Automated X-ray object recognition using an efficient search algorithm in multiple views

Domingo Mery, Vladimir Riffo, Irene Zuccar, Christian Pieringer
Department of Computer Science – Pontificia Universidad Católica de Chile
Av. Vicuña Mackenna 4860(143) – Santiago de Chile
dmery@ing.puc.cl http://dmery.ing.puc.cl

Abstract

In order to reduce the security risk of a commercial aircraft, passengers are not allowed to take certain items in their carry-on baggage. For this reason, human operators are trained to detect prohibited items using a manually controlled baggage screening process. In this paper, we propose the use of an automated method based on multiple X-ray views to recognize certain regular objects with highly defined shapes and sizes. The method consists of two steps: 'monocular analysis', to obtain possible detections in each view of a sequence, and 'multiple view analysis', to recognize the objects of interest using matchings in all views. The search for matching candidates is efficiently performed using a lookup table that is computed off-line. In order to illustrate the effectiveness of the proposed method, experimental results on recognizing regular objects –clips, springs and razor blades– in pen cases are shown achieving around 93% accuracy for 120 objects. We believe that it would be possible to design an automated aid in a target detection task using the proposed algorithm.

1. Introduction

The ability to automatically and robustly recognize objects can be critical for many applications such as surveillance, video forensics, X-ray testing and medical image analysis for computer-aided diagnosis, to mention just a few. Our paper is dedicated to *X-ray object recognition*. As X-ray images are taken under controlled conditions, X-ray object recognition may be considered as an "easy to solve" problem in comparison with other computer vision problems related to the real world under uncontrolled conditions (*e.g*. people detection [7] or scene recognition [26]), however, this is not the case of some applications such as baggage screening, for example, where computer vision techniques are still not effective enough to be used without human interaction [29].

Even though several scientific communities are exploring a range of research directions, adopting very different principles, and developing a wide variety of algorithms for very different applications, automated X-ray object recognition remains an open question due to: *i)* the large variability of the appearance and shape of the test objects –both between and within categories–; *ii)* the large variability in terms of object sample depending on its points of view (*e.g*. top view and frontal view of a *razor blade* are very different as shown in Fig. 1); and *iii)* the appearance of a test object can vary due to the conditions of (self–)occlusion, noise and acquisition.

In our paper, we would like to make a contribution to the last two mentioned problems, in which object recognition plays a crucial role. We have based our proposal on three potent ideas: *i) detection windows*, as they obtain a high performance in recognition and detection problems in computer vision; *ii) multiple views*, as they can be an effective option for examining complex objects where uncertainty by analyzing only one angle of perspective can lead to misinterpretation; and *iii) efficient visual search*, given the speeds involved when searching for objects. We believe that our framework is a useful alternative for recognizing objects because it is based on an efficient *search* in *multiple views* using *corresponding multiple view windows*.

In this paper, we propose a *framework* based on computer vision and machine learning techniques in order to

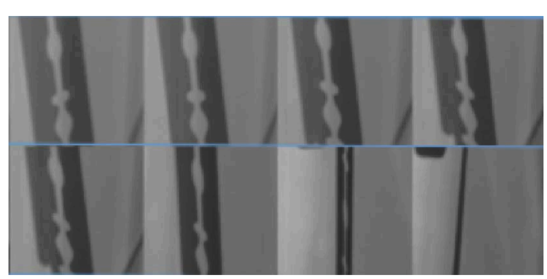

Figure 1. Large variability within a *razor blade*: some X-ray images of the same blade in different poses.

deal with the problem of 3D recognition. We believe that this solution also allows us to propose a general and adaptive methodology for X-ray testing that can be tested in several detection problems, such as the characterization of materials, and airport security. Additionally, we think that it would be possible to design an automated aid in a target detection task using the proposed algorithm.

The rest of the paper is organized as follows: the state of the art (Section 2), the proposed approach (Section 3), the results obtained in several experiments (Section 4), and some concluding remarks and suggestions for future research (Section 5).

2. State of the Art

Three-dimensional (3D) recognition from two-dimensional (2D) images is a very complex task due to the infinite number of points of views and different image acquisition conditions [22]. Nevertheless, automated recognition has been possible –in certain cases– through seminal works dedicated to obtaining highly discriminative and local invariant features related to illumination factors and local geometric constraints (see for example [17] for a good review and evaluation). In such cases, recognition of a test object can be performed by matching its invariant features with the features of a model.

Since the September 11 attacks, automated (or semi-automated) 3D recognition using X-ray images has become a very important element in baggage screening. The inspection process, however, is complex, basically because threatening items are very difficult to detect when placed in close-packed bags, superimposed by other objects, and/or rotated showing an unrecognizable view [30]. In baggage screening, where human security plays an important role and inspection complexity is very high, human inspectors are still used. Nevertheless, during peak hours in airports, human screeners have only a few seconds to decide whether a bag contains or not a prohibited item, and detection performance is only about 80-90% [16]. Before 9/11, the X-ray analysis of luggage mainly focused on capturing the images of their content: the reader can find in [18] an interesting analysis carried out in 1989 of several aircraft attacks around the world, and the existing technologies to detect terrorist threats based on Thermal-Neutron Activation (TNA), Fast-Neutron Activation (FNA) and dual energy X-rays (used in medicine since the early 70s). In the 90s, Explosive Detection Systems (EDS) were developed based on X-ray imaging [19], and computed tomography through elastic scatter X-ray (comparing the structure of irradiated material, against stored reference spectra for explosives and drugs) [25]. All these works were concentrated on image acquisition and simple image processing; however, they lacked advanced image analysis to improve detection performance. Nevertheless, the 9/11 attacks increased the security measures taken at airports, which in turn stimulated the interest of the scientific community in the research of areas related to security using advanced computational techniques. Over the last decade, the main contributions were: analysis of human inspection [29], pseudo-coloring of X-ray images [1, 4], enhancement and segmentation of X-ray images [24] and detection of threatening items in X-ray images, based on texture features (detecting a 9mm Colt Beretta automatic (machine) pistol) [21], neural networks and fuzzy rules (yielding about 80% of performance) [10], and SVM classifier (detecting guns in real time) [20].

In baggage screening, the use of multiple view information yields a significant improvement in performance as certain items are difficult to recognize using only one viewpoint. As reported in a study that measures the human performance in baggage screening [28], (human) multiple view X-ray inspection leads to a higher detection performance of prohibited items under difficult conditions, however, there are no significant differences between the detection performance (single vs. multiple view) for difficult-easy multiple view conditions, *i.e.* two *difficult* or two *easy* views are redundant. We observed that for intricate conditions, multiple view X-ray inspection is required.

Recently, some algorithms based on multiple X-ray views were reported in the literature. For example: synthesis of new X-ray images obtained from Kinetic Depth Effect X-ray (KDEX) images based on SIFT features in order to increase detection performance [2]; an approach for object detection in multi-view dual-energy X-ray with promising preliminary results [8]; X-ray active vision that is able to adequate the viewpoint of the target object in order to obtain better X-ray images to analyze [23]; and tracking across multiple X-ray views in order to verify the diagnoses performed using a single view [13, 14].

In the literature review, we observed that there are few papers on 3D recognition with multiple X-ray views. This paper wishes to contribute to this field.

3. Proposed Method

In this section, we explain in further detail the proposed method. The strategy consists of two main stages: *off-line* and *on-line*.

3.1. Off-line stage

The first stage, performed off-line, consists of two main steps: *i)* learning a model that is used for the recognition and *ii)* estimation of a multiple view geometric model that is used for data association.

3.1.1 Learning

In this step, we learn a classifier h to recognize parts of the objects that we are attempting to detect. It is assumed that there are $C + 1$ classes (labeled as '0' for non-object class, and '1', '2', ... 'C' for C different objects). Images are taken of representative objects of each class from different points of view. In order to model the details of the objects from different poses, several keypoints per image are detected, and for each keypoint a descriptor $\mathbf{d}$ is extracted using, for example, LBP, SIFT, HOG, and SURF, among others [17]. In this supervised approach, each descriptor $\mathbf{d}$ is manually labeled according to its corresponding class $c \in \{0, 1, \ldots C\}$. Given the training data $(\mathbf{d}_t, c_t)$, for $t = 1, \ldots, N$, where N is the total number of descriptors extracted in all training images, a classifier h is designed which maps $\mathbf{d}_t$ to their classification label c_t, thus, $h(\mathbf{d}_t)$ should be c_t. This classifier will be used in the on-line stage by monocular and multiple-view analysis.

3.1.2 Geometry

Our strategy deals with multiple monocular detections in multiple views. In this problem of data association, the aim is to find the correct correspondence among different views. For this reason, we use multiple view geometric constraints to reduce the number of matching candidates between monocular detections. For an image sequence with n views $\mathbf{I}_1 \ldots \mathbf{I}_n$, the fundamental matrices $\{\mathbf{F}_{ij}\}$ between consecutive frames $\mathbf{I}_i$ and $\mathbf{I}_{j=i+1}$ are computed for $i = 1, \ldots n-1$. In our approach, the fundamental matrix $\mathbf{F}_{ij}$ is calculated from projection matrices $\mathbf{P}_i$ and $\mathbf{P}_j$ that can be estimated using calibration or bundle adjustment algorithms [9].

The geometric constraints are expressed in homogeneous coordinates. Therefore, given a point $\mathbf{m}_i = [x_i \; y_i \; 1]^\mathsf{T}$ in image $\mathbf{I}_i$, a corresponding point $\mathbf{m}_j = [x_j \; y_j \; 1]^\mathsf{T}$ in image $\mathbf{I}_j$ must fulfill: *i)* epipolar constraint: $\mathbf{m}_j$ must lie near the epipolar line $\ell = \mathbf{F}_{ij}\mathbf{m}_i$, and *ii)* location constraint: for small variations of the point of views between $\mathbf{I}_i$ and $\mathbf{I}_j$, $\mathbf{m}_j$ must lie near $\mathbf{m}_i$. Thus, a candidate $\mathbf{m}_j$ must fulfill:

$$\frac{|\mathbf{m}_j^\mathsf{T}\mathbf{F}_{ij}\mathbf{m}_i|}{\sqrt{\ell_1^2+\ell_2^2}} < e \quad \text{and} \quad ||\mathbf{m}_i - \mathbf{m}_j|| < r. \qquad (1)$$

In order to accelerate the search of candidates, we propose the use of a lookup table as follows: Points in images $\mathbf{I}_i$ and $\mathbf{I}_j$ are arranged in a grid format with rows and columns. For each grid point (x, y) of image $\mathbf{I}_i$, we look for the grid points of image $\mathbf{I}_j$ that fulfill (1), as illustrated in Fig. 2. Therefore, the possible corresponding points of (x, y) will be the set $\mathbf{S}_{xy} = \{(x_p, y_p)\}_{p=1}^{q}$, where $x_p = X(x, y, p)$, $y_p = Y(x, y, p)$ and $q = Q(x, y)$ are stored (off-line) in a lookup table. In the on-line stage, given a point $\mathbf{m}_i$ (in image $\mathbf{I}_i$), the matching candidates in image $\mathbf{I}_j$ are those that lie near to $\mathbf{S}_{xy}$, where (x, y) is the nearest grid point to $\mathbf{m}_i$. This search can be efficiently implemented using k-d tree structures [3].

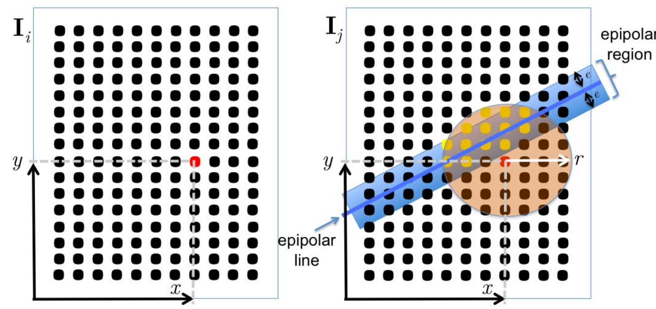

Figure 2. Given the grid point illustrated as the red point at (x, y), in image $\mathbf{I}_i$, the set of possible corresponding points in image $\mathbf{I}_j$ can be those grid points (yellow points) represented by the intersection of the epipolar region (blue rectangle) and neighborhood around (x, y) (orange circle with radius r centered at red point). The use of grid points allows us to use a lookup table in order to search the matching candidates in $\mathbf{I}_j$ efficiently.

In a controlled and calibrated environment, we can assume that the fundamental matrices are stable and we do not need to estimate them in each new image sequence, *i.e.* the lookup tables are constant. Additionally, when the relative motion of the point of view between consecutive frames is the same, the computed fundamental matrices are constant, *i.e.* $\mathbf{F}_{ij} = \mathbf{F}$, and we need to store only one lookup table.

3.2. On-line stage

The on-line stage is performed in order to recognize the objects of interest in a test image sequence of n images $\{\mathbf{I}_i\}$, for $i = 1, \ldots n$. The images are acquired by rotation of the object being tested at β degrees (in our experiments we used $n = 4$, and $\beta = 10^0$). This stage consisted of two main steps: monocular and multiple view analysis that will be described in further detail as follows.

3.2.1 Monocular Analysis

This step is performed in each image $\mathbf{I}_i$ of the test image sequence, as illustrated in Fig. 3 in a real case. The whole object contained in image $\mathbf{I}_i$ is segmented from the background using threshold and morphological operations. SIFT–keypoints –or other descriptors–, are only extracted in the segmented portion. The descriptor $\mathbf{d}$ of each keypoint is classified using classifier $h(\mathbf{d})$ trained in the off-line stage, and explained in Section 3.1.1. All keypoints classified as class c, where c is the class of interest, with $c \in \{1 \ldots C\}$ are selected. As we can see in Fig. 3 for the classification of 'razor blade', there are many keypoints misclassified. For this reason, neighbor keypoints are clustered in the 2D space using Mean Shift algorithm [5]. Only those clusters that have a large enough number of keypoints are selected. They will be called *detected monocular keypoints*.

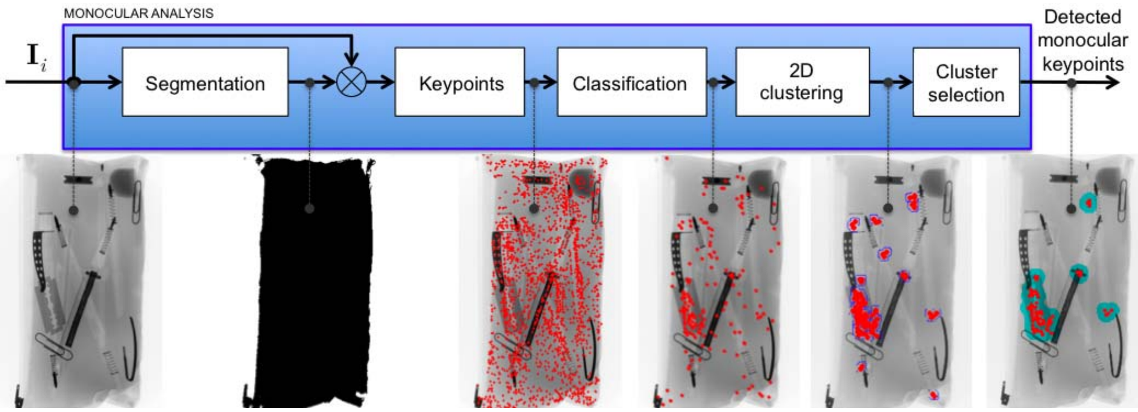

Figure 3. Monocular analysis for each image of the sequence, *i.e*. for $i = 1, \ldots n$. In this example, the class of interest is 'razor blade'.

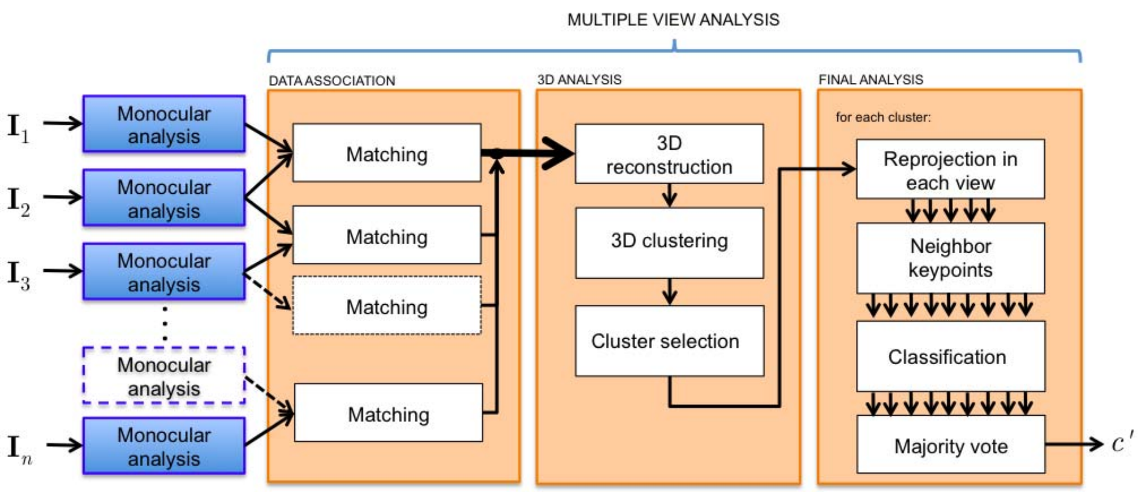

Figure 4. Multiple view analysis. An explanation of last step (final analysis) is illustrated in Fig. 5.

3.2.2 Multiple View Analysis

Multiple view analysis performs the recognition of objects of interest in three steps (see Fig. 4): *i)* data association, *ii)* 3D analysis, and *iii)* final analysis. The input is the detected monocular keypoints obtained by the mentioned monocular analysis of Section 3.2.1. The output is c', the assigned class for each detected object.

• **Data Association:** In this step, we find matchings for all detected monocular keypoints in all consecutive images $\mathbf{I}_i$ and $\mathbf{I}_{j=i+1}$, for $i = 1, \ldots n-1$, as follows:

+ For each detected monocular keypoint in image $\mathbf{I}_i$ (located at position (x_i, y_i) with descriptor $\mathbf{d}_i$), we seek in a dense grid of points, the nearest point (x, y) (see red point in Fig. 2-left) using a k-d tree structure.

+ We determine $\mathbf{S}_{xy}$, the set of matching candidates in image $\mathbf{I}_{j=i+1}$ arranged in a grid manner by reading the lookup table explained in Section 3.1.2 (see yellow points in Fig. 2-right).

+ We look for the detected monocular keypoints in image $\mathbf{I}_j$ that are located in the neighborhood of $\mathbf{S}_{xy}$, again using a k-d tree structure. They will be called *neighbor keypoints*. When no neighbor keypoint is found, no match is established for (x_i, y_i).

+ From neighbor keypoints, we select that one (located at position (x_j, y_j) with descriptor $\mathbf{d}_j$) with minimum distance $||\mathbf{d}_i - \mathbf{d}_j||$. In order to ensure the similarity between matching points, the distance should be less than a threshold ϵ. If this constraint is not satisfied, again no match is established for (x_i, y_i).

• **3D analysis:** From each pair of matched keypoints (x_i, y_i) in image $\mathbf{I}_i$ and (x_j, y_j) in image $\mathbf{I}_{j=i+1}$ established in the previous step, a 3D point is reconstructed using the projection matrices $\mathbf{P}_i$ and $\mathbf{P}_j$ of our geometric model mentioned in Section 3.1.2 (see triangulation algorithm in [9]). Similarly to the monocular detection approach, neighbor 3D points are clustered in the 3D space using Mean Shift algorithm [5], and only those clusters that have a large enough number of 3D points are selected.

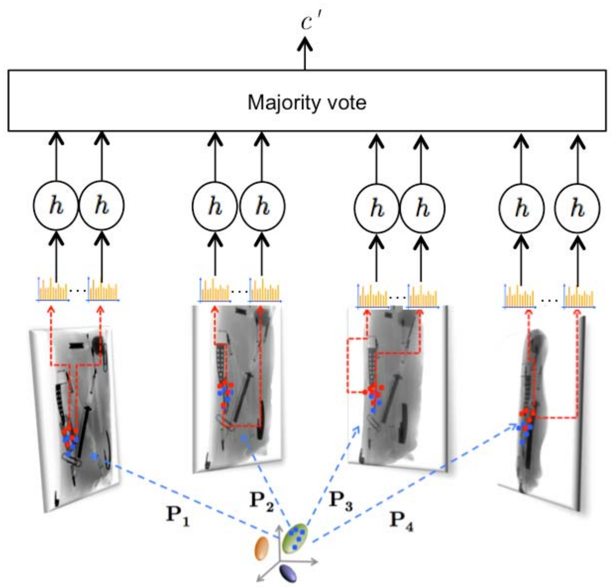

Figure 5. Final analysis: using the geometric model, the reconstructed 3D points in each cluster are reprojected in each view (blue points). The keypoints that are near to the reprojected points are identified (red points). The descriptors of these keypoints (orange histograms) are classified using trained classifier h. The class c' of this cluster is determined by majority vote. In this example of $n = 4$ views, only the green cluster is represented.

• **Final analysis:** For each selected 3D cluster, all 3D reconstructed points belonging to the cluster are re-projected onto all images of the sequence using the projection matrices of geometric model (see Fig. 5). The extracted descriptors of the keypoints located near these re-projected points are classified individually using classifier h (defined in Section 3.1.1). The cluster will be classified as class c' if there is a large number of keypoints individually classified as c', and this number represents a majority in the cluster.

This majority vote strategy can overcome the problem of false monocular detections when the classification of the minority fails. A cluster can be misclassified if the part that we are trying to recognize is occluded by a part of another class. In this case, there will be keypoints in the cluster assigned to both classes; however, we expect that the majority of keypoints will be assigned to the true class if there are a small number of keypoints misclassified.

4. Experiments and Results

In our experiments, the task was to recognize three different classes of objects that are present in a pencil case (see for example a sequence in Fig. 6a). These classes are: 'clips', 'springs' and 'razor blades'. We followed the recognition approach explained in Section 3.

In the off-line stage we used a structure from a motion algorithm in order to estimate the projection matrices of each view[1]. Additionally, in the learning phase, we used only 16 training images of each class. Due to the small intra-class variation of our classes, this number of training images was deemed sufficient. The training objects were posed at different angles. SIFT descriptors were extracted as explained in [11], and a k-Nearest Neighbor (KNN) classifier with $k = 3$ neighbors was ascertained using the SIFT descriptors of the four classes[2]. Other descriptors (like LBP and HOG) and other classifiers (like SVM or KNN with other values of k) were also tested, although the best performance was achieved with the aforementioned configuration.

In order to illustrate step by step the on-line stage, the recognition of a razor blade is illustrated in Fig. 6a–d for monocular analysis and in Fig. 6e–g for multiple view analysis[3]. It is worth mentioning that in monocular detection there are false alarms, however, they can be filtered out after multiple view analysis. The reason is because false alarms cannot be tracked in the sequence or because the tracked points, when validating the corresponding points in other views of the sequence, do not belong to the class of interest. Other results with some degree of overlap, where the task was the recognition of springs and clips, are illustrated in Fig 7.

Testing experiments were carried out by recognizing the three mentioned classes ('clips', 'springs' and 'razor blades') in 45 different sequences of 4 views (15 sequences for each class)[4]. The size of an individual image was 1430 $\times$900 pixels. In these experiments there were 30 clips, 75 springs and 15 razor blades to be recognized. A summary of the results using the proposed algorithm is presented in Table 1, in which the performance in the recognition of each class is presented in two different parts of our algorithm: after monocular analysis (Mono) and after multiple view analysis (Multi). These parts are illustrated in Fig. 6d and 6g respectively for a razor blade. In this table, ground truth (GT) is the number of existing objects to be recognized. The number of detected objects by our algorithm is D = TP + FP, including false positives (FP) and true positives (TP). Ideally, FP = 0 and TP = GT. In our experiments, precision (PR), computed as PR=TP/D, is 71.4% and 95.7% in each part; and recall (RE), computed as RE=TP/GT, is 90.8% and 92.5% in each step. If we compare single versus multiple view detection, both precision and recall are incremented. Precision, however, is drastically incremented because our approach achieves good discrimination from false alarms.

The amount of time required in our experiments was

[1]We use in our experiments a fast implementation of multiple view geometry algorithms from BALU Toolbox [12]

[2]We use in our experiments fast implementations of SIFT and KNN (based on k-d tree) from VLFeat Toolbox [27].

[3]We use in our experiments a fast implementation of Mean Shift from PMT Toolbox [6].

[4]The images tested in our experiments come from public GDXray database [15].

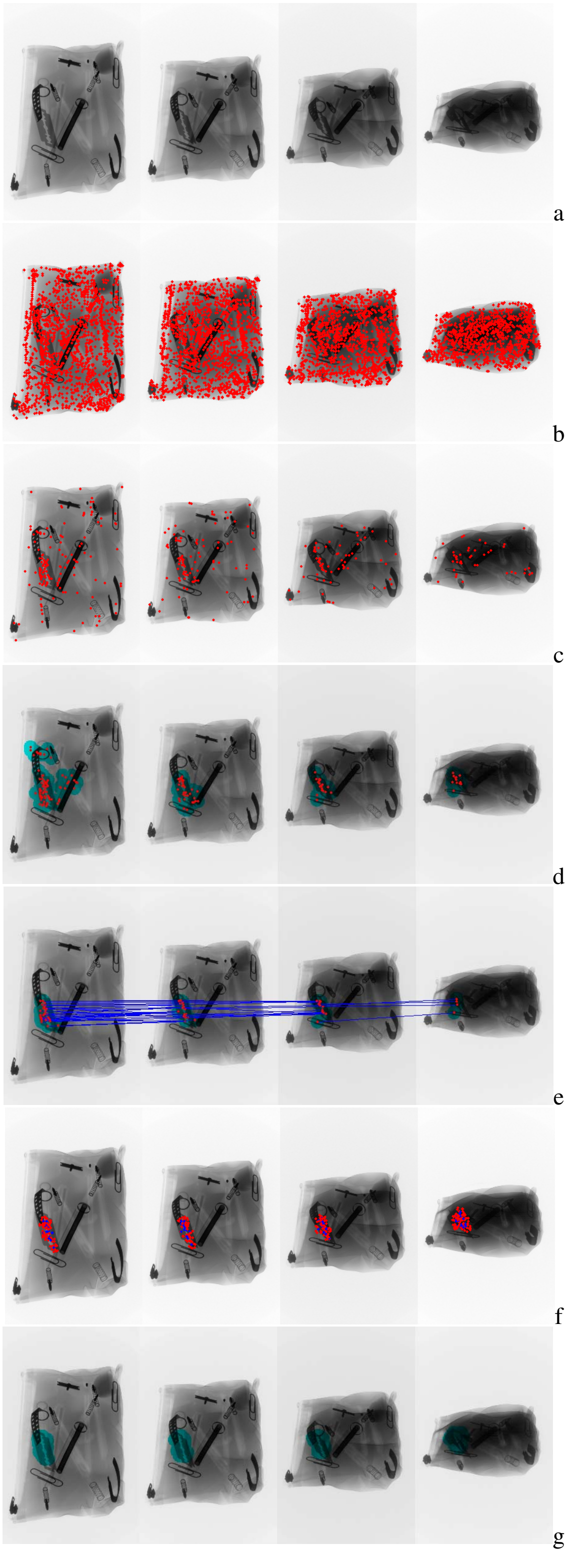

Figure 6. Recognition of a razor blade using our approach. a) original sequence, b) keypoints, c) classified keypoints, d) detected monocular keypoints, e) matched keypoints, f) reprojected 3D points (blue) and neighbor keypoins (red), g) final detection.

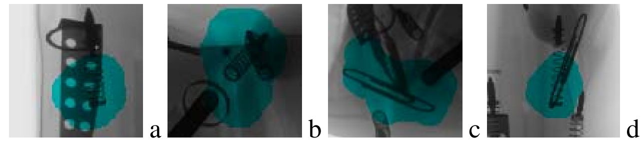

Figure 7. Recognition using our approach in cases with some degree of overlap: a) one spring, b) two springs, c) one clip, d) one clip. Each figure shows a part of one image of the whole sequence.

Table 1. Recognition performance.

	Mono			Multi		
Class	TP	FP	GT	TP	FP	GT
Clip	114	127	120	26	2	30
Spring	263	30	300	71	3	75
Blade	59	18	60	14	0	15
Total	436	175	480	111	5	120
PR[%]	71.4			95.7		
RE[%]	90.8			92.5		

about 15 minutes for the off-line stage and about 16s for testing each sequence on a iMac OS X 10.7.3, processor 3.06GHz Intel Core 2 Duo, 4GB 1067 MHz DDR3 memory. The code of the program –implemented in Matlab– is available on our web site.

5. Conclusions

In this paper, we presented a new method that can be used to recognize certain parts of interest in complex objects using multiple X-ray views. The proposed method filters out false positives resulting from monocular detection performed on single views by matching information across multiple views. This step is performed efficiently using a lookup table that is computed off-line. In order to illustrate the effectiveness of the proposed method, experimental results on recognizing regular objects –clips, springs and razor blades– in pen cases are shown achieving around 93% accuracy in the recognition of 120 objects. We believe that it would be possible to design an automated aid in a target detection task using the proposed algorithm. In our future work, the approach will be tested in more complex scenarios recognizing objects with a larger intra-class variation.

Acknowledgments

This work was supported by Fondecyt grant 1130934 from CONICYT–Chile.

References

[1] B. Abidi, Y. Zheng, A. Gribok, and M. Abidi. Improving weapon detection in single energy x-ray images through pseudocoloring. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(6):784 – 796, nov. 2006.

[2] O. Abusaeeda, J. Evans, D. D., and J. Chan. View synthesis of KDEX imagery for 3D security X-ray imaging. In *Proc. 4th International Conference on Imaging for Crime Detection and Prevention (ICDP-2011)*, 2011.

[3] J. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[4] J. Chan, P. Evans, and X. Wang. Enhanced color coding scheme for kinetic depth effect X-ray (KDEX) imaging. In *Security Technology (ICCST), 2010 IEEE International Carnahan Conference on*, pages 155 –160, oct. 2010.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[6] P. Dollár. Piotr's Image and Video Matlab Toolbox (PMT). `http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html`.

[7] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):743–761, 2011.

[8] T. Franzel, U. Schmidt, and S. Roth. Object detection in multi-view x-ray images. *Pattern Recognition*, pages 144–154, 2012.

[9] R. I. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, second edition, 2003.

[10] D. Liu and Z. Wang. A united classification system of X-ray image based on fuzzy rule and neural networks. In *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, volume 1, pages 717 –722, nov. 2008.

[11] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[12] D. Mery. BALU: A toolbox Matlab for computer vision, pattern recognition and image processing. `http://dmery.ing.puc.cl/index.php/balu`.

[13] D. Mery. Automated detection in complex objects using a tracking algorithm in multiple X-ray views. In *Proceedings of the 8th IEEE Workshop on Object Tracking and Classification Beyond the Visible Spectrum (OTCBVS 2011), in Conjunction with CVPR 2011, Colorado Springs*, pages 41–48, 2011.

[14] D. Mery, G. Mondragón, V. Riffo, , and I. Zuccar. Detection of regular objects in baggage using multiple x-ray views. *Insight*, 55(1):16–20, 2013.

[15] D. Mery, V. Riffo, G. Mondragón, I. Lillo, I. Zuccar, and H. Löbel. GDXray – the GRIMA database of X-ray images. `http://dmery.ing.puc.cl/index.php/material/gdxray/`. Department of Computer Science, Universidad Católica de Chile.

[16] S. Michel, S. Koller, J. de Ruiter, R. Moerland, M. Hogervorst, and A. Schwaninger. Computer-based training increases efficiency in X-Ray image interpretation by aviation security screeners. pages 201–206, Oct. 2007.

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[18] E. Murphy. A rising war on terrorists. *Spectrum, IEEE*, 26(11):33 –36, nov 1989.

[19] N. Murray and K. Riordan. Evaluation of automatic explosive detection systems. In *Security Technology, 1995. Proceedings. Institute of Electrical and Electronics Engineers 29th Annual 1995 International Carnahan Conference on*, pages 175 –179, oct 1995.

[20] S. Nercessian, K. Panetta, and S. Agaian. Automatic detection of potential threat objects in X-ray luggage scan images. In *Technologies for Homeland Security, 2008 IEEE Conference on*, pages 504 –509, may 2008.

[21] C. Oertel and P. Bock. Identification of objects-of-interest in X-Ray images. In *Applied Imagery and Pattern Recognition Workshop, 2006. AIPR 2006. 35th IEEE*, page 17, oct. 2006.

[22] T. Poggio and S. Edelman. A network that learns to recognize 3d objects. *Nature*, 343(6255):263–266, 1990.

[23] V. Riffo and D. Mery. Active X-ray testing of complex objects. *Insight*, 54(1):28–35, 2012.

[24] M. Singh and S. Singh. Optimizing image enhancement for screening luggage at airports. In *Computational Intelligence for Homeland Security and Personal Safety, 2005. CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on*, pages 131 –136, 31 2005-april 1 2005.

[25] H. Strecker. Automatic detection of explosives in airline baggage using elastic X-ray scatter. In *Medicamundi*, volume 42, pages 30–33, jul. 1998.

[26] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.

[27] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[28] C. von Bastian, A. Schwaninger, and S. Michel. *Do Multi-view X-ray Systems Improve X-ray Image Interpretation in Airport Security Screening?*, volume 52. GRIN Verlag, 2010.

[29] A. Wales, T. Halbherr, and A. Schwaninger. Using speed measures to predict performance in X-ray luggage screening tasks. In *Security Technology, 2009. 43rd Annual 2009 International Carnahan Conference on*, pages 212 –215, oct. 2009.

[30] G. Zentai. X-ray imaging for homeland security. *Imaging Systems and Techniques, 2008. IST 2008. IEEE International Workshop on*, pages 1–6, Sept. 2008.