FOREGROUND AND BACKGROUND RECONSTRUCTION IN POISSON VIDEO

Eric C. Hall and Rebecca M. Willett

Department of Electrical and Computer Engineering, Duke University, NC 27708

ABSTRACT

Image foreground and background separation is an essential step in a variety of image processing, video analysis, and computer vision tasks. Typically, these methods accept streaming video data, compute an estimate of the background, and subtract this from the observed frames to generate a foreground scene. While such methods are very effective in high SNR regimes, they face serious limitations in low-light settings occurring in night vision surveillance and astronomy. Existing methods cannot be easily modified to yield good results. Therefore, new methods must be created to deal with the low light setting. This paper specifically addresses the problem of foreground and background separation and reconstruction in the case of Poisson distributed observations. The proposed approach builds upon recent advances in both the online learning community and sparse reconstruction methods for Poisson images. To aid in the reconstruction and separation tasks, the method learns and incorporates the dynamics of objects in both the background and foreground in real time.

***Index Terms*—** Photon-limited imaging, Background estimation, Object tracking, Online optimization

1. INTRODUCTION

Many imaging applications such as night vision, infrared imaging, and certain astronomical imaging systems are characterized by limited amounts of available light. In these and other settings, the goal is to reconstruct spatially distributed and dynamic phenomena from data collected by counting discrete independent events, such as photons hitting a detector. More specifically, we can model our observations at time t as

$$y_t \sim \text{Poisson}(\lambda_t), \tag{1}$$

where $y_t \in \mathbb{Z}_+^n$ is the vector of photon counts across n detectors and $\lambda_t \in \mathbb{R}_+^n$ is the intensity of interest (i.e., the n-pixel scene) [1].

We are interested in the case where λ_t has two components: a dynamic foreground ϕ_t which occupies a relatively small portion of the scene, and a static or slowly-varying background β_t, so that

$$\lambda_t = \phi_t + \beta_t.$$

The goal is to recover accurate estimates of ϕ_t and β_t from y_t, especially when the photon counts are very low and when the underlying dynamics are unknown.

There exists a rich literature on image estimation and background subtraction methods, and a wide variety of effective tools in high SNR regimes. For instance, a common method for object tracking is to form an estimate of the background scene, and subtract this from the observation to get an estimate for the foreground [2]. Many of these methods make the assumption that the observed pixel values are the true scene corrupted with white Gaussian noise distributed around the true, slowly varying background mean [3], which is untrue both by the Poisson observation model and in settings with dynamic backgrounds. Alternatively, another technique is to learn and track a low-dimensional subspace representation of the background [4]. While such a method can be modified for the Poisson setting, simply subtracting this background estimate from the observations will still not yield an accurate foreground estimate in the low-light setting. In fact, even if the background were known exactly, subtraction will not give a very accurate estimate of the foreground, as shown in Figure 1.

The photon-limited image estimation problem is particularly challenging because it introduces intensity-dependent Poisson statistics which require specialized algorithms and analysis for optimal performance. Simply transforming Poisson data to produce data with approximately Gaussian noise (via, for instance, the variance stabilizing Anscombe transform [5, 6] or Fisz transform [7, 8]) can be effective when the number of counts is sufficiently high [9, 10]. However, applying these methods to foreground estimation is a difficult problem due to the non-linearities induced by the transforms. Specifically, these tools may make it possible to estimate λ_t effectively, but the inverse problem of estimating ϕ_t and β_t is significantly more challenging because of the nonlinear relationship between the unknowns and variance stabilized observations.

In addition, the dynamic setting presents significant opportunity for improved photon-limited surveillance. Consider the case in which the temporal dynamics are known exactly. For the Gaussian noise setting, the Kalman filter has proved enormously effective. The known dynamics can effectively act as a prior probability model for the scene at time t, and once y_t has been observed, this prior knowledge can dramatically improve reconstruction accuracy even when the number of available photons is small.

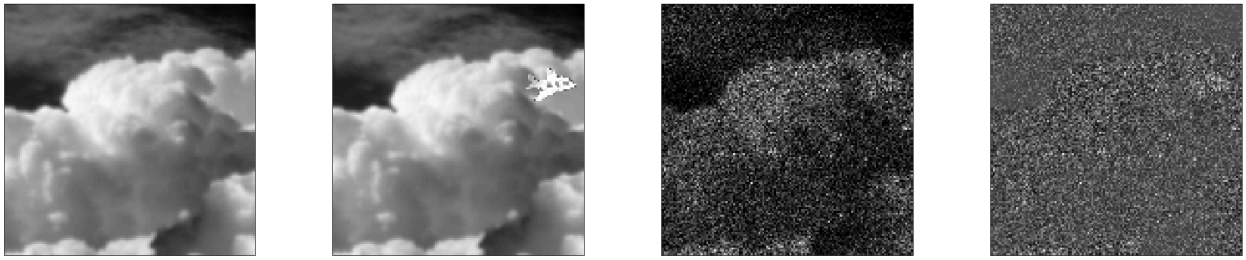
Generalizations of this approach to Poisson noise are possible with particle filters [11], but particle degeneracy is a major practical challenge. Furthermore, classical stochastic filtering methods typically assume an accurate, fully known dynamical model; if a dynamical model is learned from data, it is typically assumed not to change over time.

We present an online method which estimates the underlying, time-varying dynamical model, and uses this estimate to generate online estimates of the foreground and background video sequences. Our approach is based on recent advances in online convex programming and online learning [12, 13, 14, 15]. In particular, we use a variant of Mirror Descent [12, 13] which incorporates dynamical model estimates [16].

2. PROBLEM FORMULATION

We model the data as Poisson observations of a scene which is mostly background with some dynamic foreground. In order to distinguish foreground from background, we assume that the two have

We gratefully acknowledge the support of the awards AFOSR FA9550-11-1-0028, NSF CCF-06-43947, and DARPA FA8650-11-1-7150.



(a) Known Background (b) True Scene (c) Poisson Observation (d) Observation minus true background

Fig. 1: Challenges of background subtractions for photon-limited video. The background (a) and a foreground object in the top right corner form the true scene (b). Poisson observations are then collected from the true scene (c). Even if the background was known exactly and subtracted from the observations, the resulting image (d) is still very noisy, making acurate inference about foreground objects challenging.

discernibly different underlying dynamics, and that the foreground obscures only a small fraction of the background. We denote the observation at time t as y_t, the background as β_t and the foreground as ϕ_t. Therefore the probability density function of the observation is given as:

$$p(y_t|\phi_t,\beta_t) = \prod_{i=1}^{d} \frac{(\phi_{t,i}+\beta_{t,i})^{y_{t,i}}}{y_{t,i}!} \exp\left[-(\phi_{t,i}+\beta_{t,i})\right]. \quad (2)$$

Here, t indicates time index, and i indicates pixel location. Notice that this model assumes that the observed scene is the superposition of background and foreground at every pixel. In actuality every pixel would either be completely foreground or completely background, but it is difficult to model this explicitly because the locations of the foreground pixels would need to be known exactly *a priori*. Using this model we wish to reconstruct β_t and ϕ_t as accurately as possible in a time-efficient manner.

3. DYNAMIC FIXED SHARE ALGORITHM

In order to solve the problems of background and foreground estimation, we will use an algorithm called Dynamic Fixed Share (DFS) [16]. In this section, we describe the DFS method in a general setting, and it's application to background subtraction problems will be described in the next section.

DFS takes in streaming observations and a family of candidate dynamic models $\{\Phi^{(1)}, \Phi^{(2)}, ...\Phi^{(N)}\}$ to produce a sequence of estimates (denoted $\hat{\theta}_t$) with provably low loss. Specifically, at time t we make an observation y_t, and it induces a convex loss function

$$\ell_t(\theta) = f_t(\theta) + r(\theta),$$

where $f_t(\theta)$ describes how well a candidate estimate θ fits the observation y_t and $r(\theta)$ is a regularization function.

DFS works in two steps, the first being to produce an estimate for each candidate dynamical model $\Phi^{(i)}$ at each time step in the following way:

$$\tilde{\theta}_{t+1} = \arg\min_{\theta\in\Theta} \eta_t \langle \nabla f_t(\hat{\theta}_t), \theta\rangle + \eta_t r(\theta) + D(\theta\|\hat{\theta}_t) \quad (3)$$

$$\hat{\theta}_{t+1} = \Phi^{(i)}(\tilde{\theta}_{t+1}). \quad (4)$$

Here, η_t is a step size parameter and $D(\cdot\|\cdot)$ is a Bregman Divergence. These equations effectively update the previous estimate by taking a step in the direction of the negative gradient of f_t, while also ensuring that the new estimate is well regularized and close to the previous estimate. Once this intermediate estimate is found (3), the dynamical model is applied to get the next estimate (4). The second part of DFS is to produce a single estimate from all of the sub-estimates produced by individual dynamic models. It does this by taking a weighted average of the sub-estimates, with weights based on the accumulated loss of each candidate model.

We characterize the performance of this approach via a *regret bound*, which quantifies the difference between the accumulated loss of our method and the accumulated loss of any comparator sequence θ_t which might be output by a competing, potentially batch, method. It is shown that the estimate produced by the DFS method satisfies the following regret bound:

$$\sum_{t=1}^{T}\ell_t(\hat{\theta}_t) - \min_{\theta_1,\theta_2,...,\theta_T}\sum_{t=1}^{T}\ell_t(\theta_t) \leq$$
$$O\Big(\sqrt{T}\Big[(m+1)(\log(N)+1)$$
$$+\log\frac{1}{\alpha^m(1-\alpha)^{(T-m-1)}}$$
$$+\min_{t_2,...,t_{m+1}}\sum_{k=1}^{m+1}\min_{i_k\in\{1,...,N\}}\sum_{t=t_k}^{t_{k+1}-1}\|\theta_{t+1}-\Phi^{(i_k)}(\theta_t)\|\Big]\Big),$$

where N is the number of dynamic models considered, m is the maximum amount of times the optimal dynamic models used to describe the comparator sequence can switch, and α is a parameter used in the algorithm between 0 and 1, which is an estimate on the fraction of times the underlying dynamic model should switch (approximately m/T). The final line of the bound measures how well the comparator sequence, $\theta_1, \theta_2, ..., \theta_T$ follows the dynamics on $m+1$ optimally chosen time segments. This variation term finds the best dynamical model in our family and the optimal time points such that the variation term is minimized. This means that if the comparator sequence can be appropriately described as a series of a few subsequences which each closely follow one of the dynamical models , then the regret bound will be low. For more details see [16].

It is important to note that we use the DFS algorithm for the background instead of a moving average:

$$\hat{\beta}_t = \frac{\sum_{s=1}^{t}\alpha^{t-s}y_s}{\sum_{s=1}^{t}\alpha^{t-s}}$$

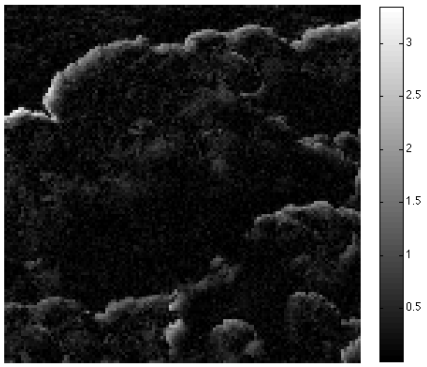


Fig. 2: Absolute difference between moving average and true background with $\alpha = .99$. The true background has max value of 5, meaning the errors are relatively large. Notice that this image contains both errors at the leading edge due to motion, and noise errors from the observation model. Both of these errors would adversely affect the foreground estimation performance

for some $\alpha \in [0, 1]$. This is important because if the background has some dynamic motion, the moving average would perform poorly. If α were set too low, then the background estimate would be heavily corrupted by Poisson noise artifacts. On the other hand, if α were very close to 1, the motion of the background would cause blur in the estimate. Even if α is chosen in between these two extremes, the estimate would not reflect the true background very well as shown in figure 2.

4. METHOD

Our first step is to estimate the background, so we must first find a loss function for estimating β_t. We use the negative Poisson log-likelihood function of the observation omitting the $y_i!$ term since it is an offset not dependent on β:

$$f_{\beta,t}(\beta) = \sum_{i=1}^{d} \left(\beta_i - y_i \log(\beta_i + \gamma)\right). \tag{5}$$

A small constant, γ is added to ensure numerical stability. Notice that this is the same loss function that would be used if the video sequence was assumed to only have background content.

We then wish to estimate ϕ_t. We again start by using the negative Poisson log-likelihood as a basis for the loss function for ϕ_t, but now assume access to an estimate of the background, $\hat{\beta}_t$.

$$-\log(p(y_t|\phi, \hat{\beta}_t)) = \sum_{i=1}^{d} \left(\phi_i + \hat{\beta}_{t,i} - \log\left(\frac{(\phi_i + \hat{\beta}_{t,i})^{y_{t,i}}}{y_{t,i}!}\right)\right) \tag{6}$$

Assuming that the background estimate has already been found, this leads to the following data fit function for the foreground:

$$f_{\phi,t}(\phi; \hat{\beta}_t) = \sum_{i=1}^{d} \left(\phi_i - y_{t,i} \log\left(\frac{\phi_i}{\hat{\beta}_{t,i} + \gamma} + 1\right)\right). \tag{7}$$

This loss function comes from the negative log-likelihood function by subtracting $\sum_{i=1}^{d} \hat{\beta}_{t,i} + \log(y_{t,i}!) - y_{t,i} \log(\hat{\beta}_{t,i})$ which is independent of ϕ_t. Again, a small positive constant γ is used to ensure numerical stability.

Finally, we include regularization penalties, $r_\beta$and r_ϕ. For this application, we use a total variation penalty [17, 18], which insures that the estimates are somewhat smooth, as would be expected in natural images. This makes the overall loss functions the following:

$$\ell_{\beta,t}(\beta) = \quad f_{\beta,t}(\beta) + \tau_\beta \|\beta\|_{\mathrm{TV}} \tag{8}$$

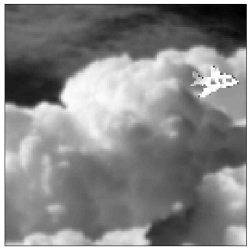
$$\ell_{\phi,t}(\phi; \beta) = \quad f_{\phi,t}(\phi; \beta) + \tau_\phi \|\phi\|_{\mathrm{TV}}. \tag{9}$$

where τ_β and τ_ϕ are tradeoff parameters between data fidelity and regularization for the background and foreground respectively. Notice how this process essentially tries to find a coarse estimate for the underlying scene in the background, and then tries to find a foreground which fine tunes this estimate. These loss functions, as constructed, are convex which means we can use online convex optimization techniques. We will also use the fact that the background and foreground should have different dynamics to help with separation and reconstruction.

Algorithm 1 Background and Foreground Estimation

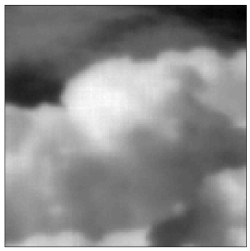
for $t = 1, ..., T$ **do**
  Observe y_t
  for $i = 1, ...N_1$ **do**
    $\tilde{w}^{\beta}_{i,t+1} = w^{\beta}_{i,t} \exp\left(-\eta \ell_{\beta,t}(\hat{\beta}_{i,t})\right)$
    $w^{\beta}_{i,t+1} = \frac{\lambda}{N_1} \sum_{j=1}^{N_1} \tilde{w}^{\beta}_{j,t+1} + (1-\lambda)\tilde{w}^{\beta}_{i,t+1}$
    $\tilde{\beta}_{i,t+1} = \arg\min_{\beta \in B} \eta_t \langle \nabla f_{\beta,t}(\hat{\beta}_{i,t}), \beta \rangle + \tau_\beta \|\beta\|_{\mathrm{TV}} + ...$
      $...\|\beta - \hat{\beta}_{i,t}\|^2$
    $\hat{\beta}_{i,t+1} = \Phi^{(\beta)}_i(\tilde{\beta}_{i,t+1})$
  end for
  $\tilde{\beta}_{t+1} = \sum_{i=1}^{N_1} w^{\beta}_{i,t+1} \tilde{\beta}_{i,t+1} / \sum_{i=1}^{N_1} w^{\beta}_{i,t+1}$
  $\hat{\beta}_{t+1} = \sum_{i=1}^{N_1} w^{\beta}_{i,t+1} \hat{\beta}_{i,t+1} / \sum_{i=1}^{N_1} w^{\beta}_{i,t+1}$
  for $k = 1, ..., N_2$ **do**
    $\tilde{w}_{k,t+1} = w^{\phi}_{k,t} \exp\left(-\eta \ell_{\phi,t}(\hat{\phi}_{k,t}; \tilde{\beta}_{t+1})\right)$
    $w^{\phi}_{k,t+1} = \frac{\lambda}{N_1} \sum_{j=1}^{N_2} \tilde{w}^{\phi}_{j,t+1} + (1-\lambda)\tilde{w}^{\phi}_{k,t+1}$
    $\tilde{\phi}_{k,t+1} = \arg\min_{\phi \in F} \eta_t \langle \nabla f_{\phi,t}(\hat{\phi}_{k,t}; \tilde{\beta}_{t+1}), \phi \rangle + ...$
      $\tau_\phi \|\phi\|_{\mathrm{TV}} + \|\phi - \hat{\phi}_{k,t}\|^2$
    $\hat{\phi}_{k,t+1} = \mathrm{SoftThresh}(\Phi^{(\phi)}_k(\tilde{\phi}_{k,t+1}))$
  end for
  $\tilde{\phi}_{t+1} = \sum_{k=1}^{N_2} w^{\phi}_{k,t+1} \tilde{\phi}_{k,t+1} / \sum_{k=1}^{N_2} w^{\phi}_{k,t+1}$
  $\hat{\phi}_{t+1} = \sum_{k=1}^{N_2} w^{\phi}_{k,t+1} \hat{\phi}_{k,t+1} / \sum_{k=1}^{N_2} w^{\phi}_{k,t+1}$
end for

The overall procedure is described in algorithm 1. For both of the inner loops, the minimization was found by using the FISTA algorithm of Beck and Teboulle [18]. Additionally, a small amount of soft thresholding is applied to the foreground estimate at each time step, to ensure that ambiguous areas that could be considered either background or foreground are removed from the foreground estimate. Without this thresholding, these ambiguous areas would appear in the foreground estimate as an underlying haze. It is important to notice that for the background and foreground we have two slightly different estimates. The values denoted $\tilde{\beta}_t$ and $\tilde{\phi}_t$ are the filtering estimates, meaning that they are reconstructions for time t using all the observations up to time t. The values $\hat{\beta}_{t+1}$ and $\hat{\phi}_{t+1}$ are the prediction values, meaning they use all the data up to time t to predict the observation at time $t + 1$.

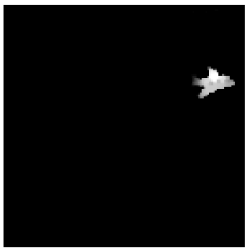


(a) True Scene

(b) Poisson Observation



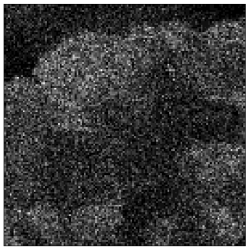
(c) Background Estimate



(d) Foreground Estimate

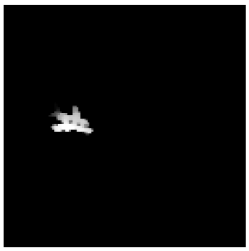
Fig. 3: Foreground and background reconstruction at $t = 250$. The true image (a) has foreground and background content, and the observations (b) are extremely noisy. We form a background estimate (c) and use it to obtain a foreground estimate (d). Notice the details visible in the foreground estimate such as the windows and tail structure of the plane.

(a) True Scene



(b) Poisson Observation

(c) Background Estimate



(d) Foreground Estimate

Fig. 4: Foreground and background reconstruction at $t = 925$. Again notice how the foreground object in (a) is basically imperceptible in the observations (b). By estimating the background (c) an accurate foreground estimate can be constructed (d)

5. EXPERIMENTAL RESULTS

To test this method, we created a data set that featured an object moving across a slowly varying background in the following way:

$$\begin{aligned}
\phi_t^* =& \Phi_{t-1}^{(\phi)}(\phi_{t-1}^*) \\
\beta_t^* =& \Phi_{t-1}^{(\beta)}(\beta_{t-1}^*) \\
x_t =& \sum_{i=1}^{d} e_i^T \left[\mathbb{1}_{\phi,t}(i)\phi_t^* + (1 - \mathbb{1}_{\phi,t}(i))\beta_t^*\right] e_i \\
y_t \sim& \text{Poisson}(x_t),
\end{aligned}$$

where $\mathbb{1}_{\phi(t),t}(i)$ is the indicator of pixel i being foreground or not and e_i is the i^{th} standard basis vector . This process shows a foreground object being translated through the function $\Phi_t^{(\phi)}$ on top of a background image moving with dynamics $\Phi_t^{(\beta)}$. The images were compiled by letting certain pixels be designated as foreground object, and everything else being background. Notice, that the algorithm assumes every pixel is the addition of foreground and background, but the data used is more realistic in that each pixel is either one or the other.

Each image is 150×150, and no pixel has mean value greater than 5, so the video is extremely photon limited. For the background, the true underlying dynamics was a subpixel shift of 1/50th of a pixel to the top left at every time step. For the foreground the true dynamics is a full pixel shift to the top right for the first 500 frames and bottom right for the second 500 frames. The candidate dynamic models used for the background were subpixel shifts of 1/50th of a pixel shift in directions of $k\pi/4$ for $k = 1, 2, ...8$ and stationary ($\Phi^{\beta} = I$). The foreground candidate dynamics were full pixel shifts in the same directions as well as a stationary dynamic.

Figures 3 and 4 show examples of the DFS algorithm taking the series of Poisson observations, and making accurate representations of the foreground and background. It is especially important to notice that including the foreground and background dynamics allows for the foreground image to become clear. Without incorporating dynamics and regularization, the additional foreground image would simply be the transient errors of the background estimation. By including the dynamics in the optimization process, the systematic difference of the background estimate can be found to be the foreground object.

6. CONCLUSIONS

We show how methods developed for photon rich video foreground and background separation cannot be simply translated to work well in the photon limited case, such as Poisson noise. In order to successfully handle the Poisson case, the statistics of low light observations must be explicitly accounted for. We show one such way of doing this, by actually incorporating a negative log-Poisson loss function into our optimization process, in order to reconstruct and separate video foreground and background. Such object tracking in low light situations has important applications such as surveillance and astronomy. Additionally, we show that by incorporating dynamical models into the process, we can find real objects moving in the foreground instead of just spurious noise artifacts.

7. REFERENCES

[1] D. Snyder, *Random Point Processes*, Wiley-Interscience, New York, NY, 1975.

[2] M. Piccardi, "Background subtraction techniques: a review," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, Oct. 2004, vol. 4, pp. 3099 – 3104.

[3] C. Stauffer and W.E.L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 1999, vol. 2.

[4] L. Balzano, R. D. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 704–711.

[5] F. J. Anscombe, "The transformation of Poisson, binomial and negative-binomial data," *Biometrika*, vol. 35, pp. 246–254, 1948.

[6] M. Mäkitalo and A. Foi, "Optimal inversion of the Anscombe transformation in low-count Poisson image denoising," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 99–109, 2011.

[7] M. Fisz, "The limiting distribution of a function of two independent random variables and its statistical application," *Colloquium Mathematicum*, vol. 3, pp. 138–146, 1955.

[8] P. Fryźlewicz and G. P. Nason, "Poisson intensity estimation using wavelets and the Fisz transformation," Tech. Rep., Department of Mathematics, University of Bristol, United Kingdom, 2001.

[9] J. Boulanger, C. Kervrann, P. Bouthemy, P. Elbau, J-B. Sibarita, and J. Salamero, "Patch-based nonlocal functional for denoising fluorescence microscopy image sequences.," *IEEE Trans. Med. Imag.*, vol. 29, no. 2, pp. 442–454, 2010.

[10] B. Zhang, J. Fadili, and J-L. Starck, "Wavelets, ridgelets, and curvelets for Poisson noise removal," *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1093–1108, 2008.

[11] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering*, Springer, 2009.

[12] A. S. Nemirovsky and D. B. Yudin, *Problem complexity and method efficiency in optimization*, John Wiley & Sons, New York, 1983.

[13] A. Beck and M. Teboulle, "Mirror descent and nonlinear projected subgradient methods for convex programming," *Operations Research Letters*, vol. 31, pp. 167–175, 2003.

[14] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient descent," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2003, pp. 928–936.

[15] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning and Games*, Cambridge University Press, New York, 2006.

[16] E. Hall and R. Willett, "Dynamical models and tracking regret in online convex programming," *arXiv:1301.1254*, 2013, To appear in *Proc. ICML*.

[17] T. Chan and J. Shen, *Image Processing And Analysis: Variational, PDE, Wavelet, And Stochastic Methods*, Society for Industrial and Applied Mathematics, 2005.

[18] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions Image Processing*, vol. 18, no. 11, pp. 2419–34, 2009.