

Sample Consensus Fitting of Bivariate Polynomials for Initializing EM-based Modeling of Smooth 3D Surfaces

Christian Nissler, Zoltan-Csaba Marton and Michael Suppa

Abstract—This paper presents a method for finding the largest, connected, smooth surface in noisy depth images. The formulation of the fitting in a Sample Consensus approach allows the use of RANSAC (or any other similar estimator), and makes the method tolerant to low percentage of inliers in the input. Therefore it can be used to simultaneously segment and model the surface of interest. This is important in applications like analyzing physical properties of carbon-fiber-reinforced polymer (CFRP) structures using depth cameras.

Employing bivariate polynomials for modeling turns out to be advantageous, allowing to capture the variations along the two principle directions on the surface. However, fitting them efficiently using RANSAC is not straightforward. We present the necessary pre- and post-processing, distance and normal direction checks, and degree optimization (lowering the order of the polynomial), and evaluate how these improve results. Finally, to improve the initial estimate provided by RANSAC and to stabilize the results, an Expectation Maximization (EM) strategy is employed to converge to the best solution.

The method was tested on high-quality data and as well on real-world scenes captured by a RGB-D camera.

I. INTRODUCTION

Carbon-fiber-reinforced polymer (CFRP) structures are promising materials for light-weight production by providing good material characteristics at a low weight. CFRP structures are composite structures and therefore feature very diverse material properties. A possibility to estimate those properties is to induce a known motion on the material and visually observe the behavior of it. This leads to the challenge of obtaining a three-dimensional map of the structure, segmenting it from the background, and at the same time finding a mathematical model describing this structure.

Obtaining a three-dimensional image of a CFRP structure is complex because of several reasons: A CFRP element typically possesses a dark, glossy, fabric-like surface with regular patterns. This makes it hard for passive sensors to find non-ambiguous points. Another problem is that because of the glossy surface and especially with parabolic alignments like in our setup (Fig. 1), specular reflections can occur. The mentioned difficulties led us to use a Microsoft Kinect RGB-D sensor, because it is an active sensor emitting a pseudo-random infrared laser pattern [1]. This reduces the difficulties with the very challenging surface. Still reflections creating blind spots on the surface and the limited precision of a Kinect sensor of several millimeters [2] can be a problem.

In our setup the CFRP element is attached to two KUKA KR210 robots (as shown in Figure 1) and three-dimensional

data is acquired by a Microsoft Kinect observing the structure. Bivariate polynomials, describing the depth map of the CFRP surface at each pixel, are especially useful in this case, as different order of variations can be specified (or found) along the horizontal and vertical directions.

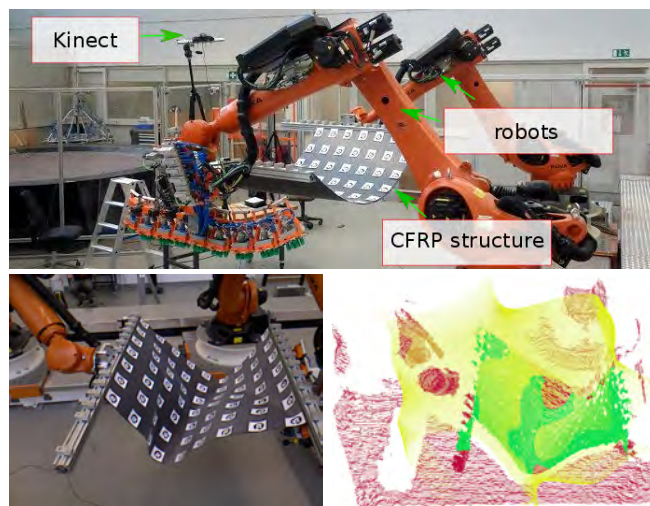


Fig. 1. Top: setup of the experiment. Bottom: for each deformation of the surface (left), the surface is identified in the scan, and a polynomial approximation is computed (right). The yellow points are sampled from the estimated surface model and the corresponding inliers are green.

Similar challenges like described before can come up also in different settings, where the segmentation and modelling of smooth surfaces has to be performed simultaneously. This is the case when high noise or spurious points are present in a scan, or when out of multiple arbitrary smooth surfaces the largest needs to be found and modeled.

Our approach to this problem is an algorithm based on Expectation Maximization (EM) which is initialized by Random Sample Consensus (RANSAC) [3]. RANSAC is used widely in the field of object recognition/segmentation and model fitting. It has the advantages of obtaining good model estimates with very noisy data, neglecting the outliers. The basic approach is to sample as little data as minimally needed to calculate a model that fits the sampled data. Then the distance of every data point to the estimated model is calculated. If the distance is within a certain threshold, the data point is considered an inlier. These steps are repeated until the best model is found with a given probability of success, i.e. the one that maximizes the number of inliers. The probabilistic estimate of the necessary number of iterations ensures fast runtimes, adapting to the actual percentage of the true inliers in the point cloud by taking the best estimate

so far as a lower bound. In the end, the method returns the best model coefficients and the list of inliers that supports it.

Since the models are estimated based on a limited number of points, the final model can be refined further by making a least-squares optimization of the distance of the model to the found inliers. Complementary to this, EM algorithms start off with an initial estimate of the unknown solution (i.e. the set of inliers), then iteratively compute the best matching model that explains the data, as well as the data that matches the model, until convergence. Typically this is formulated in a probabilistic way, maximizing the log-likelihood of the data given the model, but both EM and RANSAC have many versions, maximizing discrete as well as continuous metrics. Since EM can converge into local maxima, various meta-heuristics can be used. We opted to initialize it with the result of RANSAC. This allows for an efficient computation, guided by the user given probability of success.

For fitting geometric models, RANSAC-like algorithms employ simple shapes, for which the point-to-surface distance is easily computable [4]. Increased accuracy can be achieved if also (estimated) surface normals are considered and checked. However, this is more difficult in the case of polynomials, where finding the exact orthogonal projection can be costly, and simple approximations might be too inaccurate. Moreover, the coefficients that explain the samples best might not always be the most optimal. Large high-order terms make the surface rapidly take extreme values outside the sampled region, producing false inliers, so high orders should be avoided. We present solutions to these problems that are efficient to compute and work well in practice.

Overall, the main novelties of this paper are:

- a sampling based identification of the largest smooth surface that is robust to a large percent of outliers;
- optimizing the model coefficients through EM, such that lower order terms are encouraged;
- methods for inlier detection through approximating the point-to-polynomial distance and comparing the estimated point normal with the model normal.

The next chapter describes the related approaches and motivates our choice for the proposed solution.

II. RELATED WORK

Fitting of simple geometric shapes is a widely used technique, especially in the related task of object model reconstruction. In [5], an exhaustive nonlinear optimization is employed for reconstructing boxes, cylindrical and spherical items. This approach is relatively slow, thus often RANSAC is preferred [6]. RANSAC and modifications of it (e.g. [7]) have broad applications for model fitting of lines [8], planes [4], [9], circles [9], spheres [4], cylinders [4], [10] and many other regular geometries, with open source implementations available in the Point Cloud Library (PCL) [11].

An alternative approach is Hough voting, which has been also used to fit simple shapes like cylinders [12] or planar patches in [13]. However, the downside is that the parameter space's size needs to be relatively small. In the case of polynomials, the number of parameters can be very large,

in our application scenario we used over 30. Moreover, the range of values needs to be typically limited as well. These limitations do not hold in the case of RANSAC approaches.

For the modeling of smooth surface patches often spline structures are used. In [14] a frame from the Kinect is segmented into parts using normal-based clustering, and the patches are modeled using b-splines (NURBS with equal weights for the control points). This however requires an initial segmentation, and the modeling itself is depending on the control points. Specifically, it is not possible to limit the degree of variation in different directions.

Another way of describing surfaces is superquadrics, as used to model segmented objects [15]. While superquadrics can be fit to point clouds, they are closed surfaces, and, unlike polynomial surfaces, can not model arbitrary shapes.

Fitting polynomials locally to point clouds is performed in the moving least squares (MLS) method as in [16]. There, the goal is not to distinguish the real surface from background and noise, but to find a (weighted) least squares fit to all the points in a local neighborhood. Nonetheless, the fitting and normal estimation part is similar to the one employed in this work, but we use it in the RANSAC's fitting step and the least squares optimization part.

All in all, the method proposed in this paper offers necessary advantages for solving the problem of simultaneously identifying and modelling smooth surfaces in noisy data.

III. ALGORITHM

The algorithm can be divided in several subprocesses according to Figure 2. As a first step the input data point cloud gets smoothed and normals of the points are estimated, see III-A for details. After that a RANSAC step is performed, in which possible inliers and outliers are calculated according to their angular and euclidean distances to a parallel estimated model, see III-B and Figure 5. After a possible model is found in the RANSAC procedure, an optimal solution in a least-square sense is sought (III-C). Because the highest order polynomial model is not essentially the best model, an order optimization is performed hereafter, where the order of the polynomial is iteratively reduced and the number of inliers are calculated after every step (III-D) according to the process in Figure 6. After this step a model fitting the data is found and the data is segmented into inliers and outliers.

As a final step, this solution can be again optimized according to III-C. This is the outer loop in Figure 2, which is an EM loop, that gets repeated until the model converges.

A. Smoothing

Since Kinect data is noisy, and partially influenced also by the visual markers on the surface (see Figure 3), an initial smoothing step is necessary. This allows us to use a lower inlier distance threshold and to obtain better surface normal estimates, thus improving the results.

The MLS-based surface reconstruction proved to be very useful for smoothing large, noisy, real-world datasets [16]. By deriving the polynomial, accurate surface normals can be computed for each point. As in our case, the larger the order

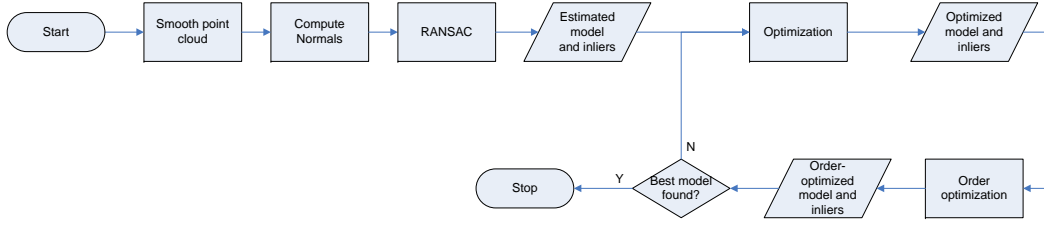


Fig. 2. Flow chart of the algorithm.

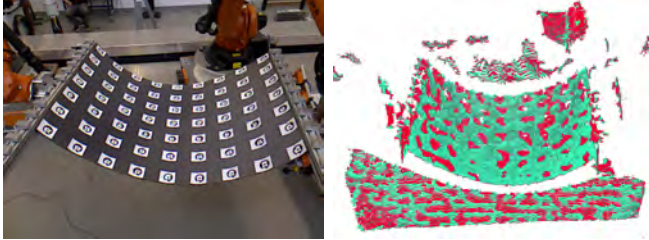


Fig. 3. Left: RGB image showing the location of the markers. Right: overlaid raw (red) and smoothed (green) point clouds, showing the depth noise and the deviations between the marked, and uncovered surface.

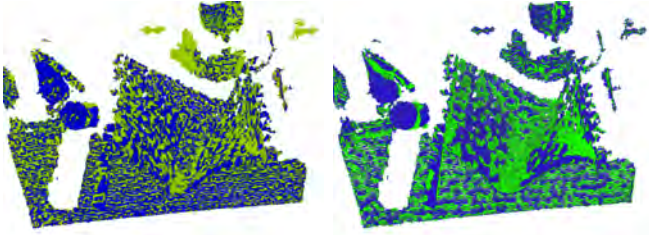


Fig. 4. Comparison of the degree 2 MLS smoothing (blue), and plane-based (i.e. degree 0) smoothing. Plane-based smoothing was first applied using a search radius of 3 cm (dark yellow in the left image) and then of 5 cm (green in the right image). The same scene as in Figure 1 is used.

of the polynomial, the more closely it can follow the surface. However, in the case of Kinect data we observed a wave-like noise pattern, sometimes above, other times below the real surface. This can be seen on the floor in Figure 3, but also on the CFRP structure’s surface.

As even a second order MLS reconstruction will follow this pattern, as seen in Figure 4, we perform an even simpler (and therefore faster) smoothing by simply projecting each point onto the least squares plane that describes its local neighborhood. This corresponds basically to fitting a plane (polynomial of degree 1) to the local neighborhood, as in MLS, but estimating the height map on it with a constant (degree 0). If the weighting of the points is skipped, this is always 0, so it can be omitted. As shown in Figure 4 a relatively large radius for finding nearest neighbors needs to be used in order to get rid of the wave-like pattern.

This heavy smoothing is not a problem in our case, since our goal is to approximate the whole surface with a smooth polynomial. The high-curvature region (closest to the ground) of the CFRP structure got smoothed out slightly, but offsetting it only by around 1 cm.

B. RANSAC-based initialization

The initial step of the RANSAC algorithm is to choose random sample points to estimate the model. With this

sample points a possible model is computed according to III-B.1. For this model the distance between every sample point and the model is computed. This distance computation can be divided in an euclidean distance and an angular distance of the normal vectors of the sample point and the normal of the corresponding point on the model function, see III-B.2.

If the euclidean distance is smaller than a certain threshold and the angle between the two normals is also under a second threshold, the sample point is considered as an inlier and added to the consensus set. After this inlier/outlier selection is done for every sample point, a new set of randomly chosen points are used to estimate a new model function and the inlier/outlier selection process is repeated. If more inliers are found than in the previous iterations, a better model function is apparently found and this function and the corresponding inliers are stored as the best found model so far.

This steps get repeated until a model is found with a certain predefined probability.

1) *Model estimation:* We use a bivariate polynomial,

$$f(x, y) = \sum_{i,j=0}^{n,m} a_{i,j} x^i y^j, \quad (1)$$

which has $(n+1) \cdot (m+1)$ unknowns, i.e. the polynomial’s coefficients $a_{i,j}$. These can be calculated from:

$$Ax = b$$

$$\text{with } A = \begin{bmatrix} x_1^0 y_1^0 & x_1^0 y_1^1 & \dots & x_1^n y_1^m \\ \vdots & \vdots & \ddots & \vdots \\ x_k^0 y_k^0 & x_k^0 y_k^1 & \dots & x_k^n y_k^m \end{bmatrix} \quad (2)$$

$$\text{and } b = [z_1 \ \dots \ z_k]^T$$

$$\text{and } x = [a_{0,0} \ \dots \ a_{n,m}]^T$$

Here A is the coefficient matrix of the linear system, consisting of known x - and y -values of k sample points, the right-hand side b accordingly consists of the z -values of the points. From the linear system in 2 we obtain a solution x for a minimum amount of $(n+1) \cdot (m+1)$ sample points. If there are more than the minimal amount of sample points used, the linear system is overdetermined and therefore an optimal solution in a least-squares sense can be calculated.

2) *Distance to model estimation:* To evaluate if a point is considered an outlier or an inlier for a given model, two quantities can be used, the *euclidean distance* of the point to the model function and the *angle* between the normal vector of the point and the normal vector on the surface.

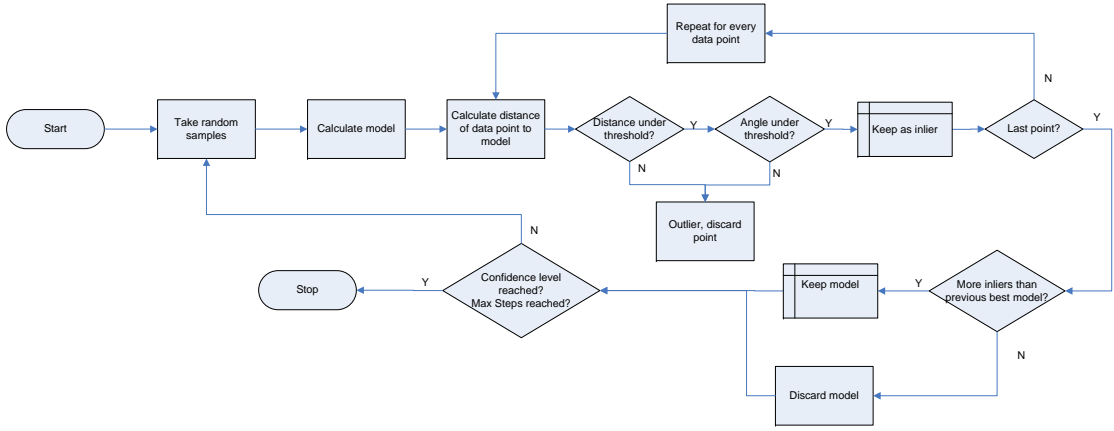


Fig. 5. Flow chart of the RANSAC model estimation.

a) *Euclidean distance*: The euclidean distance is the distance of the point P with the intersection point of its normal \vec{n}_p to the bivariate polynomial function:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \vec{p} + r\vec{n}_p \quad (3)$$

where x , y and z are points on the polynomial. This is:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} + r \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} \quad (4)$$

and with equation 1 it yields:

$$\sum_{i,j=0}^{n,m} a_{i,j} (x_p + rn_x)^i (y_p + rn_y)^j = z_p + rn_z \quad (5)$$

which is a function with only one unknown, the distance r .

We can treat the point's position ($r = 0$) as an initial guess and then explore its neighborhood to detect a sign change. The calculation of the minimal distance of a point to polynomial, which would be a nonlinear, two dimensional minimization is therefore reduced to a 1D root finding.

b) *Normal angle comparison*: The function $f(x, y) = z$ in equation 1 can be written implicitly:

$$g(x, y, z) = f(x, y) - z = 0 \quad (6)$$

For the normal of a point on the given implicit model function $g(x, y, z)$ we can write:

$$\vec{n}_f = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ -1 \end{bmatrix} = \vec{t}_x \times \vec{t}_y \quad (7)$$

In equation 7 \vec{t}_x and \vec{t}_y are the tangents in x- and in y-direction. The normal \vec{n}_p of a sample point is computed by principal component analysis decomposition in a fixed radius around this point during the smoothing step in section III-A. The angle between the two normal vectors is then calculated:

$$\alpha = \arccos \frac{|\vec{n}_p \cdot \vec{n}_f|}{|\vec{n}_p| |\vec{n}_f|} \quad (8)$$

C. Optimization

After the last RANSAC iteration a set of inliers is available. The found model can be optimized analog to equation 2 by replacing the randomly chosen sample points with the found inliers. This yields to a coefficient matrix of $j \times ((n + 1)(m + 1))$. This overdetermined system can be solved by standard least square methods.

D. Order Optimization

The highest order of a polynomial function doesn't have to be necessarily the one which fits the data set the best. In order to check for lower order terms, the order in x and y is iteratively reduced and the resulting reduced model function is tested against the full data set. After each reduction step an euclidean clustering is executed and the number of inliers is counted. In case an higher amount of inliers is found, the current model is stored as the best-fitting so far. This process is repeated until the order of 0 is reached in x and y . See Figure 6 for an overview of the algorithm.

E. Expectation Maximization

Unlike in the case of simple geometric shapes [6], [8], [4], [9], [10] a single optimization step is not enough, even with the selection of the best polynomial order. Because the polynomial tries to match the current inliers as well as possible, outside the region covered by the inlier cluster it tends to increase/decrease in value rapidly. This means that after an optimization step the number of inliers increases somewhat, but not necessarily close to the maximal value.

This is why the iterative EM steps need to be performed, as presented in Figure 2. Thereby, the current model is used to get the largest inlier cluster, which in turn is used in the model optimization step. The process is repeated until the inlier count increases. In some cases the first optimization step did not improve over the RANSAC result, therefore the model coefficients were used directly as the initial model.

IV. EXPERIMENTS

Having described the algorithm, in the following we present results for high-quality (synthetic) and CFRP data.

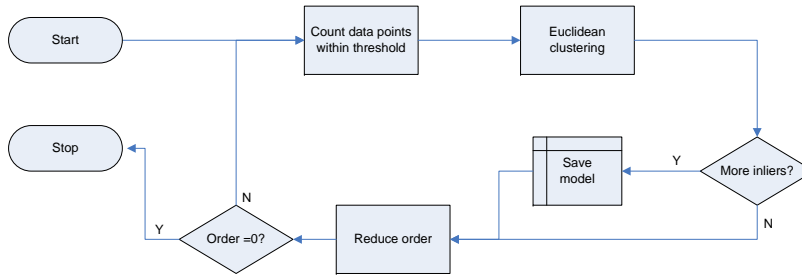


Fig. 6. Flow chart of the order optimization.

A. High-Quality Data

To test the algorithm with low-noise data, experiments were conducted with the *Stanford Bunny* dataset [17] and the *curticle.pcd* dataset of the Point Cloud Library test data [11]. In these datasets the noise level is very low and they consist only of a single surface, therefore all the points should be considered inliers. The resulting errors of the fitted model and the percentage of the found inliers are presented in Table I and Table II. Figure 7 shows a screenshot of the estimated model function and its inliers for the bunny.

TABLE I

RMS ERRORS AT DIFFERENT STEPS FOR HIGH QUALITY DATA

Model	RANSAC	optim.	order optim.	EM
bunny	0.0079415	0.0038188	0.0038188	0.0087256
cturtle	0.0032041	0.0025000	0.0025983	0.0025925

TABLE II

INLIER PERCENTAGES AT DIFFERENT STEPS FOR HIGH QUALITY DATA

Model	RANSAC	optim.	order optim.	EM
bunny	93%	93%	93%	93%
cturtle	97%	99.9%	100%	100%

Figure 8 (left) shows the result of the order optimization for the *curticle.pcd* dataset. It shows that the fitting performs very poorly for orders lower than 2, whereas the best fit is achieved with a order of 2 in x and y . Because of the smooth/noiseless surface, orders of 2 and above perform very well. This makes sense, as the surface is almost spherical. In the case of the bunny the highest order (5, 5) was the most optimal, and the method would have probably benefited if even higher orders would have been allowed.

B. CFRP Structures

During a continuous movement of the CFRP structure point clouds of the structure are obtained by a Kinect sensor. For every timeframe of the Kinect recording, the algorithm presented in section III is applied. Figure 9 shows the average root mean square errors for the found inliers for several Kinect frames at every step of the algorithm, and Figure 10 shows the average amount of inliers found for every step. Note that the RMS error can get bigger when more inliers are found, because we want to maximize the amount of inliers, independent of their distance to the model (as long as they match the distance and normal angle thresholds).

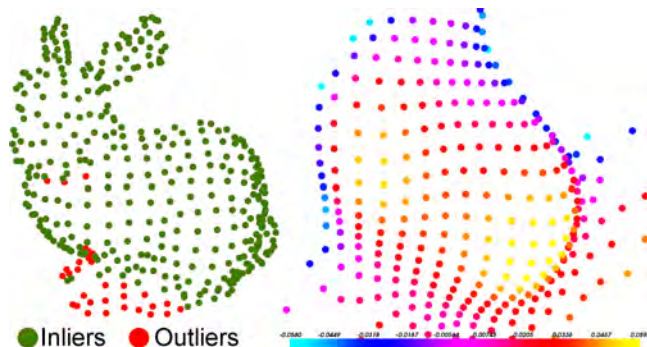


Fig. 7. Results of the fit for the Stanford bunny dataset showing the found inliers vs outliers (left) and the resulting model fitting the data (right).

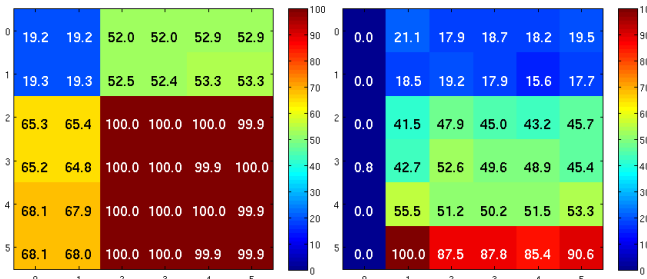


Fig. 8. Order Optimization results. Left: *curticle*. Right: CFRP. Horizontal axis shows the order of the y -component, while the vertical axis shows the order of the x -component. 100% represents the maximum number of inliers (in the case of the *curticle.pcd* this represents all the points, while in the CFRP scene it is roughly 50% of all points). Note, for better visualization, values are truncated after the first decimal place.

Figure 11 shows a screenshot of the fitted polynomials and the found inliers at different timesteps, Figure 8 (right) shows the result of the order optimization step. The fitting performs best for the highest order in x , and an order of 1 in y . This is because the CFRP is suspended on two bars, making the surface linear along y most of the time.

V. DISCUSSION

The experimental results show the advantages of the proposed method, highlighting the added advantage of each of its components. The number of inliers on the smooth polynomial surface gets maximized, all respecting the distance and normal angle thresholds.

The main parameters of the method are the order of the polynomial, the inlier distance and normal angle thresholds, and the probability of success to be used by RANSAC. The

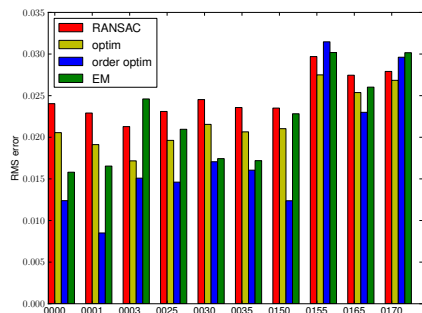


Fig. 9. Average root mean square error of the different algorithm steps.

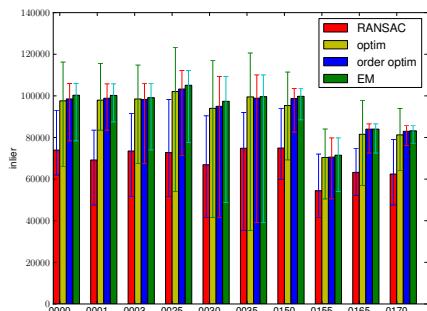


Fig. 10. Average inlier numbers of the different algorithm steps. The error bars show the minimum and maximum number of inliers.

probability of success influences the number of iterations and depends highly of the number of sampled points. In our case we would like a relatively small number of RANSAC iterations, because around half of the points are inliers, so a good model should be found relatively quickly. Since we sample the minimum of 36 points to estimate 5th order bivariate polynomials, the probability of success had to be very low in order to avoid excessively large runtimes. However, even at such low values the precision and recall of the method was good enough, as summarized in Table III.

TABLE III
PRECISION AND RECALL VALUES

	5.0E-014	1.0E-013	1.5E-013	2.0E-013
precision	90.70%	92.40%	91.44%	92.26%
recall	94.46%	88.71%	93.78%	93.88%

Alternative methods would be to apply a different segmentation method and to perform a least-squares fit to the largest/closest segment, or to model it using a spline. The advantage of polynomials over splines was already discussed in the Related Work section, as well the option to use the Hough transform for segmentation. Therefore, in this section we evaluate alternative segmentation methods.

First, though, a small note on the presence on markers on the surface. Avoiding the segmentation step would require a marker-based identification of the CFRP structure. This was attempted by placing the visual tags onto the surface, however their detection was not trivial. Because of motion blur, occlusions, and the non-planar surface, this method

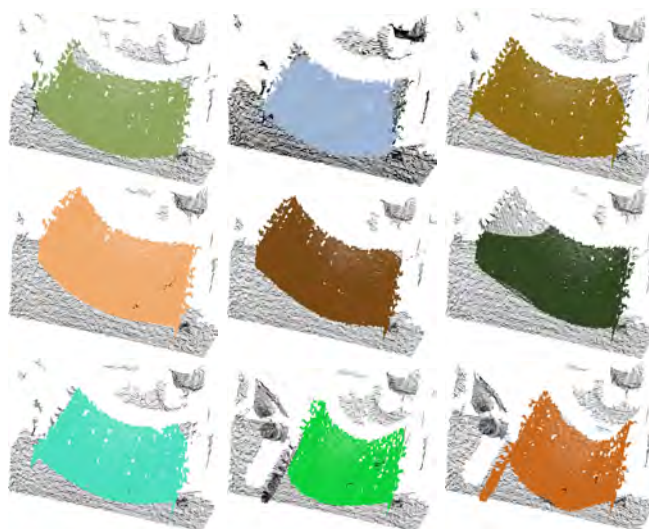


Fig. 11. Inliers for some of the fitted polynomial surface models.

failed to give robust results. Image processing methods face the problem that the surface is rather glossy, as shown in Figure 12. The visual segmentation method from [18] was tested, with the results shown in the bottom row of Figure 12, but the results were also not satisfying. Therefore, a 3D (or depth image based) approach is necessary.

The simplest 3D segmentation would be euclidean clustering, selecting the largest (or closest) connected component. This approach would obviously fail if there are objects connected to the surface. In our application scenario this could be the robots, but in other scenarios things can get even more complicated. We considered two potentially working approaches that operate on the 3D data: boundary-based segmentation and normal-based segmentation. Representative results are shown in Figure 13.



Fig. 13. Left: normal-based segmentation, displaying clusters in random color (with very small clusters deleted). Right: boundary-based segmentation, with a seed point in the center of the image. Please note that the red speckle in the middle is in the foreground, not part of the surface. Again, the same depth image as in Figure 1 is used. See text for details.

Detection of boundaries was performed as described in [19], and region growing performed, starting with a point in the depth image's center and adding points to the cluster until boundary points are not encountered. Due to the missing data points in the depth image however, some parts of the surface appear to be full of holes, stopping the region growing prematurely, as it can be seen in Figure 13. If the boundary estimation's search radius was increased in order to not produce boundaries inside the surface, the results reduced to those of a simple euclidean clustering. The detection

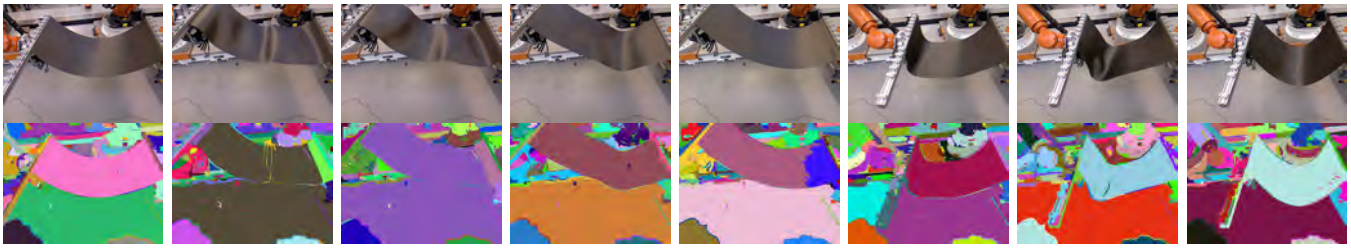


Fig. 12. A sequence of frames showing the reflectiveness of the CFRP surface, which complicated image-based segmentation. The best results using [18] were obtained using the parameters $\sigma = 0.5$, $k = 500$, $\min = 20$, so those were used for generating the results in the bottom row.

of high-curvature boundaries could be added, but that also does not guarantee a meaningful segmentation. Moreover, the large variation in depth also poses problems, as the great changes in density influence the results of both boundary detection and curvature estimation.

Additionally, the segmentation methods alone do not produce a model, so polynomial fitting, order optimization and the EM steps have to be applied nonetheless.

VI. CONCLUSIONS

In this paper we introduced a novel approach to robustly estimate two-dimensional polynomial model functions. This algorithm consists of a sample consensus step, which was used to initialize an Expectation-Maximization loop. It was shown that the method performed very well in the case of high-qualitative synthetic data and in the case of noisy CFRP structure data acquired by a Kinect sensor.

The latter case is a very challenging scenario because of the regular, dark and glossy surface of a CFRP element and reflections which can result from that. It was shown that the presented technique is robust against such complications.

The presented approach can be transferred to all model fitting applications which can be modeled as bivariate polynomials and expands the possibilities of sample consensus fitting of regular structures. Apart from this necessity, which is valid for all smooth surfaces, no previous knowledge of the surface is required. On the other hand, if there is information on the surface, a predefined order for the bivariate polynomial can be introduced. This would avoid the order optimization and speed up the calculation significantly.

The code will be made available in PCL after publication.

ACKNOWLEDGMENT

The authors would like to thank Fabio Bracci, Ferenc Balint-Benczedi, Stefan Hartweg, Florian Krebs, Andreas Klingler, Manfred Schönheits, Karan Sharma and Dejan Pangercic for their help with the data and evaluation.

REFERENCES

- [1] T. Leyvand, C. Meekhof, Y.-C. Wei, J. Sun, and B. Guo, "Kinect identity: Technology and experience," *Computer*, vol. 44, no. 4, pp. 94–96, 2011.
- [2] K. Khoshelham, "Accuracy analysis of kinect depth data," in *ISPRS workshop laser scanning*, vol. 38, 2011, p. 1.
- [3] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [4] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Computer Graphics Forum*, vol. 26, no. 2. Wiley Online Library, 2007, pp. 214–226.
- [5] G. Taylor and L. Kleeman, "Chapter 4: 3D Object Modelling and Classification," in *Visual Perception and Robotic Manipulation – 3D Object Recognition, Tracking and Hand-Eye Coordination*, ser. Springer Tracts in Advanced Robotics. Springer Berlin Heidelberg, 2006, vol. 26, pp. 57–83.
- [6] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Human Environments," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, October 11–15 2009.
- [7] P. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, pp. 138–156, 2000.
- [8] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.
- [9] O. Gallo, R. Manduchi, and A. Rafii, "Cc-ransac: Fitting planes in the presence of multiple surfaces in range data," *Pattern Recognition Letters*, vol. 32, no. 3, pp. 403–410, 2011.
- [10] R. C. Bolles and M. A. Fischler, "A ransac-based approach to model fitting and its application to finding cylinders in range data," in *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, 1981, pp. 637–643.
- [11] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9–13 2011.
- [12] T. Rabbani and F. V. D. Heuvel, "Efficient Hough Transform for Automatic Detection of Cylinders in Point Clouds," in *Laser Scanning Workshop of the International Society for Photogrammetry and Remote Sensing (ISPRS), WG III/3, III/4, V/3*, 2005.
- [13] G. Vosselman and S. Dijkman, "3d building model reconstruction from point clouds and ground plans," in *International Archives of Photogrammetry and Remote Sensing, Volume XXXIV-3/W4 pages 37-43, Annapolis, MD, 22-24 Oct. 2001*, 2005.
- [14] A. Richtsfeld, T. Morwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4791–4796.
- [15] K. Duncan, S. Sarkar, R. Alqasemi, and R. Dubey, "Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [16] Z. C. Marton, R. B. Rusu, and M. Beetz, "On Fast Surface Reconstruction Methods for Large and Noisy Datasets," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12–17 2009.
- [17] G. Turk and M. Levoy, "The stanford bunny," 2005, Stanford University Computer Graphics Laboratory. [Online]. Available: <http://graphics.stanford.edu/data/3Dscanrep>
- [18] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004.
- [19] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz, "Towards 3D Point Cloud Based Object Maps for Household Environments," *Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge in Robotics)*, vol. 56, no. 11, pp. 927–941, 30 November 2008.