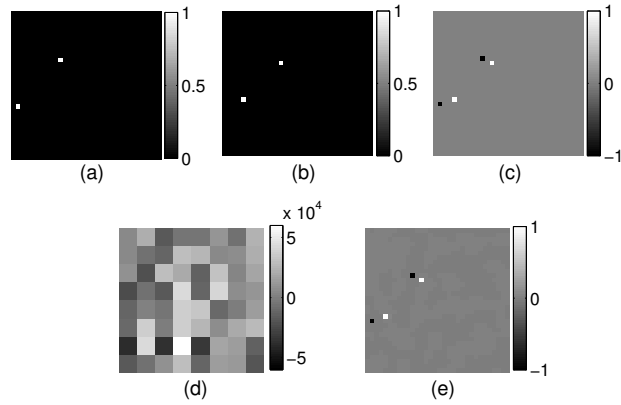


Fig. 10. (Media 1) Frame 1 of a reconstruction video of a 32×32 scene with two movers of equal amplitude on a black background. (a) ground-truth scene 1 (b) ground-truth scene 2 (c) ground-truth frame difference and (d) measured 8×8 frame difference, scaled so that it is discernible (e) reconstructed 32×32 difference frame

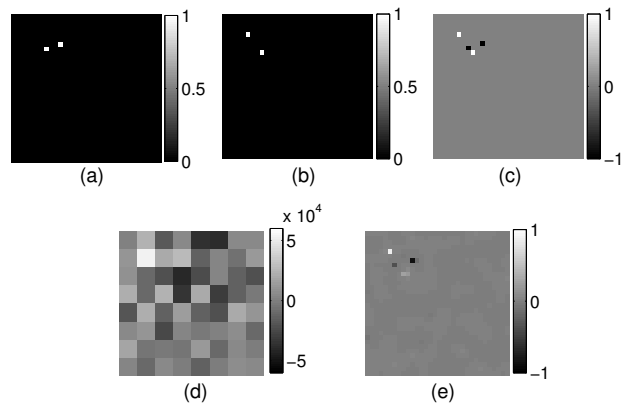


Fig. 11. (Media 1) Frame 9 of a reconstruction video of a 32×32 scene with two movers of equal amplitude on a black background. This frame shows the results when the past and present mover locations are adjacent in the difference frame. (a) ground-truth scene 1 (b) ground-truth scene 2 (c) ground-truth frame difference and (d) measured 8×8 frame difference, scaled so that it is discernible (e) reconstructed 32×32 difference frame

ibration techniques and introducing additional degrees of freedom in our architecture which may overcome this problem.

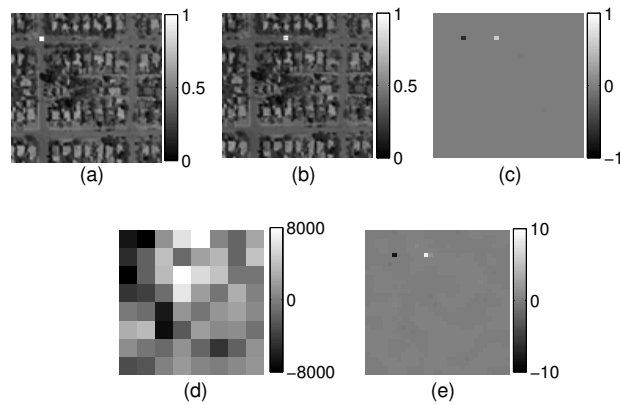


Fig. 12. (Media 2) Frame 1 of a video demonstration of compressive tracking of a 32×32 difference scene with a nonzero background. Scene background ©2012 Google. (a) ground-truth scene 1 (b) ground-truth scene 2 (c) ground-truth frame difference and (d) measured 8×8 frame difference, scaled so that it is discernible (e) reconstructed 32×32 difference frame

A video with the results for a single mover on a nonzero background is shown in Media 2. Elements relating to Frame 1 of this video are shown in Fig. 12. Figure 12(a) and Fig. 12(b) show two sequential ground-truth scenes while Fig. 12(c) shows the ground-truth difference scene. Figure 12(d) shows the experimental measurement difference at 8×8 resolution, and the reconstructed difference frame at 32×32 resolution is displayed in Fig. 12(e).

As seen in Fig. 12(c), with the nonzero background, the amplitude of the past and present mover locations in the ground-truth are lower than the zero background case. Therefore there is also less contrast in the experimental difference measurements in Fig. 12(d), which makes this case more sensitive to noise. However, the most notable feature is the lack of quantitative agreement with the ground-truth, even when the calibration matrix is scaled according to Eq. (5). This can be traced to nonlinearities in the overall system response that result from a nonlinear monitor “gamma” (mapping from pixel value to output brightness) and inter-pixel interactions that effect brightness. These effects are not captured during calibration as that is performed point-by-point (thus avoiding inter-pixel effects) and with pixels that are fully-on or -off (thus avoiding effects from monitor gamma). Quantitative agreement was possible in the zero background case of Figs. 10 and 11 as that also involved only a small number of fully-on or -off locations. The lack of quantitative agreement in the nonzero background experiment therefore does not result from something fundamental in the SCOUT architecture or processing, but rather from specific details of the source used for the experiment. Despite the lack of quantitative agreement, qualitative agreement is excellent and the movers are clearly identifiable against the background in Fig. 12(e).

It is important to note that we are using a generic ℓ_1 -norm minimization algorithm [13] which is not specialized or tuned for the block-circulant system matrix structure exhibited in the SCOUT architecture. The problem of reconstructing sparse signals from system matrices with this particular property is not well understood, so further research could yield significant improvements in reconstruction performance. The observed system performance with this algorithm serves to demonstrate the architecture’s viability without the need for significant optimization in the reconstruction stage.

6. Future work and conclusion

A typical parallel compressive imager would require as many encoding optical elements as simultaneous measurements. The SCOUT architecture eliminates this scaling issue by giving up the ability to implement arbitrary projections. Using a pair of masks at different distances to create a block-circulant system matrix, the system makes compressive measurements and reconstructs frame differences. The system can be optimized by adjusting system parameters such as mask pitch and defocus distance. Simulations demonstrated the use of the compressive sensing coherence parameter as an efficient predictor of system matrix performance to jointly optimize these parameters. An experimental system based on the SCOUT architecture successfully performed compressive motion tracking on scenes with zero and nonzero backgrounds in most instances. However, the reconstruction of difference scenes with adjacent mover locations caused issues due to the design or calibration of the system matrix. The system showed promising results using a general ℓ_1 -norm minimization algorithm and we believe that further research on sparse reconstruction with block-circulant system matrices may decrease reconstruction error. We also believe that non-isomorphic calibration techniques and adding further degrees of freedom in the design parameters could result in significant performance gains.