Interactive Exploration of 3D Scanned Baggage

Michael Traoré and Christophe Hurter
French Civil Aviation University

Volumetric data are available in a variety of scientific fields, from engineering and astrophysics, to medical imaging and security screening. The exploration of volumetric datasets is not trivial, however, and it is heavily impacted by the users' context-specific needs. In most airports, for example, security agents deal with such data during baggage inspections. This domain specifically uses two types of fluoroscopic systems: x-ray and tomography scanning. An x-ray system provides a flattened 2D baggage scan, whereas a tomography scanning system produces transversal scans called slices. Thanks to data processing (such as the Radon transform[1]), using these slices, we can produce a full 3D scan (a set of voxels with their corresponding density). The input data is a uniformly sampled multivariate field F: $V \to D$, $V \subset \mathbb{R}^n$, $D \subset \mathbb{R}^m$, where $n = 3$ (volume), $m = 1$ (a scalar), V is a subset of space (x, y, z coordinates), and D is the data attributes (density).

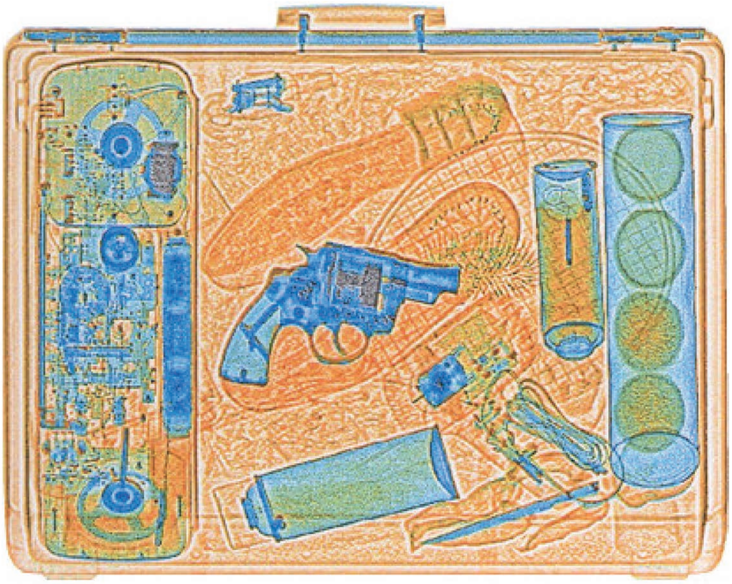

Figure 1. X-ray scan with the three standard colors. Orange indicates low-density materials, blue indicates high-density items, and green corresponds to either the superposition of different kinds of materials or average density materials.

Because the resulting x-ray scanned image only contains densities, it cannot display the material's original colors. The standard color visual mapping uses three colors (orange, green, and blue) to display the data density. Orange corresponds to low densities (mainly organic items), and blue indicates high-density items (such as metals). In x-ray systems, green corresponds to the superposition of different kinds of materials or average density materials (see Figure 1).

These 2D scanned images suffer from four issues.

- *Superposition*: A threat (such as a prohibited object like a knife) may be sheltered behind dense materials. It's sometimes possible to see through these blind shields using functionalities such as high penetration (enhanced x-ray power) or image processing (contrast improvement).
- *Location*: Depending on its location inside the luggage, a threat can be difficult to detect. For example, objects located in the corners, at the edges, or inside the luggage's frame are difficult to identify.
- *Dissociation*: One way to disguise a threat is to disassemble it and spread the parts throughout the luggage. For example, a weapon or explosive may consist of many separate items, such as a trigger, cannon, and so forth.
- *Lure*: An ill-intentioned individual may use a lure to hide a real threat. For instance, a minor threat (such as a small scissors) may be clearly visible to distract a security agent from a more important, but better hidden threat.

One potential solution that attempts to address these limitations is 3D scanned baggage explora-

Published by the IEEE Computer Society

tion. However, few studies have investigated implementing interactive volumetric exploration tools in this domain.[2] Extensive work has been done in medical 3D scan exploration and manipulation,[3] but there is a great opportunity to adapt and develop new interaction and data manipulation techniques to support 3D baggage exploration.

In this article, we present an interactive image-based visualization system[4] that supports volumetric data exploration. Although most existing systems use a combination of direct volume rendering (DVR) and its corresponding transfer function (TF), our goal was to develop new interaction paradigms and new volumetric dataset manipulations.

We began our study with a one-day onsite observation at a major French airport. We then conducted a brainstorming session with four security practitioners. Based on these analyses, we defined relevant use cases and extracted a set of top-level needs and requirements:

- Users need to visualize luggage content.
- Users have to explore the baggage using an interactive navigation system.
- The system must provide tools to address occlusion issues, such as the superposition of items inside the luggage, that hinder visualization.
- Users must be able to explore luggage with simple interactions because their knowledge of volumetric data processing too limited to implement involved techniques and their parameters.

As a result of this user-centered brainstorming and participatory design process, we were able to mock up numerous working prototypes, which we then validated and aggregated into a working system. Final users were involved into the full design process and thus also validated the final tool. Our system provides a set of interaction techniques that combine the direct manipulation of the voxels with their interactive visualization.

Top-Level Structure

Our system consists of one main view (volume visualization) and five subviews that allow user control and customization. Our interactive system does not include a menu, and every feature is directly accessible from the main view (see Figure 2). The volume visualization can contain up to two views to ease interactions with the 3D scan. These views display the baggage, and users can navigate through them (zoom, pan, or rotate) and manipulate the baggage contents (brush, select, or delete). These two views can be linked or disconnected, which allows user to inspect selected objects with or without their context. A smaller view called the Overview shows the location of the investigated area. The Overview is one-eighth the size of the main window and is located in the bottom right of the GUI.

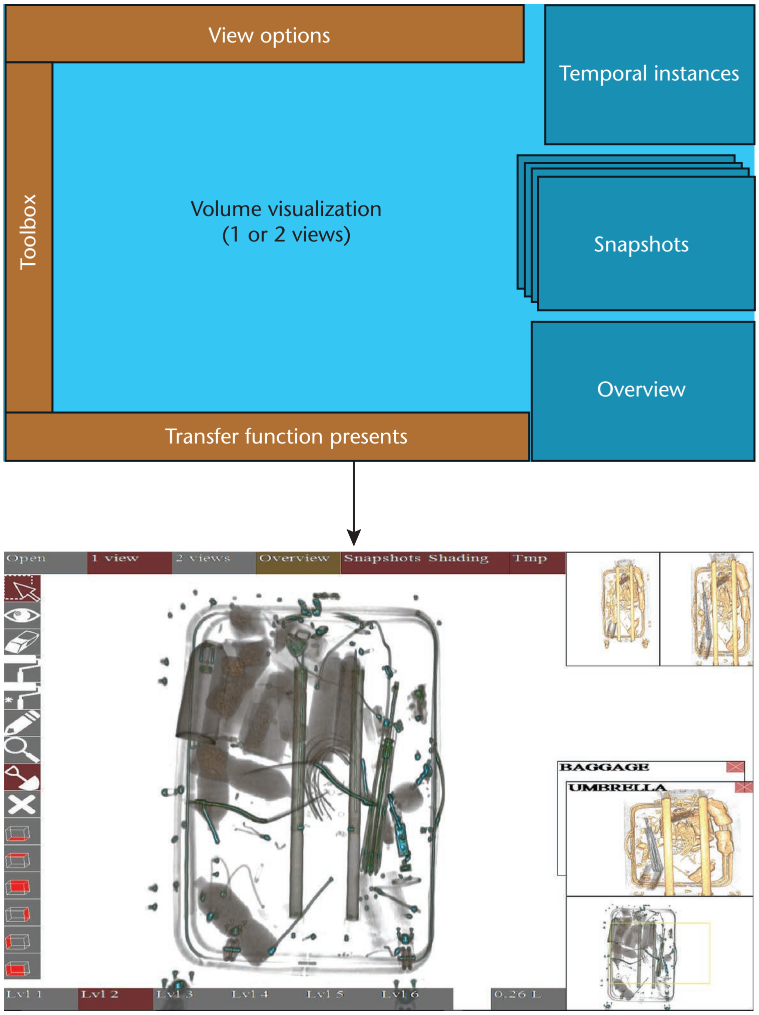

Figure 2. Top-level layout and screenshot of our interactive GUI. All the features are available through this main view. The bottom right corner shows the Overview, which indicates the location of the investigated area.

A temporal instances window, located in the top right of the interface, shows the current and previous volume visualization settings. The current setting is modified when the user changes the TF or manipulates an object. The snapshot view shows saved instances with their settings (pan, zoom, rotate, transfer function, select, and brush). The toolbox contains every interactive tool available for exploring the baggage (brushing, selecting, erasing, density picking, and navigating). The transfer function presets contain six predefined settings ordered by their filtering power. The low level shows every density, and the high level only shows high-density values.

Users can customize this global layout using the view options. For example, they can display one or

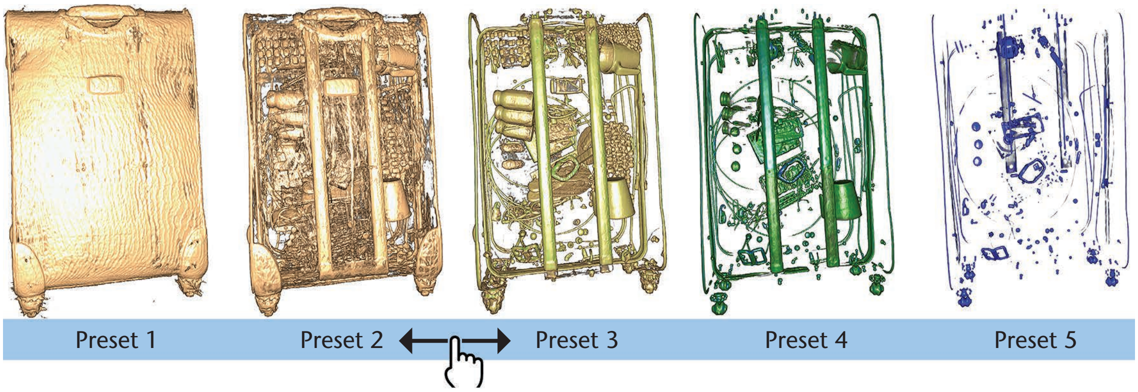

Figure 3. Transfer function presets and their continuous interaction. The high-density materials are revealed by dragging from left to right on the presets.

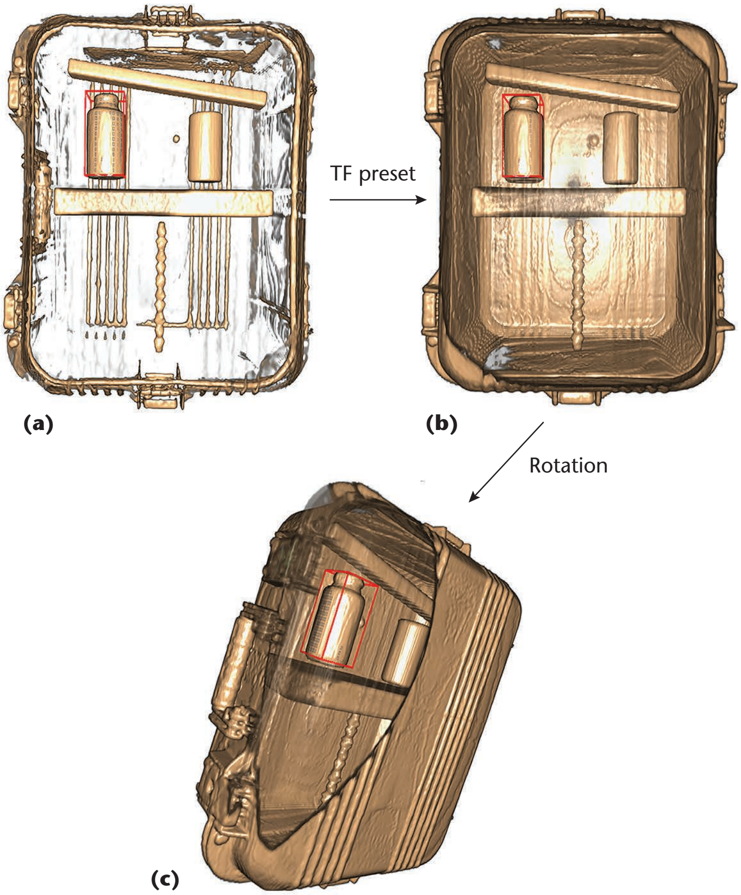

Figure 4. Inspecting an object from different perspectives. (a) When the user defines an object of interest for further inspection, this object is placed inside a box with red borders. (b) The user can then modify the transfer function to see the target in another context. (c) The user can also rotate the baggage to look at the potential threat from a different perspective. The object will stays visible regardless of the selected manipulation.

two views of the volume visualization, temporal instances, snapshot, and overview.

Interaction Techniques

To implement our interface, we used a Wacom 24HD touchscreen equipped with a stylus. Most of the developed interactions can be performed with every available modality: a mouse, hand, or stylus.

Transfer Function Presets

Datasets such as 2D raster images or 3D voxel-based representations are often processed for representation using a TF defined by a curve. Because airport security agents have a short timeframe for baggage exploration and limited knowledge of technical constraints, we defined six TF presets (see Figure 3). These presets modify only the TF transparency curve while keeping the same color mapping. We ordered these presets by their density-filtering power: the first preset displays every density, and the last one shows only the highest-density volumes (such as metals).

When a user clicks on a preset, the TF changes quickly, with a 1-second transition. The user can also select a TF located between two presets. To do so, the user drags from any location within the TF preset area until the volume visualization displays interesting features.

Objects Selection and Investigation

To investigate a specific object in detail, users can isolate it, remove surrounding items to address occlusion issues, or find a suitable point of view (see Figure 4).

Object Selection. We developed three different ways to select objects with the three different modalities: hand, mouse, or pen. When using a hand, the user double taps the desired object with a finger. When using the pen, the user presses the biggest button on the stylus while pointing at the target. Otherwise, the user double clicks on the target with the mouse pointer.

After the target is selected, our system tries to find a better point of view to display the selected object with the minimum occluded parts. We use a smooth transition to rotate and zoom the baggage.

If the user has activated the two volume visualization views by selecting the "2 views" button in the GUI's top view option panel, the selected object is isolated in the second view (on the right side of the main view in Figure 5). The selection is incremental and many objects can be selected, one after another. To undo a selection, the user has to select the desired object in the second view .

Occlusion Management. To address occlusion issues, the eraser tool can interactively remove selected objects from one view and place them in a second view. This interaction is similar to the selection interaction, and again all the modalities are available. When using a hand, the user double taps the desired object with a finger. When using the pen, the user presses the biggest button on the stylus while pointing at the target. The user can also double click on the target with the mouse pointer.

To restore a deleted object from the primary view, the user erases it from the second view. This simple principle guarantees that every item in the baggage is always visible while addressing occlusion issues (see Figure 6).

Extended Brushing Techniques

Baggage generally consists of both low- and high-density items. Low-density items (such as fabrics, clothes, papers, and organic object) are commonly more numerous and surrounded high-density items (such as metals and electronic components). To perceive a metallic object, a user can remove or hide its surrounding low-density items using an adequate TF. However, because the TF applies to the whole volume visualization, the visualization of a metallic bottle will prevent us from visualizing its likely lower-density content. That is, it is not possible to determine if a metallic bottle is empty.

To solve this problem, we developed two brushing techniques to explore the baggage by locally removing specific densities without modifying the TF (see Figures 7 and 8).

The first technique, range-based brushing, removes all voxels that are bounded in a user-defined range (see the range slider in Figure 8). The user enables the brush tool in the toolbox by clicking on it and then clicking the right button of the mouse to remove the voxels beneath the mouse pointer. The old brushing technique removed every voxel that was bounded in a density range and beneath the mouse pointer. When exploring a metallic bottle full of liquid, this technique removed both the bottle and its content (see Figure 7b). To address this issue, our brushing technique stops when a dense layer is encountered. This dense layer (for example, the outer layer of a metallic bottle) will act as a shield to protect voxels located behind it (see Figure 7c).

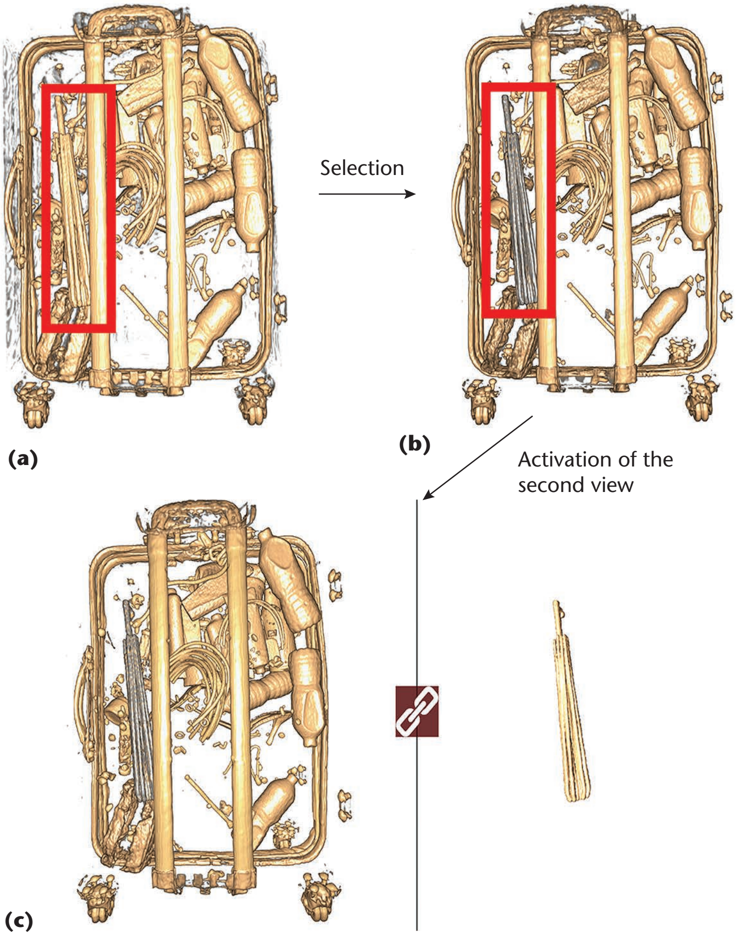

Figure 5. Selection of an umbrella for further inspection. (a) In this case, the user wants to investigate an object that looks like an umbrella. (b) When the user double clicks on the object, it then turns gray to show the selection. (c) This activates a second view, in which the user can manipulate the umbrella with or without the whole baggage. All selected items are available on this second view.

The second technique, magic brushing, has almost the same behavior as the first. The main difference is in the automatic range definition. Magic brushing removes all voxels with a lower density than the first one encountered at the beginning of the brushing process (see Figure 8b). This technique lets users directly define the densities they want to brush. This technique does not require multiple interactions with the histogram and its range slider when users define the range of brushable densities.

To restore previously brushed areas, our system also lets users cancel brushing. To do so, the user selects the brushing tool in the toolbox and holds the shift key while brushing over a given area. The restored densities are defined by the histogram range slider. This restoration process brings back every voxel that is bounded in a density range and

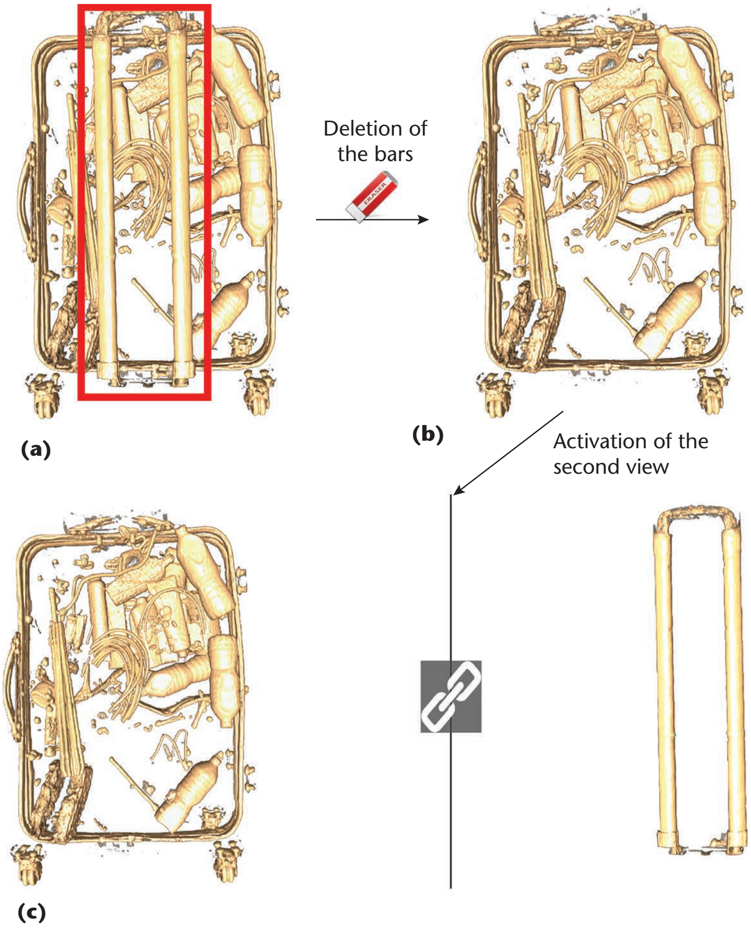

Figure 6. Addressing occlusion by removing objects. (a) In the case, the user wants to remove the bars in the baggage. (b) By double clicking with the deletion tool, the user can remove the bars from the main view. (c) This activates a second view, which shows the removed bars outside the baggage.

beneath the mouse pointer. All voxels within the range are restored so as not to confuse the user.

Snapshot

Our system can store visual configurations using snapshots. These snapshots, which are displayed on the right side of the main view, record the current rotation, zoom, pan, and TF. To take a snapshot, the user presses the space bar on the keyboard. Each snapshot can be given a name, which will be displayed on the top left of the snapshot's thumbnail. The user can restore a saved object by clicking on its snapshot. When a snapshot is selected, the system transitions (via an animation) from the current view to the selected snapshot (pan, zoom, and TF linear interpolation). A snapshot can be deleted by clicking on the close icon located on its top right.

Dual Temporal Instance Navigation

Our system provides many animated transitions when the user explores the baggage—for example, when changing the point of view, modifying the TF, or selecting an item.[5] To ease user navigation, we added two temporal instances representing the states before and after an automatic modification of the visual configuration. This interaction is based on the undo-redo paradigm. On the top right of the main window, we added two views representing the previous state before the transition and the final one after the transition. Users can navigate through this transition by dragging the cursor from one state to the other one. The closer the cursor moves to the center of a state, the closer the current view gets to its configuration. The user can also click on the desired state to directly transition to its visual configuration.

Technical Constraints and Implementation

Tomographs generate files that contain all the densities encountered by the x-rays through the baggage. These files usually contains approximately 30 million values. To load and display such a dataset in an interactive system, we used the parallel computation power of a GPU. We used a program written in C# combined with the CUDA parallel computing platform and a set of general-purpose GPU (GPGPU) techniques. To display the dataset, we used a standard ray-casting algorithm.

Object Selection

Our software features interactive selection and voxel manipulation functionalities. To implement these interactions, we had to find a way to extract objects from the volume data. To do so, we explored many object-detection algorithms such as contour tree[6] and branch tree[7] algorithms. These algorithms are computationally expensive and time-consuming, however, especially during preprocessing steps. In addition, such algorithms require many settings and modifications to process the variety of possible items present in baggage.

For these reasons, we developed our own selection algorithm that is based on a simple brute-force multithreaded propagation algorithm. The first step consists of casting a single ray toward the volume data where the selection needs to start (we stop its progression as soon as it hits a visible voxel). Second, we propagate the selection in every direction. For each encountered voxel, we check whether it is visible according to the current TF. If the voxel is visible, we go on with the selection propagation. We spread within the baggage by launching many threads, starting from the location where the first visible voxel was hit. The number of threads depends on the number of logical processors inside the current hardware. Each thread checks whether its neighbors are visible and then keeps spreading toward the visible voxels. This propagation algorithm stops when there aren't any more visible voxels to spread into.

Occlusion Minimization

Once the item of interest is selected, our system animates the visual configuration to display a new point of view for the selection that minimizes object occlusions. To do so, the system first computes a selection bounding box that consists of six faces. Then, the system computes the six possible points of views. For each of them, the system counts the number of visible pixels that display the selected object. This is done using the ray-casting algorithm, which propagates though the data cube and thus can test if the ray hits the selected items without any occlusion. Finally, the system defines the appropriate point of view as the bounding-box face with the highest number of visible pixels for the selected item. The system can then animate the volume visualization toward this new visual configuration and update the two temporal instances (before and after the pan, zoom, and rotate modifications).

Extended Brushing Technique

The extended brushing techniques help the user explore the volume without modifying the TF. This interaction is also based on the ray-casting algorithm. We only cast the rays located inside the radius of the brush in a parallel way. When a ray hits a visible voxel with a density that is within the brushing range, that voxel is removed. If the voxel density is not in the brushing range, the ray-casting algorithm stops.

The cancellation of the brushing is similar to the occlusion minimization algorithm. The rays start from the back of the volume and moves toward the front rather than moving from front to back. The second difference is that the cancelling technique restores the voxels with densities that are inside the range.

User Feedback

Because we developed this system with four baggage security practitioners, we had the opportunity to validate the usefulness of each developed technique. As part of an iterative design process, the users gave their feedback throughout the development process, which helped us assess and guide the proposed interaction techniques.

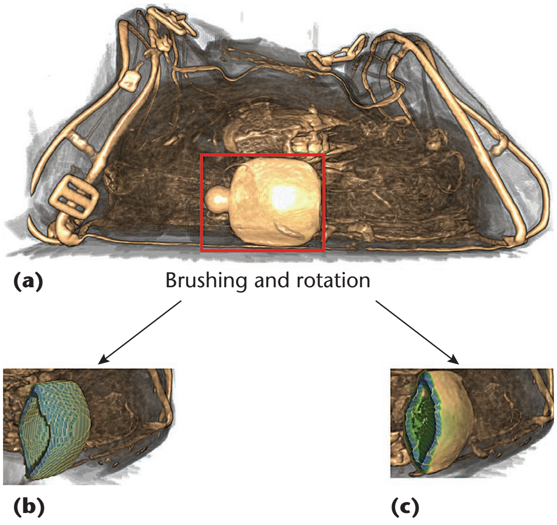

Figure 7. Brushing a bottle to see its content. (a) The initial baggage before brushing. (b) Using the old brushing technique, the bottle's content is removed as well. (c) Using our brushing technique, the content is still visible. This brushing technique removes specific densities until it encounters an object out of the selected density range.

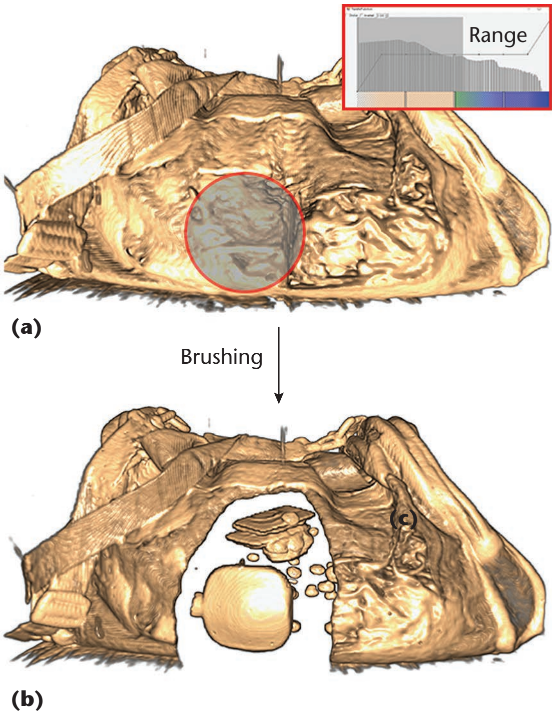

Figure 8. Brushing low-density materials to see a metallic object hidden inside the baggage. (a) The baggage before brushing. (b) After a density range is defined, the users brushes a part of the baggage, which reveals a metallic bottle.

The feedback was mainly positive, and the users did not face any specific difficulty when using our system. They appreciated the simple interface with few widgets, reduced set of interactions, smooth transitions, and incremental investigations of baggage. They did mentioned the need to display the actual density (the numerical value) of a selected object.

Surprisingly, the users were very interested in using the transfer function. The histogram and its TF were also appreciated, even though the corresponding technique was not simple to understand. We suppose that our interface motivates users to learn more regarding the technique behind it.

The users also asked many times if the displayed color corresponded to the one currently in use in operational settings. This confirms that users are willing to keep some existing features and prefer to use a system they are already familiar with.

Our system is fully functional and interactive enough to perform baggage explorations. The next step is to validate the positive user observations and qualitative feedback through formal evaluations. Future work could also involve improvements in terms of new interactions and time reductions. Technically speaking, the selection of the adequate point of view could be improved with more than six investigated faces, but in practice, this simple paradigm remains suitable. If the computed point of view is not fully satisfactory, users can manual rotate the baggage. Nevertheless, although the developed algorithm is not necessary the best solution (the lowest occlusion), it is satisfactory.

Our goal was to develop innovative interactions to support baggage exploration, but we did not attempt to optimize the exploration duration time. Manipulating the 3D volume and new interaction techniques have great potential, but they do take time. Existing investigation techniques that utilize 2D flattened images are suitable for quickly and efficiently detecting "clean" baggage. Because our tools require additional time, they can be a good solution for further investigating a potential threat and thus are most suitable for exploring suspicious baggage in more detail.

Acknowledgments

This work was supported by the TERANoVA project ANR JCJC 2014 (Agence Nationale de la Recherche, Jeunes Chercheuses Jeunes Chercheurs).

References

1. S.R. Deans, The Radon Transform and Some of Its Applications, Dover Publications, 2007.
2. W. Li et al., "Luggage Visualization and Virtual Unpacking," Proc. Workshop at ACM SIGGRAPH Asia (WASA), 2012, pp. 161–168.
3. B. Preim and C.P. Botha, Visual Computing for Medicine: Theory, Algorithms, and Applications, 2nd ed., Morgan Kaufman, 2013.
4. C. Hurter, Image-Based Visualization: Interactive Multidimensional Data Exploration, Morgan & Claypool Publishers, 2015.
5. B. Tversky, J.B. Morrison, and M. Betrancourt, "Animation: Can It Facilitate?" Int'l J. Human-Computer Studies, vol. 57, no. 4, pp. 247–262.
6. H. Carr, J. Snoeyink, and U. Axen, "Computing Contour Trees in All Dimensions," Proc. 11th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA), 2000, pp. 918–926.
7. V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli, "Multi-resolution Computation and Presentation of Contour Trees," tech. report UCRL-PROC-208680, Lawrence Livermore Nat'l Laboratory, 2005.

Michael Traoré *is a PhD student in data visualization at the French Civil Aviation University (ENAC). He investigates interactive pixel-based techniques to support big data exploration. Contact him at traore.s.michael@gmail.com.*

Christophe Hurter *is a professor in the Data Economics and Interactive Visual Laboratory (DEVI) at the Civil Aviation University (ENAC) and an associate researcher at the French Military Air Force Test Center (CReA). Contact him at christophe.hurter@enac.fr.*

Contact department editor Mike Potel at potel@wildcrest.com.