AN AERIAL CHANGE DETECTION SYSTEM USING MULTIPLE DETECTOR FUSION AND ADABOOST CLASSIFICATION

Yi Tan, Subhodev Das, Ali Chaudhry
SRI International, 201 Washington Road, Princeton, New Jersey 08540, USA

ABSTRACT

Change detection in aerial imagery is challenging due to the effects of platform movements, atmospheric degradation and unpredictable photometric or thermal variations of the scene. We introduce an aerial change detection system built around multiple change detectors, each exploiting a different feature set, and fusion of their output in order to overcome the challenges. A supervised classifier screens the fused results for false detections. In particular, we have investigated three change detectors: Histogram of Oriented Gradients (HOG), Clustering-Based Anomaly Detection (CBCD), and Local Binary Similarity Pattern (LBSP). A set of change regions (blobs) are retained following fusion. Next, a set of features are extracted from each blob and fed into an AdaBoost classifier for *true* change acceptance test. Ours is a multi-threaded system performing change detection in real time. Test results on a variety of EO and IR imagery demonstrate the efficacy our system in real-world applications.

Index Terms — Change Detection, Fusion, Classification, Aerial Image

1. INTRODUCTION

Change detection is the process of identifying differences in the state of a scene entity by observing it in images captured at different times. Image change detection has been an active area of computer vision research over the past few decades [1]. IEEE Change Detection Workshops [2][3] have spearheaded the release of benchmark datasets using a variety of sensors that are augmented with ground truth labels. Among the state-of-the-art (SoA) methods is the flux tensor [4] with split Gaussian models (FTSG), which uses a three-step process: (1) moving object detection on the trace of flux tensor and a variant of Gaussian mixture model (GMM), (2) fusing motion detection results, and (3) removal of ghost artifacts. SuBSENSE [5], a top performer in 2014 change detection challenge, uses color and local binary similarity patterns (LBSP) to make pixel-level decisions and automatically tunes parameters to adapt to changing input. These methods have proven successful in moving object detection and tracking in ground-based color video from stationary cameras. However, they primarily handle "short-term change detection" among successive frames of a single video. They are not very effective for the general change detection problem involving a pair of images (*before-after*) with arbitrary time gap, particularly in the aerial context.

In this paper, we describe a change detection approach and its implementation in a real-time system for aerial imagery. The types of changes investigated are often unconventional, such as disturbed earth, tire tracks, or small ground targets. The temporal difference between *before* and *after* images can vary from seconds to months. There are many challenges in such scenarios: first, the interesting changes may appear weak, say in IR images w.r.t. environmental thermal variations; second, the outline of the changes may not be well defined; third, a fast moving aerial platform may image a scene point only once, thereby, offering no opportunity to gather pixel-level statistics; fourth, motion parallax, shadows, and image compression artifacts may introduce significant false alarms. Due to these challenges, the traditional change detection approaches, such as background modeling and subtraction or foreground segmentation, based on pixel intensity differences [1][5][6], may not be viable.

The primary contributions of our work are the following: we combine the complementarity of multiple change detectors operating in pixel gradient and intensity spaces to form an initial set of robust hypotheses (blobs of connected pixels); we utilize an AdaBoost classifier trained on multi-dimensional feature vectors extracted from the blobs to enhance the robustness. These contributions lend to the viability of ours as a practical system with a high detection rate and low false alarms.

The paper is organized as follows. Section 2 describes the various change detection algorithms with our variations implemented in the system. Section 3 provides the details of the implementation. Section 4 presents experimental results. The final section has the concluding remarks.

2. ALGORITHMS

A robust change detection algorithm should be capable of detecting changes, irrespective of signal strength. However, in most application scenarios, it is observed that a single feature (e.g. intensity) may not achieve the degree of robustness needed for real-world change detection. In this section we describe an approach that utilizes multiple change detectors, each exploiting a different feature space, and combines the output of these detectors. To reduce false alarms, a trained AdaBoost classifier is applied to the fused output for change detection acceptance test.

2.1. Change detection by HOG feature matching

Histogram of Oriented Gradients (HOG) has proved to be an effective feature for human and vehicle detection [7] [8]. In our implementation, we use the first degree of spatial derivatives of pixel intensities to compute the gradients. (We experimented with more complex derivatives, such as 3×3 and 5×5 DoG kernels, but found little performance improvement.) To handle intensity or contrast reversals in IR imagery, the use of *un-signed gradients* proved to be accommodating. A set of control variables is defined for the HOG descriptor: {*numBins, Signed, minThre, pyrLevel, R*}, where *numBins* is the number of gradient bins, *Signed* (Boolean) specifies use of signed/unsigned gradients, *minThre* is the minimal threshold for valid derivatives, *pyrLevel* is the image pyramid level, and *R* is the radius for computing HOG. The default control settings are {*18, false, 5.0, 1, 9*}. Control variables *pyrLevel* and *R* are adjusted according to the scale of changes sought. In computing HOG, each pixel within *R* vote for a histogram bin weighted by the inverse of square of the pixel distance from the patch center. The uniqueness of our algorithm is the geometric distance-based weighting that emphasizes spatial structural changes over derivative strengths. This helps to reduce the detector sensitivity to contrast and intensity changes. L_1 normalization is applied to the HOG descriptor $\{v_i\}$, such that $\sum v_i = 1.0$.

Given an aligned image pair, dense HOG is computed on each image pixel (*i, j*). Let $\mathbf{b}(i,j)=\{b_k\}$ ($k=1, \ldots, N$) denote HOG descriptor of *before* image at pixel (*i, j*) and $\mathbf{a}(i,j)=\{a_k\}$ ($k=1, \ldots, N$) that of *after* image. Here *N* is the length of the HOG descriptor. Instead of using HOG differences to detect changes [9], we formulate change detection as problem of matching HOG descriptors using *Bhattacharyya* distance:

$$M(a,b) = \sum_{k=1}^{N} \sqrt{a_k * b_k} \tag{1}$$

If corresponding image patches are similar, then $M(a, b) \approx 1$; if they are very dissimilar, then $M(a, b) \approx 0$. To identify image changes, our algorithm looks for pixels with low matching scores. A 2-D *heat map* is derived by obtaining pixel *heat value* (=*1-M*) and scaling it to 8 bits followed by low-pass filtering *G*(*x*), as shown in Eq. 2. The filtering removes spurious and isolated noises and improves connectivity of detected change regions.

$$H(i,j) = G([1.0 - M(a,b)] * 2^8) \tag{2}$$

2.2. Change detection by pixel intensity clustering

An intensity-based method utilizes pixel intensity and contrast distributions to detect changes in the scene using the well-known *Reed-Xiaoli* (*RX*) [10] anomaly detector. In this method, we apply *k*-means clustering to characterize the *background* – the before image. Let μ_k and Σ_k denote the mean and covariance of the *k*-th background cluster, the change detection process obtains the *Mahalanobis* distance, as shown in Eq. 3, for each pixel x in the *after* image against the corresponding background cluster in the *before* image. We call this the clustered-based change detection (CBCD). To make the detection less sensitive to the intensity variations between two images, a pre-processing step of intensity normalization of *before* image using *after* image as the reference is applied. This is summarized in Eq. 4, where μ_a and μ_b are mean pixel intensities of *after* and *before* images, δ_a and δ_b are the corresponding standard deviations, and B' is the normalized *before* image.

$$RX(x) = (x - \mu_k)^T {\Sigma_k}^{-1} (x - \mu_k) \tag{3}$$

$$B'(x,y) = \frac{\delta_a}{\delta_b}\left(B(x,y) - \mu_b\right) + \mu_a \tag{4}$$

The *RX* scores are used to compose a 2-D change heat map of pixel anomaly between *before* and *after* images.

2.3. Change detection using local similarity measures

Local Binary Similarity Pattern (LBSP) [11] of pixel and its vicinity is used to measure the *intra* and *inter* differences of a before-after image pair. This method responds well to changes in both texture and intensity. It uses a $n \times n$ pixel neighborhood (typically, *n*=5) as shown in Figure 1. The *local similarity* binary code – a 16-bit binary vector – of a center pixel and its neighborhood is obtained using Eq. (5) and (6), where i_c is center pixel intensity, i_p are intensity values of the neighborhood pixels. T_d is similarity threshold.

In our method, to make T_d adaptive to local image area contrast and intensity variations, the image is divided into *N*×*N* (*N*=8~16) sub regions and a threshold T_d^r is obtained for each sub region *r*. This is shown in Eq. 7, where T_d^0 (=20) is a preset base threshold and μ_a^r μ_b^r are local mean intensities of the sub region *r* in *after* and *before* images, respectively.

Dense LBSP is derived on *before* image (intra) and *after* image (inter). In computing the inter LBSP, the center *x* from the *before* image is used. The change detection is measured by computing the *Hamming* distance between the intra- and inter-LBSP binary vectors.

o		o		o
	o	o	o	
o	o	x	o	o
	o	o	o	
o		o		o

Figure 1: LBSP 5×5 pattern for computing the binary code: ***x*** is the center pixel ***c***; ***o*** are the *P* neighbor pixels.

$$LBSP_R(x_c, y_c) = \sum_{p=0}^{P-1} d(i_p - i_c) 2^p \tag{5}$$

with

$$d(x) = \begin{cases} 1 & |x| \le T_d \\ 0 & |x| > T_d \end{cases} \tag{6}$$

$$T_d^r = T_d^0 + (|\mu_a^r - \mu_b^r|) \tag{7}$$

2.4. Multi detection fusion

A two-stage fusion process combines the change detection results from multiple detectors as shown in Figure 2. The first stage fuses changes at the pixel level, while the second stage does it at the blob level. In pixel-level fusion, the pixel-wise product is computed on the change detection heat maps $\{H^k\}$, followed by the selection of the maximum response within a neighborhood, as shown in Eq. 8, where $\odot$ denotes the pixel-wise product and w denotes the local window size for maximization operation.

$$H^f_{(i,j)} = argmax_w(H^1_{(i,j)} \odot H^2_{(i,j)} \odot \dots) \quad (8)$$

If none of the detectors generate strong enough response, then the detection probability at pixel (i, j) diminishes. The pixel-level fusion ensures a balance among multiple detectors – it helps to reduce potential false detections caused by strong response from one detector, and it helps to maintain the real changes if stronger responses arise from more detectors.

Following the pixel-level fusion, a dynamic threshold process is applied to the fused heat map $H^f_{(i,j)}$ which results in an initial binary change detection map B. The binary map is subjected to a template T_{3x3} morphological operations – erosion $B \ominus T$ followed by dilation $B \oplus T$ – to eliminate small detection sparks and to fill in holes. It is then followed by a connected component analysis and a blob split-merge process to derive the final change detection blobs.

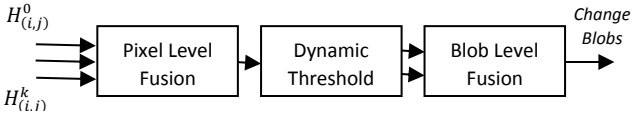

Figure 2: A two-stage fusion process to combine change detections from multiple detectors.

2.5. Classification for changes

To eliminate false detections, we have implemented an AdaBoost classifier based on Schapire and Freund's algorithm [12] with the following modifications:

1. The *cost sensitive ratio* [13] [14] is used to handle the imbalanced data set (the ratio of positive and negative samples could be well over 1:1000).
2. A set of weak classifiers are used as the decision stump with optional bi-level decision trees.

AdaBoost classifier is trained on a large labeled IR dataset of 100,000+ IR image pairs, and over 1 million detection blobs with a ratio of positive and negative samples ~ 1:1000.

For change detection blob classification, a multi-dimensional feature vector is obtained for each blob. The features are crafted such that they capture various aspects of a blob. The feature vector contains a) saliency measures from various change detectors, b) geometric measures such as size and aspect ratio, c) first and second order of pixel intensity statistics from *before* and *after* images, d) blob statistical distributions within an image, and e) local *self-similarity* measures between *before* and *after* images. It is observed that any single feature may not decisively discriminate a change as being either true or false. However, a combination of multiple features is able to serve as an efficient and reliable discriminator, as shown in Eq. 9, where x is blob feature vector, $G_m()$ is m-th weak classifier and α_m is its weight. A threshold T_A in [-0.5, 0.5] – the actual value is determined by choosing an operating point in the ROC curve (*Pd* vs. *FAR*) – is used in the final classifier decision, as shown in Eq. 10.

$$G(x) = sign\left(\sum_{m=1}^{M} \alpha_m G_m(x)\right) \quad (9)$$

$$Classifier\ decision = \begin{cases} True & if\ G(x) \geq T_A \\ False & if\ G(x) < T_A \end{cases} \quad (10)$$

3. IMPLEMENTATION

The above change detection approach has been implemented in a real-time pipeline illustrated in Figure 3. The system ingests live video and platform metadata on the input side. The pipeline initiates by finding a match of the current location indicated by the metadata in previous video collections. The system detected changes are presented to an image analyst for confirmation. The key processes include:

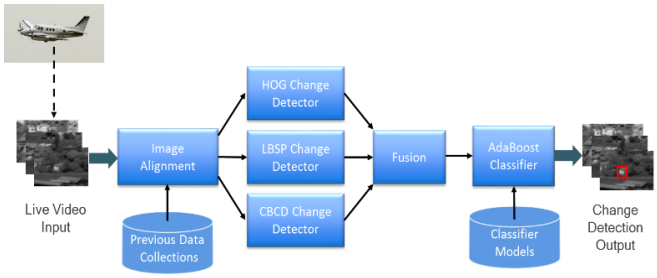

Figure 3: Real-time change detection pipeline.

Image quality assessment: In a real-world scenario, image blur and compression artifacts are common sources of image degradation. Hence, an efficient and effective blurriness and blocky-ness detection [15] is applied as a pre-filtering step.

Image pairing: For a given frame of the live video, corresponding image frames are extracted from the previous collections. These previous images must have significant overlap with the current image based on telemetry information. The sharpest among these previous images is retained for change detection pairing.

Image alignment: The image pair is aligned using a two-step process: 1) telemetry metadata is used to estimate an initial transformation; 2) fine-scale alignment is performed using HOG/ORB features selected by a RANSAC process.

Change detection: Multiple detectors are applied to an aligned pair. The detected changes are fused and are screened by the classifier for false alarms.

The multi stages of the pipeline are run in a multi-threaded environment as the live video is processed in real-time.

4. RESULTS

The training and testing of the change detection system are based on a rich set of real-world data collected at different times of the year – day and night. The classifier training and test data sets are mutually exclusive. The training set contains 42,700 image pairs of which 1353 are positive samples while 2.7 million are negative. The test set contains 102,660 image pairs and 930 nominated ground truth changes.

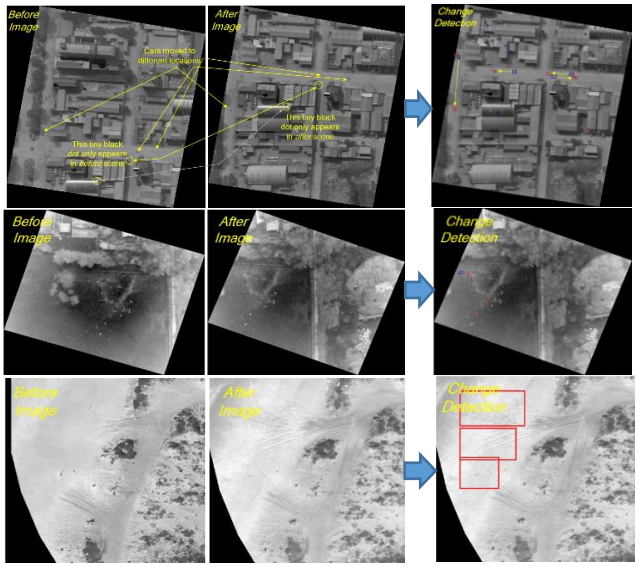

Figure 4. Examples of change detection in IR imagery.

Figure 4 illustrates a few change detection examples of IR images from a MIST-G sensor [16]. The top row images are of a city block, in which the changes caused by a few moving vehicles are identified. The middle row images are from a rural scene, in which cattle movements are picked up by the change detector. The bottom row images are from a beach area, where tire tracks are detected.

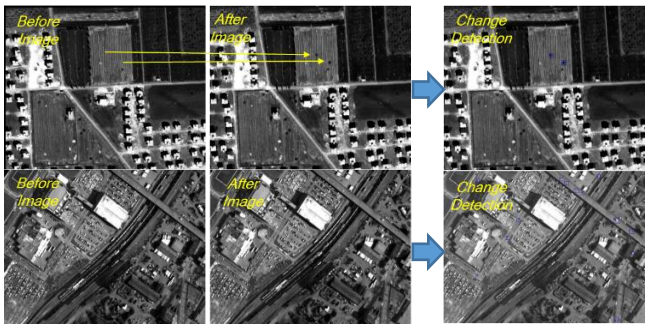

Figure 5. Change detection in satellite EO imagery

Figure 5 illustrates change detections in EO imagery. The top row, obtained from satellite data, shows detection of two craters (highlighted by blue boxes) caused by exploding ordnance. The bottom row, obtained from a high-resolution aerial reconnaissance and remote sensing imaging system, shows traffic change detections in a busy metropolitan area.

Regression tests involved more than 100,000 images. The AdaBoost models were trained on *disturbed earth* imagery with positive/negative sample ratio of 1:1000. Figure 6 compares the change detection performance of our system and other approaches. Probability of detection (P_D) is defined as the percentage of human-nominated *true* changes automatically identified by our pipeline. False alarm rate (*FAR*) is expressed as the average number of false blobs per image. The pink curve represents the detection performance of the pipeline that utilizes HOG and CBCD detectors and their fusion along with AdaBoost classifier.

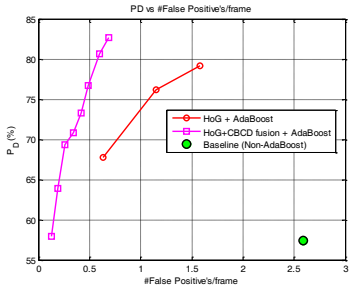

Figure 6. Change detection performance

The red curve indicates performance using HOG alone together with the AdaBoost classifier. The green circle in the lower right corner is a baseline, which uses only the HOG change detector and a simple rule-based false blob rejection method. The results demonstrate large performance gain when using multiple detectors together with fusion and AdaBoost classifier. Compared to the baseline P_D of 58% and 2.6 false blobs per image, the advanced change detection pipeline achieved a P_D of 83% and 0.6 false blobs per image.

5. CONCLUSION

In this paper, we presented a novel change detection method that uses multiple change detectors, each based on a different set of features, and a fusion process to combine the detection heat maps; it utilizes a learning-based classifier for change blob acceptance/rejection. The experimental results indicate that the method significantly improves the performance over a single detector-based approach using challenging aerial IR/EO imagery. Our on-going work explores deep-learning framework to enhance both the change detection and the classification processes without sacrificing any of the current capabilities.

6. REFERENCES

[1] Richard J. Radke, Srinivas Andra, Omar Al-Kofahi, and Badrinath Roysam. "Image Change Detection Algorithms: A Systematic Survey", *IEEE Trans. On Image Processing*, Vol. 14, No. 3, pp. 294–307, March 2005.

[2] Nil Goyette, Pierre-Marc Jodoin, Fatih Porikli, Janusz Konrad, and Prakash Ishwar, "Changedetection.net: A New Change Detection Benchmark Dataset", *IEEE Change Detection Workshop in Conjunction with CVPR*, pp. 1-8, June 2012.

[3] Yi Wang, Pierre-Marc Jodoin, Fatih Porikhli, Janusz Konrad, Yannick Benezeth, and Prakash Ishwar, "CDnet 2014: An Expanded Change Detection Benchmark Dataset", *IEEE Change Detection Workshop in conjunction with CVPR*, pp. 387-394, June 2014.

[4] R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, "Static and moving object detection using flux tensor with split Gaussian models", *IEEE Workshop on Change Detection*, *CVPR*, pp. 414-418, June 2014.

[5] P.-L. St-Charles, G.-A. Bilodeau and R. Bergevin, "Flexible background subtraction with self-balanced local sensitivity", *IEEE Workshop on Change Detection*, *CVPR*, pp. 408-413, June 2014.

[6] Bin Wang, and Piotr Dudek, "A Fast Self-tuning Background Subtraction Algorithm", *IEEE Workshop on Change Detection*, *CVPR*, pp. 395-398, June 2014.

[7] Navneet Dalal and Bill Triggs, "Histograms of Oriented Gradients for Human Detection", *IEEE Conf. on CVPR*, Vol. 1, pp. 886-893, June 2005.

[8] Yi Tan, Feng Han, and Faroog Ibrahim, "A Radar Guided Vision System for Vehicle Validation and Vehicle Motion Characterization", *IEEE Conf. on ITSC*, pp. 1059-1066, Sept. 2007.

[9] L.L. He and I. Laptev, "Robust Change Detection in Dense Urban Areas via SVM Classifier", *URBAN Remote Sensing Event*, pp. 1-5, 2009.

[10] R. Reed and X. Yu, "Adaptive multi-band CFAR detection of an optical pattern with unknown spectral distribution", *IEEE Trans. Acoustics, Speech, and Signal Process*, Vol. 38, No. 10, pp. 1760–1770, 1990.

[11] G.-A. Bilodeau, J.-P. Jodoin and N. Saunier, "Change detection in feature space using Local Binary Similarity Patterns", *IEEE International Conf. on Computer and Robot Vision*, pp. 106-112, May 2013.

[12] R. E. Schapire, Y. Freund, "Boosting: Foundations and Algorithms", *MIT Press*, 2014.

[13] Paul Viola and Michael Jones, "Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade", *Advances in Neural Information Processing System*, Vol. 2, pp. 1311-1318, 2002.

[14] Y. Sun, M. S. Kamel, A. Wong, Y. Wang, "Cost-sensitive boosting for classification of imbalanced data", *Pattern Recognition* Vol. 40, pp. 3358-3378, 2007.

[15] J. Ko and C. Kim, "Low cost blur image detection and estimation for mobile devices", *IEEE International Conference son Advanced Communication Technology*, Vol. 3, pp. 1605-1610, 2009.

[16] MIST-G: Multi-spectral Airborne Imaging System for UAVs & Manned Aircraft http://www.visionmap.com/Airborne_Imaging_Systems/126/MIST_G