

# ACTIVE TARGET DETECTION WITH MOBILE AGENTS

*Sunav Choudhary, Naveen Kumar, Srikanth Narayanan and Urbashi Mitra*

Ming Hsieh Department of Electrical Engineering  
University of Southern California, Los Angeles, CA-90089  
{sunavcho, komathnk, ubli}@usc.edu, shri@sipi.usc.edu

## ABSTRACT

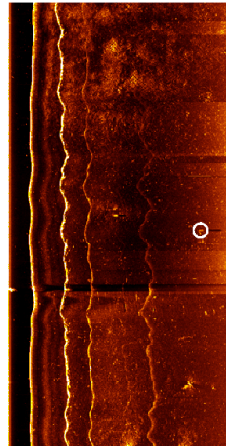
A strategy for active target detection suitable for the use of mobile agents in a field is presented. In particular, there is an interest in autonomous underwater vehicles. By exploiting notions from group testing, the proposed algorithm decides when to collect new samples depending on whether the mobile agent perceives the sensor measurements correspond to noise or a target pattern. Under suitable assumptions about the field emanated by the target, *i.e.* the target signature is locally low rank in the field, one can efficiently sample the field to locate the target using  $O(m \log m \log n)$  samples on an  $n \times n$  grid where  $m \ll n$  is a parameter specifying the group size.

**Index Terms**— Target Detection, Matrix Completion, Adaptive Group Testing

## 1. INTRODUCTION

Target detection in a field, is a long-standing problem of interest in a variety of applications from environmental to military. In this paper, we consider a framework in which a single vehicle, an autonomous underwater vehicle (AUV), physically samples a field to determine the location of the target. For underwater applications such as chemical plume detection and tracking or mine detection, the challenge of detection is further exacerbated by the severe limitations of resources such as power, sensing capabilities and the density of sensors available – the ocean is vast. Thus, the trade-offs between robustness, efficiency and energy consumption is of significant interest. For example, in tasks like mine-countermeasure operations, the hit rate of the algorithm might be considered more important compared to the energy spent. In contrast, operations that require the AUV to stay underwater for a longer duration making efficient use of resources for measurements or communication is important. Hence, a desirable exploration strategy is one that can get the best of both. The problem of searching for a target in an underwater field is akin to finding a needle in a haystack. The target signature is often only present locally and hard to detect without a brute force scan of the entire field. An exhaustive scan is often prohibitive because of the associated actuation and communication costs. Target detection methods that consider each sample independently such as over fixed size windows suffer from this drawback. A better informed decision can often be taken if the collected measurements are considered in unison and adaptively.

In this paper, we propose an adaptive sampling strategy similar to that used for group testing. The field is progressively sampled such that more samples are collected around the target. Our method explicitly exploits the low rank structure of the field around a target to decide a direction in which the AUV should move next. While



**Fig. 1.** An example of a target (white circle) against the seabed in a real sidescan sonar image from the NSWC dataset. Note the relative size of the target in the field.

a naive application of low-rank matrix completion principles [1, 2] would require the number of samples that grow super-linearly in the grid dimensions ( $O(n \log^2 n)$  for  $n \times n$  grid), we propose combining the basic theory with a group testing approach to make the growth logarithmic in the size of the grid ( $O(m \log m \log n)$  where  $m$  denotes size of sub-grid/group). We demonstrate the performance of our method on actual side scan sonar images with a synthetic spread on the target signature.

Notational Conventions: Lowercase boldface alphabets to denote column vectors (*e.g.*  $\mathbf{z}$ ) and uppercase boldface alphabets to denote matrices (*e.g.*  $\mathbf{A}$ ). Special sets are denoted by uppercase blackboard bold font (*e.g.*  $\mathbb{R}$  for real numbers). The MATLAB<sup>®</sup> indexing rules will be used to denote parts of a vector/matrix.

Active detection and classification have been previously examined. In [3], a data-adaptive approach is taken to select features and the training data is used to design a kernel based classifier. As in our work, the underlying target model is derived from the observations, in contrast, we do not seek to model a wide class of targets using exemplars from the data. Additionally, we seek to solve the exploration-exploitation problem. In principle, the model-based methods of [3] could be adapted to be incorporated in our approach.

In [4–6], the authors take a dynamic programming (DP) based approach for deciding the optimal Bayesian sampling policy with suboptimal strategies to mitigate computational hardness associated with the DP approach. In contrast, we take a random sub-sampling approach in the same spirit as matrix completion [1, 2] that is very easy to implement and has negligible computational overhead. Further, we

This research has been funded in part by the following grants and organizations: ONR N00014-09-1-0700 and NSF CCF-1117896.

do not assume knowledge of the probability distributions.

Our work shares some features with that of distilled sensing [7–9] as far as algorithmic approaches are concerned but we have a low-rank structure instead of a sparsity prior. In [7], the objective is to detect the locations of non-zero elements of a sparse vector by iteratively discarding unlikely locations. In contrast, we assume that our target’s signature has some spatial structure relative to the background clutter or noise. We do not throw away measurements, but rather focus our attention on spatial locations where we should take additional measurements. Thus, we actively determine regions to ignore and focus our attention on the regions of interest. We take this approach since underwater sensing is extremely expensive. Thus, versus [7–9], we do not have the luxury of repeating measurements and the structure we assume in our target (low rank) enables our approach.

## 2. PROBLEM DESCRIPTION

### 2.1. Assumptions

To describe our notation we assume for the time being that the target is located at the origin and noise is absent. Let  $H: \mathbb{R}^2 \rightarrow \mathbb{R}$  denote the *scalar valued* field induced by the target, *i.e.* the target’s signature, and  $\mathbf{x} = (x_c, x_r) \in \mathbb{R}^2$  denote an arbitrary location in the search space. Thus, a mobile agent measuring the field value at location  $\mathbf{x} \in \mathbb{R}^2$  would record the value  $H(\mathbf{x}) = H(x_c, x_r) \in \mathbb{R}$ . We shall make the following key assumptions on the field  $H(\mathbf{x})$ :

(A1) The field is separable in some known basis of  $\mathbb{R}^2$ .

(A2) The magnitude of the field is a monotonically decreasing function of the distance from the target in *any* given direction.

(A3) The field is spatially invariant relative to the target’s position.

An example of (A1) is separability in the  $x_c$  and  $x_r$  directions (*i.e.* in the canonical basis  $\{(1, 0), (0, 1)\}$ ). This means that there exist functions  $F: \mathbb{R} \rightarrow \mathbb{R}$  and  $G: \mathbb{R} \rightarrow \mathbb{R}$  such that  $H(\mathbf{x}) = F(x_c)G(x_r)$ ,  $\forall (x_c, x_r) \in \mathbb{R}^2$ . Notice that if  $H(\mathbf{x})$  is instead separable in the rotated directions  $\Sigma(1, 0)^T$  and  $\Sigma(0, 1)^T$  for some *known*  $\Sigma \in \mathbb{R}^{2 \times 2}$ , then we can work in this rotated coordinate system. Thus, without loss of generality, we shall assume separability of  $H(\mathbf{x})$  in the canonical basis. Assumption (A2) is intuitively clear and can be mathematically described by the inequality:

$$|H(t_1\mathbf{x})| > |H(t_2\mathbf{x})|, \forall \mathbf{x} \in \mathbb{R}^2, t_1, t_2 \in \mathbb{R}, |t_2| > |t_1|. \quad (1)$$

Assumption (A3) implies that if we keep the origin fixed, and the target were moved to  $\mathbf{x}_0 \in \mathbb{R}^2$ , then the new field at location  $\mathbf{x}$  would be given by  $H(\mathbf{x} - \mathbf{x}_0)$ . This assumption ensures that (A1) holds in the canonical basis, regardless of the target’s position  $\mathbf{x}_0$ . In this sense (A3) is stricter than necessary for our purposes, but we retain it for intuitive clarity.

Scalar fields can correspond to intensity measurements or a transformation of the true field. The following types of commonly assumed intensity fields satisfy our assumptions (A1) and (A2):

1. Gaussian fields:  $H(\mathbf{x}) = H_0 \exp(-\|\mathbf{D}\mathbf{x}\|_2^2)$  for any  $2 \times 2$  diagonal matrix  $\mathbf{D}$  and some constant  $H_0 > 0$ .
2. Laplacian fields:  $H(\mathbf{x}) = H_0 \exp(-\|\mathbf{D}\mathbf{x}\|_1)$  for any  $2 \times 2$  diagonal matrix  $\mathbf{D}$  and some constant  $H_0 > 0$ .
3. Power Law fields:  $H(\mathbf{x}) = H_0 |x_c|^{-p_1} |x_r|^{-p_2}$  for constants  $H_0, p_1, p_2 > 0$ .
4. Any *multiplicative* combination of fields satisfying (A1) and (A2), *e.g.*

$$H(\mathbf{x}) = H_0 |x_c|^{-1} |x_r|^{-2} \exp(-c_1 x_c^2) \exp(-c_2 |x_r|)$$

for constants  $H_0, c_1, c_2 > 0$ .

### 2.2. Formulation

In light of our notation and assumptions in the previous section, we can state the active target detection problem as the following task: *To determine the location of the peak in the field  $H(\mathbf{x})$  from its values in only a few locations  $\mathbf{x} \in \mathbb{R}$ , without incurring a heavy computational cost.* Both assumptions (A1) and (A2) contribute towards reducing the number of samples required for active target detection, albeit in different ways. This is apparent from our algorithm in Section 3.1. By virtue of assumption (A2), detecting the target is synonymous with locating the peak of the induced field. This property reduces the required computational effort as shown in Section 3.3.

Before presenting the sampling and reconstruction strategies in Section 3, we demonstrate how (A1) implies a low-rank structure on the field. We’ll use the *lifting* technique from optimization [10]. Suppose that  $H(\mathbf{x}) = F(x_c)G(x_r)$ . The discretized version of the field, sampled on a regular  $n \times n$  grid, can be arranged in the form of a rank one matrix  $\mathbf{H}$ , whose  $(i, j)^{\text{th}}$  entry  $\mathbf{H}(i, j)$  is given by

$$\mathbf{H}(i, j) = H(x_c^i, x_r^j) = F(x_c^i)G(x_r^j). \quad (2)$$

where  $(x_c^i, x_r^j)$  is the physical location of the  $(i, j)^{\text{th}}$  point. The matrix  $\mathbf{H}$  is clearly of rank one since we can express it as the outer product  $\mathbf{H} = \mathbf{f}\mathbf{g}^T$  where

$$\begin{aligned} \mathbf{f} &= [F(x_c^1), F(x_c^2), \dots, F(x_c^n)]^T \\ \mathbf{g} &= [G(x_r^1), G(x_r^2), \dots, G(x_r^n)]^T \end{aligned} \quad (3)$$

Without loss of generality, we assume that both  $x_r^1, x_r^2, \dots, x_r^n$  and  $x_c^1, x_c^2, \dots, x_c^n$  are increasing sequences, corresponding respectively to traversing the grid from top to bottom and from left to right.

## 3. SAMPLING AND RECONSTRUCTION STRATEGY

To illustrate the basic idea, we shall initially assume that  $H(\cdot)$  is a positive scalar field and that all sources of noise are absent. In Section 3.4, we mention modifications to the algorithm in the presence of noise. Note that although we assume a regular square  $n \times n$  grid to illustrate our algorithm, the same arguments apply, almost verbatim, to rectangular grids as well.

### 3.1. Algorithm Description

Let the first round of sampling be on the Cartesian product of the index sets  $T_c, T_r \subseteq \{1, 2, \dots, n\}$  such that  $|T_c| = |T_r| = m$  for some  $m \in \mathbb{Z}^+$  with  $m \ll n$ . Without loss of generality, assume that the indices in  $T_c$ , denoted by  $T_c(1), T_c(2), \dots, T_c(|T_c|)$ , are in increasing order and for notational convenience we shall let  $T_c(0) = 1$  and  $T_c(|T_c| + 1) = n$  to denote the boundary indices of the actual  $n \times n$  grid. We collect  $O(m \log^2 m)$  random measurements on the sub-matrix  $\mathbf{H}(T_r, T_c)$  and use the optimal solution to the tractable convex heuristic

$$\begin{aligned} &\underset{\mathbf{X}}{\text{minimize}} && \|\mathbf{X}\|_* \\ &\text{subject to} && \mathcal{P}(\mathbf{X}) = \mathcal{P}(\mathbf{H}(T_r, T_c)) \end{aligned} \quad (P_1)$$

as our reconstruction  $\widehat{\mathbf{H}}(T_r, T_c)$ . In  $(P_1)$ ,  $\|\mathbf{X}\|_*$  is the sum of the singular values of  $\mathbf{X}$  (nuclear norm), and  $\mathcal{P}$  denotes the projection operator on the sampled indices. Let  $\mathbf{u}$  and  $\mathbf{v}$  denote respectively, the left and right singular vectors corresponding to the largest singular value

of  $\widehat{\mathbf{H}}(T_r, T_c)$ , with  $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$ . Let the largest magnitude element of  $\mathbf{u}$  be at the index  $j_0 \in \{1, 2, \dots, |T_c|\}$ , and that of  $\mathbf{v}$  be at index  $i_0 \in \{1, 2, \dots, |T_r|\}$ . The next round of sampling is on the Cartesian product of the index sets  $T'_c \subseteq \{T_c(j_0 - 1), \dots, T_c(j_0 + 1)\}$ , and  $T'_r \subseteq \{T_r(i_0 - 1), \dots, T_r(i_0 + 1)\}$ , with  $|T'_c| = |T'_r| = m'$  for some  $m' \in \mathbb{Z}^+$  with  $m' \ll n$ .

### 3.2. Proof of Correctness

*Matrix completion* [1, 2] implies that  $O(m \log^2 m)$  random measurements are sufficient to exactly reconstruct the unit rank matrix  $\mathbf{H}(T_r, T_c)$  with high probability by solving the convex heuristic (P<sub>1</sub>). Thus, in the absence of noise, we have  $\widehat{\mathbf{H}}(T_r, T_c) = \mathbf{H}(T_r, T_c) = \sigma \mathbf{u} \mathbf{v}^T$  as the singular value representation with  $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$ . This expression is necessarily unique in the triplet  $(\sigma, \mathbf{u}, \mathbf{v})$  up to a global sign flip for the pair  $(\mathbf{u}, \mathbf{v})$ . By the positivity assumption on the field  $H(\cdot)$ , the matrix  $\mathbf{H}(T_r, T_c)$  is element-wise positive implying that element-wise positivity of the singular vectors  $\mathbf{u}$  and  $\mathbf{v}$  can be ensured (after making a global sign flip if necessary).

If the true location of the target is between the horizontal spatial coordinates  $x_c^{T_c(j)}$  and  $x_c^{T_c(j+1)}$ , then  $\mathbf{u}(1 : j)$  would be a monotonically increasing sequence and  $\mathbf{u}(j + 1 : |T_c|)$  would be a monotonically decreasing sequence. In case the target is located to the left of  $x_c^{T_c(1)}$  (respectively right of  $x_c^{T_c(|T_c|)}$ ) then the entire sequence  $\mathbf{u}(1 : |T_c|)$  would be monotonically decreasing (respectively monotonically increasing). Thus, the index  $j_0 \in \{1, 2, \dots, |T_c|\}$  at which  $\mathbf{u}$  attains a maximum gives the estimate  $x_c^{T_c(j_0)}$  of the horizontal location of the target. Furthermore, if  $\mathbf{u}(j_0)$  is a strict local maximum, *i.e.*  $\mathbf{u}(j_0) > \mathbf{u}(j_0 + 1)$  and  $\mathbf{u}(j_0) > \mathbf{u}(j_0 - 1)$  then the target is definitely located between the horizontal positions  $x_c^{T_c(j_0-1)}$  and  $x_c^{T_c(j_0+1)}$  (due to the spatial monotonicity assumptions on the field  $H(\cdot)$ ). In case,  $\mathbf{u}(j_0)$  is not a strict local maximum and one has  $\mathbf{u}(j_0) = \mathbf{u}(j_0 + 1)$  (respectively  $\mathbf{u}(j_0) = \mathbf{u}(j_0 - 1)$ ) then the target is definitely located between horizontal positions  $x_c^{T_c(j_0)}$  and  $x_c^{T_c(j_0+1)}$  (respectively  $x_c^{T_c(j_0-1)}$  and  $x_c^{T_c(j_0)}$ ). In either scenario, the target is between  $x_c^{T_c(j_0-1)}$  and  $x_c^{T_c(j_0+1)}$  and only the space between these two horizontal coordinates needs to be explored in the next round of sampling. The preceding argument applies almost verbatim in the vertical spatial direction, *i.e.* to the index set  $T_r$ . The target is located between  $x_r^{T_r(i_0-1)}$  and  $x_r^{T_r(i_0+1)}$  and the algorithm explores it in the next round of sampling. After a few rounds of sampling we are able to zero in on the location of the target.

### 3.3. Key Trade-offs

For a quantitative comparison of the trade-offs involved, we present the following analysis. Suppose that in each round of sampling, we collect  $O(m \log^2 m)$  random samples on an  $m \times m$  sub-matrix of  $\mathbf{H}(\cdot, \cdot)$ . If we pick the sub-matrix using  $m$  uniformly spaced rows and columns of  $\mathbf{H}$ , then using the algorithm described above, we reduce the grid size to be searched in the next round of sampling to  $4n^2/m^2$ . If  $N_R$  denotes the number of rounds of sampling (stopping time) needed to zero in on the target then we have,

$$N_R \leq \frac{\log n^2}{\log m^2 - \log 4} = \frac{\log n}{\log m - \log 2} \approx \frac{\log n}{\log m} \quad (4)$$

Let the total number of samples collected be  $N_S$ . Since we are collecting  $O(m \log^2 m)$  samples in each round of sampling, we have

$$N_S = N_R \cdot O(m \log^2 m) = O(m \log m \log n) \quad (5)$$

Let us denote the runtime of the algorithm by  $N_T$ . To compute this, we denote the run time of the  $m \times m$  matrix completion problem from  $O(m \log^2 m)$  random samples, by  $R(m)$ . It is well known that the maximum of an  $m$  length sequence can be computed in  $m - 1 < m$  comparisons. Since each round of sampling involves one matrix completion problem and one problem of determining a sequence maximum, we have

$$N_T \leq N_R \cdot (R(m) + m) \approx N_R \cdot R(m) = R(m) \frac{\log n}{\log m} \quad (6)$$

where the approximation of the iteration cost to  $R(m)$  is valid because matrix completion algorithms are known to scale super-linearly in dimension (if using general purpose Semidefinite Program Solvers like SeDuMi with CVX [11, 12] then  $R(m) = O(m^6)$ ). If we tried to reconstruct the entire field matrix  $\mathbf{H}(\cdot, \cdot)$  instead we would have needed  $O(n \log^2 n) \gg N_S$  number of samples and  $R(n) \gg N_T$  computation time. Thus, reconstructing the entire field turns out to be much worse from both sampling and computational viewpoints.

### 3.4. Handling Noise

When measurement and/or background noise is present we make the following modifications to our basic algorithm. Instead of (P<sub>1</sub>), we now use the optimal solution to a different convex heuristic

$$\begin{aligned} & \underset{\mathbf{X}}{\text{minimize}} && \|\mathbf{X}\|_* \\ & \text{subject to} && \|\mathcal{P}(\mathbf{X}) - \mathcal{P}(\mathbf{H}(T_r, T_c))\|_F \leq \epsilon \end{aligned} \quad (\text{P}_2)$$

as our reconstruction  $\widehat{\mathbf{H}}(T_r, T_c)$ . Problem (P<sub>2</sub>) is the stable matrix completion formulation [13].

The choice of  $m$  becomes important in the presence of noise. We shall numerically demonstrate the effect of  $m$  in our simulation results.

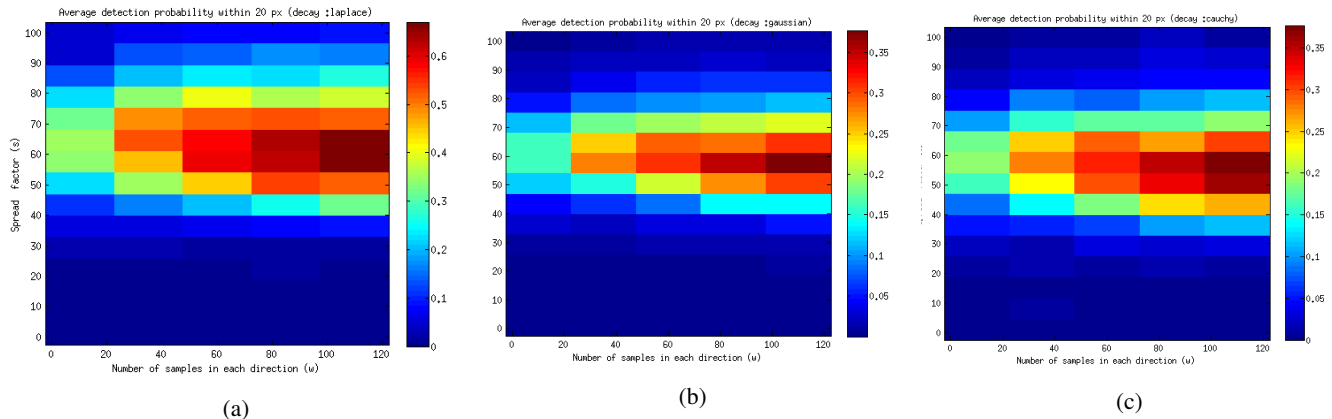
## 4. SIMULATION RESULTS

### 4.1. Setup

We test the performance of our algorithm on synthetic images with a single target and a background comprising random Gaussian noise. The input parameters to the simulation are the decay profile of the target and its spread factor( $\sigma$ ). We perform simulations on three target decay profiles *viz.* Gaussian, Laplacian and Cauchy as shown in (7). The true location of the target is denoted by  $x_0, y_0$  and its decay profile on the 2-D field is given as  $I(x, y)$ . Random *i.i.d.* Gaussian noise is then added to the intensity values at each location  $x, y$ .

$$\begin{aligned} I(x, y) &= A_0 e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{\sigma^2}} \\ I(x, y) &= A_0 e^{-\frac{\sqrt{(x-x_0)^2 + (y-y_0)^2}}{\sigma}} \\ I(x, y) &= \frac{A_0 \sigma^2}{\sigma^2 + (x-x_0)^2 + (y-y_0)^2} \end{aligned} \quad (7)$$

In each such image for a particular decay type we perform 10 monte carlo trials of our localization algorithm. The size of the sampled window  $w$  is varied as an algorithm parameter. The localization results are averaged over these 10 trials to obtain a probability of target localization for the particular image. This is averaged over 50 different images with different target locations to obtain the average



**Fig. 2.** Figure shows the variation in the average probability of localization within 20 pixels of target location with the size of sampled window  $w$  and the spread factor  $\sigma$ . From left to right are the results for the decay types a) Laplacian b) Gaussian c) Cauchy decay profiles for the target. A Gaussian distributed background noise profile was used for the simulation.

probability of target localization within a radius of 20 pixels target location.

To solve the low-rank matrix completion problem ( $P_2$ ) we used the implementation LMaFit [14, 15].

#### 4.2. Discussion

In Figure 2 the spread factor  $\sigma$  for each decay type varies along the y-axis. Number of samples  $w$  collected across each direction varies along the x-axis. For this simulation, we collect the same number of samples in both horizontal and vertical directions.

The results in Figure 2 suggest, as expected, that the accuracy of localization increases when the number of samples collected  $N_s$  at each iteration is increased. The same is true for the spread factor  $\sigma$  except that if for a large values of  $\sigma$  the target signature decays too slowly the algorithm confuses it for background noise. Thus, for each decay type there seems to be an optimal range of spread factors for which the algorithm is able to reliably locate the target. Overall, the average probability of detection appears to be highest when the target decay signature is Laplacian which decays slower compared to Gaussian but faster than the Cauchy decay profile.

This re-iterates the fact that the algorithm performs optimally over a range of decay rate or spread factors  $\sigma$  of the target. Hence, depending on the spread of the field, the algorithm might be required to collect a higher number of samples at each iteration.

#### 5. CONCLUSION

In this paper, we present an active target detection strategy for localizing a target in the field using samples collected by a single vehicle. We adopt a group testing like technique for adaptively collecting more samples around the region of interest. An algorithm is presented for fields that can be assumed to be separable locally around the target, which allows us to exploit its low rank nature.

Simulation results on synthetically generated fields containing targets with varying decay profiles suggest that there exists a trade-off between the number of samples collected by the vehicle at each iteration and the spread of the target. In the future, we would like to additionally adaptively select the number of samples  $w$  at each iteration depending on the local gradient.

#### 6. REFERENCES

- [1] E. J. Candès and T. C. Tao, “The Power of Convex Relaxation: Near-Optimal Matrix Completion,” *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.
- [2] D. Gross, “Recovering Low-Rank Matrices From Few Coefficients in Any Basis,” *IEEE Trans. Inf. Theory*, vol. 57, no. 3, pp. 1548–1566, 2011.
- [3] E. Dura, Y. Zhang, X. Liao, G. J. Dobeck, and L. Carin, “Active learning for detection of mine-like objects in side-scan sonar imagery,” *Oceanic Engineering, IEEE Journal of*, vol. 30, no. 2, pp. 360–371, 2005.
- [4] M. Naghshvar and T. Javidi, “Active Sequential Hypothesis Testing,” October 2012, arXiv:1203.4626v3.
- [5] —, “Two-Dimensional Visual Search,” in *International Symposium Information Theory (ISIT)*, July 2013.
- [6] D. Wei and A. O. Hero, “Multistage Adaptive Estimation of Sparse Signals,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 783–796, April 2013.
- [7] J. Haupt, R. M. Castro, and R. Nowak, “Distilled sensing: Adaptive sampling for sparse detection and estimation,” *IEEE Transactions on Information Theory*, vol. 57, p. 6222–6235, September 2011.
- [8] M. L. Malloy and R. D. Nowak, “Near-Optimal Adaptive Compressed Sensing,” June 2013, arXiv:1306.6239.
- [9] J. Haupt, R. Baraniuk, R. Castro, and R. Nowak, “Sequentially designed compressed sensing,” in *IEEE Statistical Signal Processing Workshop (SSP)*, August 2012, pp. 401–404.
- [10] E. Balas, “Projection, lifting and extended formulation in integer and combinatorial optimization,” *Ann. Oper. Res.*, vol. 140, pp. 125–161, 2005.
- [11] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, [http://stanford.edu/~boyd/graph\\_dcp.html](http://stanford.edu/~boyd/graph_dcp.html).
- [12] I. CVX Research, “CVX: Matlab Software for Disciplined Convex Programming, version 2.0,” <http://cvxr.com/cvx>, Aug. 2012.

- [13] E. J. Candes and Y. Plan, "Matrix Completion With Noise," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 925–936, 2010.
- [14] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Frontiers of Mathematics in China*, vol. 7, no. 2, pp. 365–384, 2012.
- [15] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optimization Methods and Software*, no. ahead-of-print, pp. 1–25, 2012.