BLOOD CELL DETECTION AND COUNTING IN HOLOGRAPHIC LENS-FREE IMAGING BY CONVOLUTIONAL SPARSE DICTIONARY LEARNING AND CODING

Florence Yellin, Benjamin D. Haeffele, and René Vidal

Center for Imaging Science, Johns Hopkins University. Baltimore MD, USA.

ABSTRACT

We propose a convolutional sparse dictionary learning and coding approach for detecting and counting instances of a repeated object in a holographic lens-free image. The proposed approach exploits the fact that an image containing a single object instance can be approximated as the convolution of a (small) object template with a spike at the location of the object instance. Therefore, an image containing multiple non-overlapping instances of an object can be approximated as the sum of convolutions of templates with spikes. Given one or more images, one can learn a dictionary of templates using a convolutional extension of the K-SVD algorithm for sparse dictionary learning. Given a set of templates, one can efficiently detect object instances in a new image using a convolutional extension of the matching pursuit algorithm for sparse coding. Experiments on red blood cell (RBC) and white blood cell (WBC) detection and counting demonstrate that the proposed method produces promising results without requiring additional post-processing.

Index Terms— Holography, lens-free imaging, convolutional dictionary learning and coding, blood cell counting

1. INTRODUCTION

Lens-free imaging (LFI) is emerging as a promising technology for biological applications due to its minimal hardware requirements and large field of view compared to conventional microscopy [1]. One such application is high-throughput cell detection and counting in an ultra-wide field of view [2]. However, detecting objects in a lens-free image is particularly challenging, because standard holographic reconstruction algorithms, such as forward propagation of the hologram using the wide-angular spectrum (WAS) approximation [3], produce significant artifacts, commonly referred to as twin-image artifacts (see Fig. 1 for an example). Consequently, simple object detection methods such as thresholding can fail, because reconstruction artifacts may appear as dark as the object being imaged, producing many false positives.

Template matching (TM) is a classical algorithm for detecting objects in images by finding correlations between an image and one or more pre-defined object templates (see Fig. 1, top). TM is typically more robust to reconstruction artifacts

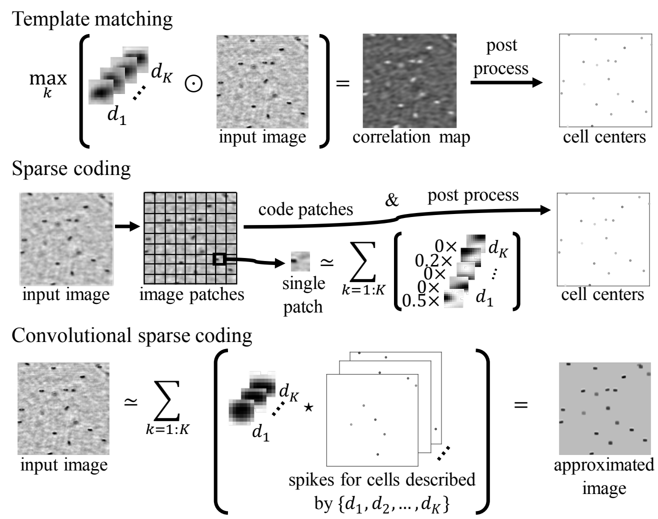

Fig. 1: Three methods for detecting cells in an LFI image obtained via standard holographic reconstruction, which is plagued by reconstruction artifacts: TM (top), SC (middle), and CSC (bottom).

which do not resemble the templates. However, TM requires the user to pre-specify the object templates: usually templates are patches that must be extracted by hand from an image, and the number of templates can be large if one must capture large variability among object instances. Furthermore, TM requires post-processing via non-maximal suppression and thresholding, which are sensitive to several parameters.

Sparse dictionary learning is an unsupervised method for learning templates of typical image patches, and sparse coding (SC) expresses each image patch in terms of these templates (see Fig. 1, middle). This method has been recently applied to cell detection and counting in LFI with great results [4]. However, SC can be computationally demanding as it requires a highly redundant number templates to accommodate the fact that an object need not be centered within a patch. Moreover, SC requires every image patch to be coded even though objects might only appear in a handful of patches. Furthermore, SC must be used in conjunction with other object detection methods, like thresholding.

In this paper we propose a convolutional sparse dictionary learning and coding approach to object detection and counting in LFI. Our approach exploits the fact that an image with multiple object instances can be written as the sum of a few images, each formed by convolving an object template with a sparse location map (see Fig. 1, bottom). Since an image contains a small number of instances relative to the total number of pixels, object detection can be done efficiently using convolutional sparse coding (CSC), a greedy approach that extends the matching pursuit algorithm to convolutional coding. Moreover, the collection of templates can be learned automatically using convolutional sparse dictionary learning (CSDL), a generalization of K-SVD to the convolutional case.

The proposed approach overcomes many of the limitations of other object detection methods while retaining their strengths. Similar to TM, CSC is not fooled by reconstruction artifacts. However, unlike TM, CSC does not use image patches of predefined example objects as templates; instead it learns the templates directly from the data. Another advantage over both TM and SC is that CSC is a stand-alone method for object detection that does not depend on post-processing steps because the coding step directly locates objects. Moreover, if the number of objects in the image is known a-priori, CSC is entirely parameter free, and if the number of objects is unknown, there is a single parameter to be tuned. In addition, CSC does not suffer from the inefficiencies of SC, whose complexity scales with the number of patches and the number of highly redundant templates. In contrast, the runtime of CSC scales with the number of objects in the image and the number of templates needed to describe only centered objects. Existing convolutional methods [5, 6, 7, 8] are computationally intensive and not well-suited to detecting sparse objects in large field of view images. Greedy methods for convolutional coding [9, 10] are not fully developed and have not been designed for LFI applications. These advantages make CSC ideal for cell detection and counting in LFI, particularly when the cells are sparse.

2. METHODS

2.1. Problem Formulation

Let $I : \Omega \to \mathbb{R}^+$, $\Omega \subset \mathbb{R}^2$ be an observed image obtained using, e.g., forward propagation of a hologram using the WAS approximation [3]. Specifically, if H is the recorded hologram, the image I is reconstructed as $I = |T(z) \star H|$ where $\star$ is the 2D convolution operator and $T(z)$ is the wide-angular spectrum transfer function at a focal depth z. Assume that the image contains N instances of an object at locations $\{(x_i, y_i)\}_{i=1}^N$. Both the number of instances and their locations are assumed to be unknown. Suppose also that we have K object templates $\{d_k : \omega \to \mathbb{R}^+\}_{k=1}^K$, $\omega \subset \Omega$, that capture the variations in shape of the object across multiple instances. Let I_i be an image that contains only the i^{th} instance of the object at location (x_i, y_i) and let k_i be the template that best approximates the i^{th} instance. We then have

$$I_i(x, y) \approx d_{k_i}(x - x_i, y - y_i) = d_{k_i}(x, y) \star \delta(x - x_i, y - y_i), \quad (1)$$

We can decompose I as $I \approx \sum_{i=1}^N I_i$, so that

$$I(x, y) \approx \sum_{i=1}^{N} \alpha_i d_{k_i}(x, y) \star \delta(x - x_i, y - y_i), \quad (2)$$

where the variable $\alpha_i \in \{0, 1\}$ is such that $\alpha_i = 1$ if the i^{th} instance is present and $\alpha_i = 0$ otherwise, and is introduced to account for the possibility that there are fewer object instances in I when N is an upper bound for the number of objects. In practice, we can relax the constraint on α_i to $\alpha_i \geq 0$ so that the magnitude of α_i measures the strength of the detection. Observe that the same template can be chosen by multiple object instances, so that good approximations can be obtained with $K \ll N$. Fig. 1 provides a pictorial description of (2).

2.2. Cell Detection by Convolutional Sparse Coding

Assume for the time being that the templates $\{d_k\}_{k=1}^K$ are known. Given an image I, our goal is to find the number of object instances N (object counting) and their locations $\{(x_i, y_i)\}_{i=1}^N$ (object detection). As a byproduct we also need to estimate which template k_i best approximates the i^{th} instance. We can formulate this problem as

$$\min_{\{\alpha_i, k_i, x_i, y_i\}} \|I - \sum_{i=1}^{N} \alpha_i d_{k_i} \star \delta_{x_i, y_i}\|_{\ell_2}^2 \quad (3)$$

where δ_{x_i, y_i} is a shorthand notation for $\delta(\cdot - x_i, \cdot - y_i)$. Since solving (3) for all N objects in the image in one step is very difficult, we use a greedy method to detect objects one at a time (N steps are needed). This approach is an application of matching pursuit for sparse coding [11] to a convolutional objective. At step j, let $R_j = I - \sum_{i=1}^{j} \alpha_i d_{k_i} \star \delta_{x_i, y_i}$ be the part of the input image that has not yet been coded, called the residual image. Initially, none of the image has been coded so $R_0 = I$. After all N objects have been coded, R_N will contain background noise but no objects. The basic object detection step that is used to locate the i^{th} object can be formulated as

$$\min_{\alpha_i, k_i, x_i, y_i} \|R_{i-1} - \alpha_i \delta_{x_i, y_i} \star d_{k_i}\|_{\ell_2}^2. \quad (4)$$

For a fixed α_i, it can be shown that the minimization problem in (4) is equivalent to the maximization problem

$$\max_{k_i, x_i, y_i} \langle R_{i-1} \odot d_{k_i}, \delta_{x_i, y_i} \rangle, \quad (5)$$

where $\odot$ denotes correlation and $\langle \cdot, \cdot \rangle$ denotes the inner product. Notice that the solution to problem (5) is to compute the correlation of R_{i-1} with all templates d_k and select the template and the location that give the maximum correlation (similar to TM). Given the optimal k_i, x_i, y_i, solving for α_i in (4) is a simple quadratic problem, whose solution can be computed in closed form. These observations lead to the CSC method in Algorithm 1.

Algorithm 1 (Convolutional Sparse Coding)

procedure CSC(I, D)
  Choose threshold T
  Initialize $R_0 = I$, $\hat{\alpha}_0 = \infty$, and $i = 0$
  Compute correlation matrix $Q_0 = R_0 \odot [d_1, ..., d_K]$
  while $\hat{\alpha}_i > T$ **do** ▷ Termination criteria
    $(x_{i+1}, y_{i+1}, k_{i+1}) \leftarrow \arg\max_{x,y,z} Q_i$ ▷ Detect 1 object
    $\alpha_{i+1} \leftarrow \max Q_i$
    $\hat{\alpha}_{i+1} \leftarrow \alpha_{i+1}/\alpha_1$
    $R_{i+1} \leftarrow R_i - \alpha_{i+1} d_{k_{i+1}} \star \delta_{x_{i+1},y_{i+1}}$ ▷ Update residual
    $Q_{i+1} \leftarrow R_{i+1} \odot [d_1, ..., d_K]$ ▷ Update correlations
    $i \leftarrow i + 1$

Efficient Implementation of CSC. To obtain an efficient implementation of Algorithm 1, let the size of each of the K templates be m^2 and the size of the image be M^2, where we have $m \ll M$. Initially, we need to perform K large convolutions of size $[m^2] \star [M^2]$. However, during each subsequent iteration i, only a small (m^2) patch of R_i, centered at (x_i, y_i), must be updated. Consequently, only $K(2m-1)^2$ elements of Q_i must be updated during each iteration, which can be done quickly. Further efficiency is gained by noticing that one can use a max-heap implementation to store the large (KM^2) matrix Q. The $\max(Q)$ operation that is performed during each iteration is $O\left(KM^2\right)$ when Q is stored as a matrix but is only $O\left(1\right)$ when Q is stored as a max-heap. The computational gain of using a max-heap to store Q (eliminating N operations, each $O\left(KM^2\right)$) far outweighs the computational cost of maintaining the heap's structure whenever Q is updated (adding $NK(2m-1)^2$ operations, each $O\left(\log\left(KM^2\right)\right)$).

Termination Criteria. Because one object is located at each iteration of the CSC algorithm, the key to accurate counting is to terminate the algorithm at the right time. The sparse coefficients $\{\alpha_i\}$ decrease with i as the chosen objects in the image decreasingly resemble the templates. The algorithm is terminated when $\hat{\alpha}_N = \alpha_N/\alpha_1 \leq T$, where T is a threshold chosen by cross validation. This termination criteria allows CSC to be used to code N objects when N is not known a priori.

2.3. Template Training with Convolutional Sparse Dictionary Learning

Consider now the problem of learning the templates $\{d_k\}_{k=1}^K$. The CSDL method minimizes the objective in (3), but now also with respect to $\{d_k\}_{k=1}^K$ subject to the constraint $\|d_k\|_2 = 1$. In general this would require solving a non-convex optimization problem, so here we employ a greedy approximation that uses a convolutional version of the K-SVD algorithm [12], which alternates between CSC and updating the dictionary. During the coding update step, the dictionary is fixed, and the sparse coefficients and object locations are updated using the CSC algorithm. During the dictionary update step, the sparse coefficients and object locations are fixed, and the object templates are updated one at a time. Specifically, the template d_p is updated as

Algorithm 2 (Convolutional Sparse Dictionary Learning)

procedure CSDL(I)
  Choose numbers of iterations J and templates K
  Initialize D_0 with random, normalized patches of I
  for $j = 0 : (J-1)$ **do**
    $\{x_i^{j+1}, y_i^{j+1}, k_i^{j+1}, \alpha_i^{j+1}\}_{i=1}^N \leftarrow$ CSC(I,D^j) ▷ Do CSC
    for $p = 1 : K$ **do** ▷ Update each template
      $\Delta_p \leftarrow \{i : k_i^{j+1} = p\}, n = \|\Delta_p\|_0$
      if $n > 1$ **then** ▷ n is number patches coded with d_p
        $E_p \leftarrow I - \sum_{i \notin \Delta_p} \alpha_i^{j+1} d^j_{k_i^{j+1}} \star \delta_{x_i^{j+1}, y_i^{j+1}}$
        $\{e_l\}_{l=1}^n \leftarrow E_p$ patches around $\{(x_i^{j+1}, y_i^{j+1})\}_{i \in \Delta_p}$
        $\left(d_p^{j+1}, \{\alpha_i^{j+1}\}\right) \leftarrow$ SVD($[e_1(:), ..., e_n(:)]$)
      else
        $d_p^{j+1} \leftarrow$ max reconstruction error patch, normalized

$$\min_{d_p} \|E_p - \sum_{i \in \Delta_p} \alpha_i \delta_{x_i,y_i} \star d_p\|_{\ell_2}^2, \tag{6}$$

where $E_p = I - \sum_{i \notin \Delta_p} \alpha_i d_{k_i} \star \delta_{x_i,y_i}$ is a residual image associated with the template d_p and $\Delta_p = \{i : k_i = p\}$. Since d_p affects only patches from E_p of the same size as the templates and centered at $\{(x_i, y_i)\}_{i \in \Delta_p}$, we can reduce problem (6) to the standard patch-based dictionary update problem, which is solved using the singular value decomposition [12]. This leads to the method described in Algorithm 2. Once a dictionary has been learned from training images, it can be used for object detection and counting via CSC in new test images.

3. RESULTS

The proposed CSDL and CSC methods were applied to the problem of detecting and counting RBCs and WBCs in holographic lens-free images reconstructed using a single, forward propagation of the hologram via WAS approximation [3]. Two different lens-free imaging setups were used to image anti-coagulated human blood samples from five different healthy donors each (ten donors in total). From each donor and each imaging setup, two types of blood samples were imaged: (1) diluted (300:1) whole blood, which contained primarily RBCs (in addition to a smaller number of platelets and even fewer WBCs); and (2) diluted (5:1) lysed whole blood, containing primarily WBCs and lysed RBC debris. WBCs were more difficult to detect due to the lysed RBC debris. All blood cells were imaged in suspension while flowing through a micro-fluidic channel. Hematology analyzers were used to obtain "ground truth" RBC and WBC concentrations from each of the ten donors. Using the known dimensions of the micro-fluidic channel and the known dilution ratio, we were able to convert between cell counts and concentrations.

CSDL was used to learn four dictionaries, each learned from a single image: A dictionary was learned for each imaging setup (I1 and I2) and each blood sample type (RBC and

Table 1: This table shows percents error of cell counts obtained using CSDL and CSC, compared to extrapolated cell counts from a hematology analyzer. Data came from ten donors; two LFI setups (I1, I2) were used; and both RBC and WBC samples were imaged.

Donor #	I1-RBC	I2-RBC	I1-WBC	I2-WBC
1	-1.8%	–	-10.6%	–
2	-6.3%	–	-3.3%	–
3	-4.6%	–	-43.0%	–
4	2.7%	–	-28.4%	–
5	10.6%	–	-36.2%	–
6	–	-9.2%	–	-8.1%
7	–	10.1%	–	-24.6%
8	–	0.7%	–	11.4%
9	–	8.7%	–	12.1%
10	–	4.4%	–	-5.3%
Mean |%Error|	5.2%	6.7%	24.3%	12.3%

WBC). Ten iterations of the CSDL algorithm were used to learn six RBC templates and seven WBC templates. The RBC and WBC templates were 7x7 and 9x9 pixels, respectively (WBCs are typically larger than RBCs). CSC was then applied to all data sets, approximately 2700 images in all (about 240, 50, 200, and 50 images per donor from datasets I1-RBC, I2-RBC, I1-WBC, and I2-WBC, respectively). Table 1 shows the error rate of the mean cell counts compared to cell counts extrapolated from a hematology analyzer.

Finally, the results obtained using CSC are compared to results obtained from SC, TM, and thresholding in Fig. 2. Notice the tradeoff between image reconstruction time and reconstruction quality when using SC. Notice also that the runtime of CSC is dependent on the number of cells and the number of templates required to describe the variation expected among cells (more variation means more templates are required). Typical RBC images contain about 2500 cells, while WBC images only contain around 250 cells. In the case of the sparse WBCs, CSC is both more accurate and faster than SC; in the case of the denser RBCs, CSC is slower but sometimes more accurate than SC. CSC performs worse than TM for WBC images but better for RBC images; this could be due to TM's sensitivity to several post-processing parameters as well as the pre-processing step of selecting by hand example cells to use as templates. In contrast, CSC is reliant on only a single parameter and requires no pre- or post-processing steps. Lastly, CSC outperforms thresholding, which cannot distinguish between image reconstruction artifacts and cells.

4. CONCLUSIONS

We presented a method based on convolutional sparse dictionary learning and coding for detecting and counting objects in reconstructed holographic lens-free images. We demonstrated the advantages of our approach in counting red and white blood cells. However, our method should be applicable to holographic images of any objects with similar appearance.

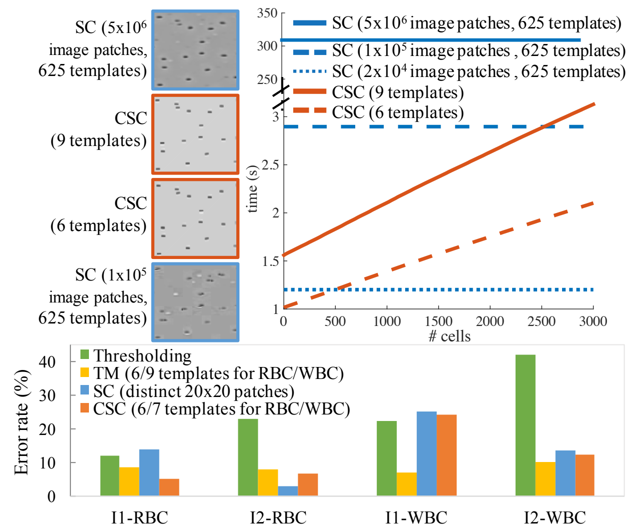

Fig. 2: There is a tradeoff between reconstruction accuracy and runtime inherent in the patch-based SC. In contrast, the runtime of CSC scales with the number of cells in the image (top). The counting error rates for thresholding, TM, SC, and CSC are compared (bottom).

Acknowledgements. We thank Sophie Roth and Lin Zhou for the dataset. This work was funded by miDIAGNOSTICS.

5. REFERENCES

[1] S. B. Kim, H. Bae, K. Koo, M. R. Dokmeci, A. Ozcan, and A. Khademhosseini, "Lens-free imaging for biological applications," *Journal of Lab Automation*, vol. 17, no. 1, pp. 43–49, 2013. 1

[2] Ozcan A and Demirci U, "Ultra wide-field lens-free monitoring of cells on-chip," *Lab Chip*, vol. 8, no. 1, pp. 98–106, 2008. 1

[3] M. K. Kim, *Digital Holographic Microscopy*, Springer, 2011. 1, 2, 3

[4] B. Haeffele, S. Roth, L. Zhou, and R. Vidal, "Removal of the twin image artifact in holographic lens-free imaging by sparse dictionary learning and coding," in *IEEE International Symposium on Biomedical Imaging*, 2017. 1

[5] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus, "Deconvolutional networks," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2528–2535. 2

[6] Hilton Bristow, Anders Eriksson, and Simon Lucey, "Fast convolutional sparse coding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 391–398. 2

[7] Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein, "Fast and flexible convolutional sparse coding," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015, pp. 5135–5143. 2

[8] Brendt Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 301–315, 2016. 2

[9] Arthur Szlam, Koray Kavukcuoglu, and Yann LeCun, "Convolutional matching pursuit and dictionary training," *arXiv preprint arXiv:1010.0422*, vol. 4, 2010. 2

[10] Marius Pachitariu, Adam M Packer, Noah Pettit, Henry Dalgleish, Michael Hausser, and Maneesh Sahani, "Extracting regions of interest from biological images with convolutional sparse block coding," in *Advances in Neural Information Processing Systems*, 2013, pp. 1745–1753. 2

[11] Stéphane G Mallat and Zhifeng Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on signal processing*, vol. 41, no. 12, pp. 3397–3415, 1993. 2

[12] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006. 3