Automatic Segmentation of Unknown Objects, with Application to Baggage Security

Leo Grady[1], Vivek Singh[2], Timo Kohlberger[2], Christopher Alvino[3], and Claus Bahlmann[2]

[1] HeartFlow, Inc., Redwood City, CA, USA
[2] Imaging and Computer Vision, Siemens Corporation, Corporate Research and Technology, Princeton, NJ, USA
[3] American Science and Engineering, Billerica, MA, USA

Abstract. Computed tomography (CT) is used widely to image patients for medical diagnosis and to scan baggage for threatening materials. Automated reading of these images can be used to reduce the costs of a human operator, extract quantitative information from the images or support the judgements of a human operator. Object quantification requires an image segmentation to make measurements about object size, material composition and morphology. Medical applications mostly require the segmentation of prespecified objects, such as specific organs or lesions, which allows the use of customized algorithms that take advantage of training data to provide orientation and anatomical context of the segmentation targets. In contrast, baggage screening requires the segmentation algorithm to provide segmentation of an unspecified number of objects with enormous variability in size, shape, appearance and spatial context. Furthermore, security systems demand 3D segmentation algorithms that can quickly and reliably detect threats. To address this problem, we present a segmentation algorithm for 3D CT images that makes no assumptions on the number of objects in the image or on the composition of these objects. The algorithm features a new Automatic QUality Measure (AQUA) model that measures the segmentation confidence for any single object (from any segmentation method) and uses this confidence measure to both control splitting and to optimize the segmentation parameters at runtime for each dataset. The algorithm is tested on 27 bags that were packed with a large variety of different objects.

1 Introduction

Image segmentation is a core problem in computer vision that can be used to assess material and morphological properties of the objects being imaged. Classically, the unsupervised image segmentation problem has been posed as the localization of contiguous regions in an image that satisfy some measure of appearance homogeneity and/or some shape regularity. Classical methods to perform unsupervised image segmentation include watershed [1], mean shift [2] and normalized cuts [3]. More recent work has focused on addressing the homogeneity of complex appearances (e.g., textures) and operation across scales (e.g., [4]).

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part II, LNCS 7573, pp. 430–444, 2012.

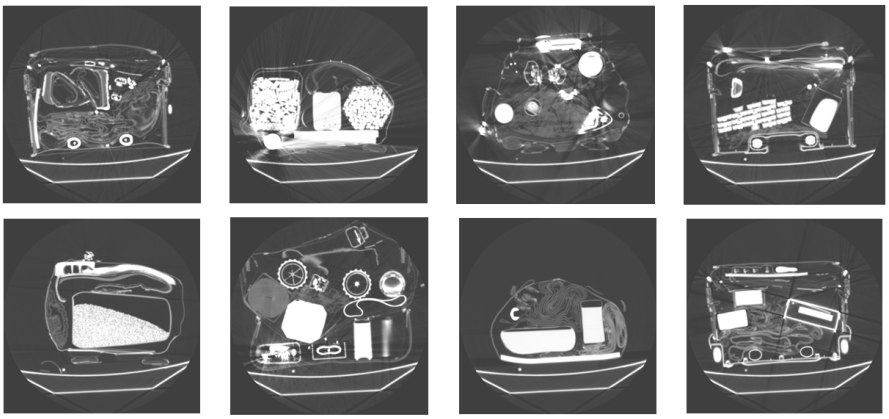

Fig. 1. A sample of 2D cross-sectional images in our 3D baggage screening dataset. Each bag contained objects with a large variety in size, shape, homogeneity and luggage exterior.

The established methods have proven effective for the creation of "superpixels" that can be used to drive further analysis by calculating and comparing localized region properties in the image (e.g., [5]). However, when considered from the standpoint of object segmentation (e.g., one segment label per object) these established methods tend to create oversegmentations that split each object into many pieces, requiring a further local merging step to reassemble the objects. This merging step is challenging to perform since the merging criterion is usually described at a global scale (e.g., to look like an "object") but the merging choices are made at a local level of assembling two adjacent superpixels. Furthermore, the results of the merging process is highly dependent on the order in which potential merges are evaluated.

To overcome these problems of the classical approach to unsupervised object segmentation, we propose a new algorithm consisting of three main steps:

1. Identification of all "object" voxels (as opposed to"background" voxels).
2. Creation of candidate splits into individual objects based on a partitioning method with global criteria.
3. Evaluation of these candidate splits using a novel Automatic QUality Assessment (AQUA) module that is trained on a wide variety of objects to recognize "good" objects in the candidate splits. Note that the AQUA module proposed here is different from the work in [6,7] which apply strictly to 2D images only and depend on having color/texture information available.

To validate the proposed approach we chose the application of security screening, more specifically the segmentation of objects in 3D CT scans of luggage. The role of security screening systems is to inspect checked and hand baggage, cargo and containers for dangerous content, such as, explosives, improvised explosive devices (IEDs), firearms, contraband, drugs, etc. These screening systems play a key role in the US Homeland Defense/Security strategy for increased safety of airports, air and sea traffic.

Security screening and explosive detection systems have been studied for several decades by a variety of groups in academia and industry (see [8] for review). A new decade for security screening started after the events of Sept. 11th, creating an elevation of security levels at airports and other hubs of mobility. Government investment and commercial interest increased the importance of highly sensitive security screening systems. Since that time, significant improvements to practical screening systems were made in terms of sensor and imaging devices [9]. However, the majority of research in the security screening area was performed in national labs and scanner vendors, and this work was kept classified in order to prevent better concealed threats.

Recently, the US government (i.e., DHS and the ALERT center of excellence[1]) began an initiative to promote academic and 3rd party research in security screening with the aim of developing next generation security screening systems. These efforts were accompanied by a series of workshops, where the latest (unclassified) state-of-the-art in security screening has been published [10]. The work described in this paper is a response to the ALERT initiative, which established the infrastructure and released appropriate security screening data to the unclassified domain.

Today's systems for CT based explosive detection [11] involve the following steps: First, during automatic screening, CT scan data is obtained and volumetric data is reconstructed, providing pseudocolor volumes that encode the main material properties such as density or, in case of dual energy scans, the effective atomic number Z_{eff}. Then, contiguous objects are segmented and physical properties such as density, Z_{eff} and volume are computed. If the combination of density, mass, and Z_{eff} are considered critical, suspicious regions are flagged and scheduled for operator inspection. The wide range of objects, sizes and appearances that can appear in baggage screening is illustrated in Figure 1. Our method for volumetric CT object segmentation is motivated by this application due to the difficulty of the problem but our algorithm could be applied to any unsupervised CT object segmentation problem.

2 Method

3D segmentation of CT images is challenging due to the unknown number of objects, variety of object sizes/shapes, objects consisting of multiple distinct parts, inhomogeneous internal density distribution (appearance) of some objects and the presence of artifacts. Segmentation in the context of baggage screening is particularly challenging due to the fact that metal is common (causing strong streaking artifacts) and the objects are packed tightly.

Each of the challenges listed above is addressed in our algorithm. To correct artifacts and identify "object" voxels, we utilize a Mumford-Shah algorithm to reduce noise. Separation of touching objects is performed using a global splitting algorithm (the isoperimetric algorithm [12,13]). Finally, we develop our

[1] `http://www.northeastern.edu/alert/`

Automatic QUality Assessment (AQUA) system to learn "good" object segments based on a large training database.

Our algorithm consists of three steps: 1) Identification of "object" voxels, 2) Generating candidate splits using a global partitioning criterion, 3) Evaluation of the candidate splits for good "objectness" using our novel AQUA method. These steps will now be described in detail.

2.1 Identification of Object Voxels

In the context of security screening with CT, objects of interest are ideally identified as consisting of voxels above a certain threshold. However, the identification of "object" voxels in this application is not quite so simple since artifacts and noise can disrupt CT imagery by creating false separations within objects or by merging nearby objects. These disruptions can be ameliorated by reducing artifacts and image denoising. In our application we do both by first applying the metal artifact reduction of Tuy [14] and then applying a version of the Mumford-Shah algorithm to perform denoising. The Mumford-Shah functional [15,16,17] is a core model in computer vision, for which variants can be used for image denoising, image segmentation and many other problems. We take advantage of the speed and robustness of recent advances in combinatorial optimization by applying the Mumford-Shah algorithm described in [18] to the artifact-reduced image in order to quickly identify all "object" voxels. Since this optimization is nonconvex it requires an initialization, which we provided in our application using the idealized target threshold for an object of interest (-500HU).

Following the artifact reduction and application of the Mumford-Shah technique in [18], each voxel in the image is labeled as either "object" or "background". We now describe how the "object" voxels can be split into candidate objects for evaluation with the AQUA.

2.2 Object Splitting with the Isoperimetric Algorithm

The output of the previous stage is to label each voxel as object or background. If each object in the image were spatially isolated, then it would be possible to simply return each connected component as a different object and conclude the segmentation. Unfortunately, real images often contain objects which are significantly touching each other. The problem of touching objects is particularly acute in the baggage screening application in which objects are deliberately packed tightly to conserve space. Consequently, it is necessary to adopt an algorithm that can be used to split touching objects by using a global criterion.

One global algorithm, the isoperimetric algorithm [12,13], was adapted to operate in linear time by operating on a *subgraph* (the *distance tree*) of the lattice graph representing a connected component [19]. The isoperimetric method searches for a split within a component, $\mathcal{M} \subseteq \mathcal{V}$ for the set of voxels $\mathcal{V}$ that minimizes

$$E(\mathcal{S}, \bar{\mathcal{S}}) = \frac{\text{cut}(\mathcal{S}, \bar{\mathcal{S}})}{\min(\text{Volume}(\mathcal{S}), \text{Volume}(\bar{\mathcal{S}}))}, \tag{1}$$

where $\mathcal{S} \subset \mathcal{M}$ and $\bar{\mathcal{S}} = \mathcal{M} - \mathcal{S}$. Due to the low overhead in speed and memory, we applied this isoperimetric distance tree algorithm [19] to find object separations in each connected component of the objects. This algorithm was applied recursively on each connected component until it was unable to find a high quality separation (or the component was too small) at which point the algorithm terminated. Taking advantage of the speed of the isoperimetric distance tree method, we ran it several times to produce candidate splits by using randomly chosen reference nodes in the connected component (see [19]). These splits were evaluated and the best split was chosen to continue the recursion if the split produced objects that were considered to have sufficiently high quality. The accurate assessment of a "high quality" object is key to the successful termination of this method and the avoidance of over- or under-segmentations. Our novel approach to this problem is detailed in the next section.

2.3 Automated QUality Assessment

We developed a novel confidence measure to automatically compute the quality of an image segmentation without *a priori* knowledge of the object being segmented. Such an **Automated QUality Assessment (AQUA)** is of great significance to the image segmentation problem, since it can be used to compare different segmentations and autonomously select the best one. Furthermore, such a measure can be used within a segmentation algorithm to iteratively improve the overall segmentation. An AQUA can also be used to flag the user of an automatic segmentation algorithm that the solution needs to be visually examined and fixed before the segmentation results should be used.

To obtain the confidence measure, we identified 92 good object segments (both ground truth and algorithm-generated) and employed a data-driven approach for model learning. A key challenge is to design or select appropriate features to allow us to learn which segments are "high quality". We determined these features by considering a set of features inspired by the literature on global methods for object segmentation. Effectively, we took each objective function as a feature and applied machine learning to create a model of "high quality".

Features. The main motivation behind the choice of our 42 features was to use a variety of segmentation metrics as indicators about whether a segmentation is correct. The purpose of this variety was to remain agnostic about which feature choices worked well for the final classifier in any given situation. The features we used can be broken down into major categories: (weighted or unweighted) geometric features, intensity features, gradient features, and ratio features.

Before beginning our exposition of the features, note that all weights in these descriptions refer to the Cauchy distribution function applied to the appropriate image intensities differences, i.e.,

$$w(i,j) = \frac{1}{1 + \beta \left(\frac{I_i - I_j}{\rho} \right)^2}, \tag{2}$$

where I_i and I_j are image intensities of neighboring voxels, where I_i and I_j are image intensities of neighboring voxels, v_i and v_j, β, which was set to 10^4 for all feature computations, is intended to control the sensitivity of the weight to intensity difference, and $\rho = \max_{(x,y)\in\mathcal{M}} \|\nabla I(x,y)\|_1$ was the maximum L1 norm of all intensity gradients within the segmentation mask, $\mathcal{M} \subseteq \mathcal{V}$. The purpose of ρ is to normalize the weights. Similarly, we define $w_-(I_i, I_j) = 1$ when $I_i > I_j$ and $w_-(I_i, I_j) = w(I_i, I_j)$ otherwise.

We used geometric features to capture some measure of size and regularity of the segmentation mask $\mathcal{M} \subset \mathcal{V}$, a concept dating back to some of the earliest works on image segmentation [20,15,21]. Of these, we chose:

$$\text{Volume}(\mathcal{M}) = |\mathcal{M}|\,, \quad \text{Surface Area}(\mathcal{M}) = \sum_{i,j:v_i\in\mathcal{M},v_j\in\bar{\mathcal{M}}} 1,$$

$$\text{Total Curvature}(\mathcal{M}) = \sum_{i,j:v_i\in\mathcal{M},v_j\in\bar{\mathcal{M}}} H(i,j),$$

where $H(i,j)$ is the discretely computed mean curvature on the surface of $\mathcal{M}$, which was locally computed as in [22].

Weighted geometric features are similar to the geometric features, but the geometric measures are locally emphasized when intensity values were similar to each other and suppressed when dissimilar. This concept that has been pervasive in image segmentation since the work of Caselles *et al.* [23] and has been seen in many other recent works [3,24,25]. The weighted geometric features we used were

$$\text{Weighted Volume}(\mathcal{M}) = \sum_{i,j:v_i\in\mathcal{M},v_j\in\mathcal{M}} w(i,j), \tag{3}$$

$$\text{Weighted Cut}(\mathcal{M}) = \sum_{i,j:v_i\in\mathcal{M},v_j\in\bar{\mathcal{M}}} w(i,j), \tag{4}$$

$$\text{Total Weighted Curvature}(\mathcal{M}) = \sum_{i,j:v_i\in\mathcal{M},v_j\in\bar{\mathcal{M}}} w(i,j)H(i,j), \tag{5}$$

$$\text{Low-High Weighted Cut}(\mathcal{M}) = \sum_{i,j:i\in\mathcal{M},j\in\bar{\mathcal{M}}} w_+(I_i, I_j), \tag{6}$$

$$\text{High-Low Weighted Cut}(\mathcal{M}) = \sum_{i,j:i\in\mathcal{M},j\in\bar{\mathcal{M}}} w_-(I_i, I_j). \tag{7}$$

Our intensity features employed various measures of the direct image intensities. These features were either intended to measure absolute intensity or intensity spread. Without exception, only intensities inside the segmentation mask $\mathcal{M}$ were included. Of these, we chose Mean Intensity($\mathcal{M}$)= $\frac{1}{|\mathcal{M}|}\sum_{v_i\in\mathcal{M}} I_i$; Median Intensity($\mathcal{M}$)= median($\{I_i : v_i \in \mathcal{M}\}$); Total Intensity($\mathcal{M}$)= $\sum_{v_i\in\mathcal{M}} I_i$; Minimum Intensity($\mathcal{M}$)= $\min_{v_i\in\mathcal{M}} I_i$; Maximum Intensity($\mathcal{M}$)= $\max_{v_i\in\mathcal{M}} I_i$; and Standard Deviation ($\mathcal{M}$)= $\frac{1}{|\mathcal{M}|-1}\sum_{v_i\in\mathcal{M}} (I_i - \text{Mean Intensity}(\mathcal{M}))^2$; Interquartile Distance, defined as half of the difference between the 75th percentile and the 25th percentile values of intensities;

Gradient features used various measures of the intensity gradients (local intensity changes). All intensity derivatives comprising these gradients were computed via central differences. Of these, we chose: Total L1 Gradient Norm($\mathcal{M}$)= $\sum_{v_i \in \mathcal{M}} \|\nabla I(v_i)\|_1$; Total L2 Gradient Norm($\mathcal{M}$)= $\sum_{v_i \in \mathcal{M}} \|\nabla I(v_i)\|_2$; Mean L1 Gradient Norm($\mathcal{M}$)= $\frac{1}{|\mathcal{M}|} \sum_{v_i \in \mathcal{M}} \|\nabla I(v_i)\|_1$; Mean L2 Gradient Norm($\mathcal{M}$) = $\frac{1}{|\mathcal{M}|} \sum_{v_i \in \mathcal{M}} \|\nabla I(v_i)\|_2$; Median L2 Gradient Norm($\mathcal{M}$) = median($\{\|\nabla I(v_i)\|_1 : v_i \in \mathcal{M}\}$); Min L1 Gradient Norm($\mathcal{M}$)= $\min_{v_i \in \mathcal{M}} \|\nabla I(v_i)\|_1$; Max L1 Gradient Norm($\mathcal{M}$)= $\max_{v_i \in \mathcal{M}} \|\nabla I(v_i)\|_1$; L1 Norm Interquartile Distance($\mathcal{M}$); L1 Norm Standard Deviation($\mathcal{M}$); and L2 Norm Standard Deviation($\mathcal{M}$).

We opted to explicitly include a selection of features that were ratios of our other features. The intent was not to be completely comprehensive, but rather to use domain knowledge of segmentation problems to explicitly choose combinations that the literature and our experience told us would be good indicators of segmentation performance. The ratio features were simply the ratio of two features above. Several fall into the category of cut divided by volume, a concept that has appeared in various forms [3,13,12]. Of these, we chose: *all four weighted and unweighted combinations of cut divided by volume*; *all four combinations of low-high weighted cut or high-low weighted cut divided by unweighted or weighted volume*; *weighted cut divided by unweighted cut*; *all four combinations of low-high weighted cut or high-low weighted cut divided by unweighted or weighted cut*; *blur index* defined as *sum the L2 norms of the gradients* divided by *sum of the L1 norms of the gradients*; *normalized cut* as defined in [3]; *curvature over unweighted cut*; and *weighted curvature over unweighted cut*.

We note that some of the features, such as the geometric features and most of the intensity-based features, were not meant to be discriminative alone. Rather, they were intended to lend context about the expected values for some of the other more discriminative features for a given candidate segmentation. All features mentioned above are both translation and rotation invariant.

Model Learning. We used a data-driven approach to model learning by annotating a number of "high quality" object segments and fitting a generative model over these segments in the feature space. A generative model for "high quality" segments was chosen over a discriminative model due to our expectation that the variability in feature space of "low quality" segmentations is much higher than the variability for "high quality" segmentations. Specifically we used a Gaussian mixture model (GMM) to approximate the distribution of the good segments in the feature space which we fit via Expectation Maximization. Prior to fitting the GMM model, we normalized each feature by subtracting the mean (over all segments) and dividing by the standard deviation.

We used Principal Component Analysis (PCA) to reduce the dimensionality of the data. This step is essential since the number of features is large compared to the size of the training set. Consequently, the distribution of the "high quality" object segments in the training set may be too sparse in the original feature space. Furthermore, the features are likely to be highly correlated and therefore redundant. The number of PCA dimensions was chosen based on the

performance on a validation set. We fit the GMM model (using eight Gaussians) over the PCA coefficients of all the segments in the training set using the EM algorithm.

We define the AQUA of an object segment $\mathcal{S} \subseteq \mathcal{V}$ as

$$\text{AQUA}_{\text{seg}}(\mathcal{S}) = \sum_{i=1}^{8} w_i \mathcal{N}(f(\mathcal{S}); \mu_i, \mathbf{\Sigma}_i) \tag{8}$$

where $f(\mathcal{S})$ is the PCA projected feature vector for segment $\mathcal{S}$ and w, μ, σ are GMM parameters of the ith Gaussian in the mixture.

We also define the AQUA of the cut of a connected component mask $\mathcal{M}$ and a component object $\mathcal{R} \subset \mathcal{M}$ as

$$\text{AQUA}_{\text{cut}}(\mathcal{R}, \mathcal{M}) = \max\left(\text{AQUA}_{\text{seg}}(\mathcal{R}), \text{AQUA}_{\text{seg}}(\mathcal{M} - \mathcal{R})\right). \tag{9}$$

One challenge in using the AQUA derived from a GMM is that it is often the case that initial connected components may contain many objects which are fused together. Splitting a connected component with many fused objects into two pieces may create two components which are each individually composed of several fused objects. Since the AQUA is trained to recognize individual objects, a split of several fused objects into two smaller sets of fused objects may be rejected as "poor quality". Since we want intermediate stages of the recursion to split an aggregate of fused objects into two smaller aggregates of fused objects, we use a combination of the AQUA and the isoperimetric ratio of the cut [13]. More formally:

$$\text{SP}(\mathcal{R}, \mathcal{M}) > \text{SP}(\mathcal{S}, \mathcal{M}) \equiv \begin{cases} \text{AQUA}(\mathcal{R}, \mathcal{M}) > \text{AQUA}(\mathcal{S}, \mathcal{M}) & \text{if AQUA}(x, \mathcal{M}) > \gamma, \\ \text{ISO}(\mathcal{R}, \mathcal{M}) < \text{ISO}(\mathcal{S}, \mathcal{M}) & \text{otherwise,} \end{cases} \tag{10}$$

where $x \in \mathcal{R}, \mathcal{M}$, $\text{ISO}(\mathcal{R}, \mathcal{M})$ is the isoperimetric ratio of split $\text{SP}(\mathcal{R}, \mathcal{M})$ (see [13]) and γ is a free parameter set to determine if the AQUA score of the split indicates that one piece of the split represents an object.

AQUA Usage. The AQUA is used in two stages of the algorithm: First, to assess the quality of the candidate splits from the isoperimetric distance tree method using randomized reference points (to determine whether to continue recursion) and second, to select the best segmentation result obtained by running the complete segmentation algorithm at three different parameter settings.

To guide our segmentation, we use the AQUA in the splitting phase of our segmentation algorithm. Recall from Section 2.2 that the splitting method iteratively splits under-segmented objects into two segments by considering multiple splits (derived from different reference voxels) and selects the best split using (10). Moreover, the recursive splitting also stops if one of three conditions are met: 1) The original mask has a very large probability of being a single object (as determined by the AQUA), 2) None of the multiple splits has a sufficiently

high AQUA (greater than a threshold γ), 3) The original mask is too small. The first condition is derived from the fact that the GMM was trained to recognize that an object is high quality, so when the GMM registers an object as being very high quality we assume that no further splitting is required. In the context of the baggage screening application we selected a 50mL size as "too small" to continue recursion.

The AQUA can also be used to evaluate an overall segmentation of the entire image by averaging the AQUA of each individual segmented object. This overall AQUA can be used generally to compare different segmentation results obtained using different parameter settings of a segmentation algorithm, or quite possibly a different segmentation algorithm altogether. In this work, we use the overall image AQUA to evaluate the segmentations obtained using our segmentation algorithm at different parameter settings and then selected the segmentation with the greatest overall AQUA to be the final segmentation. Specifically, we experimented with α (a free parameter in the edge weight function in isoperimetric segmentation [12]) and γ in (10); a lower value of α and/or higher value of γ encourages further splitting and avoids undersegmentation. Based on several experiments on 4 baggage datasets, we selected the following parameter sets to yield the best performance: $(\alpha, \gamma) \in \{(100, -250), (100, -350), (10, -350)\}$. For validation, we compared the selected best segmentation with other segmentation results by visual inspection.

2.4 Summary

Our segmentation algorithm of a CT volume is comprised of the following steps:

1. Metal artifact reduction using [14].
2. Applying the Mumford-Shah method in [18] to identify all object voxels.
3. For each connected component, recursively apply the isoperimetric distance tree algorithm with several randomly chosen reference voxels and use (10) to determine the best split. End the recursion as determined by comparing the AQUA of the best split, (8) to parameter γ.

To improve the execution speed, we applied the algorithm with a coarse-to-fine segmentation, using the segmentation from the previous level as the initial masks for further splitting. This entire pipeline running single-threaded code requires about 4 minutes per parameter set on an Intel Core 2 Duo 2.8GHz machine. Therefore, we can change the two parameters and re-run the entire pipeline, or run them in parallel processes, several times and select the segmentation with the best average AQUA for all objects. Note that this algorithm would be applied to checked baggage, so 4 minutes is acceptable.

3 Results

We evaluate our algorithm on the challenging problem of baggage screening. The data was generated to mimic a real baggage screening application by creating 27

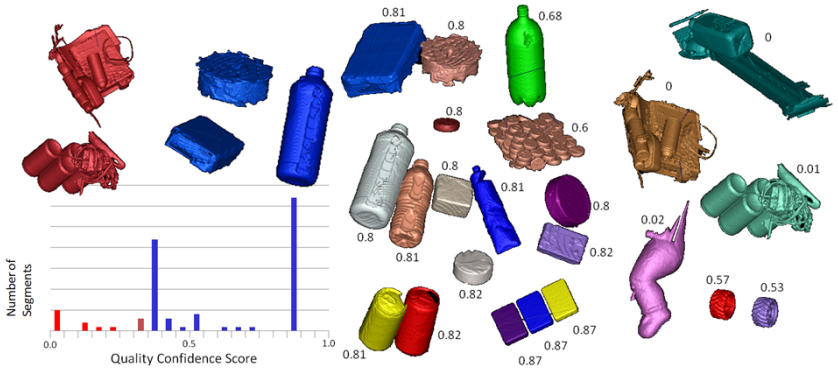

Fig. 2. Left: Histogram showing the number of segments placed into each bin of the AQUA. Red bars represent "bad" segmentations and blue bars represent "good" segmentations. Note that the good and bad segmentations are completely separable by thresholding the AQUA. Top: Examples of object segments labeled "good" (blue) and object segments labeled "bad" (red). Right: A variety of object segments and the corresponding AQUA score.

baggage scans using a medical CT scanner. Each scan contained a single piece of luggage, such as backpacks, duffel bags and soft-shell suitcases. Within each piece of luggage, nine to twenty different objects were packed that had a large variability in size, shape, orientation and material (appearance) composition. Some objects consisted of several parts (e.g., a large model car with clearly defined wheels and battery). The produced images had resolution 0.98mm $\times$ 0.98mm $\times$ 1.2mm. Depending on the size of the suitcase, the number of slices ranged from 665 to 953 and each slice consisted of 512×512 voxels. For each scan, almost all objects inside a luggage piece were manually segmented by baggage screening experts to provide a ground truth. Figure 1 shows several examples of the data that we used for evaluation, which illustrates the difficulty in segmenting these datasets in meaningful objects.

Our experiments are aimed at answering the following questions: 1) Does the AQUA qualitatively match our intuitive conception of a "good" object segment? 2) Does the AQUA quantitatively allow us to discriminate high quality segmented objects from poor quality segmented objects? 3) Does the overall segmentation algorithm produce segmentations that qualitatively agree with our conception of high quality segmentations? 4) How does the overall segmentation algorithm quantitatively match the manual (ground truth) segmentations? 5) How much improvement is gained by using our AQUA to end recursion as opposed to using the conventional isoperimetric ratio from [12]?

3.1 AQUA Evaluation

In this section, we perform an experiment to evaluate the quantitative and qualitative accuracy of the AQUA. We collected 60 object segments from the

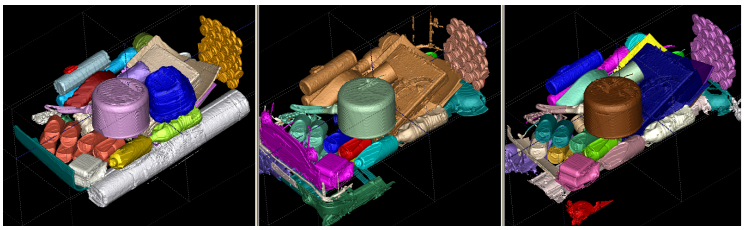

Fig. 3. Segmentation of a luggage CT Scan. Segments with different labels are rendered in different colors. (left) Ground truth segmentation, (middle) segmentation obtained using iso-ratio, and (right) segmentation obtained using AQUA.

intermediate and final results of the segmentation algorithm and labeled each "high quality" or "poor quality". Some of the poor quality object segmentations were undersegmented and others were oversegmented. Our AQUA was applied to each object. Figure 2 shows a histogram for the AQUA and the number of "high quality" and "low quality" objects that were placed into each AQUA bin. The figure also gives several examples of object segments that were labeled "high quality" or "low quality" and the values of the AQUA for a range of object segments (good and bad) that we tested on. Since perfect discrimination was achieved by the AQUA for segments of "high quality" and "low quality", we can conclude that the AQUA is accurately evaluating the quality of each object segment.

It is somewhat surprising that the AQUA performs so well for such a large variety of objects. However, the features describing each object were derived from the objective functions that have been optimized in the literature for different segmentation algorithms. The assumption behind each of these algorithms is that the best segmented object in an image is the segmented object that optimizes the criterion established by the objective function. Although these segmentation algorithms were all intended for general-purpose image segmentation, we are not aware of any previous examination of whether the values of the objective function for an object can be used to *discriminate* "good" object segmentations from "bad" object segmentations across images and application domains. Our experiment demonstrates that using these objective function evaluations as features for a generative GMM model does allow us to identify "good" segmentations.

3.2 Segmentation of the Baggage Datasets

A qualitative result from our segmentation system is given by Figure 3. In order to quantify the segmentation quality we developed error metrics that are based on mutual overlaps between ground truth (manual segmentations) and the segments produced by our algorithm. In our evaluation, we assumed that each computed label/segment is assigned to exactly one ground truth label/segment and vice-versa. Therefore, even if a computed segment overlaps with more than one ground truth segment the overlap is compared with only one of the ground truth segments, and not compared with any of the other ones. Conversely, if one ground truth segment overlaps with several computed segments, the overlap

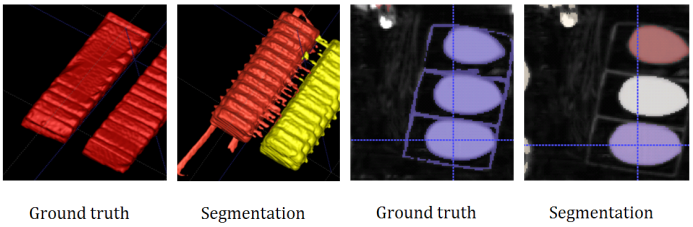

Fig. 4. Oversegmentations relative to ground truth. The ground truth annotations of the battery pack (left) and the soaps (right) assigned a single label to the whole package. However, the segmentation algorithm assigned several labels to sub-components, i.e., the two separate groups of batteries and the three soaps, respectively. Consequently, our overlap measure was poor for these objects, even though our segmentations could be considered to simply represent different notions of the target object.

with only one of them is compared and all of the others were ignored. Therefore, differences in object definition between the ground truth and the segmentation can sometimes appear as a segmentation having a very poor score using this evaluation method. Figure 4 illustrates this situation.

The one-to-one assignments of ground truth segments to computed segments was determined automatically and such that the total sum of all overlaps (in voxels) was maximal. Since this problem is equivalent to the matching of two independent graphs with equal or differing number of nodes, here we were able to employ the Hungarian algorithm to find the optimal matching. Technically, this matching is realized by first computing the mutual overlaps (in voxels) between all ground truth and all computed segments, which results in an overlap matrix whose row index the list of ground truth labels and whose column index the list of computed labels. The Hungarian algorithm then finds the optimal assignment that maximizes the total overlap.

Based on this optimal matching of ground truth segments/labels to computed segments/labels, we computed the relative overlap per ground truth segment by dividing its overlap with the assigned computed segment (in voxels) by the size of the individual ground truth label (in voxels). The resulting overlap percentages are then plotted for each ground truth label. The overlap measures for each object in one dataset is shown in Figure 5. Both quantitatively and qualitatively, the use of our AQUA system to perform quality assessment significantly improved the quality of the segmentation (per-object and overall) as compared to the conventional use of the isoperimetric ratio from [12] to end recursion.

Besides these per-ground truth segment measures, as overall overlap metric we computed the average overlap percentage per dataset. The mean of the average overlap percentage over all 27 datasets with ground truth was 54% ($\sigma = 10$) when using AQUA in the splitting algorithm, and only 44% ($\sigma = 12$) when using just the isoperimetric ratio. As noted previously, this score appears worse than it is as a result of differences in "object philosophy" between the algorithm and the ground truth (see Figure 4).

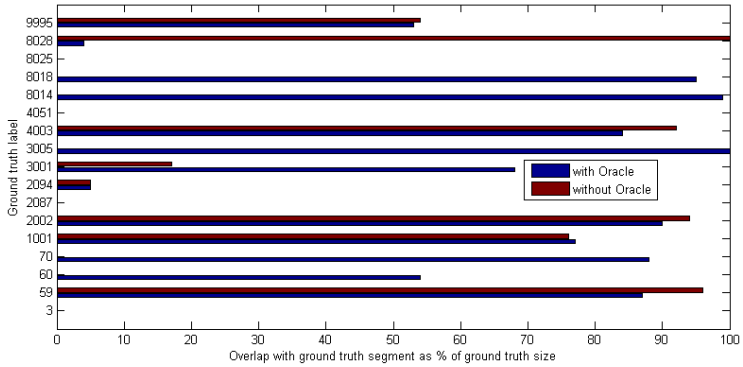

Fig. 5. Relative overlaps for one dataset. Average overlap percentage: 53% (with AQUA) and 31% (with isoperimetric ratio). Red: Overlap using isoperimetric ratio. Blue: Overlap using AQUA. Using AQUA to end recursion is a major improvement over using the isoperimetric ratio of [12] to end recursion. Note that most of the low object scores were due to partial matches that resulted from a different "object philosophy" between the algorithm and ground truth (see Figure 4).

4 Conclusion

We presented a new algorithm for unsupervised image segmentation that introduces a novel Automatic QUality Assessment (AQUA) module which is trained as a generative model (for "good" object segments) using a combination of different objective functions dispersed in the literature. The AQUA is then used both iteratively to end the splitting recursion (and choose between candidate splits) and at the end to assess the overall quality of the segmentation.

Our segmentation algorithm was applied to the challenging and important application of the unsupervised segmentation of baggage screening images acquired from 3D computed tomography scans. The challenge of image segmentation in these images is a product of the unknown number of objects, significant imaging artifacts, the wide variety of shapes and sizes, inhomogeneity and the tight packing of the objects against each other. General segmentation algorithms into superpixels of region homogeneity are unsuited for these images in which the goal is to make measurements about total object composition. A further challenge of segmenting these images is their large size and the speed requirements necessary for an algorithm to be useful.

We tested our algorithm on a novel dataset designed to mimic a real baggage screening data application. The evaluation demonstrated that our new AQUA can accurately determine the quality of an object segment *even for objects that were not in the training set*. Furthermore, our overall algorithm could be used to match a manually-derived ground truth for these 27 images. These experiments demonstrate the suitability and future prospects for applying this algorithm to improve the baggage security screening capabilities of today's airports.

Acknowlegement. This material is based upon work supported by the U.S. Department of Homeland Security under Task Order Number HSHQDC-10-J-00396. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. Datasets owned and provided by ALERT DHS Center of Excellence, Northeastern University, Boston MA.

References

1. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE PAMI 13(6), 583–598 (1991)
2. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE PAMI 24(5), 603–619 (2002)
3. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE PAMI 22(8), 888–905 (2000)
4. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE PAMI 33(5), 898–916 (2011)
5. Li, Y., Sun, J., Tang, C., Shum, H.: Lazy snapping. In: Proc. of ACM SIGGRAPH 2004, pp. 303–308 (April 2004)
6. Endres, I., Hoiem, D.: Category Independent Object Proposals. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 575–588. Springer, Heidelberg (2010)
7. Carreira, J., Sminchisescu, C.: Cpmc: Automatic object segmentation using constrained parametric min-cuts. IEEE PAMI 34, 1312–1328 (2012)
8. Singh, S., Singh, M.: Explosives detection systems (EDS) for aviation security: A review. Signal Process 83(1), 31–55 (2003)
9. Smith, R., Connelly, J.: CT technologies (chapter 7). In: Aspects of Explosives Detection, pp. 131–145. Elsevier (2009)
10. ALERT (Awareness and Localization of Explosives-Related Threats), Northeastern University, ed.: Algorithm Development for Security Applications (ADSA) Workshops 1-6 (2008–2011)
11. Martz Jr., H.E., Crawford, C.: Overview of deployed EDS technologies. Technical report, LLNL-TR-417232, Lawrence Livermore National Laboratory (September 2009)
12. Grady, L., Schwartz, E.L.: Isoperimetric graph partitioning for image segmentation. IEEE PAMI 28(3), 469–475 (2006)
13. Grady, L., Schwartz, E.L.: Isoperimetric partitioning: A new algorithm for graph partitioning. SIAM J. on Scientific Computing 27(6), 1844–1866 (2006)
14. Tuy, H.K.: A post-processing algorithm to reduce metallic clip artifacts in CT images. Eur. Radiol. 3, 129–134 (1993)
15. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. PAM 42, 577–685 (1989)
16. Tsai, A., Yezzi, A., Willsky, A.: Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification. IEEE Trans. on Image Proc. 10(8), 1169–1186 (2001)
17. Chan, T., Vese, L.: A level set algorithm for minimizing the Mumford-Shah functional in image processing. In: Workshop on VLSM, pp. 161–168. IEEE (2001)

18. Grady, L., Alvino, C.: The piecewise smooth Mumford-Shah functional on an arbitrary graph. IEEE TIP 18(11), 2547–2561 (2009)
19. Grady, L.: Fast, Quality, Segmentation of Large Volumes – Isoperimetric Distance Trees. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 449–462. Springer, Heidelberg (2006)
20. Blake, A., Zisserman, A.: Visual Reconstruction. MIT Press (1987)
21. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces based object segmentation. IEEE Trans. on PAMI 19(4), 394–398 (1997)
22. El-Zehiry, N., Grady, L.: Fast global optimization of curvature. In: Proc. of CVPR. IEEE Computer Society, IEEE (2010)
23. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22, 61–79 (1997)
24. Boykov, Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In: Proc. of ICCV, pp. 105–112 (2001)
25. Grady, L.: Random walks for image segmentation. IEEE PAMI 28(11), 1768–1783 (2006)