2D Contour Reconstruction for Industrial Computed Tomography using Crease Cluster

Maximilian Wattenberg\textsuperscript{1,2}, Maik Stille\textsuperscript{1}, Philipp Klein\textsuperscript{2}, Thorsten M. Buzug\textsuperscript{1}

\textsuperscript{1}Institute of Medical Engineering, Universität zu Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany, e-mail: \{wattenberg, stille, buzug\}@imt.uni-luebeck.de

\textsuperscript{2}YXLON International GmbH, Essener Bogen 15, 22419 Hamburg, Germany, e-mail: philipp.klein@hbg.yxlond.com

Abstract

One task of industrial computed tomography in dimensional metrology is to provide surface models of examined objects. Currently, in the standard approach polygonal surface meshes from reconstructed tomograms are extracted. Precise surface meshes require a high resolution tomogram, leading to time-consuming reconstruction algorithms. In order to overcome this problem, we propose a method to approximate the contour and folds in the surface by a point cloud, which resemble the most characteristic parts of the object. These creases are reconstructed from filtered projection images directly in Cartesian space. This cloud of creases can serve as a basis for the creation of meshes. Initial results of a 2D phantom study indicate that the method reaches reconstruction accuracies comparable to standard methods while the number of necessary projections is reduced significantly, leading to a notable decrease in scan time.

Keywords: surface, point cloud, reconstruction, crease, cluster, edge, fold

1 Introduction

Industrial computed tomography (CT) in dimensional metrology is used to examine if manufactured products comply with the specifications defined by a computer-aided design (CAD) model. Primarily surfaces are detected in the object’s representation, the reconstructed tomogram, and compared with CAD models. The surface mesh accuracy depends strongly on the image quality of the reconstructed slices, whereby a high detector and tomogram resolution, a large number of projection images, and advanced algebraic reconstruction algorithms support the quality. These parameters rise the expense of calculation during reconstruction, demanding for algorithms providing high-quality surface meshes on low calculation costs. The number of projections can drastically be reduced by the method of Jinnouchi et al., which detects and binarizes edges in projection images and reconstructs a tomogram from a few preprocessed projections \[1\]. The tomogram then outlines the object surface-regions of strong and rapid curvature, referred to as creases. These creases mark the most prominent structures of the scanned object and might be sufficient for some measurement tasks. However, the accuracy of surface vertices calculated using the method of Jinnouchi et al. still relies on the resolution of the tomogram. The presented algorithm aims at overcoming this drawback by calculating the crease locations in Cartesian space without extracting points from a tomogram. Instead, a clustering approach is proposed.

![Figure 1: a) The rays mark the way from the X-ray source s to an edge in the projection image \(p_i\). The ray crossings from two different read-out angles (\(\phi\) and \(\psi\)) resemble possible locations for object creases. Figure b) shows the ray intersections for another angle pair. Figure c) contains all intersections from a) and b), whereas valid creases (larger dots) are visible in multiple projection pairs.](image)

2 Material and Methods

The presented approach uses binarized edges of the projection images of pairs of projections. The ray paths from the focus of the X-ray source to the binarized edges are calculated for two system geometries. The intersections of the rays from two
projections for different pairs will cross more frequently at the object’s crease locations. Therefore, the creases can be grouped by a clustering approach described below. The algorithm is compared with a tomogram-based crease detection approach based on a digital step wedge phantom.

2.1 Crease Cluster Reconstruction

The direct reconstruction of creases in Cartesian space is performed by an edge-detection in the projection profile, calculation of possible crease locations, grouping of possible crease locations into clusters, and thresholding the cluster by the number of creases. This procedure is described in detail in the sections below and outlined by the pseudocode in Algorithm 1.

Data: projection_data
Result: creases of the scan object
cluster_list = [];
while max number of reconstruction steps not reached do
    projection_a = select_projection_randomly(projection_data);
    projection_b = select_projection_randomly(projection_data);
    while projection_b == projection_a do
        projection_b = select_projection_randomly(projection_data);
    end
    lines_a = calculate_source_closest_rays(projection_a.focus_position, projection_a.creases);
    lines_b = calculate_source_closest_rays(projection_b.focus_position, projection_b.creases);
    crease_candidate_list = calculate_all_line_intersections(lines_a, lines_b);
    if cluster.empty then
        temp_cluster_list = [];
        forall crease_candidate in crease_candidate_list do
            new_cluster = convert_to_cluster(crease_candidate);
            temp_cluster_list.add_cluster(new_cluster);
        end
        cluster_list.add_cluster_list(temp_cluster_list);
    else
        temp_cluster_list=[];
        crease_added_to_existing_cluster = false;
        forall crease_candidate in crease_candidate_list do
            forall cluster in cluster_list do
                if crease_candidate in cluster.acceptance_area then
                    cluster.add_crease(crease_candidate);
                    crease_added_to_existing_cluster = true;
                end
            end
            if not crease_added_to_existing_cluster then
                new_cluster = convert_to_cluster(crease_candidate);
                temp_cluster_list.add_cluster(new_cluster);
            end
        end
        cluster_list.add_cluster_list(temp_cluster_list);
    end
end

Algorithm 1: Pseudocode of the 2D crease reconstruction algorithm. Object creases are detected by calculating ray intersections from two different projections.

2.1.1 Detection of Projection of Creases

Strong and rapid curvature in the surface of a scanned object, referred to as a crease, is characterized by strong gradient changes in the projection. In order to reconstruct creases directly in the Cartesian space these edges in the projection profile need to be detected with high accuracy. Therefore, the projection is re-sampled by the factor of 5 using Akima cubic spline interpolation [2]. The edges are detected with the second cubic spline derivative and binarized with a subsequent thresholding. The threshold \( \tau \) was set to

\[
\tau = \min \left( f' \right) + 0.1 \cdot \left( \max \left( f' \right) - \min \left( f' \right) \right),
\]

where \( f' \) marks the filtered projection profile. Although, most industrial CT systems have a static source-detector geometry with a manipulator moving the scanned object according to the specified trajectory, our reconstruction approach will be explained in
a static object geometry. Thus, the thresholded pixels are converted from detector coordinates to object coordinates by taking the projection angle into account. These projection creases in the object coordinate system resemble the projection of the object creases on a detector rotated around the scanned object (see Figure 1).

2.1.2 Crease Candidates

In the paper of Jinnouchi et al. the binarized projection images are backprojected, accumulating high intensities in voxels, which underlay the object’s crease locations [1]. This principle is adapted to avoid using a tomogram by calculating ray intersections from two different projection images. Therefore, the coordinate of the X-ray source is converted, respecting the projection angle, into the object coordinate system. This guarantees, that the projection creases and the source focus are virtually positioned relative to a steady object, similar to the setup of medical CTs. The ray paths from the source \( s^{\phi_1} \) to all projection creases \( p^{\phi_1}_{i} \) are calculated as well as the rays from source \( s^{\psi_1} \) to the projection creases \( p^{\psi_1}_{j} \). Subsequently, all intersections between the rays under the projection angle \( \phi_1 \) and the rays under projection angle \( \psi_1 \) are calculated. These crease candidates arise at real object’s crease locations or are false positives. For multiple projection angle pairs more crease candidates will be present at locations of real object creases. In contrast, false positive crease candidates are more likely to be located solitarily. In order to distinguish false positive from real creases a clustering approach is used.

2.1.3 Crease Cluster

Clusters are a common structure in big data analysis and are used to group data points with similar characteristics like location and distinguish them from other groups. K-means [3] and other algorithms suffer from the need to specify the number of clusters in advance. As the structure of the object, which can be unknown, defines the necessary number of creases in a reconstruction the number of clusters cannot be determined beforehand. Therefore, the object creases will be detected out of all crease candidates by a clustering method with a variable number of clusters. A single cluster is defined by a set of creases, each representing a coordinate in the object coordinate system. The centroid of the cluster is the mean coordinate of all creases.

![Figure 2](image.png)

Figure 2: A section of the field of view is shown for different projection pairs. The crease candidate (dot) of a ray intersection is converted to a cluster (a). Crease candidates from other reconstruction steps build clusters on their own, when they lie outside the circular acceptance area (b) or are added to an existing cluster otherwise (c). This relocates the centroid (cross) marking the object crease with increasing accuracy. Non-valid crease clusters can be removed by applying a threshold to the number of intersections (d).

For one step in the crease cluster reconstruction, a set of crease candidates is calculated from two projection angles. In the first reconstruction step, each crease candidate is used to initialize a separate cluster. Thus, each cluster only contains a single crease, which defines the centroid location solely (see Figure 2). The crease candidates of following projection pairs are evaluated for their distance to any existing cluster. If a crease candidate is in the acceptance area around a centroid, it is added to the corresponding cluster. If it is inside the acceptance area of multiple clusters, it is added to the closest cluster. In the case that a crease candidate is in no range of acceptance to any cluster, it will initialize a new cluster. Many distance calculations between crease candidates and centroids are necessary, and the number of false positive crease candidates rises constantly. In order to reduce the number of false positives the clusters are equipped with a counter. Whenever a crease candidate is converted to a cluster its counter is set to a pre-defined time of probation. The counter will be decreased in every reconstruction step. During the time of probation, creases can be added to the cluster. When the counter reaches zero and no additional crease was added to the cluster, the cluster will be removed from the set of clusters. For probation, a number of 5 reconstruction steps has shown to be sufficient.

2.1.4 Thresholding Clusters

Even though new clusters are equipped with a probation, not all false positive clusters might be rejected and remain in the set with a small number of elements. As these false positive clusters do not resemble an object crease and will separable from clusters at real object crease locations by the number of creases, they can be distinguished. In order to separate clusters located at real object creases from false positives, the clusters are thresholded by the number of creases they contain.
2.2 Harris Corner Detection-based Crease Extraction from Tomogram

In comparison to the proposed algorithm, creases were also extracted with subpixel accuracy from two reconstructed images using Harris corner detection algorithm of OpenCV 3.4.1 [4, 5]. For this purpose, two images of the size 600 × 600 were reconstructed with Volume Graphics Studio Max 3.2 from 720 projections. The first slice was reconstructed using filtered backprojection (FBP) approach, the second using algebraic reconstruction technique with 8 iterations and a relaxation of 0.1.

![Draft of the CAD model of the digital step wedge phantom with the length, height and width given in mm. All steps have a height of 4 mm. The red crosses mark the crease locations to be reconstructed.](image_url)

2.3 Evaluation

The tomogram-based crease detection and the direct crease reconstruction were evaluated on a digital step wedge model (see Figure 3. The step wedge has the dimensions 200 × 40 × 40 mm (length, height, width) and consists of ten evenly spaced steps so that each step has a length of 20 mm and a height of 4 mm. The step wedge was scanned with a scan simulation for X-rays, called aRTist, presented by Bellon et al. [6]. The simulation was performed with a magnification of 3, a detector with 600 elements, and a pixel size of 0.12 mm, using a circular trajectory of 360° with either 720 and 36 sampling points along the trajectory. The two projection sets of 720 and 36 projections were reconstructed with the ART and FBP implementation of Visual Studio Max leading to four reconstruction images. The crease locations were determined with the above-described method of Harris corner detection. Creases were also reconstructed with the proposed crease cluster reconstruction using the same numbers of projections. Subsequently, the clusters of the crease clusters reconstruction were manually thresholded by their number of creases. The Euclidean distance (see Table 1) from the corners of the CAD model to the closest calculated creases was evaluated for all three methods. Only creases with a distance below 1.5 pixel were considered as detected. The number of detected creases was counted and listed in Table 1.

3 Results

Figure 4 shows the reconstructed images for ART and FBP method with the overlaying detected crease locations of the Harris corner algorithm visualized by red crosses. The reconstructed images of the ART algorithm from 720 and 36 projections both show a circular image artifact. The Harris corner method detects some of these artifacts as creases (see Figure 4 a). The reconstructed images of ART and FBP from 720 projections have a high similarity, neglecting the artifacts just described. In contrast, the reconstruction results of the two methods from 36 projections in contrast, differ significantly. The gray value between material and background pixels have a higher contrast in the ART reconstruction compared to the FBP result. Figure 4 also provides the centroid locations of the reconstructed clusters. As no image is reconstructed using this method, the CAD model contour is provided for visual guidance.

Consulting Table 1 all three methods reach very high accuracy within subpixel precision between 0.26 and 0.28 pixel. Using all 720 projections, Harris corner-based approach was able to detect 21 of the 22 corners of the CAD model, whereas the crease cluster reconstruction was able to detect 20 creases. Both image-based creases detections falsely located creases at image artifacts or at straight edges of the object contour, which do not resemble object creases. False positive clusters of the crease cluster reconstruction were successfully discarded by the thresholding process.

For the image-based crease detection from only 36 projections only 2 creases for the ART-based and only one for the FBP-based approach were successfully detected. In contrast, the crease reconstruction detected the same amount of creases from 720 as well as for 36 projections. However, the precision of the detected creases decreased with the reduced number of projections from 0.28 pixel to 0.53 pixel.

4 Conclusion

The present study compared a tomogram-based approach to find object creases with a tomogram independent algorithm, which aims to detect creases by a clustering approach. For densely sampled projection angles all presented methods showed a similar accuracy in finding creases. For a sparse trajectory with only 36 projections the image-based algorithms only found very few
<table>
<thead>
<tr>
<th>Number of Projections</th>
<th>Distance in pixel</th>
<th>Number of Detected Creases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FBP</td>
<td>ART</td>
</tr>
<tr>
<td>720</td>
<td>0.28±0.21</td>
<td>0.26±0.21</td>
</tr>
<tr>
<td>36</td>
<td>0.26±0.00</td>
<td>0.92±0.63</td>
</tr>
</tbody>
</table>

Table 1: Mean distances of the creases to the real CAD model’s corner locations and the number of successfully detected creases, for the different methods are listed. Only creases with a distance closer than 1.5 pixel to a real CAD model corner were considered as successfully detected. The result is shown for different numbers of projections and images, as well as clusters, based on the different crease detection approaches.

corners of the CAD model, whereas the clustering approach achieved similar results with only a slightly worse precision. Therefore, the clustering approach gains precision with an increasing number of projections but shows good results in an early stage of reconstruction. Furthermore, the independence from a tomogram motivates for high-resolution detectors, which will increase the precision even further. Even though the same argument would be valid for a tomogram-based method, it is directly connected to high computational costs, as the tomogram resolution would have to increase similarly.
The presented method indicates the potential for dimensional measuring in non-destructive testing. Such a tomogram-independent approach might handle the combination of high detector resolution with reasonable reconstruction times. In the future, the algorithms suitability for more complex object geometries and multi-material scan objects must be investigated.

Figure 4: Calculated crease positions of the step wedge are shown. Figures a) and d) show the image reconstructed with ART. Figures b) and e) show the creases detected in the image reconstructed using FBP. c) and f) visualize the centroids of the reconstructed crease clusters overlayed to the ideal CAD model contour (blue line). The first row shows the results calculated from 720 projections, whereas the lower row shows the results from 36 projections.
References


