aesm-32-06-03

*Feature Article:*

DOI. No. 10.1109/MAES.2017.160132

# Dynamic, Data-Driven Processing of Multispectral Video Streams

***Honglei Li, Kishan Sudusinghe, Yanzhou Liu,*** **University of Maryland College Park, College Park, MD, USA**
***Jinsung Yoon, Mihaela van der Schaar,*** **University of California Los Angeles, Los Angeles, CA, USA**
***Erik Blasch,*** **Air Force Research Laboratory, Rome, NY, USA**
***Shuvra S. Bhattacharyya,*** **University of Maryland College Park, College Park, MD, USA and Tampere University of Technology, Tampere, Finland**

## INTRODUCTION

Video analytics plays an important role in a wide variety of defense-, monitoring- and surveillance-related systems for air and ground environments. In this context, multispectral video processing is attracting increased interest in recent years, due in part to technological advances in video capture. Compared with monochromatic video, multispectral video offers better spectral resolution, and different bands of multispectral video streams can enhance video analytics capabilities in different ways. For example, the infrared bands can provide better separation of shadows from objects, and improved spatial resolution in scenes that are impaired by fog or haze [16].

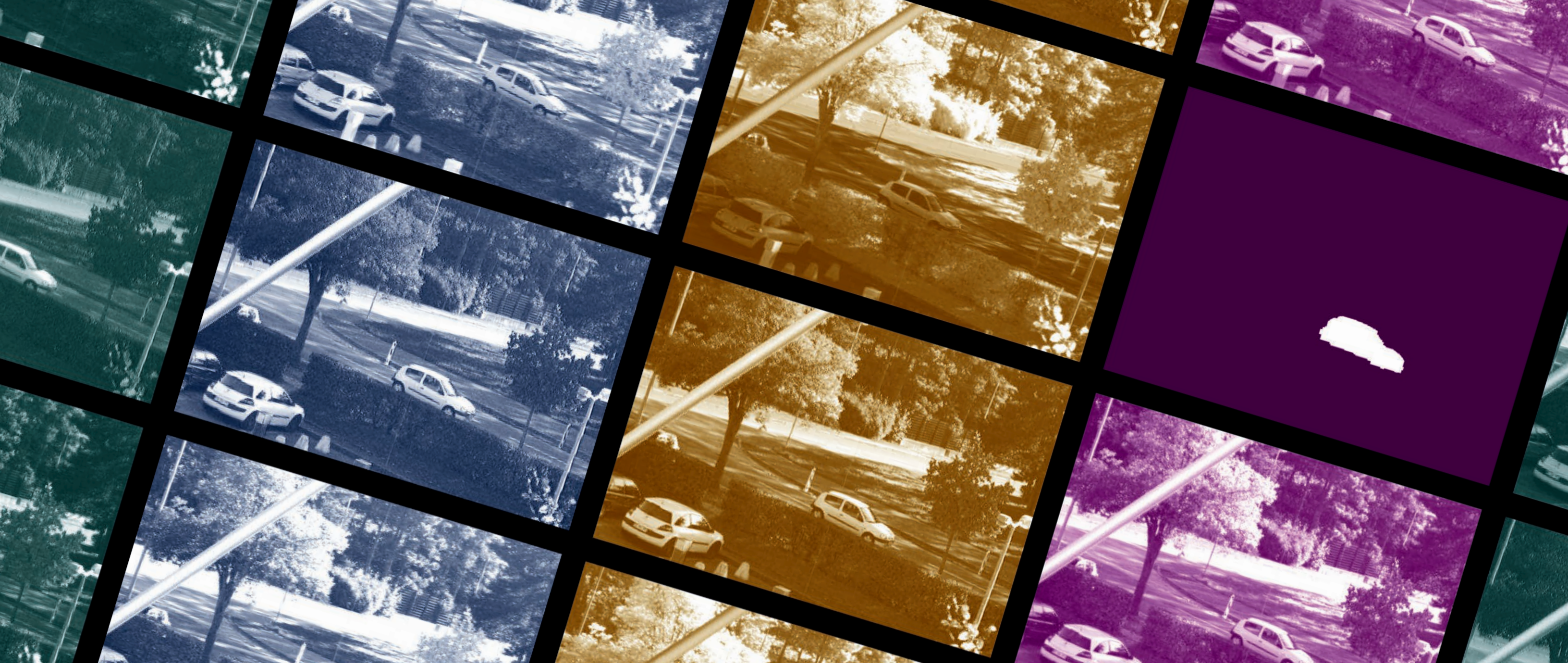
Multispectral video acquisition technology introduces novel opportunities and challenges for applying the paradigm of dynamic, data-driven applications systems (DDDAS) [5] to design and implementation of video analytics systems. The subset of available multispectral bands that is stored and processed, and the hardware and software configurations that are used to perform the processing introduce a complex design space. Furthermore, the most effective operating point in this design space is dependent on the specific application scenario and data characteristics that are encountered at a given point in time during system operation. For example, when system accuracy is of greatest importance, it may be desirable to operate on the full set of available multispectral bands, while in situations where resource constraints are critical (e.g., due to failures in certain subsystems or limited energy capacity), it may be most effective to select a proper subset of the available bands and process the selected bands in a way that optimizes accuracy subject to the given resource limitations.

Based on this view of selectively processed bands from multispectral video data, we introduce in this article a novel system design framework for dynamic, data-driven video processing. A central part of our framework is the application of model-based design methods based on dataflow techniques to represent and transform the functionality of multispectral video processing systems. This allows us to leverage existing knowledge on dataflow techniques, which are employed for design optimization in a wide variety of signal processing application areas, including speech processing, wireless communications, and video processing (e.g., see [2]).

The approach that we discuss supports the development of new DDDAS methods to dynamically select subsets of multispectral bands to process, and dynamically reconfigure the dataflow within the targeted video processing system to achieve the required processing on the selected subset of bands. DDDAS is a paradigm that unifies computational and instrumentation aspects of applications systems, and thereby promotes deeply integrated approaches to modeling, sensing, control, and data processing. DDDAS principles have great relevance to aerospace applications (e.g., see [4], [11], [19]).

Multispectral imaging is related to hyperspectral imaging in that both provide increased spectral discrimination compared with traditional imaging methods. The difference is primarily in the number of bands employed and the degree of spectral resolution (e.g., see [7]). Whereas multispectral imaging generally refers to a number of bands in the range of about 3–10, hyperspectral imaging uses significantly larger numbers of bands—e.g., hundreds, thousands, or more—and narrower bandwidths. Thus, this article is complementary to tutorials in IEEE AESM that have covered aspects of hyperspectral imaging. For example, Birk and McCord provide a review of many different airborne hyperspectral sensing systems, and also provide a detailed comparison of their system

Authors' current addresses: H. Li, K. Sudusinghe, Y. Liu, S. S. Bhattacharyya, University of Maryland College Park, 7950 Baltimore Ave., College Park, MD 20742, USA. E-mail: (honglei@umd.edu). J. Yoon, M. van der Schaar, University of California Los Angeles, Los Angeles, CA; E. Blasch, Air Force Research Laboratory, Rome, NY; S. S. Bhattacharyya, University of Maryland College Park, 7950 Baltimore Avenue, College Park, MD 20742, and with the Department of Pervasive Computing, Tampere University of Technology, Tampere, Finland.
Manuscript received June 8, 2016, revised October 11, 2016, and ready for publication December 12, 2016.
Review handled by M. Cardinale.



specifications [3]. Matteoli, Diani, and Corsini present a survey of methods for processing hyperspectral imagery to detect small human-made anomalies that are relevant in defense and surveillance applications [10].

We refer to our proposed new approach for multispectral image processing as LDspectral, where "LD" here stands for lightweight dataflow [18], [17]. LD is a lightweight design methodology that facilitates cross-platform prototyping, experimentation, and design optimization of signal processing systems. LD is "lightweight" in the sense that it is based on a compact set of application programming interfaces that can be retargeted to different platforms and integrated into different design processes relatively easily. The lightweight dataflow environment (LIDE) is a software tool that supports the LD design methodology, and that we apply in this work [17].

We prototype LDspectral using LIDE together with OpenCV, and present results of extensive experimentation with this prototype to demonstrate the utility of LDspectral. OpenCV provides a large library of software components for video processing (e.g., see [14]), including specialized capabilities that are complementary to the capabilities of LIDE for model-based design and implementation. In particular, the dataflow-based embedded software components (actors) that we employ to implement LDspectral incorporate calls to relevant OpenCV functions to perform specific image processing operations.

We demonstrate and evaluate the performance of LDspectral capabilities using a background subtraction application, along with a recently introduced data set for experimenting with multispectral background subtraction techniques [1]. As compared with a standard image processing pipeline, the dynamic, integrated adjustment of data flow and spectral band selection provides systematic trade-off optimization among computational efficiency and multispectral video processing accuracy.

## RELATED WORK

Benezeth et al. [1] present a publicly available collection of multispectral video sequences that includes ground truth annotation of moving objects. They also apply this data set to demonstrate improvements in background subtraction accuracy when using multispectral video streams compared with RGB (Red, Green, and Blue) streams. Additionally, they provide an evaluation of alternative background subtraction techniques that operate on multispectral video.

This article demonstrates the capabilities of the proposed DDDAS-motivated LDspectral system using the multispectral data set introduced in [1]. Figure 1 shows an example of the data associated with a single video frame within the employed multispectral data set. Specifically, Figure 1 shows 7 different images corresponding to the 7 different bands for the same scene and the foreground result of this scene.

Our work on LDspectral is different from the methods discussed in [1] in its emphasis on integrating DDDAS methods into multispectral video processing, and specifically, on supporting flexible optimization involving the subset of available multispectral bands that is processed, and the associated trade-offs between accuracy and computational cost. Additionally, pixel-level fusion in the front-end of the processing chain for background subtraction is investigated. Pixel-level fusion improves computational efficiency, and reduces the execution time costs incurred by incorporating additional bands (i.e., larger subsets of the available bands) into the video processing pipeline.

While pixel-level fusion is applied in our demonstration of LDspectral, the LDspectral framework does not require use of pixel-level fusion, nor any other specific form of multiband processing. This flexibility allows for integration and experimentation with alternative methods for fusion and analysis of video data across multiple bands (e.g., see [15], [13], [16]) that may enhance the available operating points and overall system adaptivity in terms of accuracy, throughput, and other relevant metrics. Exploration of such alternative methods in the context of LDspectral is an interesting direction for further study to develop multispectral image fusion systems with user interaction [9].

The developments in this article provide new models and methods that are promising for integration in cloud-computing

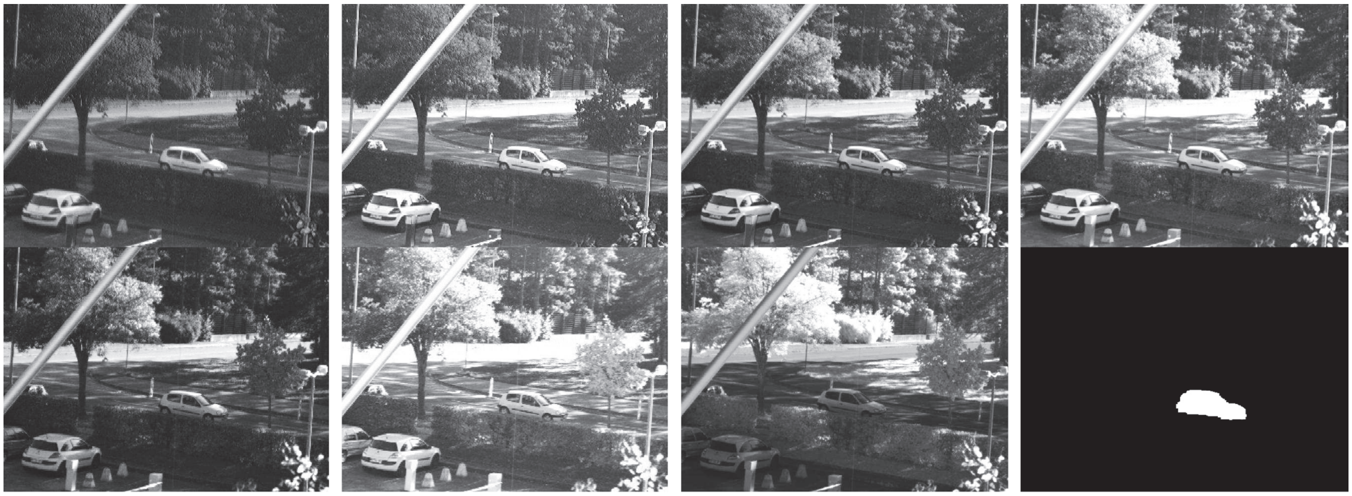


**Figure 1.**
An example of a single video frame in the employed multispectral data set. Images 1–6 show the 6 visible bands, image 7 corresponds to the near-infrared band, and image 8 is the corresponding foreground result that is derived using LDspectral.

frameworks for information fusion, such as the class of frameworks reviewed in [8]. Exploration of such integration is another useful direction for further investigation.

## DESIGN METHODOLOGY

Compared with traditional imaging methods, multispectral imaging provides increased spectral discrimination, which can exploit increasing spectral resolution and spectral diversity. Conventional approaches assume that all of the available bands are employed for the video processing tasks. When system accuracy is of the greatest importance, it may be desirable to use all bands. However, it may be most effective to select a proper subset of all the available bands in situations where resource constraints are critical due to failures in certain subsystems or limited energy capacity.

In the DDDAS-driven video processing system design problem that we target in this article, we assume the availability of multispectral data that comes from a set $Z = \{B_1, B_2, \ldots, B_N\}$ of spectral bands, where $N$ denotes the total number of available bands. In resource- or heavily performance-constrained scenarios where it may not be desirable or feasible to process all bands, this leads a problem of strategically selecting a subset $S \in 2^Z$, where $2^Z$ is the power set of $Z$, that is, the set of all subsets of $Z$.

We assume that we are given a constraint $C_r$ (in units of time) on execution time performance for a particular video processing scenario. Our problem then is to select the set $S \in 2^Z$ to store and process, and the associated strategy to process this selected subset of bands such that video analysis accuracy is maximized subject to the constraint $C_r$. In this article, we focus on the former aspect of this problem—the selection of $S \in 2^Z$—while laying a foundation for incorporating the second aspect as a useful direction for future work.

Figure 2 illustrates our first version system design for LDspectral, which is designed to address the design optimization problem described above. Here, video processing configurations are reevaluated periodically with the period of reevaluation being equal to the value of the reconfiguration interval parameter $T_r$. Lower values of $T_r$ correspond to the possibility for more frequent reconfiguration at the expense of increased overhead due to more frequent operations for reconfiguration management. The reconfiguration management overhead includes computations for dynamically determining whether or not to reconfigure the system, and determining and applying the new operational parameters, including the band subset $S$, when reconfiguration is to be performed.

The block in Figure 2 labeled band subset selection (BSS) is invoked at time intervals determined by the reconfiguration interval parameter $T_r$, subject to application specifications. The BSS block attempts to optimize the subset of bands that is to be employed during the next interval of video processing. In this optimization process, offline data (subset selection profiles) pertaining to the effectiveness of selected subsets of bands is considered along with recent results from performance evaluation, and the current operational constraints $C_r$ and $C_e$.

The output of BSS is a vector indicating the bands $S = \{B_{s1}, B_{s2}, \ldots, B_{sm}\}$ ($m \leq N$ or equivalently, $S \subset Z$) that are to be processed during the next video processing interval.

We perform pixel-level fusion, where the selected bands in a given multispectral image are combined pixel-by-pixel into a single image. In our fusion approach, each pixel in the combined image is derived from a weighted sum of the corresponding pixels in the individual bands. Compared with feature-level fusion, pixel-level fusion can have significantly reduced computational cost since features are extracted from the combined image rather than separately from each individual band (e.g., see [12], [20]). On the other hand, feature-level fusion allows for optimization of feature extraction algorithms for each band [6]. Extension of the LDspectral framework to include feature-level fusion and adaptive selection between pixel- and feature-level fusion is a useful direction for future work.

The video processing functionality performed on the selected bands is represented by the block in Figure 2 labeled band subset

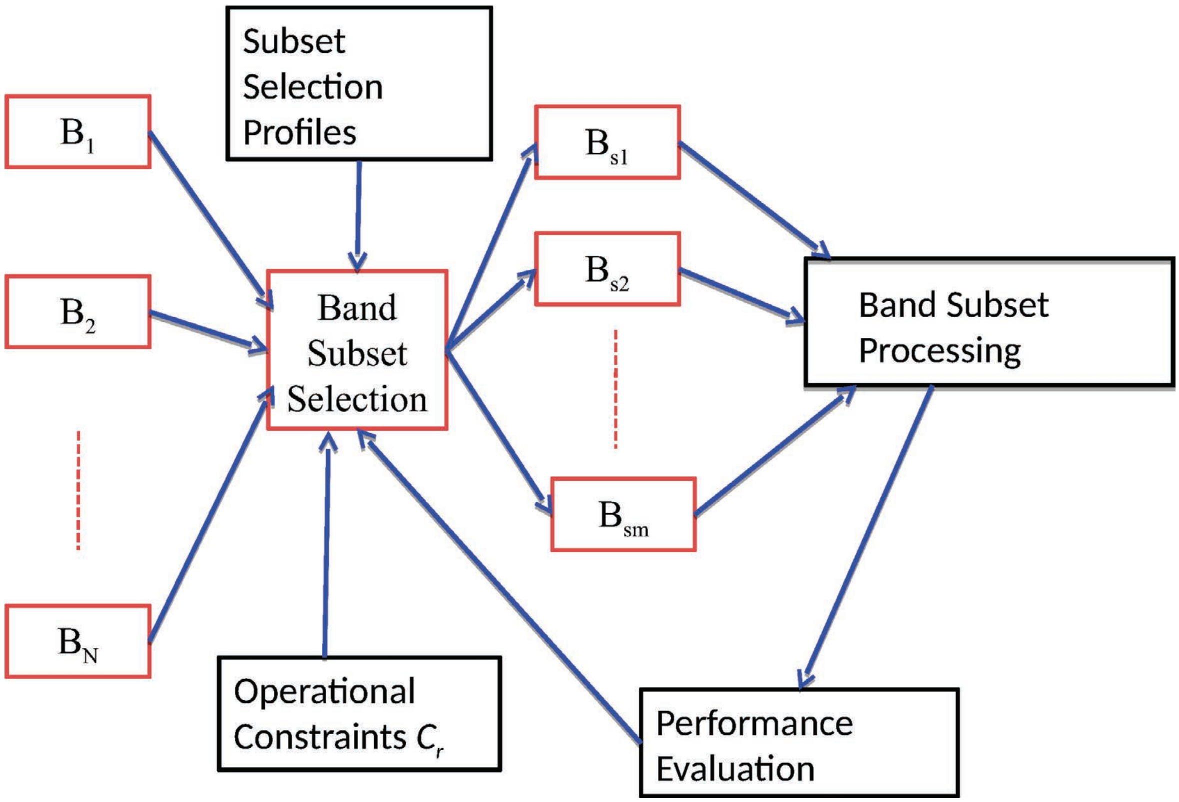



**Figure 2.**
Block diagram of the design flow in LDspectral.

processing. Further discussion on band subset processing in this work is given later in this article.

## CASE STUDY: BACKGROUND SUBTRACTION

We demonstrate the importance of careful BSS, which is a core aspect of the LDspectral design methodology discussed previously. We demonstrate this through a case study involving background subtraction. The two metrics that we consider in this evaluation are the accuracy $F_{measure}$ of the background subtraction (foreground extraction) results, and the average execution time $t_{ave}$ to extract the foreground. We focus on quantifying trade-offs consisting of singleton (one-band) and two-band subsets, and demonstrate significant variations in performance trade-offs among different subsets. Analysis and optimization of BSS trade-offs among larger subsets (i.e., where the subset size exceeds 2) are motivated through this preliminary study as useful directions for future work.

The band subset processing subsystem for this multispectral background subtraction case study is illustrated in Figure 3. In the context of this case study, this illustration represents the internal functionality associated with the block in Figure 2 that is labeled band subset processing. In the dataflow graph subsystem depicted in Figure 3, each actor reads a pointer to an image from its input buffer, and outputs a pointer to the image that results from the image processing operation performed by the actor.

We use LIDE to develop a prototype implementation of the band subset processing subsystem in Figure 3, and we apply calls to selected OpenCV functions in some of the actors within this implementation. The image read actor in Figure 3 is used to inject

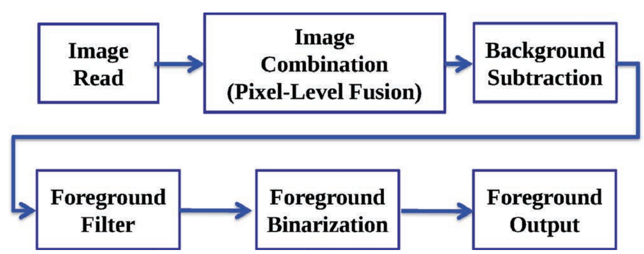



**Figure 3.**
Block diagram of band subset processing in the background subtraction system.

a stream of pointers to successive images into the subsystem so that background subtraction can be performed separately on each image that is referenced (pointed to) in the stream. At the output of the image read actor, each image contains a set of *m* separate components, where each component corresponds to one of the selected spectral bands (i.e., an element of the set *S*, as defined previously). The image combination actor then performs pixel-level fusion to combine the components associated with the selected bands into a singled "fused" image. We provide more details on the fusion operation performed by this actor later in this article.

The background subtraction actor then computes an initial background subtraction result and passes the extracted foreground through the image pointer produced on its output. The core background subtraction operation applied by this actor is carried out by the OpenCV function called BackgroundSubtractorMOG2, which applies a Gaussian mixture model (GMM) [21], [22].

The foreground filter actor in Figure 3 is designed to remove noise from the output of the background subtraction actor. In the foreground filter actor, we use two morphological operations—erosion and dilation—through their respective implementations in OpenCV. Intuitively, the erosion function helps to remove objects in the foreground that are smaller than the filter size (a parameter of the erosion function), and dilation helps to more completely identify boundaries of detected objects. Erosion may lead to distortion in object boundaries; dilation is applied after erosion as a corrective operation to address this potential for distortion.

The foreground binarization actor takes the output of the foreground filter, and converts it into a binary form, where each pixel is classified as being either a foreground or background pixel. This conversion is performed by applying a threshold, and classifying pixels as foreground whenever the corresponding pixel values exceed the threshold. The specific threshold that is employed is determined empirically (off-line) in an effort to enhance classification accuracy. The resulting binary image is then processed by the foreground output actor to store the classification results for each image as a separate file in a given output directory. The files generated in this output directory are indexed so that they can easily be matched up with their corresponding input frames from the given multispectral data set.

We performed experiments applying LDspectral with the background subtraction subsystem shown in Figure 3. These experiments were performed using a laptop computer equipped with an AMD A8-4500M CPU, 4GB RAM, and the Ubuntu 14.04 LTS operating system. Results from these experiments are discussed in the following section.

## EXPERIMENTS

In our experiments involving BSS in conjunction with background subtraction, we applied the novel data set for multispectral background subtraction that was published recently by Benezeth et al. [1]. From this data set, we experimented with multispectral video input that contains 1102 images, where each image contains separate components in 7 different spectral bands. Among these 7 bands, 6 are in the visible spectrum and the remaining one is in the near-infrared spectrum. We divided this set of images into 735 images (approximately 2/3) for training and 367 images for testing.

Here, the training phase is applied to optimize the performance of each two-band subset. Given a band subset $\{b_{s1}, b_{s2}\}$, training is used to optimize the relative weightings for these bands when they are fused in the image combination actor described previously. More specifically, suppose that $x_1$ and $x_2$ are two corresponding pixel values [pixel values at the same image coordinates $(a, b)$] in bands $b_{s1}$ and $b_{s2}$, respectively, and let $y$ denote the pixel value at coordinates $(a, b)$ in the output of the image combination actor. Then $y$ is derived by

$$y = \alpha \times x_1 + (1-\alpha) \times x_2, \tag{1}$$

where $\alpha$ ($0 \le \alpha \le 1$) is a parameter of the image combination actor that is used to control the relative weightings of the two input bands. We refer to this parameter $\alpha$ as the pairwise band combination (PBC) parameter.

Based on this formulation of pixel-level fusion for a two-band subset, our training phase is used to optimize the image combination parameter $\alpha$. This training process is carried out for each distinct pair $\{b_{s1}, b_{s2}\}$ of bands to yield a corresponding PBC parameter value $A(s1, s2)$ that controls the relative weighting of pixels when combining bands $b_{s1}$ and $b_{s2}$.

For each distinct pair $\{b_{s1}, b_{s2}\}$ of bands, the training phase involves performing an exhaustive search across $\alpha \in \{0, 0.1, 0.2, \ldots, 1\}$, and then selecting a value for the PBC (with ties broken arbitrarily) that leads to the highest average accuracy for the background subtraction subsystem of Figure 3. This selected value is then used in the testing phase to assess the accuracy produced by using the band subset $\{b_{s1}, b_{s2}\}$ for background subtraction.

The measure of accuracy employed in these experiments is the harmonic mean performance measure of background subtraction accuracy, which is motivated, for example, in [1]. This measure is defined as

$$F_{measure} = 2 \times \frac{recall \times precision}{recall + precision} \tag{2}$$

Here, *precision* and *recall* are defined by

$$precision = \frac{n_c}{n_f}, \text{ and } recall = \frac{n_c}{n_g}, \tag{3}$$

where $n_c$ is the number of correctly classified foreground pixels, $n_f$ is the number of pixels classified as foreground, and $n_g$ is the number of foreground pixels in the ground truth.

Table 1 and Table 2 show experimental results using the off-line analysis capabilities of LDspectral to evaluate processing trade-offs among different one- and two-band combinations (i.e., where the set of selected bands is restricted to contain only one or two elements). Table 1 shows the background subtraction accuracy that is experimentally observed for different one- and two-band combinations, while Table 2 shows the processing times for different combinations. In each of these tables, the diagonal entries give the results for single-band processing, while each entry at row $a$ and column $b$ when $a \neq b$ gives the results from joint processing of the bands indexed by $a$ and $b$. In each of these tables, elements below the diagonal are not shown since they are symmetric with respect to the diagonal. As mentioned above, we employ 1102 images in each of these experiments. These 1102 images form the complete set of images from the employed multispectral data set [1] that have ground truth available as part of the data set.

From Table 1, we see that the accuracy provided by LDspectral is significantly higher on average compared with the results presented in [1] for the same video data set. This demonstrates the effectiveness of LDspectral in optimizing the accuracy of background subtraction.

Experimentally derived data of the form shown in Table 1 and Table 2 can be used as the subset selection profiles to guide BSS, as illustrated in Figure 2. Additionally, the results in Table 2 define lower limits on how short the reconfiguration interval $T_r$ can be.

Table 3 shows the optimized values for the PBC parameters that were derived through the training procedure for processing of

Table 1.

| Accuracy Results for Different One- and Two-Band Combinations using LDspectral, and (in the last three rows) the Results from [1] with Three Different Algorithms | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Band** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | 0.934 | 0.940 | 0.944 | 0.945 | 0.943 | 0.943 | 0.933 |
| 2 | — | 0.931 | 0.942 | 0.936 | 0.942 | 0.937 | 0.930 |
| 3 | — | — | 0.939 | 0.939 | 0.943 | 0.940 | 0.939 |
| 4 | — | — | — | 0.929 | 0.940 | 0.932 | 0.930 |
| 5 | — | — | — | — | 0.942 | 0.938 | 0.937 |
| 6 | — | — | — | — | — | 0.919 | 0.925 |
| 7 | — | — | — | — | — | — | 0.886 |
| **Mahalanobis distance** | — | — | — | — | — | — | 0.689 |
| **Spectral angle** | — | — | — | — | — | — | 0.897 |
| **Spectral information divergence similarity** | — | — | — | — | — | — | 0.896 |

Table 2.

| Execution Times for Different Single- and Dual-Band Combinations. | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Band** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | 62.5 | 68.4 | 67.2 | 66.9 | 67.5 | 68.4 | 66.5 |
| 2 | — | 62.3 | 68.2 | 68.4 | 68.0 | 69.7 | 68.2 |
| 3 | — | — | 62.8 | 67.4 | 66.9 | 68.5 | 66.9 |
| 4 | — | — | — | 63.1 | 67.4 | 69.0 | 66.8 |
| 5 | — | — | — | — | 63.0 | 68.8 | 66.8 |
| 6 | — | — | — | — | — | 62.2 | 68.4 |
| 7 | — | — | — | — | — | — | 63.4 |
| **NOTE**: The results here are given in milliseconds. Each entry in the table represents the average time to perform background subtraction (including the entire processing chain shown in Figure 3 on a single input image). | | | | | | | |

Table 3.

| Derived Values for PBC Parameters (rounded to tenths) | | | | | | |
|---|---|---|---|---|---|---|
| **Band** | **2** | **3** | **4** | **5** | **6** | **7** |
| 1 | 0.7 | 0.4 | 0.6 | 0.3 | 0.7 | 0.9 |
| 2 | — | 0.3 | 0.7 | 0.4 | 0.5 | 0.8 |
| 3 | — | — | 0.9 | 0.5 | 0.7 | 0.9 |
| 4 | — | — | — | 0.4 | 0.5 | 0.9 |
| 5 | — | — | — | — | 0.5 | 0.9 |
| 6 | — | — | — | — | — | 0.8 |

two-band subsets. The rows and columns of Table 3 correspond, respectively, to $x_1$ and $x_2$ in (1). For example, when $S$ consists of bands 2 and 3, we use $\alpha = 0.3$, and when $S$ consists of bands 1 and 4, we use $\alpha = 0.6$. The diversity of the values in this table demonstrates the utility of optimizing the PBC parameter separately for each two-band subset rather than using a balanced weighting ($\alpha = 0.5$) or some other uniform PBC parameter setting for all subsets.

We also see from Table 3 that the optimized $\alpha$ values are relatively high when $x_2$ is taken to be band 7, which is the near-infrared band. This results in correspondingly low weightings given to band 7. This trend matches intuitively with the data in Table 1, which shows that band 7 in isolation has significantly lower accuracy compared with all of the other one-band subsets.

Overall, the results in Table 1 and Table 2 help to motivate the utility of careful selection of band combinations as there is significant variation in accuracy among different pairs of bands. The results also help to quantify the trade-off—in terms of increased execution time—when a single band is augmented with a second band to help increase background subtraction accuracy. For the implemented pixel-level fusion approach, this increase is found to be relatively low (within 13% in all cases). This is because increasing the number of bands increases the computational load for only a small subset of the actors in Figure 2 and Figure 3—in particular, the actors for band selection and image combination.

## CONCLUSION

In this article, we have introduced a novel system design framework for dynamic, data-driven processing of multispectral video streams using LD techniques. The framework is motivated by the need for efficient and accurate video processing in a wide variety of systems for air and ground environments. This framework, called LDspectral, is designed to incorporate selection of subsets of bands as a core, front-end step in the video processing process. This emphasis on BSS opens up a large design space for data-driven adaptation that influences key metrics, including accuracy and computational efficiency. We have demonstrated a prototype implementation of LDspectral applied to a background subtraction application. Through experiments with this prototype on a relevant data set, we have demonstrated the utility of flexible, optimized BSS in the navigation of operational trade-offs for multispectral video processing systems.

The current version of LDspectral is developed for input streams in which the multispectral images are well aligned across the different bands. The band subset processing subsystem in LDspectral can readily be extended to incorporate image registration, which would be useful to extend the capabilities of the overall system to handle images that are not aligned. Such extension together with the integrated optimization of associated operational trade-offs is a useful direction for future work. ◆

## ACKNOWLEDGMENT

This research was supported in part by the U.S. Air Force Office of Scientific Research under the DDDAS Program.

## REFERENCES

[1] Benezeth, Y., Sidibé, D., and Thomas, J. B. Background subtraction with multispectral video sequences. In *Proceedings of the Workshop on Non-classical Cameras, Camera Networks and Omnidirectional Vision,* 2014.

[2] Bhattacharyya, S. S., Deprettere, E., Leupers, R., and Takala, J. (Eds.). *Handbook of Signal Processing Systems* (2nd ed.). New York: Springer, 2013. ISBN: 978-1-4614-6858-5 (Print); 978-1-4614-6859-2 (Online).

[3] Birk R. J., and McCord, T. B. Airborne hyperspectral sensor systems. *IEEE Aerospace & Electronic Systems Magazine,* Vol. 9, 10 (1994), 26–33.

[4] Blasch, E. Enhanced air operations using JView for an air-ground fused situation awareness UDOP. In *Proceedings of the IEEE/AIAA Digital Avionics Systems Conference,* 2013, pp. 5A5–1–5A5–11.

[5] Darema, F. Grid computing and beyond: The context of dynamic data driven applications systems. *Proceedings of the IEEE,* Vol. 93, 2 (2005), 692–697.

[6] Demars, C. D., Roggemann, M. C., and Havens, T. C. Multispectral detection and tracking of multiple moving targets in cluttered urban environments. *Optical Engineering,* Vol. 54, 12 (2015), 123106–1–123106–14.

[7] Ferrato, L. Comparing hyperspectral and multispectral imagery for land classification of the lower Don river, Toronto. Master's thesis, Ryerson University, Toronto, Canada, 2012.

[8] Liu, B., Chen, Y., Hadiks, A., Blasch, E., Aved, A., Shen, D. et al. Information fusion in a cloud computing era: A systems-level perspective. *IEEE Aerospace & Electronic Systems Magazine,* Vol. 29, 10 (2014), 16–24.

[9] Liu, Z., Blasch, E., Xue, Z., Zhao, J., Laganiere, R., and Wu, W. Objective assessment of multiresolution image fusion algorithms for context enhancement in night vision: A comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 34, 1 (2012), 94–109.

[10] Matteoli, S., Diani, M., and Corsini, G. A tutorial overview of anomaly detection in hyperspectral images. *IEEE Aerospace & Electronic Systems Magazine,* Vol. 25, 7 (2010), 5–28.

[11] Nguyen, N., Maifi, M., and Khan, H. Context aware data acquisition framework for dynamic data driven applications systems (DDDAS). In *Proceedings of the Military Communications Conference,* 2013, 334–341.

[12] Nirmala, D. E., and Vaidehi, V. Comparison of pixel-level and feature level image fusion methods. In *Proceedings of the International Conference on Computing for Sustainable Global Development,* 2015, 743–748.

[13] Patrick, J., Brant, R., and Blasch, E. Hyperspectral imagery throughput and fusion evaluation over compression and interpolation. In *Proceedings of the International Conference on Information Fusion,* 2008, 1–8.

[14] Pulli, K., Baksheev, A., Kornyakov, K., and Eruhimov, V. Real-time computer vision with OpenCV. *Communications of the ACM,* Vol. 55, 6 (2012).

[15] Ramesh, K. P., Gupta, S., and Blasch, E. P. Image fusion experiment for information content. In *Proceedings of the International Conference on Information Fusion,* 2007, 1–8.

[16] Rufenacht, D., Fredembach, C., and Susstrunk, S. Automatic and accurate shadow detection using near-infrared information. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 36, 8 (2014), 1672–1678.

[17] Shen, C., Plishker, W., and Bhattacharyya, S. S. Dataflow-based design and implementation of image processing applications. In *Multimedia Image and Video Processing,* $2^{nd}$ ed., L. Guan, Y. He, and S.-Y. Kung, Eds. Boca Raton, FL: CRC Press, 2012, ch. 2, pp. 609–629.

[18] Shen, C., Plishker, W., Wu, H., and Bhattacharyya, S. S. A lightweight dataflow approach for design and implementation of SDR systems. In *Proceedings of the Wireless Innovation Conference and Product Exposition*. Washington, DC, Nov. 2010, 640–645.

[19] Silva, W., Frew, E. W., and Shaw-Cortez, W. Implementing path planning and guidance layers for dynamic soaring and persistence missions. In *Proceedings of the International Conference on Unmanned Aircraft Systems,* 2015, 92–101.

[20] Zhou, Y., Mayyas, A., Qattawi, A., and Omar, M. Feature-level and pixel-level fusion routines when coupled to infrared night-vision tracking scheme. *Infrared Physics & Technology,* Vol. 53 (2010), 43–49.

[21] Zivkovic, Z. Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings of the International Conference on Pattern Recognition,* 2004, 28–31.

[22] Zivkovic, Z., and van der Heijdenb, F. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters,* Vol. 27, 7 (2006), 773–780.


Author: This article has been lightly edited for grammar, style, and usage. Please compare against your original document and make changes on these pages. Please limit your corrections to substantive changes that affect meaning. If no change is required in response to a question, please write "OK as set" in the margin. Copy editor