Automatically Detecting Changes and Anomalies in Unmanned Aerial Vehicle Images

Automatically Detecting Changes and Anomalies in Unmanned Aerial Vehicle Images

M. Ugliano, L. Bianchi, A. Bottino, W. Allasia

***Abstract*—The use of unmanned aerial vehicles (UAVs) in civil aviation is growing up quickly, enabling new scenarios, especially in environmental monitoring and public surveillance services. So far, Earth observation has been carried out only through satellite images, which are limited in resolution and suffer from important barriers such as cloud occlusion. Microdrone solutions, providing video streaming capabilities, are already available on the marketplace, but they are limited to altitudes of a few hundred feet. In contrast, UAVs equipped with high quality cameras can fly at altitudes of a few thousand feet and can fill the gap between satellite observations and ground sensors. Therefore, new needs for data processing arise, spanning from computer vision algorithms to sensor and mission management. This paper presents a solution for automatically detecting changes in images acquired at different times by patrolling UAVs flying over the same targets (but not necessarily along the same path or at the same altitude). Change detection in multi-temporal images is a prerequisite for land cover inspection, which, in turn, sets up the basis for detecting potentially dangerous or threatening situations.**

***Index Terms*—change detection, computer vision, remote sensing, surveillance, spatiotemporal features, video signal processing, unmanned aerial vehicles.**

I. Introduction

UNMANNED aerial vehicles (UAVs), also referred to as remotely piloted aerial systems (RPAS), are quite large aircraft without a human pilot aboard. They are capable of flying at heights of a few hundred to thousand feet with many hours of autonomy and are thus well suited for the surveillance of areas of several square kilometers. Originally developed for military missions, their use in civil missions is still at the very beginning, and the civil normative rules concerning their employment are still under refinement. The research project SMAT-F2, co-funded by Regione Piemonte, tried to take advantage of such aircraft with the aim of developing an advanced environmental monitoring system.

The capability of UAVs of boarding several kilograms of payload sensors enables the adoption of many very high definition cameras, usually mounted on dedicated and stabilized gimbals. The huge amount of HD videos recorded is demanding new processing techniques and innovative methodologies, in order to provide the ground station with an intelligent support system. When a multi-sensor setting is available, the remote pilot usually pays attention to the front camera only. Thus, other video streams must be automatically processed in order to highlight potential frames and scenes which may contain anomalies, that can be roughly defined as situations arising from changes in the scene compared to previously acquired videos at the same position. Automatic detection of such changes allows to capture the attention of the operator on situations that might require further human evaluation.

This paper introduces a first attempt for automatically detecting changes from high definition videos recorded from UAVs flying over the same area during different patrolling missions. Clearly, albeit important ones, anomaly and change detection are only pre-processing steps for subsequent operations, such as tracking, classification or estimation of the change.

The processing flow is made up of several steps. First of all, exploiting the telemetry information, candidate images from previous missions are extracted. Thresholds and ranges have been established for getting the widest overlapping areas. Then, image registration is performed, followed by image enhancing steps such as color correction. In order to be able to quickly detect main changes and limit false negatives, we applied a texture descriptor providing compound information of image sub-regions. In change detection contexts, this kind of descriptors have already been used in [1] and [2], where edge-based descriptors have been adopted to analyze textural dissimilarity. In [3] Edge Histogram Descriptors served to the same purpose of image pre-processing for quickly purging false signals, allowing the use of a refinement algorithm on a small number of Regions of Interest (ROI).

This paper is structured as follows. Section II gives an overview of the related works. Section III describes the data set employed and the pre-processing steps. The algorithm for change detection is described in Section IV, while Section V provides the related results. Eventually, Section VI wraps up the proposed methodology and introduces the future work.

II. Related Works

To date, many algorithms have been developed to automatically detect changes in images taken at different times. Such algorithms perform well for some classes of

The presented work has been partially supported by SMAT-F2, a research project co-funded by Regione Piemonte according to the call for proposal POR FESR 2007/2013, linea di attività I.1.1. "Piattaforme innovative" AEROSPAZIO FASE II. The authors want to thank Selex ES Caselle Plant [23] for their useful support and their synthetic aerial images dataset availability.

M. Ugliano, L. Bianchi and W. Allasia are with the R&D Department, EURIX Group, via Carcano 26, 10153, Torino, Italy (e-mail: ugliano@eurixgroup.com, bianchi@eurixgroup.com, allasia@eurixgroup.com).

A. G. Bottino is with the Department of Control and Computer Engineering, Politecnico di Torino, corso Duca degli Abruzzi 24, 10129, Torino, Italy (e-mail: andrea.bottino@polito.it).

images, but there is no single algorithm that seems to be able to simultaneously address all the key challenges that accompany real-world (non-synthetic) videos.

Many works focus their attention on the issue of building recognition from aerial images, such as those presented in [4], [5], and [6]. Reference [4] uses SIFT features [7] and changed line segments to detect buildings which were not present in the reference image. In [5] a survey on building detection in aerial and satellite images is provided, pointing out how a Marked Point Process framework is suitable for building extraction. Reference [6] presents a detailed damage assessment on an individual building basis, which makes use of supervised classification.

Many efforts for detecting changes or damages in buildings, such as [1], [2], [8], and [9], are pixel based and only provide information regarding patches of the images which could have changed. Other works exploit the Light Detection and Ranging (LiDAR) data, from which the Digital Elevation Model (DEM) of surfaces is derived and fused onto corresponding aerial images [10]. In [11] high resolution aerial images are processed with a kernel partial least squares method for features correlation.

Another field where change detection algorithms are being actively developed is that of object recognition in videos recorded from fixed cameras in urban areas. In that context, algorithms have to focus on the issue of background identification and subtraction. In [12], an innovative Pixel-Based Adaptive Segmenter (PBAS) models the background through the history of recently observed pixel values. Improved Local Binary Patterns (LBP) have been exploited in [13]; further improvement of this method lead to the development of SuBSENSE (Self-Balanced SENsitivity Segmenter), an algorithm based on pixel-level change detection using comparisons of colors and Local Binary Similarity Pattern (LBSP) features [14][15]. Other researchers tried to exploit the integration of remote sensing data together with GIS techniques in order to analyze and classify the changing patterns of lands during a long time period.

Originally adopted in military context, nowadays UAVs are going to be employed in civil environments, especially in surveillance services. They can provide many more images for evaluating and assessing algorithms compared to the few that have been available so far, with the important advantage that such images rarely suffer from cloud occlusion, often experienced by satellite acquisition. At the same time, the use of UAV missions to record surveillance images introduces new challenges. During different missions, aircraft path and attitude may change, requiring additional processing of the images in order to compare them with previously recorded ones. Furthermore, different on-board instruments may record information at different frequencies (e.g., typically videos are recorded at 25 frames per second, while information concerning the aircraft position and attitude are recorded only once or twice per second), and additional steps are required in order to put together data from different sensors.

Change detection in videos streamed by UAVs is usually performed by human operators at their ground station, an operation which is time consuming, prone to mistakes and does not take into account the archive of images from previous missions. The aim of this work is to provide a support system to the ground station operator during the UAV mission.

III. Image Pre-Processing

When comparing images acquired by UAVs during different missions, one must take into account that the aircraft position and attitude may change between one mission and the other. Therefore, a preliminary step is required to match the newly acquired images with those recorded during previous missions. We describe the procedure that we adopted, together with the characteristics of the data set employed, in Section III.A.

Furthermore, the images have been acquired not only at different times, but also in different conditions. Changes in the weather condition, in the global illumination, in the sensors used and in the camera position and orientation are heavily affecting the application of any change detection algorithm. Thus, different pre-processing steps are required in order to reduce the errors in the algorithm output. In particular, we focused on two processes, namely image registration, described in Section III.B, and color balancing, described in Section III.C.

A. Data Sources and Image Matching

The images that we used in this paper come from synthetic videos of UAV missions which were provided by the Caselle plant of Selex ES [23]. The videos are compressed in the MPEG-2 transport stream format (MPEG-2 TS) [24], which is a standard container format for transmission and storage of audio, video and program-specific data, embedded into a multitrack stream. Information about the aircraft flight plan and attitude is provided in the MPEG-2 TS metadata track and is compliant with the standard defined in the NATO STANAG 4609 [25]. This metadata format is currently widely adopted within military environments and recently started to be employed in civil aviation as well. Acquisition frequency of the several payload sensors is different: attitude from IMU device, position from GPS and video stream from cameras are running at different speeds that in our case implies different frequencies of video frames and metadata in the MPEG-2 TS. In our experimentation sample set there is one set of metadata every 10-15 frames, but this ratio can vary between different videos. Therefore, associating the correct GPS coordinates to each image and finding matching images between videos acquired at different times is not a trivial task.

We proceed in the following way. First, we extract the individual frames and the metadata sets from the MPEG-2 TS video file. Then, since the frequency of images and metadata is different, we associate the same coordinates to all the frames between one metadata set and the next. Finally, we search through videos acquired during previous missions for all the images which were taken in the same area. Once we have selected a group of possible matches, we can perform image registration.

B. *Image Registration*

Image registration is a necessary pre-requisite for any change detection algorithm and it consists of transforming two images into a common reference system. The relevance of this step is obvious, since misregistration necessarily leads to incorrect results (i.e., false change area detected in the images). However, image registration is a computationally demanding task; therefore, it is important to limit the registration step to pairs of images acquired in the same area. This is done through the image matching procedure described in Section III.A.

We have implemented a standard image registration process, based on the identification of correspondences between the images and then, based on such correspondences, on the computation of an affine transformation matrix which aligns the two images. In details, we first detect and extract local features from each image by means of the Oriented BRIEF (ORB) local image descriptor [18]. These features are then matched using an approximate k-Nearest Neighbor search algorithm (FLANN [19]) combined with a constrained random sampling consensus (RANSAC [20]) algorithm for discarding outliers. The result of the point-pair subset selection is the best matching of interest points and can be used to compute the final homography required to align images.

For each frame in a given video, we compute registration with all the candidate matches found in the previous step (see Section III.A) and we select the best match, i.e. the one with the widest overlapping area.

C. *Color Balancing*

Automatic color balancing is the process of transferring the color characteristics of a source image to a target image in order to reduce the illumination and chromatic differences between them. Such differences can be related to several factors, such as the meteorologic conditions, different positions of the sun, different sensors used to acquire the images and difference in the viewing directions that, in turn, reflects into different component of the light reflected by objects in the scene.

The approach to color balancing used in this work stems from the method presented in [17]. This algorithm has been specifically developed to transfer color between geometrically aligned multi-temporal images. Such images can contain both changed and unchanged regions and the aim of the method is to transfer local image characteristics in the unchanged regions and global image characteristics in changed regions, smoothing the transition between regions where local and global characteristics are transferred in order to avoid possible artifacts.

Color transfer works as follows. Consider an input image t that should be transformed into a new image t^{new} after transferring the color information from a source image s. First, the RGB images are transformed into the CIE Lab color space, in order to decorrelate the color channels. Then, for each image pixel, we apply a local color transfer according to the following formula:

$$t^{\text{new}}(\mathrm{i},\mathrm{j}) = \mu^{k}_{s(\mathrm{i},\mathrm{j})} + \frac{\sigma^{k}_{s(\mathrm{i},\mathrm{j})}}{\sigma^{k}_{t(\mathrm{i},\mathrm{j})}} \left(t(\mathrm{i},\mathrm{j}) - \mu^{k}_{t(\mathrm{i},\mathrm{j})} \right) \quad (1)$$

where *(i,j)* are the coordinates of the current pixel, μ_t^k and μ_s^k are the means of the target and source image in a neighbourhood of size k x k around the pixel and σ_t^k and σ_s^k are the standard deviations of the source and target images in the same window. The algorithm is adaptive, since the value of k is optimized per pixel. The optimal value is the one in an interval $[k_{min}, k_{max}]$ that maximizes the normalized cross correlation between the corresponding regions surrounding the pixel in the source and target images.

An example of the application of the color balancing can be seen in Fig. 1.

Fig. 1. Example application of the color balancing algorithm. Source image (top left), target image (top right), and result of application of local color transfer (bottom).

IV. Change Detection Algorithm

After the alignment and color correction steps described in the previous Section, on each pair of matched images we applied our change detection algorithm which is composed of two steps. In the first step we aim at identifying areas inside the images were there are significant changes. In order to do so, we apply the edge detection algorithm described in Section IV.A. The detection is then refined in the second step, described in Section IV.B.

A. *Edge Detection*

Our edge detection algorithm is based on the MPEG-7 Edge Histogram Descriptor [21][22]. Each RGB image is first converted into a gray scale one and divided into 16 non-overlapping blocks of equal size. Each block is further divided into a fine grid of cells and edge information is calculated in five categories: horizontal, vertical, 45°, 135° and non-oriented. Each block in the target image is compared with the corresponding one in the reference image: if the difference in the total number of edges is greater than a given threshold t_E, the block is considered changed. At this stage we prefer to adopt a relatively low threshold (usually a few units), in order to avoid discarding too many real changes. This means that we

retain a higher number of false positives, which will be discarded in the next step. An example of the impact of different values for t_E is shown in Fig. 2.

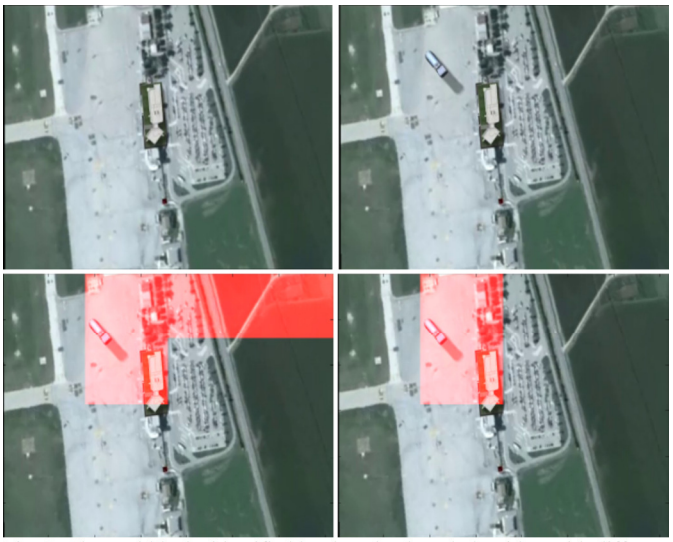

Fig. 2. Changed blocks identified by the edge-based algorithm with different thresholds. *Top left:* reference image. *Top right:* target image. *Bottom left:* changed blocks, $t_E = 5$. *Bottom right:* changed blocks, $t_E = 10$. (Images from Set A described in Section V, courtesy of Selex ES [23].)

B. *Feature Extraction*

The second step of our algorithm has the purpose of refining the results of the first step, discarding false positives and localizing the exact position of the change inside the image. First, we compute the difference of the two images. Since we applied color correction, we removed most differences in the background and illumination, thus relevant changes in the images are highlighted. In order to localize such changes, we compute once more ORB-based features, but this time we only extract them in the regions that were flagged as changed by the first step of the algorithm. If we find that a significant fraction of the features are clustered in a small area, we consider this as a real change.

Once we have localized the changes, we try to reduce the number of false positives according to the following criterion. Since we typically have many images (about 10) for the same target and we are not searching for fast-moving objects but rather permanent changes, we expect that any real change will appear in many images of our sample set. Therefore, we define a new threshold, t_F, representing the minimum percentage of frames with the same coordinates where the change must appear in order to consider it a true change. We define t_F as a percentage because in our sample set the number of images corresponding to the same coordinates is not constant, therefore a fixed threshold would need to be adjusted for each group of images.

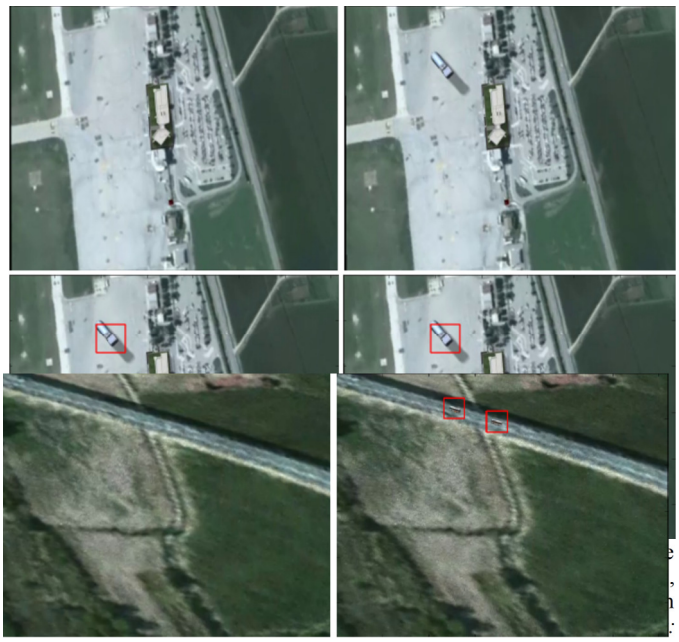

Fig. 4. Example of change detection in a pair of images from Set B (see Section V). *Left:* reference image. *Right:* target image. Detected changes are marked with red squares. (Images courtesy of Selex ES [23].)

V. Results and Discussion

We have tested our algorithm on two sets of images, which we will indicate as 'Set A' and 'Set B'. Each set consists of two synthetic videos which simulate different flights over the same targets. Each video consists of about 1500 frames with a resolution of 720x576 pixels; artificial changes were introduced between the two flights: one truck in Set A, and two buses in Set B. The change appears in about 10% of the frames. Information about the flight plan and the camera field of view is encoded in the video metadata as described in Section III.A.

Since the videos come from different flights, we applied the pre-processing steps described in Section III before attempting to identify changes. Afterwards, we have applied the change detection algorithm.

We have tested the robustness of our algorithm against the variation of the two thresholds t_E and t_F defined in Section IV. Below we will discuss the results obtained.

A. *Set A*

After the pre-processing steps described in Section III, set A is composed of 1175 pairs of registered images. Out of these, 277 image pairs contain changes (one truck was added to the scene in one of the two videos, see Figs. 2 and 3), while the remaining 898 image pairs do not contain changes. Thus, the number of real positives in the sample is 277, while the number of real negatives is 898. We ran different cases with increasing values for the most relevant thresholds and we computed different metrics for evaluating the performance of our algorithm. Overall, the algorithm was able to find the change in a good fraction of the images. As expected, increasing the thresholds leads to increasing precision (less false positives), but also to lower recall. We show the values of precision and recall obtained for the different cases in Fig. 5 and we list them, together with other evaluation metrics, in Table I.

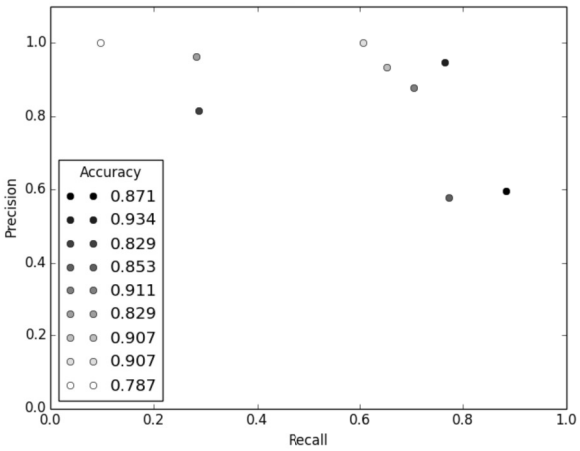

Fig. 5. Precision as a function of recall for the different cases computed for Set A. Different gray values indicate different values of accuracy.

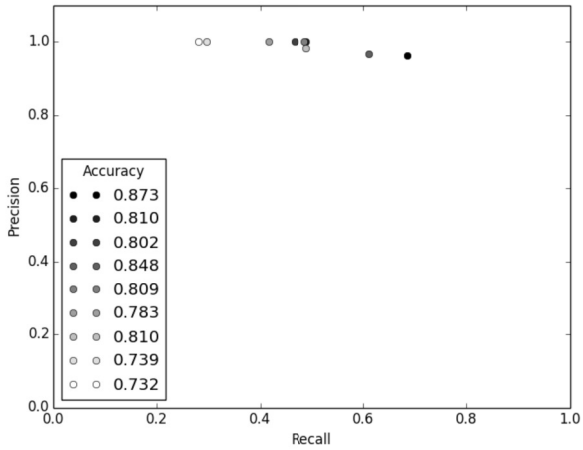

Fig. 6. Precision as a function of recall for the different cases computed for Set B. Different gray values indicate different values of accuracy.

TABLE I
STATISTICS OF SET A

t_E	t_F	Accuracy	Precision	Recall	F-measure
5	0 %	0.871	0.595	0.884	0.711
5	50 %	0.934	0.946	0.765	0.846
5	100 %	0.829	0.816	0.286	0.424
8	0 %	0.853	0.577	0.773	0.661
8	50 %	0.911	0.878	0.704	0.781
8	100 %	0.829	0.963	0.282	0.436
10	0 %	0.907	0.933	0.653	0.768
10	50 %	0.907	1.000	0.606	0.755
10	100 %	0.787	1.000	0.097	0.177

Statistics of Set A for different values of the two thresholds t_E and t_F defined in Section IV.

TABLE II
STATISTICS OF SET B

t_E	t_F	Accuracy	Precision	Recall	F-measure
1	0 %	0.873	0.962	0.685	0.800
1	10 %	0.810	1.000	0.489	0.657
1	20 %	0.802	1.000	0.467	0.637
2	0 %	0.848	0.966	0.612	0.749
2	10 %	0.809	1.000	0.486	0.654
2	20 %	0.783	1.000	0.417	0.589
3	0 %	0.810	0.982	0.489	0.653
3	10 %	0.739	1.000	0.297	0.458
3	20 %	0.732	1.000	0.281	0.439

Statistics of Set B for different values of the two thresholds t_E and t_F defined in Section IV.

B. Set B

After the pre-processing steps described in Section III.B, set B is composed of 1208 pairs of registered images. Out of these, 276 image pairs contain changes (two buses were added to the scene, see Fig. 4), while the remaining 932 image pairs do not contain changes. Thus, the number of real positives in the sample is 552, while the number of real negatives is 932. Since the two buses are significantly smaller than the truck of Set A, we had to increase the number of cells for the edge detection and use lower values for the thresholds in order to detect them. We list the different cases computed, together with evaluation metrics, in Table II and we show the values of precision and recall in Fig. 6.

VI. CONCLUSIONS AND OUTLOOK

We have presented a new method for automatically detecting changes in aerial images recorded from high resolution cameras on unmanned aerial vehicles (UAVs). We have tested our method on synthetic videos provided by Selex ES [23] and the proposed algorithm is capable of finding even small changes with quite good confidence levels. Our next goal is to extend the algorithm in order to allow automatic classification of identified changes. The adoption of UAVs in civil aviation will require novel dedicated software in order to provide the operator at the ground station with a support system, either for detecting changes or for capturing specific information from the embarked sensor suite. We believe that change detection as well as other image processing capabilities are a new potential business to be investigated and exploited by software vendors and providers in the current aerial marketplace.

REFERENCES

[1] B. J. Adams, C. K. Huyck, B. Mansouri, R. T. Eguchi, and M. Shinozuka, "Application of High-Resolution Optical Satellite Imagery for Post-Earthquake Damage Assessment: The 2003 Boumerdes (Algeria) and Bam (Iran) Earthquakes," MCEER, Buffalo, NY, USA, 2004.

[2] M. Matsuoka, T. T. Vu, and F. Yamazaki, "Automated damage detection and visualization of the 2003 Bam, Iran earthquake using high-resolution satellite images," 25th Asian Conf. Remote Sens., Chiang Mai, Thailand, 2004.

[3] W. Allasia, F. Gallo, M. Ferraro, M. Vigilante, and C. Culeddu, "Automatic Video Processing for Traffic Sign Recognition," The Fifth IASTED European Conference on Internet and Multimedia Systems and Applications, Cambridge, United Kingdom, Jul. 2009.

[4] W. Li, X. Li, Y. Wu, and Z. Hu, "A novel framework for urban change detection using vhr satellite images," 18th International Conference on Pattern Recognition, 2006.

[5] C. Benedek, X. Descombes, and J. Zerubia, "Building Extraction and Change Detection in Multitemporal Aerial and Satellite Images in a Joint Stochastic Approach," *Research Report RR-7143*, 00426615v3, 2009.

[6] J. Thomas, A. Kareem, and K. Bowyer, "Automated poststorm damage classification of low-rise building roofing systems using high-resolution aerial imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 7, pp. 3851–3861, July 2014.

[7] D. G. Lowe, "Object recognition from local scale-invariant features," International Conference on Computer Vision, Washington, DC, USA, Volume 2, pp. 1150–, 1999.

[8] C. F. Barnes, H. Fritz, and J. Yoo, "Hurricane disaster assessments with image-driven data mining in high-resolution satellite imagery," *Proc. IEEE Trans. Geosci. Remote Sens.*, pp. 1631–1641, 2007.

[9] V. Vijayaraj, E. Bright, and B. Bhaduri, "Rapid damage assessment from high resolution imagery," *Proc. IEEE Geosci. Remote Sens. Symp.*, pp. 499–502, 2008.

[10] J.C. Trinder and M. Salah, "Aerial images and LiDAR data fusion for disaster change detection," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume I-4, 2012.

[11] K. Zong, A. Sowmya, and J. Trinder, "Kernel partial least squares based hierarchical building change detection using high resolution aerial images and lidar data," International Conference on Digital Image Computing: Techniques and Applications (DICTA), 3 – 978-1-4799-2126, 2013.

[12] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, June 2012, pp. 38–43. 5

[13] P.-L. St-Charles and G.-A. Bilodeau, "Improving background subtraction using local binary similarity patterns," IEEE Winter Conference on Applications of Computer Vision (WACV), March 2014, pp. 509–515.

[14] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Subsense: A universal change detection method with local adaptive sensitivity," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 359–373, Jan 2015.

[15] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "Flexible background subtraction with self-balanced local sensitivity," IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2014, pp. 414–419.

[16] Carmelo Riccardo Fichera, Giuseppe Modica and Maurizio Pollino, "Land Cover classification and change-detection analysis using multi-temporal remote sensed imagery and landscape metrics," *European Journal of Remote Sensing,* 2012, 45, 10.5721/EuJRS20124501

[17] J. Thomas, K.W. Bowyer and A. Kareem, "Color balancing for change detection in multitemporal images," IEEE Workshop on Applications of Computer Vision 2012, pp. 385-390

[18] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, "ORB: an efficient alternative to SIFT or SURF," IEEE International Conference on Computer Vision 2011, pp. 2564-2571

[19] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," International Conference on Computer Vision Theory and Applications 2009

[20] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Comm. of the ACM 24(6), 1981, pp. 381-395

[21] MPEG-7: ISO/IEC 15938-3 Information technology – multimedia content description interface – Part 3: Visual, March 2003

[22] T. Sikora, "The MPEG-7 Visual Standard for Content Description - An Overview," *IEEE Trans. Circuits Syst. Video Techn.*, 11(6):696–702, 2001

[23] Selex ES – website: selex-es .com - Last visited: 25 May 2015

[24] ISO/IEC 13818-1, ITU-T Recommendation H.222.0

[25] Website: nato.int/structur/ac/224/standard/4609/4609.htm – Last visited: 25 May 2015