Inspection of Complex Objects Using Multiple X-ray Views

Domingo Mery, *Member, IEEE*

***Abstract*—This paper presents a new methodology for identifying parts of interest inside of a complex object using multiple X-ray views. The proposed method consists of five steps: A) *image acquisition*, that acquires an image sequence where the parts of the object are captured from different viewpoints; B) *geometric model estimation*, that establishes a multiple view geometric model used to find the correct correspondence among different views; C) *single view detection*, that segment potential regions of interest in each view; D) *multiple view detection*, that matches and tracks potential regions based on similarity and geometrical multiple view constraints; and E) *analysis*, that analyzes the tracked regions using multiple view information, filtering out false alarms without eliminating existing parts of interest. In order to evaluate the effectiveness of the proposed method, the algorithm was tested on 32 cases (five applications using different segmentation approaches) yielding promising results: precision and recall were 95.7% and 93.9%, respectively. Additionally, the multiple view information obtained from the tracked parts was effectively used for recognition purposes. In our recognition experiments, we obtained an accuracy of 96.5%. Validation experiments show that our approach achieves better performance than other representative methods in the literature.**

***Index Terms*—X-ray testing, computer vision, tracking, automated visual inspection, baggage screening.**

I. Introduction

X-ray imaging is used for both medical imaging and non-destructive testing (NDT) of materials and objects. The purpose of the latter application, called X-ray testing, is to analyze internal parts that are undetectable to the naked eye. X-ray radiation is passed through a test object and a detector senses variations in the intensity of the radiation exiting the object. The individual parts within an object can be recognized because they modify the expected radiation received by the sensor according to the differential absorption law [1].

There are numerous areas in which X-ray testing can be applied. In many of them, however, automated X-ray testing remains an open question and still suffers from: i) *loss of generality*, which means that approaches developed for one application may not transferred to another; ii) *deficient detection accuracy*, which means that there is a fundamental tradeoff between false alarms and miss detections; iii) *limited robustness* given that requirements for the use of a method are often met for simple structures only; and iv) *low adaptiveness* due to the fact that it may be very difficult to accommodate an automated system to design modifications or different specimens.

This paper proposes a *multiple view* methodology for automated X-ray testing that can contribute to reducing the four problems mentioned above. This methodology is useful for examining complex objects in a more *general*, *accurate*, *robust* and *adaptive* way given that this method analyzes an X-ray image sequence of a target object from several viewpoints automatically and adaptively. We observe that multiple view analysis has not yet been exploited in areas in which vision systems have typically been focused on single view analysis. This is the case of baggage screening, where certain items are very difficult to inspect from a single viewpoint because they could be placed in densely packed bags, occluded by other objects or rotated. For example, in Fig. 1d, it is clear that the part ① (a pencil sharpener) could not be identified using a single (intricate) projection, however, it could be possible to recognize it if multiple projections of the part are available, as shown in Fig. 1e. Thus, multiple view analysis is used by our approach because it can be a powerful tool for examining complex objects in cases in which uncertainty can lead to misinterpretation. Its advantages are not limited to 3D interpretation, as two or more views of the same object taken from different points can be used to confirm and improve the diagnostic obtained by analyzing a single image.

The main goals of our proposed multiple view methodology for detecting parts of interest in complex objects are:

A) To acquire an image sequence where the parts of the object are captured from different viewpoints (Fig. 1a).
B) To establish a multiple view geometric model used to find the correct correspondence among different views (lines in Fig. 1b).
C) To segment potential regions (parts) of interest in each view using an application-dependent method that analyzes 2D features in each single view, ensuring the detection of the parts of interest (not necessarily in all views) and allowing for false alarms (points in Fig. 1c).
D) To match and track potential regions based on similarity and geometrical multiple view constraints, eliminating those that cannot be tracked (lines in Fig. 1c).
E) To analyze the tracked regions using multiple view information, filtering out false alarms without eliminating existing parts of interest (see Fig. 1e where our approach is able to detect different parts recognizing for example a clip in ②).

The main contribution of our work is a generic multiple X-ray view methodology that can be used to inspect complex objects in which the detection cannot be performed using a single view. The approach is robust for poor monocular segmentation and some degree of occlusion. In order to illustrate the effectiveness of the proposed method, the algorithm was

D. Mery is with the Department of Computer Science, Pontificia Universidad Catolica de Chile, dmery@ing.puc.cl, http://dmery.ing.puc.cl

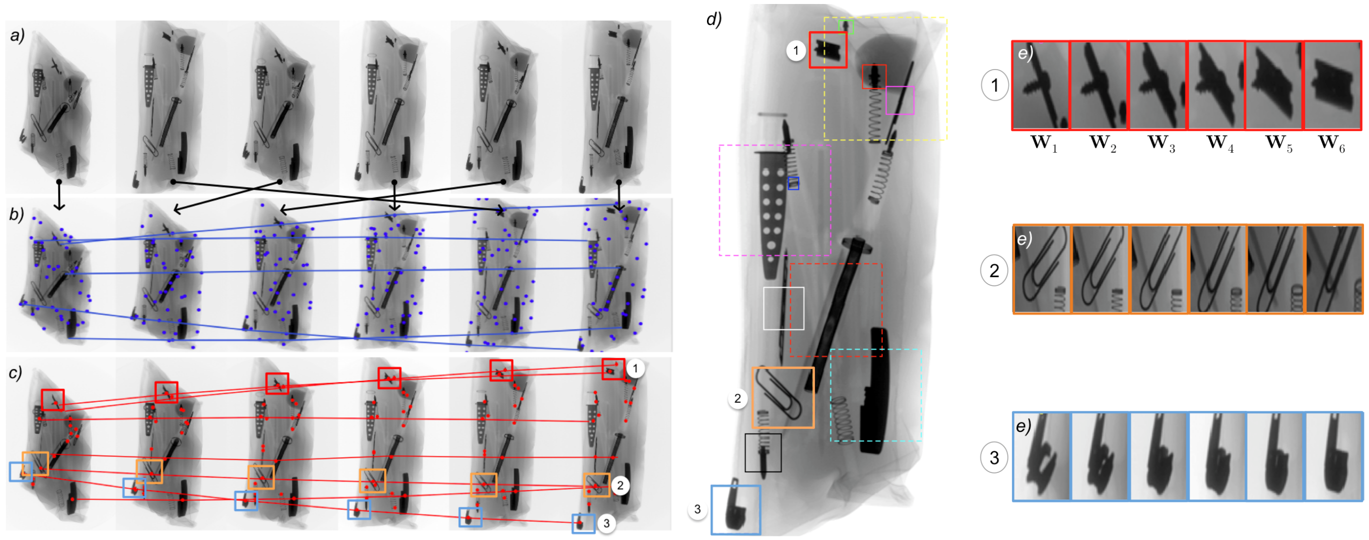

Fig. 1: Detection of objects in a pencil case using the proposed method: a) Unsorted sequence with six X-ray images. The images are sorted according to their similarity (see arrows). b) Sorted sequence, keypoints (points) and structure from motion (lines across the sequence). c) Detection in the sequence and tracked regions. d) Detection of parts of interest in the last image in the sequence (three of them are used in this example to illustrate the next sub-figures). e) Tracked example regions in each view of the sequence (1: pencil sharpener, 2: clip and 3: zipper slider body and pull-tab).

tested on several sequences yielding promising results on both detection and recognition problems.

This paper presents a literature review (Section II), the proposed approach (Section III), the experiments and results (Section IV), and concluding remarks (Section V). An early version of this paper was presented in [2]. In this paper, several new experiments are included and a recognition algorithm is presented and tested based on [3].

II. State of the Art

In order to present the state of the art of this field, this section presents a summary of the level of knowledge achieved by the scientific community in this area.

A. X-ray testing

The field of X-ray testing has five relevant areas of application[1]: *a) Baggage screening:* After 9/11, X-ray testing became an important tool for quickly identifying items that may pose a threat to security [4]. In countries where part of the economy is based on agricultural exports, baggage screening is also used to detect products of plant or animal origin in order to prevent the introduction of new diseases, pests or plagues. *b) Foods:* X-ray testing has been used in food safety procedures to detect foreign objects in packaged foods, fish bones and insect infestation. It is also used in quality inspections of fruit and grain [5]. *c) Cargo:* Cargo inspection has become increasingly important due to the growth of international trade. X-ray testing has been used to evaluate the contents of cargo, trucks, containers, and passenger vehicles in order to find smuggled goods [6]. *d) Castings:* X-ray testing is also used to verify the safety of certain automotive parts by inspecting each part of the product. Non-homogeneous regions like bubble-shaped voids or fractures can be formed in the production process and may affect the stability of components that are considered important for overall roadworthiness [7]. *e) Weldings:* X-ray-based inspection is required for objects subjected to a welding process in order to identify defects (porosity, inclusion, cracks and lack of fusion or penetration). X-ray testing is widely used for this purpose in the petroleum, chemical, nuclear, naval, aeronautics and civil construction industries [8]. We observe that X-ray testing has applications such as casting inspection where automated systems are very effective, and baggage screening, where human inspection is still used. Semi-automatic inspection procedures are used in areas such as welding and cargo. In certain applications (*e.g.* baggage screening), the use of multiple-view information can significantly improve success in recognizing items that are difficult to identify using a single viewpoint.

B. 3D object recognition

3D object recognition from 2D images is a very complex task given the infinite number of viewpoints, different lighting conditions, and objects that are deformable, occluded or embedded amidst clutter. In certain cases, automated recognition is possible through the use of approaches focused on obtaining highly discriminative and local invariant features related to lighting conditions and local geometric constraints (see for example [9] for a good review and evaluation of descriptors including the well-known SIFT [10] and SURF [11] features). A test object can be recognized by matching its invariant features to the features of a model. Over the past decade, many approaches have been proposed in order to solve the problem of 3D object recognition. Some use 3D data (points, meshes or CAD models) for 3D object category classification. In such

[1] The reader is referred to [1] for a review of this topic.

cases, the reconstructed 3D object serves as a query and is matched against the shape of a collection of 3D objects (see, for example, a review in [12]). These methods, however, are not used in practice to recognize real world objects in cluttered scenes because they cannot recognize the object's underlying structure. Other approaches focus on learning new features from a set of representative images (see, for example, visual vocabularies [13], implicit shape models [14], mid-level features [15], sparse representations [16], and hierarchical kernel descriptors [17]). These methods may fail when the learned features cannot provide a good representation of viewpoints that have not been considered in the representative images. Finally, some approaches include multiple view models (see, for example, an interconnection of single-view codebooks across multiple views [18], a learned dense multiple view representation by pose estimation and synthesis [19], a model learned iteratively from an initial set of matches [20], a model learned by collecting viewpoint invariant parts [21], 3D representations using synthetic 3D models [22], and a tracking-by-detection approach [23]). These methods may fail, however, when objects have a large intraclass variation. We observe that 3D object recognition is a challenging problem in the area of computer vision, and many studies have addressed this issue. Recognition performance can be improved by extracting relevant parts of the test object using different viewpoints, in order to build a multiple-view model.

C. Multiple view imaging

Many important contributions in computer vision using multiple-view analysis have been made over the past few years, *e.g.*, object class detection (see Section II-B), motion segmentation [24], object segmentation [25], visual motion registration [26], 3D reconstruction [27], people tracking [28], breast cancer detection [29], inspection using active vision [30], inspection using dual energy X-ray [31], and quality control [32]. We observe that the use of multiple-view information yields a significant improvement in performance in these fields.

We thus conclude that multiple view analysis is a powerful tool that can be used in X-ray testing, especially in the inspection of complex objects in cases where certain items are very difficult to recognize using only a single viewpoint (*e.g.* when they are occluded, covered by other items or positioned in difficult poses).

III. Proposed approach

Our proposed approach can be summarized as follows: *A)* a sequence of X-ray images of the object under test is acquired from different points of view; *B)* a model used to establish geometric correspondences between the views is estimated; *C)* potential parts in each single view of the sequence are segmented and described; *D)* using the geometric model and extracted descriptors, segmented parts are tracked across the views; and *E)* tracked parts are finally analyzed. In following sections, the steps will be explained in further details.

A. Image acquisition

A set of X-ray images of the object is acquired from m different points of view: $\mathbb{J} = \{\mathbf{J}_i\}_{i=1}^m$. The number of views and the viewpoint angles are analyzed in Section IV, however, in order to illustrate the explanation, a sequence of six views will be used (Fig. 1a). In this example, the rotation with respect to horizontal axis is $50^0, 10^0, 40^0, 20^0, 30^0, 0^0$ for each view.

If the images are not sorted, they can be arranged as a new sequence in order to obtain similar consecutive images that simplifies data association problem across the views. For this end, a visual vocabulary tree based on SIFT keypoints [10] of $\mathbb{J}$ is constructed for fast image indexing. The new image sequence, $\mathbb{I} = \{\mathbf{I}_i\}_{i=1}^m$, where $\mathbf{I}_i \in \mathbb{J}$, $\mathbf{I}_i \neq \mathbf{I}_j$, $\forall i \neq j$, is established by maximizing the total similarity defined as $\sum_{i=1}^{m-1} \text{sim}(\mathbf{I}_i, \mathbf{I}_{i+1})$, where the similarity function 'sim' is computed from a normalized scalar product obtained from the visual words of the images [13] (see an example in Fig. 1b, where the keypoints of each image are shown as points).

B. Geometric model estimation

Our strategy deals with 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. In our approach, the geometric constraints are established from bifocal (epipolar) and trifocal geometry [33]. Thus, for a point $\mathbf{x}_i$ in view $\mathbf{I}_i$, the corresponding point $\mathbf{x}_j$ in a second view $\mathbf{I}_j$ must lie on its epipolar line estimated by using the bifocal tensors (or fundamental matrix) $\mathcal{F}_{ij}$ of views i and j. On the other hand, for a point $\mathbf{x}_i$ in view $\mathbf{I}_i$ and its corresponding point $\mathbf{x}_j$ in view $\mathbf{I}_j$, the corresponding point $\mathbf{x}_k$ in a third view $\mathbf{I}_k$ is a point estimated by using the trifocal tensors $\mathcal{T}_{ijk}$ of views i, j and k. Multifocal tensors can be estimated from projection matrices $\mathbb{P} = \{\mathbf{P}_i\}_{i=1}^m$, where $\mathbf{P}_i$ is used to calculate the projection of a (3D) point $\mathbf{X}$ of the test object into a (2D) point $\mathbf{x}_i$ of image $\mathbf{I}_i$. The projection is computed as $\lambda \mathbf{x}_i = \mathbf{P}_i \mathbf{X}$ using homogeneous coordinates, where λ is a scale factor. The fundamental matrix $\mathcal{F}_{ij}$ is calculated from $\mathbf{P}_i$ and $\mathbf{P}_j$, and the trifocal tensors $\mathcal{T}_{ijk}$ from $\mathbf{P}_i$, $\mathbf{P}_j$ and $\mathbf{P}_k$ (see details in [33]).

The estimation of $\mathbb{P}$ can be performed by minimizing the error between real and modeled projection 3D → 2D. Two well known approaches can be used: *i) calibration*, in which known 3D points from a calibration object is used [33], [34] or *ii) bundle adjustment*, in which both 3D points and projection matrices are calculated from views of the test object itself using stable tracked keypoints across multiple views [2], [35].

C. Single view detection

Potential regions of interest are segmented in each single image of sequence $\mathbb{I}$. This is an *ad–hoc* procedure that varies depending on the application used.

Four segmentation approaches were tested in our experiments: *i)* Maximally Stable Extremal Regions (MSER) detects thresholded regions of the image which remain relatively constant by varying the threshold in a range [36]. *ii)* Spots detector (SPOTS) segments regions by thresholding the difference between original and median filtered image [37]. *iii)* SIFT matching detects regions of the image which SIFT descriptors are similar to SIFT descriptors of reference objects [10]. *iv)* Crossing line profile (CLP) detects closed and connected regions from edge image that meet contrast criteria [38].

Each segmented region, denoted as r, has a 2D centroid $\mathbf{x}_r$ and it is described using a SIFT descriptor as $\mathbf{y}_r \in \mathbb{R}^d$. The scale of the extracted descriptor, *i.e.*, the width in pixels of the spatial histogram of 4×4 bins, is set to $\sqrt{A_r}$, where A_r is the corresponding area of the region r.

D. Multiple view detection

In previous step, n_1 potential regions were segmented and described in the entire image sequence $\mathbb{I}$. Each segmented region is labeled with a unique number $r \in \mathbf{T}_1 = \{1, ..., n_1\}$. In view i, there are m_i segmented regions that are arranged in a subset $\mathbf{t}_i = \{r_{i,1}, r_{i,2}, ..., r_{i,m_i}\}$, *i.e.*, $\mathbf{T}_1 = \mathbf{t}_1 \cup \mathbf{t}_2 \cup ...\mathbf{t}_m$.

The matching and tracking algorithms combine all regions to generate consistent tracks of the objects parts of interest across the image sequence. The algorithm has the following four steps:

• *Matching in two views:* All regions in view i that have corresponding regions in the next p views are searched, *i.e.*, regions $r_1 \in \mathbf{t}_i$ that have corresponding regions $r_2 \in \mathbf{t}_j$ for $i = 1, ..., m-1$ and $j = i+1, ..., \min(i+p, m)$. In our experiments, we use $p = 3$ to reduce the computational cost. The matched regions (r_1, r_2) are those that meet *similarity* and *location* constraints. The similarity constraint means that corresponding descriptors $\mathbf{y}_{r_1}$ and $\mathbf{y}_{r_2}$ must be similar enough such that:

$$||\mathbf{y}_{r_1} - \mathbf{y}_{r_2}|| < \varepsilon_1. \tag{1}$$

The location constraint means that the corresponding locations of the regions must meet the epipolar constraint. In this case, the Sampson distance between $\mathbf{x}_{r_1}$ and $\mathbf{x}_{r_2}$ is used, *i.e.*, the first-order geometric error of the epipolar constraint must be small enough such that:

$$|\mathbf{x}_{r_2}^{\mathsf{T}} \mathcal{F}_{ij} \mathbf{x}_{r_1}| \left(\frac{1}{\sqrt{a_1^2 + a_2^2}} + \frac{1}{\sqrt{b_1^2 + b_2^2}} \right) < \varepsilon_2, \tag{2}$$

with $\mathcal{F}_{ij}\mathbf{x}_{r_1} = [a_1 \;\; a_2 \;\; a_3]^{\mathsf{T}}$ and $\mathcal{F}_{ij}^{\mathsf{T}}\mathbf{x}_{r_2} = [b_1 \;\; b_2 \;\; b_3]^{\mathsf{T}}$. In this case, $\mathcal{F}_{ij}$ is the fundamental matrix between views i and j calculated from projection matrices $\mathbf{P}_i$ and $\mathbf{P}_j$ [33]. In addition, the location constraint used is as follows:

$$||\mathbf{x}_{r_1} - \mathbf{x}_{r_2}|| < \rho(j - i), \tag{3}$$

because the translation of corresponding points in these sequences is smaller than ρ pixels in consecutive frames.
Finally, a new matrix $\mathbf{T}_2$ sized $n_2 \times 2$ is obtained with all matched duplets (r_1, r_2), one per row. If a region is found to have no matches, it is eliminated. Multiple matching, *i.e.*, a region that is matched with more than one region, is allowed. Using this method, problems like non-segmented regions or occluded regions in the sequence can be solved by tracking if a region is not segmented in consecutive views.

• *Matching in 3 views:* Based on the matched regions stored in matrix $\mathbf{T}_2$, we look for triplets (r_1, r_2, r_3), with $r_1 \in \mathbf{t}_i$, $r_2 \in \mathbf{t}_j$, $r_3 \in \mathbf{t}_k$ for views i, j and k. We know that a row a in matrix $\mathbf{T}_2$ has a matched duplet $[T_2(a, 1) \;\; T_2(a, 2)] = [r_1 \;\; r_2]$. We then look for rows b in $\mathbf{T}_2$ in which the first element is equal to r_2, *i.e.*, $[T_2(b, 1) \;\; T_2(b, 2)] = [r_2 \;\; r_3]$. Thus, a matched triplet (r_1, r_2, r_3) is found if the regions r_1, r_2 and r_3 meet the trifocal limitation:

$$||\hat{\mathbf{x}}_{r_3} - \mathbf{x}_{r_3}|| < \varepsilon_3, \tag{4}$$

This means that $\mathbf{x}_{r_3}$ must be similar enough to the re-projected point $\hat{\mathbf{x}}_{r_3}$ computed from the points in views i and j ($\mathbf{x}_{r_1}$ and $\mathbf{x}_{r_2}$), and the trifocal tensors $\mathcal{T}_{i,j,k}$ of views i, j, k calculated from projection matrices $\mathbf{P}_i$, $\mathbf{P}_j$ and $\mathbf{P}_k$ [33]. A new matrix $\mathbf{T}_3$ sized $n_3 \times 3$ is built with all matched triplets (r_1, r_2, r_3), one per row. Regions in which the three views do not match are eliminated.

• *Matching in more views:* For $v = 4, ..., q \leq m$ views, we can built the matrix recursively $\mathbf{T}_v$, sized $n_v \times v$, with all possible v-tuplets $(r_1, r_2, ..., r_v)$ that fulfill $[T_{v-1}(a, 1) \; ... \; T_{v-1}(a, v-1)] = [r_1 \;\; r_2 \; ... \; r_{v-1}]$ and $[T_{v-1}(b, 1) \; ... \; T_{v-1}(b, v-1)] = [r_2 \; ... \; r_{l-1} \;\; r_v]$, for $j, k = 1, ..., n_{v-1}$. No more geometric constraints are required because it is redundant. The final result is stored in matrix $\mathbf{T}_q$. For example, for $q = 4$ we store in matrix $\mathbf{T}_4$ the matched quadruplets (r_1, r_2, r_3, r_4) with $r_1 \in \mathbf{t}_i$, $r_2 \in \mathbf{t}_j$, $r_3 \in \mathbf{t}_k$, $r_4 \in \mathbf{t}_l$ for views i, j, k and l.
The matching condition for building matrix $\mathbf{T}_i, i = 3, ..., q$, is efficiently evaluated (avoiding an exhaustive search) by using a k-d tree structure [39] to search the nearest neighbors for zero Euclidean distance between the first and last $i-2$ columns in $\mathbf{T}_{i-1}$.

• *Merging tracks:* Matrix $\mathbf{T}_q$ defines tracks of regions in q views. It can be observed that some of these tracks correspond to the same region. For this reason, it is possible to merge tracks that have $q-1$ common elements. In addition, if a new track has more than one region per view, we can select the region that shows the minimal reprojection error after computing the corresponding 3D location. In this case, a 3D reconstruction of $\hat{\mathbf{X}}$ is estimated from tracked points [33]. Finally, matrix $\mathbf{T}_m$ is obtained with all merged tracks in the m views. See more details and examples in [2].

E. Analysis

The 3D reconstructed point $\hat{\mathbf{X}}$ from each set of tracked points of $\mathbf{T}_m$ can be reprojected in views where the segmentation may have failed to obtain the complete track in all views. The reprojected points of $\hat{\mathbf{X}}$ should correspond to the centroids of the non-segmented regions. It is then possible to calculate the size of the projected region as an average of the sizes of the identified regions in the track. In each view, a small window centered in the computed centroids is defined. These corresponding small windows, referred to as *tracked part*, will be denoted as $\mathbb{W} = \{\mathbf{W}_1, ..., \mathbf{W}_m\}$, as shown in Fig. 1e. Subsequently, each tracked part can be analyzed. Analysis of the tracked parts involves extracting features and classifying them using a supervised approach. At this stage we have the opportunity to extract features in all windows of $\mathbb{W}$, including those for which segmentation fails (or would have to be more lenient). In the analysis, a differentiation between detection and recognition is made.

1) Detection: In detection, there are only two classes: *parts* (for the parts of interest) and *no-parts* (for the other parts or background). Detection can be used when an *ad-hoc* monocular segmentation approach is designed to find the specific parts of interest, *e.g.*, a razor blade detector, as explained in Section III-C. Thus, a tracked part $\mathbb{W}$ can be used to confirm and improve the diagnostic as compared to a segmentation based on a single image.

A simple approach is to extract a contrast feature from the average image of the tracked windows: $\frac{1}{m}\sum_i \mathbf{W}_i$. Given that regions must appear as contrasted zones relating to their environment, verification is carried out as to whether or not the contrast of each averaged window is greater than ε_C. Obviously, more sophisticated features and classifiers can be used in cases in which detection is more complex.

2) Recognition: In recognition there are *parts* and *no-parts* also, however, the parts are divided into different classes. In this case, the parts could be segmented using a general purpose approach such as MSER (see Section III-C). For instance, in a pencil case the tracked parts can be clips, springs, razor blades, etc., and it is necessary to distinguish one from the other. In order to recognize parts, a multiple view strategy can be used [3]. The strategy consists of two stages: learning and testing.

In learning stage, we learn a classifier h to recognize patches or keypoints of the parts that we are attempting to detect. It is assumed that there are $C+1$ classes (labeled as '0' for no-parts class, and '1', '2', ... 'C' for C different parts of interest). 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 $\mathbf{y}$ is extracted using, for example, LBP, SIFT and SURF, among others [9]. In this supervised approach, each descriptor $\mathbf{y}$ is manually labeled according to its corresponding class $c \in \{0, 1, \ldots C\}$. Given the training data $(\mathbf{y}_t, c_t)$, 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 $\mathbf{y}_t$ to their classification label c_t, thus, $h(\mathbf{y}_t)$ should be c_t.

In the testing stage, we have to assign to which class does the tracked part $\mathbb{W}$ belong to. From each window $\mathbf{W}_i$, n_i patches $\mathbf{z}_{ij}$ and are extracted and described $\mathbf{y}_{ij} = f(\mathbf{z}_{ij})$, for $i = 1, \ldots m$, and $j = 1, \ldots n_i$. Each descriptor is classified as $c_{ij} = h(\mathbf{y}_{ij})$ according to the classifier designed in the learning stage. In order to classify a tracked part, $\hat{c} = \text{mode}(\{c_{ij}\})$, an ensemble strategy is used by computing the majority vote of the classes assigned to each patch of all windows. This strategy can overcome not only the problem of false single detections when the classification of the minority fails, but also when a part is partially occluded. For instance, a tracked part of a clip could show a window $\mathbf{W}_i$ where the clip and a spring are over imposed (see second row of clips in Fig. 4). Certainly, there will be patches in this window assigned to both classes; however, we expect that the majority of patches (of all windows) will be assigned to the class 'clip' if there are a small number of patches classified as 'spring'.

IV. Experimental results

In this Section we present the experiments and results obtained using the proposed method and some details about the implementation. The images tested in our experiments come from public GDXray database [1].

A. Evaluation of the proposed method

This section shows: 1) several experiments in which our approach can be used; 2) an analysis of the detection performance in function of the number of views of the sequence and the viewpoint angles; and 3) experiments on automated recognition. The geometric model was estimated using bundle adjustment [2].

1) Experiments on applications: We experimented on X-ray images from five different applications: a) detection of parts in general, b) detection of pen tips, c) detection of pins, d) detection of razor blades, and e) detection of discontinuities in aluminum wheels. The first four applications deal with detection of parts located inside pencil cases or bags. In this sense, the proposed approach could be used in baggage screening. The last application corresponds to a non-destructive testing that can be used in automated quality control tasks. In the applications, we used for the segmentation a) MSER, b) SPOTS, c) SPOTS, d) SIFT and e) CLP respectively as explained in Section III-C. The images used for our experiments present various characteristics, and the applications also vary. However, they do share one problem: performing the segmentation in a single image can lead to misclassification. A sequence is illustrated in Fig. 1 (other examples can be found in the early version of this paper [2]).

Table I shows statistics on 32 sequences of digital X-ray images (4 to 8 images per sequence, 185 images in total). In this table, ground truth (GT) is the number of existing parts and n_d is the number of parts detected using multiple view analysis including false positives (FP) and true positives (TP), *i.e*, n_d = FP+TP. Ideally, FP = 0 and n_d = TP = GT. In these experiments, *precision*, computed as TP/n_d, is 95.7%, and *recall*, computed as TP/GT, is 93.9%. If we compare single versus multiple view detection, the number of regions detected per image (n'_1) is drastically reduced by tracking and analysis steps to n_d (in total from 501 to 210, *i.e.*, 41.9%).

It is also interesting to observe the results for the detection of aluminum wheel discontinuities (application 'e'), where a similar level of performance is observed in [40]. In our approach, however, we avoid the calibration step [34].

2) Analysis of number of views and viewpoint angles: In order to evaluate how the number of views and the different viewpoint angles can affect the performance of our algorithm, the following experiment was carried out: X-ray images were captured from a test object –a pen case with 14 parts– from 90 different viewpoints by rotating its X and Y axes in increments of 10^0 ($\alpha = 0^0, 10^0, \ldots, 80^0$ and $\beta = 0^0, 10^0, \ldots, 90^0$) as illustrated in Fig. 2. Note that images for $\alpha > 70^0$ or $\beta > 70^0$ are quite intricate. Each viewpoint can be represented as a point in the (α, β) space. We define $\mathbb{S}$ as the set of all 90 points, and a sequence of m views as a subset of m different points $(\alpha_i, \beta_i) \in \mathbb{S}$ for $i = 1, \ldots m$. The algorithm

TABLE I: Detection using 2D analysis in 32 sequences $^{(*)}$

Application	#	size	m	n_1'	n_d	TP	GT
a) Parts in general	1	158	6	24	16	16	16
	2	158	6	19	12	12	14
	3	158	6	18	11	11	14
	4	158	6	25	20	20	20
Fig. 1→	5	158	6	19	12	13	14
	6	322	6	23	15	14	14
	7	322	6	17	10	10	13
	8	322	6	25	17	17	17
b) Pen tips	1	158	4	15	7	5	5
	2	158	5	10	5	4	5
	3	233	6	17	10	9	9
	4	158	6	14	8	5	5
	5	158	6	9	5	4	5
c) Pins	1	89	5	16	2	2	2
	2	89	6	17	2	2	2
d) Razor blades	1	81	6	2	2	2	2
	2	158	5	2	1	1	1
	3	158	6	2	1	1	1
	4	261	6	5	1	1	1
	5	322	6	6	1	1	1
e) Discon–tinuities	1	110	4	5	2	2	3
	2	439	4	24	1	1	1
	3	110	6	39	27	27	27
	4	158	6	10	5	4	4
	5	439	6	17	2	2	2
	6	439	6	15	2	2	2
	7	439	6	23	1	1	1
	8	439	6	21	3	3	3
	9	439	6	15	3	3	4
	10	439	6	16	2	2	2
	11	439	6	15	2	2	2
	12	439	8	16	2	2	2
Total	–	–	185	501	210	201	214
Precision					95.7%		
Recall							93.9%

(*) 'size': size of each image of the sequence in thousand of pixels. 'm': number of images in the sequence. 'n_1'': number of segmented regions per image. 'n_d': number of detected regions in the sequence. 'TP': true positives. 'GT': ground truth.

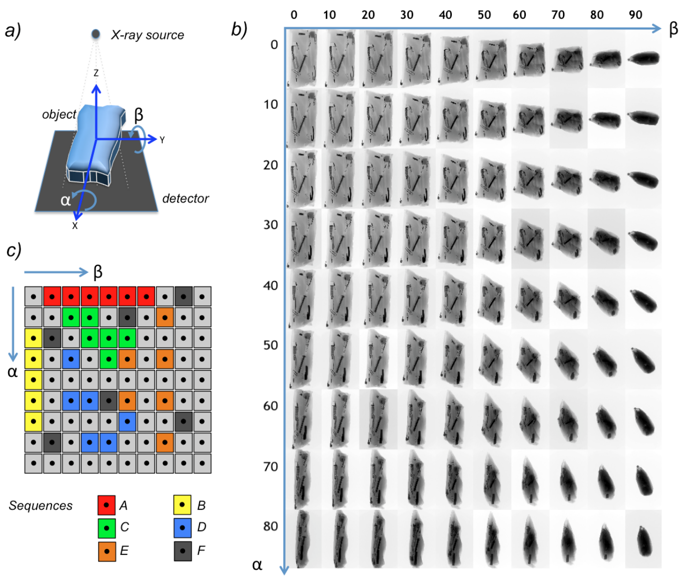

Fig. 2: Multiple views taken of an object test (a pen case) from 90 different viewpoints. The object was rotated around the X and Y axes in increments of 10^0. Each view is represented as a point in the (α,β) space. The colors in c) represent examples of different 6-views sequences.

was tested in several sequences of m views chosen from the 90 available X-ray images. The number of true and false positives were calculated for each sequence along with the performance statistics (precision and recall). Six different scenarios for choosing the m views were analyzed. In all scenarios, the first viewpoint –which corresponds to point (α_1, β_1)– was chosen randomly from $\mathbb{S}$, and the next $m-1$ viewpoints (α_i, β_i) were selected randomly (without repetition) from the following sets for $i = 2, \ldots m$: A) $\alpha_i = \alpha_1$, $\beta_i \in \mathbb{B}_1$; B) $\alpha_i \in \mathbb{A}_1$, $\beta_i = \beta_1$; C) $\alpha_i \in \mathbb{A}_1$, $\beta_i \in \mathbb{B}_1$; D) $\alpha_i \in (\mathbb{A}_1 \cup \mathbb{A}_2)$, $\beta_i \in (\mathbb{B}_1 \cup \mathbb{B}_2)$; E) $\alpha_i \in \mathbb{A}_2$, and $\beta_i \in \mathbb{B}_2$; and F) $(\alpha_i, \beta_i) \in \mathbb{S}$ where $\mathbb{A}_1 = \{\alpha_j \pm 10\}_{j=1}^{i-1}$, $\mathbb{A}_2 = \{\alpha_j \pm 20\}_{j=1}^{i-1}$, $\mathbb{B}_1 = \{\beta_j \pm 10\}_{j=1}^{i-1}$ and $\mathbb{B}_2 = \{\beta_j \pm 20\}_{j=1}^{i-1}$. See examples in Fig. 2c.

After the sequence is sorted, we observe that the viewpoint angles between consecutive views in the sequences are small in cases A, B and C (up to 10^0 on each axis); medium in cases D and E (up to 20^0 on each axis); and large in case F (up to 80^0 on each axis). Since in our approach the geometric model and tracking are estimated by matching SIFT keypoints from different views, it was expected that our methodology would not allow for large viewpoint angles. For this reason, our approach performed well in case F only when the randomly chosen images did not have a large viewpoint angle between consecutive views. For $m = 6$, we generated 150 sequences for each case (A, B, . . . F). The averages obtained (precision, recall) in percentages were as follows: $(89.5, 59.0)_A$, $(94.1, 70.9)_B$, $(88.5, 53.4)_C$, $(77.4, 38.3)_D$, $(61.8, 27.9)_E$, and $(53.6, 12.1)_F$. We observe that performance is better in case B than A because for constant β there are 1-2 (from 9) intricate images, whereas for constant α there are 1-3 (from 10) intricate images. In our experiments, the proposed method yields satisfactory performance only for small viewpoint angles in consecutive views (about 10^0). As a result, the rest of the experiments were only performed for scenarios A, B and C.

Similarly, we tested our algorithm for sequences of $m =$ 3, 4, 6, 8 and 9 images (in most of the cases for $m \geq 10$, it was not possible to estimate the geometric model because it was difficult to meet the criterion set by the bundle adjustment algorithm used in our method, which requires stable keypoints across all views of the sequence). For each case A, B and C, 150 sequences were generated, the algorithm was tested and the performance statistics were calculated and averaged. The performance is illustrated in Fig. 3. The figure also shows the performance for monocular test, *i.e.*, $m = 1$. The true positive rate against the false positive rate was fitted to a ROC curve modeled as $y = 1 - \exp(-\gamma x)$, and the area under the curve A_z was calculated (it was 0.88 for $m = 1$ and around 0.96 for m = 3, 4, 6, 8 and 9). We observe that the multiple view approach is better than the monocular approach because the ability to filter out false alarms is higher (maintaining the true positive rate almost constant). Multiple view performance is very similar for different numbers of views. However, in these

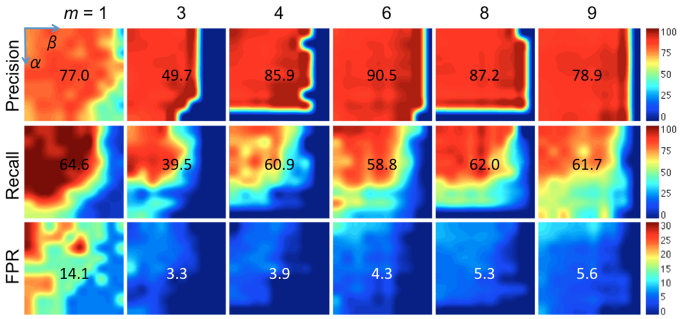

Fig. 3: Performance representation for sequences of $m = 1$, 3, 4, 6, 8 and 9 views. Each rectangle represents the defined (α,β) space as shown in Fig. 2c. The color of a point (α,β) corresponds to the average performance obtained when the X-ray image taken at rotations α and β belongs to the sequence of m views. The average of the values of each rectangle are presented in the middle.

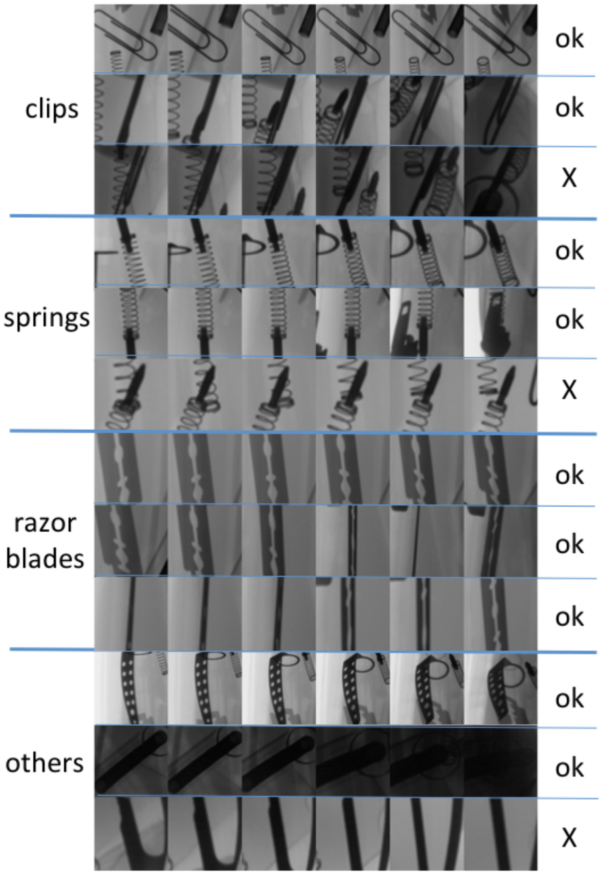

Fig. 4: Results on recognition of four classes, where the total accuracy was 93% (see Table II). The algorithm fails (see rows with 'X') in very intricate sequences or when the similarity to an incorrect class is high. Moreover, the algorithm classifies correctly (see rows with 'ok') even in sequences with occluded parts (see second rows of 'clips' and 'springs').

experiments, $m = 6$ yielded the best results.

It is clear that there is a trade-off between number of views and performance: tracking in many views makes difficult the estimation of the geometric model and it could lead to the elimination of those real regions that were segmented in few views, and tracking in fewer views increases the likelihood of false alarms. See Section IV-B for a comparison with other tracking algorithms.

3) Experiments on recognition: In our recognition experiments, the task was to distinguish between four different classes of objects that are present in the pencil case images of Fig. 2: 'clips', 'springs', 'razor blades' and 'others'. We followed the recognition approach explained in Section III-E2. In the learning phase, we used only 16 training images of each class. Due to the small intraclass variation of our classes, this number of training images was deemed sufficient. The training objects were posed in different orientations. SIFT descriptors were extracted as explained in [10], and a k-Nearest Neighbor (KNN) classifier with $k = 3$ neighbors was designed using the SIFT descriptors of the four classes. Other descriptors (like LBP [41] and HOG [42]) and other classifiers (like SVM) were also tested, although the best performance was achieved with the aforementioned configuration.

Testing experiments were carried out in scenarios A, B and C for sequences of 4, 6 and 8 views (as explained in the previous section). For each case, we tested on 200 tracked parts obtained by our tracking algorithm. In these experiments, there were 58 clips, 58 springs, 26 razor blades and 58 other objects. Some tracked parts used in our experiments on six views are shown in Fig. 4. A summary of the results is presented in Table II. We observed that the recognition of the tracked parts could be performed successfully by matching their invariant features with the features of the model. Additionally, Table II shows the accuracy of the classification of the single views (see row 'Single') in which the decision was taken using only one window of the tracked part. It is evident that the accuracy increases when using multiple views strategy (for example, see the increase from 83.1% to 92.0% in 8 views). It should be noted that the highest recognition accuracy was obtained for sequences of 4 views. The reason is that the occlusion in this case is negligible (an object that in some views could be occluded, would be difficult to be tracked with only 4 views). For this reason, for 6 and 8 views, the images of the tracked parts seem to be more intricate and occluded (see Fig. 4), and the accuracy is slightly lower. See Section IV-B3 for a comparison with other recognition algorithms.

B. Comparison with other methods

In order to compare the proposed method with other known approaches, in this Section we present comparison with other single view methods, other tracking algorithm and object recognition approaches. Finally, this section gives some comparison with human inspection.

1) Comparison with single view methods: We compared multiple view versus single view using well known segmentation approaches (Fig. 5). We found that single view approaches can detect the relevant regions of the object but that they are not able to isolate them well, *i.e.*, they cannot handle occlusion and superimposition of inner parts in a satisfactory manner.

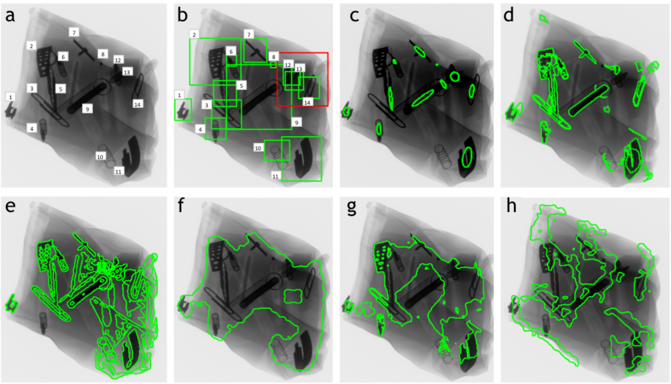

Fig. 5: Single view methods: a) original image with 14 parts of interest; b) detection using proposed method in a sequence of 6 views, all parts are detected with one false alarm; c) MSER [36], only 11 objects are detected; d) spot detector [37], it is very difficult to separate each part; e) CLP [38], there are many false alarms; f) HOG with SVM [42]; g) Otsu of HOG+LBP with SVM [41]; and h) sparse dictionary [16]. In the last three only large zones of interest are detected.

TABLE II: Accuracy and computational time in recognition.

m	Object → Method	Clip [%]	Spring [%]	Blade [%]	Others [%]	Total [%]	Time [s]
4	SIFT [10]	56.9	60.3	61.5	62.1	60.0	11.5
	S & Z [13]	44.8	74.1	73.1	70.7	64.5	0.08
	SURF [11]	84.5	29.3	80.8	67.2	63.0	0.31
	Sparse [16]	87.9	72.4	73.1	82.8	80.0	0.12
	Single	93.5	83.6	86.5	95.3	90.3	0.32
	Proposed	94.8	96.6	96.2	98.3	96.5	0.33
6	SIFT [10]	50.0	50.0	73.1	69.0	58.5	17.8
	S & Z [13]	51.7	69.0	96.2	50.0	62.0	0.12
	SURF [11]	91.4	43.1	80.8	55.2	65.5	0.47
	Sparse [16]	98.3	65.5	80.8	69.0	78.0	0.20
	Single	91.6	73.3	87.5	92.6	86.0	0.58
	Proposed	93.1	86.2	100.0	96.6	93.0	0.59
8	SIFT [10]	44.8	56.9	69.2	67.2	58.0	23.4
	S & Z [13]	58.6	77.6	92.3	62.1	69.5	0.16
	SURF [11]	93.1	55.2	96.2	46.6	69.0	0.59
	Sparse [16]	94.8	69.0	76.9	77.6	80.0	0.26
	Single	80.9	76.0	85.1	91.6	83.1	0.85
	Proposed	86.2	93.1	96.2	94.8	92.0	0.86

The multiple view approach can separate the parts of interest from each other because some views of the sequence do not suffer from these problems.

2) Comparison with other tracking algorithms: Tracking is one of the most relevant parts of our algorithm (see Section III-D). In our approach, potential objects are detected in a single view and afterwards they are tracked across multiple views. In order to compare our tracking algorithm with other known tracking algorithms, we tested only the tracking part, *i.e.*, we used the same single view detector –in order to segment the potential regions of interest– and we evaluated how well these regions can be tracked using different tracking algorithms. For this purpose, we selected randomly 20 sequences of six views from scenarios A, B and C; we sorted the sequences using algorithm explained in Section III-A; and we used MSER single detector to segment the potential objects (see Section III-C). In this experiment, the total single detections in the first view of the 20 sequences were 260. For each single detection, we counted the number of views in which it was correctly tracked. The algorithms used in this comparison were: 1) Lucas–Kanade [43]: It uses a fast registration technique based on a Newton-Raphson iteration. 2) Region Covariance [44]: It uses the covariance of certain pixel features extracted from the object to be tracked. 3) Tracking-Learning-Detection (TLD) [45]: It consists of a tracker that follows the object from frame to frame using a detector that can correct the tracker if necessary. The results are summarized in Table III. The performance, computed as the average percentage of views of the sequence in which single detections were correctly tracked, is increased by our algorithm because it considers geometric constraints by matching single detections. In addition, the table shows the average computational time by tracking one single detection across the whole sequence. The computational times are difficult to be compared because algorithm TLD is implemented in C++ (and the rest in Matlab), however, they give a good reference[2].

TABLE III: Tracking performance.

Method	[%]	time [ms]
Lucas–Kanade [43]	67.4	50
Region Covariance [44]	88.6	1250
TLD [45]	62.7	0.3
Proposed	93.2	190

3) Comparison with other recognition approaches: In this comparison we used the same training and testing sets explained in Section IV-A3. The task was to recognize four different classes of objects in 4, 6 and 8 views. For each case, 200 tracked parts were tested. The approaches used in this experiment were: 1) SIFT [10]: It matches individual features of the testing images to a database of features of the training images using the nearest-neighbor algorithm. A Hough transform is used to identify clusters that belongs to the same class. 2) Sivic & Zisserman (S & Z) [13]: It uses an efficient visual search algorithm based on a "term frequency–inverse document frequency" (tf–idf) weighted visual words frequencies. In our case, the visual vocabulary was built using SIFT descriptors of all training images. 3) SURF [11]: It finds correspondences between training and testing images based on scale and rotation invariant keypoints and descriptors. 4) Sparse [16]: It builds a sparse dictionary from SIFT descriptors of training images for each class, SIFT descriptors are extracted from testing images and classified according to the smallest reconstruction error. The final recognition in these experiments was decided by majority vote. The parameters were set so as to obtain the best performance. The results are summarized in Table II. In addition, the table gives the average

[2] For Lucas–Kanade and Region Covariance we use our own Matlab implementation according the details given in the references. In Region Covariance we used as features the position, the gray values, the gradients in each direction and its magnitude. For TLD we used the implementation available on the webpage given by the authors: https://github.com/zk00006/OpenTLD

computational time required to recognize one tracked part[3]. We observe that our algorithm considerably outperforms the other approaches because its ensemble strategy, as explained in Section III-E2, can overcome not only the problem of false single detections when the classification of the minority fails, but also when a part is partially occluded.

4) Experiments with human inspection: In this Section, different experiments on human detection are reported. The experiments were conducted by computer vision students, faculty members and staff of the Department of Computer Science of the Universidad Católica de Chile. The total number of participants was twenty. We carried out two experiments on detection tasks, and three experiments on recognition tasks. We measured precision (PR), recall (RE) or accuracy (AC) -in percentage- and inspection time (t) in seconds. The measurements are given as (X/Y), where X corresponds to the average obtained by the participants and Y by our algorithm.

In detection experiments, the participants were told to count how many objects were present in two sequences with different degrees of complexity. We used scenario C with $m = 6$ including image $(\alpha = 0^0, \beta = 0^0)$ for 'easy' case and $(\alpha = 70^0, \beta = 70^0)$ for 'difficult' case (see Fig. 2). In 'easy' case, PR = (100/90), RE = (92/95), $t =$ (36/2.6). The performance of our algorithm was similar and the computational time was only 2.6s. In 'difficult' case, PR = (95/100), RE = (64/14), $t =$ (42/2.6). Our algorithm achieved a high precision with a very low recall and computational time.

In the first recognition experiment, the participants were told to recognize how many clips, springs, razor blades and other objects were present in an 'normal' sequence of six images (including $(\alpha = 30^0, \beta = 30^0)$). In average the accuracy was AC = (91/88) with t = (46/8.4). Our method was slightly lower, however, faster than human inspection. In the second recognition experiment, the participants were told to recognize how many clips there were in a 'easy' sequence with and without a computer aid (a bounding box around each clip of the sequence computed by our algorithm). With and without computer aid the recognition was perfect, however, the inspection time with aid was in average the 67.4% of the inspection time without aid. In the third recognition experiment, the goal was to compare the performance of our system with the performance of the participants in a recognition of a tracked part. To this end, we use the same sequences of tracked parts explained in Section IV-A3. The participants were shown 200 sequences of six views. They were told to indicate which class of object was present. The accuracy was AC = (96/93) with t = (2.5/0.6) per tracked part. The performance achieved by our algorithm was slightly lower, however, the recognition time was considerable reduced.

This preliminary experiments has certainly some limitations: there were a small number of participants, who did a few experiments and the inspection task was being done for the first time. These limitations makes it impossible to draw definitive conclusions, however, five observations can be mention: *i)* In images where the difficulty is low or medium, the performance of our approach in comparison with the performance of the human inspectors seems to be similar or slightly lower. *ii)* In intricate images, the performance of human inspectors seems to be significantly higher. This observation corresponds to other computer vision problems (*e.g.*, pedestrian detection in crowd scenes). *iii)* The inspection time of our approach is considerably lower than the human inspection time. *iv)* The manual visual inspection is a tedious task, the participants have gotten fatigued after inspecting a few parts, and the obtained performance had a large variance. *v)* It seems to be possible to design an automated aid of human inspection task using the proposed algorithm.

C. Implementation of our algorithm

We used the implementation of SIFT, MSER, visual vocabulary and k-d tree from VLFeat [46]. The rest of algorithms were implemented in MATLAB. In the 'Detection' step, we used the contrast on the average window greater than 5%. For SIFT matching, the value ε_1 was set to 1000. The values ε_2 = 15 pixels and ε_3 = 25 pixels, were set by considering the epipolar distance and trifocal distance between correspondence points using our estimated geometric model; and the value ρ = 60 pixels was set by considering the maximal translation of corresponding points in consecutive views. The computational time depends on the application. In Fig. 1, as reference, the results of 2D detection were obtained in 2.6s and the recognition in 0.6s per object on a iMac OS X 10.8.5, processor 2.9GHz Intel Core i7, 8GB 1600 MHz DDR3 memory. The code of the MATLAB implementation is available on our webpage [48].

V. Conclusions

In this paper, we presented a new generic methodology that can be used to detect and recognize parts of interest in complex objects automatically and adaptively. The proposed approach filters out false positives resulting from segmentation steps performed on single views of an object by corroborating information across multiple views. The proposed methodology can be used to detect regions in images where the segmentation fails. The algorithm was tested on 32 cases (five applications using different segmentation approaches) yielding promising results: precision and recall are 95.7% and 93.9%, respectively. Best performance was achieved for small viewpoint angles between consecutive images (up to 10^0) and for sequences of six views. Additionally, the multiple view information obtained from the tracked parts can be effectively used for recognition purposes. In our recognition experiments, we obtained an accuracy of 96.5%. Preliminary experiments have shown that our approach achieves better performance than other representative methods in the literature, and the inspection time is reduced without reducing the performance significantly when comparing with human inspection. We believe that our method can be used to aid an user in an inspection task.

[3] For SIFT approach we used the object matching implementation by Li yang Ku available on File Exchange of Matlab Central http://www.mathworks.com/matlabcentral/fileexchange/34626-object-matching and SIFT descriptors extracted using VLfeat Toolbox [46]. For Sivic & Zisserman approach we used our Matlab implementation according the details given in the references. For SURF we used the Matlab implementation available in Computer Vision Toolbox version 5.1. For sparse representations we used the KSVD implementation given by the authors [47].

Acknowledgments

Fondecyt Grants No.1100830 and No.1130934 from CONICYTChile provided support for this work. The author would also like to thank Vladimir Riffo for providing the baggage X-ray images.

References

[1] D. Mery, "X-Ray Testing by Computer Vision," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 360–367.

[2] ——, "Automated detection in complex objects using a tracking algorithm in multiple X-ray views," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2011, pp. 41–48.

[3] D. Mery, V. Riffo, I. Zuccar, and C. Pieringer, "Automated X-Ray Object Recognition Using an Efficient Search Algorithm in Multiple Views," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2013, pp. 368–374.

[4] G. Zentai, "X-ray imaging for homeland security," *IEEE International Workshop on Imaging Systems and Techniques (IST)*, 2008.

[5] R. Haff and N. Toyofuku, "X-ray detection of defects and contaminants in the food industry," *Sensing and Instrumentation for Food Quality and Safety*, vol. 2, no. 4, pp. 262–273, 2008.

[6] Z. Zhu, Y.-C. Hu, and L. Zhao, "Gamma/X-ray linear pushbroom stereo for 3D cargo inspection," *Machine Vision and Applications*, vol. 21, no. 4, pp. 413–425, 2010.

[7] D. Mery, "Automated radioscopic testing of aluminum die castings," *Materials Evaluation*, vol. 64, no. 2, pp. 135–143, 2006.

[8] T. W. Liao, "Improving the accuracy of computer-aided radiographic weld inspection by feature selection," *NDT&E International*, vol. 42, pp. 229–239, 2009.

[9] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *European Conference on Computer Vision (ECCV)*, 2006.

[12] B. Bustos, D. A. Keim, D. Saupe, T. Schreck, and D. V. Vranić, "Feature-based similarity search in 3D object databases," *Computing Surveys (CSUR)*, vol. 37, pp. 345–387, 2005.

[13] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 4, pp. 591–605, 2009.

[14] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Workshop on Statistical Learning in Computer Vision (ECCV)*, vol. 2, no. 5, 2004.

[15] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2559–2566.

[16] I. Tosic and P. Frossard, "Dictionary Learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, Mar. 2011.

[17] L. Bo, K. Lai, X. Ren, and D. Fox, "Object recognition with hierarchical kernel descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1729–1736.

[18] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool, "Towards multi-view object class detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2006, pp. 1589–1596.

[19] H. Su, M. Sun, L. Fei-Fei, and S. Savarese, "Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 213–220.

[20] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation from single or multiple model views," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 159–188, 2006.

[21] S. Savarese and L. Fei-Fei, "3D generic object categorization, localization and pose estimation," in *IEEE International Conference on Computer Vision (ICCV)*, 2007.

[22] J. Liebelt and C. Schmid, "Multi-view object class detection with a 3D geometric model," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1688–1695.

[23] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 1515–1522.

[24] V. Zografos, K. Nordberg, and L. Ellis, "Sparse motion segmentation using multiple six-point consistencies," in *Asian Conference on Computer Vision (ACCV)*, 2010.

[25] A. Djelouah, J.-S. Franco, and E. Boyer, "Multi-view object segmentation in space and time," in *International Conference on Computer Vision (ICCV)*, 2013.

[26] K. Konolige and M. Agrawal, "FrameSLAM: from bundle adjustment to realtime visual mapping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077, 2008.

[27] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski, "Building Rome in a day," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 72–79.

[28] R. Eshel and Y. Moses, "Tracking in a dense crowd using multiple cameras," *International Journal of Computer Vision*, vol. 88, pp. 129–43, 2010.

[29] J. Teubl and H. Bischof, "Comparison of Multiple View Strategies to Reduce False Positives in Breast Imaging," *Digital Mammography*, pp. 537–544, 2008.

[30] V. Riffo and D. Mery, "Active X-ray testing of complex objects," *Insight*, vol. 54, no. 1, pp. 28–35, 2012.

[31] T. Franzel, U. Schmidt, and S. Roth, "Object detection in multi-view X-ray images," *Pattern Recognition*, pp. 144–154, 2012.

[32] M. Carrasco, L. Pizarro, and D. Mery, "Visual inspection of glass bottlenecks by multiple-view analysis," *International Journal of Computer Integrated Manufacturing*, vol. 23, no. 10, pp. 925–941, 2010.

[33] R. I. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. Cambridge University Press, 2003.

[34] D. Mery, "Explicit geometric model of a radioscopic imaging system," *NDT & E International*, vol. 36, no. 8, pp. 587–599, 2003.

[35] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon, "Bundle adjustment: a modern synthesis," *Vision algorithms: theory and practice*, pp. 153–177, 2000.

[36] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.

[37] R. Gonzalez and R. Woods, *Digital Image Processing*, 3rd ed. Pearson, Prentice Hall, 2008.

[38] D. Mery, "Crossing line profile: a new approach to detecting defects in aluminium castings," in *Scandinavian Conference on Image Analysis (SCIA)*, vol. 2749, 2003, pp. 725–732.

[39] J. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[40] D. Mery and D. Filbert, "Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp. 890–901, December 2002.

[41] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, Jul 2002.

[42] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[43] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International joint conference on artificial intelligence*, vol. 3, 1981, pp. 674–679.

[44] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 589–600.

[45] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.

[46] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," in *International ACM Conference on Multimedia (ICM)*, 2010, pp. 1469–1472.

[47] M. Aharon, M. Elad, and A. Bruckstein, "KSVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, 2006.

[48] D. Mery, "BALU: A toolbox Matlab for computer vision, pattern recognition and image processing (http://dmery.ing.puc.cl/index.php/balu)," 2011.