Title:
RGB-Depth Image Segmentation using Interproximation with Cubic B-Spline Curves

Authors:
Anastasia Kazadi, ansm226@g.uky.edu, University of Kentucky
Fuhua (Frank) Cheng, cheng@cs.uky.edu, University of Kentucky
Seifalla A. Moustafa, seifalla.moustafa@uky.edu, University of Kentucky
Joyce Yang, joyceyang@uky.edu, University of Kentucky

Keywords:
RGB-Depth, Segmentation, Pixel-Based Classification, Interproximation, Cubic B-Splines

DOI: 10.14733/cadconfP.2024.232-237

Introduction:
Advancements in the development of high-grade and commodity RGB-D sensors have reshaped the landscape of computer vision **[6-8].** Related goals and tasks have been changed and are constantly evolving. The integration of RGB-D data in image segmentation plays a pivotal role in advancing computer vision applications by providing a more accurate representation and a semantic interpretation of visual scenes **[9].** The importance of newly developed methods lies in the combination of RGB information from color images and direct spatial depth details derived from depth sensors, which otherwise would be hard to impossible to infer from traditional images. This paper focuses on using depth data to solve edge detection and image segmentation problems. We developed a state-of-the-art algorithm that manages both problems by employing a local pixel-based data classification and a global interproximation-based edge smoothing technique. Furthermore, we simplify the task of segmentation to the task of edge detection by outlining the boundary of selected image objects **[9-10].**
This paper presents a combined solution for edge detection and segmentation problems. The main contributions of this work are:

- Local approach to semantic pixel classification based on available information about neighboring pixels (the initial guess for boundary and the follow-up depth correction).
- Global approach for estimating edges of an object based on minimizing the energy of the curve representing the boundary.

Method Description:
Step 1. Initial Boundary Guess: Pixel-Based Classification
Pixel-based classification of depth data serves as the foundational step in formulating an initial estimation of the object boundaries within the scene. The program begins by extracting depth information from a given depth image where each pixel corresponds to a specific distance extending from a sensor to a point on the surface of an object in the scene. From the user-selected point, the program initiates an expansion process, systematically adding and traversing rows in all four directions across the depth image. During this expansion, pixels are classified according to a predefined scheme, attributing semantic labels based on the discerned characteristics. The output of the program is a comprehensive classification matrix encapsulating assigned labels for each pixel within the selected object and a bounding polygon that succinctly outlines the spatial extent of the object.

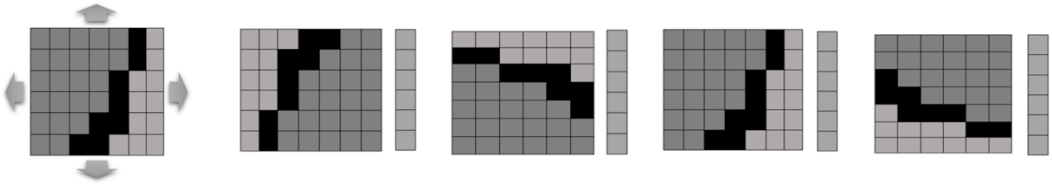

Fig. 1: Four-way expansion used in the search for object's boundary: (a) Indented directional expansion (b) Left expansion: image is rotated by -180 degrees (c) Bottom expansion: image is rotated -90 degrees (d) Right expansion: image is not rotated (e) Top expansion: image is rotated -270 degrees.

We simplified the implementation of the algorithm by treating all four search directions as a unified entity and opting for a single direction – in this case, arbitrarily chosen as *right*. Before engaging in pixel classification, the program flips the image, aligning it with the right expansion direction. Once the classification for a specific direction concludes, the image is reverted to its original orientation. Pixel-based classification heavily relies on previously categorized pixels supplemented by depth map data when existing information is insufficient. In certain scenarios, depending solely on prior classification and depth map data, it may also prove insufficient for accurately assigning a semantic label. To address this limitation, the algorithm compensates by pre-computing average depth values in the vicinity of the pixel undergoing classification.

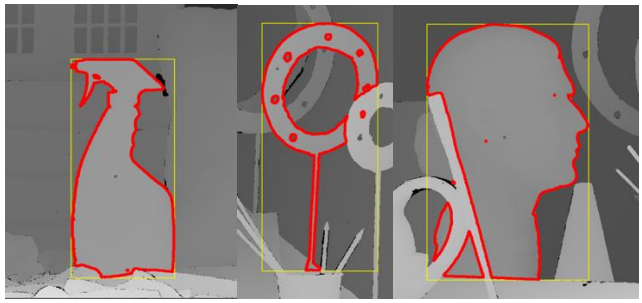

Fig. 2: Sample results for pixel-based semantic classification.

Step 2. Construct a list of points and regions from the data obtained in the previous step.
The previous step yields the classification matrix of the object pixels and the corresponding bounding polygon. In this phase, our objective is to extract an ordered list of exact points and uncertain regions. The procedural logic behind this process is straightforward and intuitively structured. Starting with the identification of the initial boundary pixel, the search for subsequent pixels is performed using a 3x3 sliding window and positioning the current boundary pixels at the center of the window. The algorithm explores the immediate vicinity of the next boundary pixel, marking each discovered pixel as visited. The classification matrix is traversed until no further boundary pixels are found. Upon completion, the algorithm returns to the starting point, verifying whether the edge extends into additional directions. The output is an ordered list of exact points and uncertain regions of the boundary representing an outline of the selected object.

Step 3. Identify points and regions to be used in interproximation based on measurement of curvature.
After constructing the ordered list of points and regions, the subsequent objective is to select only the essential points. Typically, an object's boundary may encompass hundreds of points. However, the interproximation algorithm does not necessitate the inclusion of all these points. The point selection process is guided by the curvature of the edge segment. At each boundary point, a curvature is calculated by fitting a circle to three points: a current boundary point and two additional points that are located *n*-

pixels away. The magnitude of the boundary curvature is the reciprocal of the radius of curvature, $\kappa = 1/R$, nd is calculated by finding the radius of the osculating circle using three neighboring points[12]. When curvature is pronounced, more points are chosen; conversely, fewer points are selected. This criterion specifically applies to exact points, although all uncertain regions are included in the selection.

Step 4. Interpoximation algorithm.
We propose a method to assist in edge detection and object segmentation that uses an interproximation scheme with cubic B-spline curves, aiming at producing smooth curves over a grid of given data. This method combines the two classic concepts of *interpolation* and *approximation*. On one hand, a typical interpolation is an estimation method of constructing new data points within a range of a given set of known data points. Unfortunately, interpolation on its own is not suitable for solving the problem at hand because of the uncertainty of noisy regions and the unknown range of data points in the depth map. On the other hand, approximation is a data fitting method that performs data fitting by backward error analysis and does not guarantee that the resulting curve or surface will pass through specific points. This is why we use the hybrid approach of data fitting, which uses both interpolation and approximation. The resulting curve will interpolate exact data points and pass through uncertain regions in such a way that it has minimum energy at each of its piecewise components and gives a relatively accurate estimation of object shape [1].

A specialized interproximation scheme with piecewise curve energy minimization using non-uniform cubic B-spline curves was first proposed by F. Cheng and B. Barsky [1-3] to address the problem of data uncertainty. The idea of having a curve passing through the exact points and uncertain regions is very suitable for our problem of denoising depth map data.

Problem formulation
Let $\{Depth_i \mid i = 1,2,\dots,n+m\}$ be a set of two-dimensional depth data where $Depth_i$ could be either a point or a region. We denote exact points as:

$$Depth_{i_j} = P_j = (x_j, y_j) \text{ for j=1,2, …, n} \tag{4.1}$$

Uncertain regions could be represented by an arbitrary one-dimensional or a two-dimensional shape; regardless of the specific shape or size, the region is enclosed in a bounding box as shown in Figure 4 and denoted by:

$$Depth_{i_k} = X_k \times Y_k = [x\_start_k, x\_end_k] \times [y\_start_k, y\_end_k] \text{ for k=1,2, …, m} \tag{4.2}$$

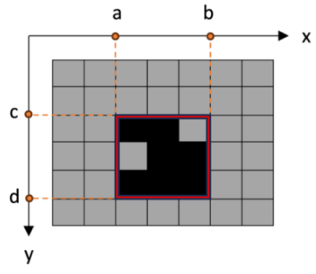

Fig. 3: Example of a bounding polygon for an uncertain region.

We adhere to the conventional interpolation scheme to construct an open or closed cubic B-spline curve comprising of n+m endpoints and n+m-1 for an open curve and n+m for a closed curve piecewise segments to fit given data. Such a B-spline requires n+m+2 contributing base functions of degree three and n+m+6 parametric knots. We denote the set of parametric knots in the range from 0 to 1 generated using the *centripetal* parameterization technique [4, 6]:

$$T = \{t_{-2}, t_{-1}, t_0, t_1 \dots, t_{n+m}, t_{n+m+1}, t_{n+m+2}, t_{n+m+3}\} \tag{4.3}$$

B-spline is constructed using contributing basis functions, which are defined by the *recurrence* relation:

(1) *Base case*. Reduction of the degree of the basis function stops when it reaches a degree of zero and is represented by a line having a value of one at the starting knot and zero elsewhere.

$$N_{i,0}(u) = \begin{cases} 1 & for\ t_i \le u < t_{i+1} \\ 0 & otherwise \end{cases} \quad (4.4)$$

(2) *Recursive case.* Computation of each of the following basis function depends on the two previous functions of lower degree:

$$N_{i,k}(u) = \frac{u-t_i}{t_{i+k}-t_i} N_{i,k-1}(u) + \frac{t_{i+k+1}-u}{t_{i+k+1}-t_{i+1}} N_{i+1,k-1}(u) \quad for\ k \ge 1 \quad (4.5)$$

For the system of equations, the dimension of piecewise B-spline curve is $n+m+2$, but we only have $n+m$ fitting conditions. The solution for that, as in traditional interpolation, is to impose *two extra* end-conditions; 2nd derivative at the first and the last knots should be equal to zero:

$$f^{(2)}(t_1) = 0 \ \ and \ \ f^{(2)}(t_{n+m}) = 0 \quad (4.6)$$

Therefore, given a set of exact points and a set of uncertain regions, *f(u),* a two-dimensional cubic B-spline curve is defined by the sum of its piecewise components with respect to the knot sequence:

$$f(u) = \sum_{i=1}^{n} \alpha_i g_i(u) + \sum_{j=1}^{m} \beta_j h_j(u) \quad for\ u \in [0,1] \quad (4.7)$$

where α_i and β_j are two-dimensional control points and g_i and h_j are basis functions of degree three contributing to the B-spline curve in the parameter range starting at t_1 up to t_{n+m} for exact points and uncertain regions, respectively. The problem of interpoximation can be finalized and formulated as follows. Given a set of data, find a B-spline curve that interpolates exact points P_j:

$$f\left(t_{i_j}\right) = P_j \qquad for\ j = 1,2, \ \ldots, \ n \quad (4.8)$$

And approximates regions $A_k \times B_k$:

$$f\left(t_{i_k}\right) \in X_k \times Y_k \quad for\ k = 1,2, \ldots, m \quad (4.9)$$

The objective is to find a curve f such that it satisfies conditions (1) and (2). But for the given data set, there are exact points and uncertain regions, therefore, there is an *infinite* set of such possible B-spline curves. We desire to find a curve $\hat{f}$ in a set of all possible curves F, that has the smoothest shape or the minimum energy [6]:

$$\left|\left|\hat{f}\right|\right| = \min\{\ ||f||\ \mid\ f \in F\ satisfies\ conditions\ (4.8)\ and\ (4.9)\} \quad (4.10)$$

Where the energy of the curve is defined by an integral:

$$||f|| = \int_0^1 \left[f^{(2)}(u)\right]^2 du \quad (4.11)$$

Step 5. Optimization

To perform optimization of the integral function, we use *fmincon* routine [12], a nonlinear programming solver which optimizes the objective function $||f||$ subject to nonlinear equality and inequality constraints. In addition, we employ an interior-point method designed for solving large-scale constrained optimization problems, which iteratively moves towards the optimal solution while staying within the feasible regions defined by customary constraints represented by boxes bounding the uncertain regions [12]. The algorithm balances the trade-off between feasibility and optimality, ensuring convergence towards a feasible solution with a minimized objective function. The output of *fmincon* is an optimal solution for β-control points which is used to compute α-control points for the final cubic B-spline curve construction.

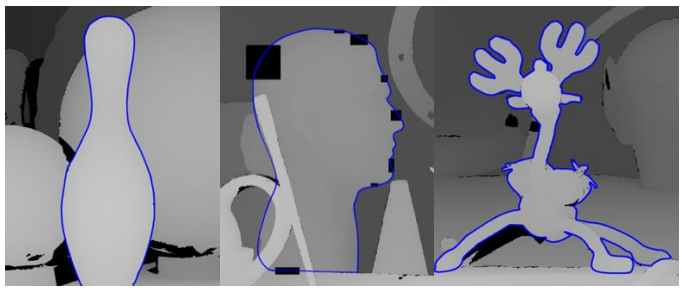

Fig. 4: Sample interpoximation results.

Step 6. Fill in holes and smooth ragged edges using neighborhood depth information.
In the last step we perform depth adjustments based on depth information. False negatives, false positives, and uncertain depth values are targeted for adjustment. False negatives are the depth values which were initially identified as not being on the object but were determined to be within the object after the interproximation. False positives are the depth values initially identified as being on the object but were determined to be outside of object's boundaries after interproximation. Uncertain depth values are the failed signals read by the depth sensor. All values adjustment is performed with using weighted differences mean algorithm: the weighing coefficient for each targeted depth value is computed as inverse mean value between the current depth and depths chosen from the closest neighborhood:

$$depth'_{(i,j)} = \frac{\sum_{i=1}^{n} w_i * depth_{i,j}}{\sum_{i=1}^{n} w_i}$$

Where $w_i = k/(\sum_{j=1}^{n} |depth_i - depth_j|)$ and k is some positive integer, in most cases chosen in range [1, n-1]. This method is selected because it is less sensitive to outliers than other variations of regular arithmetic mean[13].

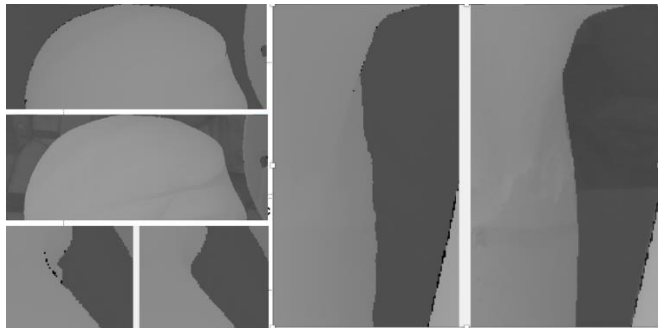

Fig. 5: Pixels identified as being within/on the boundaries of the B-Spline

Conclusions:
This paper addresses the challenge of edge detection and segmentation of the depth maps. Our innovative method integrates both local and global approaches to improve the precision of detection and formation of object edges represented by open or closed cubic B-spine curves. Interproximation results are improved by measuring the curvature of the edge segments. Thus, more points are chosen if the curvature is great and less otherwise. The issue of data uncertainty is addressed using interpolation at exact points and approximation at uncertain regions. Energy minimization is utilized to create the smoothest optimal curves. Future works in this direction include the extension of the algorithm into a three-dimensional framework, which will play a crucial role in reconstructing surfaces within the scene by utilizing comprehensive data integration from RGB images, depth data, lighting, and texture details.

References
[1] Cheng F.; Barsky, B.A.: Interproximation using Cubic B-Spline Curves. In: Falcidieno B., Kunii T.L. (eds) Modeling in Computer Graphics. IFIP Series on Computer Graphics. Springer, Berlin, Heidelberg, 1993.
[2] Cheng, F.; Barsky, B.A.: Interproximation: Interpolation And Approximation Using Cubic Splines Curves, Computer-Aided Design, 23(10), 1991, 700-706.
[3] Cheng, F.; Wang X.; Barsky, B.A.: Energy and B-spline Interproximation, Computer-Aided Design, 29(7), 1997.
[4] Lee, E.T.Y.: On Choosing Nodes in Parametric Curve Interpolation, Computer-Aided Design, 21(6), 1989, 363-370.
[5] Farin, G.E.: Curves and surfaces for computer aided geometric design: A practical guide. Academic Press, 1988. https://doi.org/10.1016/B978-0-12-460515-2.50020-2.

[6] Bartel, R.; et. al: An Introduction to Splines for Use in Computer Graphics and Geometric Modeling, Morgan Kaufmann Publishers, Inc., San Mateo, California, 1987.
[7] Holladay: Smoothest Curve Approximation, Math Tables Aids Computation, vol. 11, pp.233-243, 1957.
[8] Kirillov, A; Mintun, E.; Ravi, N; Mao, H; Rolland, C; Gustafson, L; Xiao, T.; Whitehead, S.; Berg, A.C.; Lo, W.; Dollár , P.; Girshick, R.: Segment Anything, 2023.
[9] Han, X.; Laga, H.; Bennamoun, M.: Image-Based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era, in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 43, no. 05, pp. 1578-1604, 2021. https://doi.org/10.1109/TPAMI.2019.2954885.
[10] Zollhöfer, M.; Stotko, P.; Görlitz, A,; Theobalt, C.; Nießner, M.; Klein, R.; Kolb, A.: State of the Art on 3D Reconstruction with RGB-D Cameras, Computer Graphics Forum, 2018. 37.625-652. 10.1111/cgf.13386.
[11] Bowyer, K.W.; Chang, K.; Flynn, P.: A Survey Of Approaches And Challenges In 3d And Multi-Modal 3d+2d Face Recognition, Computer Vision and Image Understanding, 101(1), 2006, 1–15.
[12] The MathWorks, Inc.: Optimization Toolbox version: 9.4 (R2022b). Accessed: January 01, 2023. Available: https://www.mathworks.com.
[13] Dodonov Y.S., Dodonova Y.A.: Robust measures of central tendency: weighting as a possible alternative to trimming in response-time data analysis, Psikhologicheskie Issledovaniya, 5(19), 2011.