# The 2D/3D Best-Fit Problem

Velislav Bodurov, Dimo Dimov, Georgi Evtimov, Ivan Georgiev,
Stanislav Harizanov, Geno Nikolov, Vencislav Pirinski

## 1. Introduction and problem description

Quality control of part production is crucial in manufacturing. Whether the produced part/detail passes the inspection or not depends on its particular usage and the answer may differ from application to application. Furthermore, the daily detail manufacture is usually huge and the quality inspection should be fast. Therefore, in order to meet the needs of industry, the process must be as automatic and as flexible as possible.

Sirma Group Holding JSC is one of the largest software groups in Southeast Europe, with a proven track record since 1992. EngView Systems Jsc is a subsidiary company for CAD/CAM software, which, among other tasks, deals with quality control via scanning. More precisely, their team wants to enrich the software of the scanner they sell on the market, so that the original CAD model of the detail is "properly" compared to the scanner's output of a given manufactured specimen. The first object usually consists of a list of "CAD primitives", that are either line or arc segments, for which the two endpoints and the circle center/radius (for arc segments) are given. The second object is a real-point cloud, whose density depends on the scanner's resolution. The two data sets lie in different coordinate systems, thus the scanned data should be translated and rotated in order to align with the CAD one. This "optimal" alignment is the main purpose of our work. Once achieved, the objects' comparison is user-dependent, but typically point-wise displacements between the two data sets at certain (again user-specified) points of interest (*control points*) are measured (see Fig. 1). When those quantities are within the user-given range, the specimen passes the quality inspection.

The optimal data alignment (a.k.a. *Best-Fit problem*) can be described as a search for the best transformation matrix to transform input measured points from their coordinate system into a CAD model coordinate system using a criteria function for optimization. The best algorithmic solution should include the following features:

---

1. Partial fit (only part of the object is scanned).
2. Different parts (these could also be measure points) can have their own

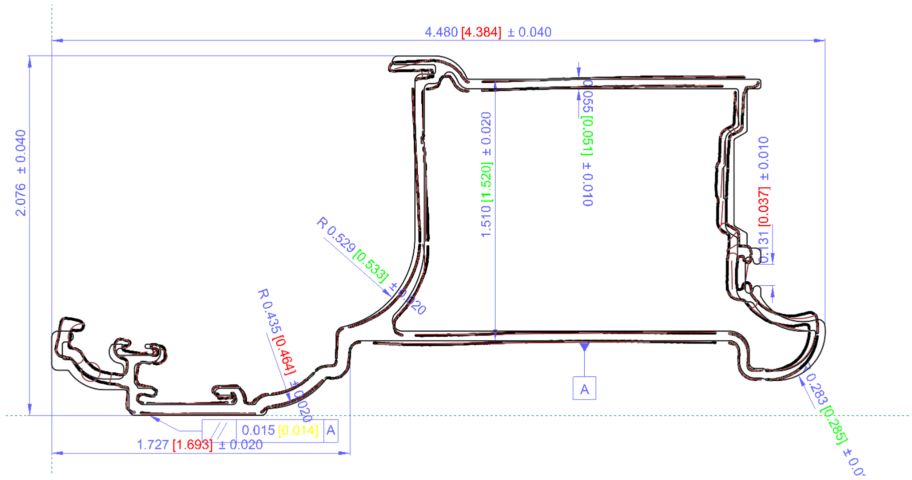


Fig. 1. CAD vs. CAM comparison of an industrial detail. Green numbers pass the quality inspection, while red ones do not

individual weights.

3. Only some of the three rotations and three translations can be applicable.
4. The algorithm can be applied on 2D or 3D data.
5. Preliminary assessment can be made if there are points that constitute noise. If such points are detected, they should be filtered out.
6. In the ideal case, the algorithm's input data – these are the data in the two coordinate systems – can appear as points, as a mesh, or as a CAD model.
7. Optimization can take place by different optimization criteria: least squares, minimum sum of deviations, mini-max, uniform deviations, minimum standard deviation, tolerance envelope, tolerance envelope mini-max.
8. The fit process should be able to accept also partially deformed parts. Even if there are discrepancies between the CAD model and the input data, the algorithm must be able to process them.
9. The computation needs to be fast and efficient.
10. An option could exist for multi-core, parallel computation.

---

In the discrete setting (pixel grid), a CAD/CAM Best-Fit algorithm for the corresponding raster images has been proposed and investigated in [1]. Here, because of the point-wise-distance criterion, the EngView Systems representative insisted on us working in the continuous setting (vector format), where each point is represented via its coordinates in $\mathbb{R}^n$, $n = 2, 3$. In [2] an algorithm that checks if two unlabeled configurations of points in $\mathbb{R}^n$ are an orthogonal transformation of one another is proposed. If they are, the transformation matrix is explicitly computed. This algorithm is also modified for noisy measurements (as is the case with our scanned data), but it assumes that the two point clouds are of the same cardinality and are one-to-one. The latter is not applicable to our problem, since the scanned data are randomized, thus we cannot extract their corresponding point cloud from the CAD model.

## 2. Our approach

The main difficulty in solving the given Best-Fit problem arises from the diversity in the data representation of the CAD model and the vectorized scanned image (*CAM data*). On the one hand, we have the CAD primitives (fully structured, continuous data), where only few points are specified (namely, the endpoints of the primitives). On the other hand, we have the real-point cloud, derived by the scanner, where the whole information is incorporated in point coordinates and no connectivity among the points is known (thus, completely unstructured, discrete data). Furthermore, the randomness of the vectorization implies that the probability for the input point cloud to contain the corresponding image of any of the CAD endpoints is zero.

Since structuring a point cloud is an NP-hard problem, we choose to discretize the CAD model and to apply techniques from Principal Component Analysis (PCA) on the two point clouds. In theory, the input data should be uniformly sampled from the specimen surface with step-size, depending on the scanner's resolution. Hence, we also uniformly sample our CAD data with respect to the arc-length parameterization of the primitives. We use a standard AutoCAD function for that. Then, for each of the discrete data sets, we compute their *energy ellipse/ellipsoid* (in 2D/3D respectively). Those ellipses define local frames, centered at the corresponding data barycenters, with axes along the directions of minimal and maximal energy. The computation is based on least-squares approach, that leads to quadratic constrained optimization problem on the unit circle. The latter is equivalent to finding the Jordan decomposition of a $n \times n$ Gramian matrix, $n = 2, 3$, which is an easy, fast, and numerically stable procedure. In signal processing, this technique is known as *the Karhunen-Loève Transform* [3] and the total mean-square error is proven to be minimized in this local (energy) basis. Finally, we map the CAM local frame onto the CAD-discretized local frame, choosing the "correct" orientation (in 2D we have 4 different options if only flips along the coordinate axes are allowed, and 8 - if we consider *mirroring*, as well) and declare the corresponding transformation matrix as optimal.

When the scanned data is denoised and uniformly sampled, while the manufactured part/detail specimen is without any defects, the transformed CAM data cloud should be perfectly contained within the continuous CAD model. Thus, the transformation matrix is optimal with respect to any optimization criteria. In practice, however, the scanned data is noisy, and the EngView Systems' team performs a denoising procedure, where untrustworthy data is erased. This, together with the possible defects of the specimen, affects the CAM local frame (mainly the coordinate axes, while the origin remains quite stable) and further "local" modifications of the derived transformation matrix are needed in order to optimize it. The latter is a subject of future work and it will be discussed in the corresponding section.

### 2.1. Karhunen-Loève transform

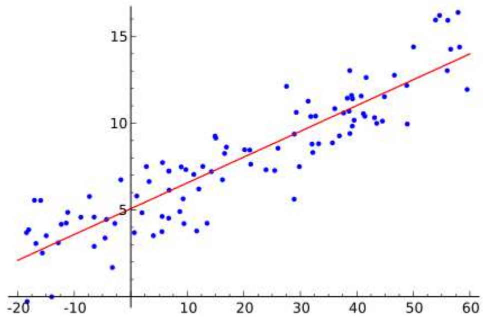


Fig. 2. Linear regression analysis. The picture is taken from Wikipedia

For a given set of $N$ points $\mathcal{M} := \{(x_i, y_i)\}_{i=1}^N$, we look for the line in $\mathbb{R}^2$ that minimizes the sum of the squared Euclidean distances from the point set to it. In statistics this procedure is known as *Linear regression*, see Fig. 2 (the picture is taken from `https://en.wikipedia.org/wiki/Linear_regression`), while in mathematics – as *Least Squares Problem*. It is easy to show that this optimal line passes through the barycenter $(x_G, y_G)$ of $\mathcal{M}$. Thus, we use the normal representation

$$\ell : A(x - x_G) + B(y - y_G) = 0$$

of the former, where $(A, B)^T$ is a unit normal vector with respect to $\ell$. We want to solve the following minimization problem

$$(1)\qquad \underset{A,B}{\operatorname{argmin}} \underbrace{\sum_{i=1}^{N} \big(A(x_i - x_G) + B(y_i - y_G)\big)^2}_{F(A,B)} \qquad s.t. \quad A^2 + B^2 = 1,$$

which is equivalent to quadratic optimization on the unit circle:

$$(2)\qquad \underset{A,B}{\operatorname{argmin}} \left\langle \begin{pmatrix} \langle x,x\rangle & \langle x,y\rangle \\ \langle y,x\rangle & \langle y,y\rangle \end{pmatrix} \begin{pmatrix} A \\ B \end{pmatrix}, \begin{pmatrix} A \\ B \end{pmatrix} \right\rangle \qquad s.t. \quad A^2 + B^2 = 1.$$

Here, $x = (x_1 - x_G, \dots, x_N - x_G)^T$, $y = (y_1 - y_G, \dots, y_N - y_G)^T$, and $\langle \cdot, \cdot \rangle$ is the standard scalar product in $\mathbb{R}^N$. The matrix

$$\mathbf{M} := \begin{pmatrix} \langle x,x\rangle & \langle x,y\rangle \\ \langle y,x\rangle & \langle y,y\rangle \end{pmatrix}$$

is Gramian, thus symmetric and positive definite (unless $\mathcal{M}$ is collinear). Problem (2) is classical. The range of the cost function $F$ is $[\lambda_1, \lambda_2]$, where $0 \le \lambda_1 \le \lambda_2$ are the eigenvalues of $\mathbf{M}$, and the minimizer is given via

$$(3)\qquad \begin{pmatrix} \bar{A} \\ \bar{B} \end{pmatrix} = \pm v_1, \qquad \mathbf{M} v_1 = \lambda_1 v_1, \quad \|v_1\|_2 = 1.$$

Note that the minimizer is unique up to sign, so additional "orientation" issues need to be considered afterwards.

In this setup, the Karhunen-Loève transform $KL : \mathbb{R}^2 \to \mathbb{R}^2$ is simply a change of basis:

$$(4)\qquad KL \begin{pmatrix} x \\ y \end{pmatrix} = \underbrace{\begin{pmatrix} v_{1,x} & v_{2,x} \\ v_{1,y} & v_{2,y} \end{pmatrix}}_{\mathcal{T}_{\mathcal{M}}} \begin{pmatrix} x + x_G \\ y + y_G \end{pmatrix}.$$

The vector $v_2$ is a normalized eigenvector for $\mathbf{M}$ w.r.t. $\lambda_2$, and it describes the direction of minimal energy of $\mathcal{M}$. On the other hand $v_1$ describes the direction of maximal energy of $\mathcal{M}$. In other words, the local frame $(G, v_1, v_2)$ places $\mathcal{M}$ along the $y$-axis, it is quite natural, and the most stable invariant of $\mathcal{M}$ w.r.t. Gaussian noise.

### 2.2. KL-transform-based Algorithm

We propose and implement in MatLab the following algorithm in $\mathbb{R}^n$, $n = 2, 3$:

**Algorithm 2.1** (2D/3D Karhunen-Loève transformation matrix)**.**
Input: ***cad_data***, ***sc_data*** Output: *Transformation matrix $\mathcal{T}$ and translation vector $\vec{\mathbf{d}}$*

1. *Compute the barycenters $G^{CAD}$ and $G^{sc}$ of **cad_data** and **sc_data**.*
2. *Compute the shift $\vec{\mathbf{d}} = G^{CAD} - G^{sc}$.*
3. *Compute $\mathcal{T}_{CAD}$ and $\mathcal{T}_{sc}$ as in Section 2.1.*
4. *Compute the orthonormal matrix $\mathcal{T} = \mathcal{T}_{CAD}\mathcal{T}_{sc}^T$.*
5. *Derive **sc_data_aligned** from **sc_data** via translation by $\vec{\mathbf{d}}$ and rotation by $\mathcal{T}$.*
6. ***FUZZ** the **cad_data**.*
7. *Axes orientation check and $\mathcal{T}$ modifications, if necessary.*

As already mentioned, the input **cad_data** is a point cloud, uniformly sampled from the CAD model via standard AutoCAD software. The barycenters are computed directly via coordinate-wise averaging. Since both $\mathcal{T}_{CAD}$ and $\mathcal{T}_{sc}$ are orthogonal, so is $\mathcal{T}$ and it rotates the CAM local frame in order to align it with the CAD one, i.e.,

$$\mathcal{T} : v_i^{cs} \to v_i^{CAD}, \qquad i = 1, \dots, n.$$

Since all the basis vectors are unique up to a sign, we have to choose the correct axes orientations for optimal data matching. The latter means that we need to consider all possible "axes flips" of the CAM data, leading to $2^n$ different choices, and take the one that best fits the CAD data. In 2D, those flips result in left multiplications of $\mathcal{T}$ by

$$\mathcal{T}_x := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \qquad \mathcal{T}_y := \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \mathcal{T}_{xy} := \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix},$$

respectively. In order to quantitatively compare the different orientations, we use another standard AutoCAD function, namely *FUZZ distance*. This works as follows: around each CAD primitive, we draw an envelope of certain width $\varepsilon$ (the width may vary from primitive to primitive, but for the moment it suffices to consider it a global parameter) and AutoCAD orthogonally projects all points within the envelope on the primitive. (A non-scientific explanation would be, that we thicken the lines of the CAD model.) We choose an appropriate $\varepsilon$, and for each of the orientations of **sc_data_aligned** we count the number of points outside of the CAD model $\varepsilon$-envelope. The optimal orientation is the one that minimizes this number.

### 3. 2D Numerical examples

We consider two numerical examples (see Fig. 3). All the CAD/CAM data are provided by EngView Systems. In both cases the CAD and CAM coordinate systems are a priori aligned (Fig. 4), which does not affect at all the Karhunen-Loève Transform and the performance of Algorithm 2.1, but allows us to compare our output with the optimal transform (which is the identity). We uniformly sample the first CAD model (Fig. 3 Left) using 1323 points and the second CAD model (Fig. 3 Right) using 1218 points. The scanned data for the first example consists of 399 points, while for the second – of 8206 points. Comparison of CAD vs. CAM local bases is shown on Fig. 4. Comparisons between CAD and

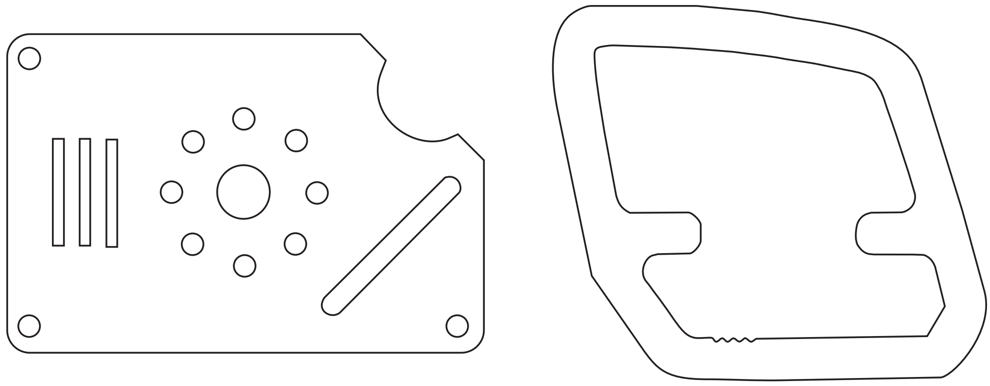

Fig. 3. The CAD models of the 2 considered parts

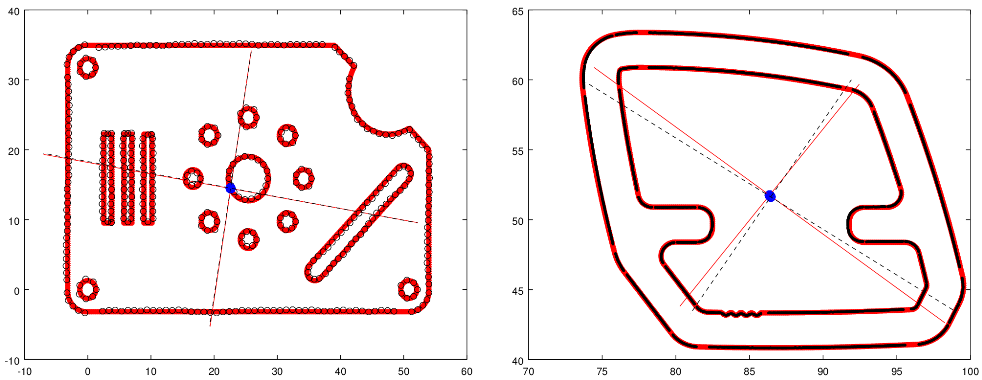


Fig. 4. Point-cloud comparison for the details: Red(solid): (Fuzzed) CAD sample and its KL frame. Black(circled): Scanned data and its KL frame. Blue dots: Barycenters

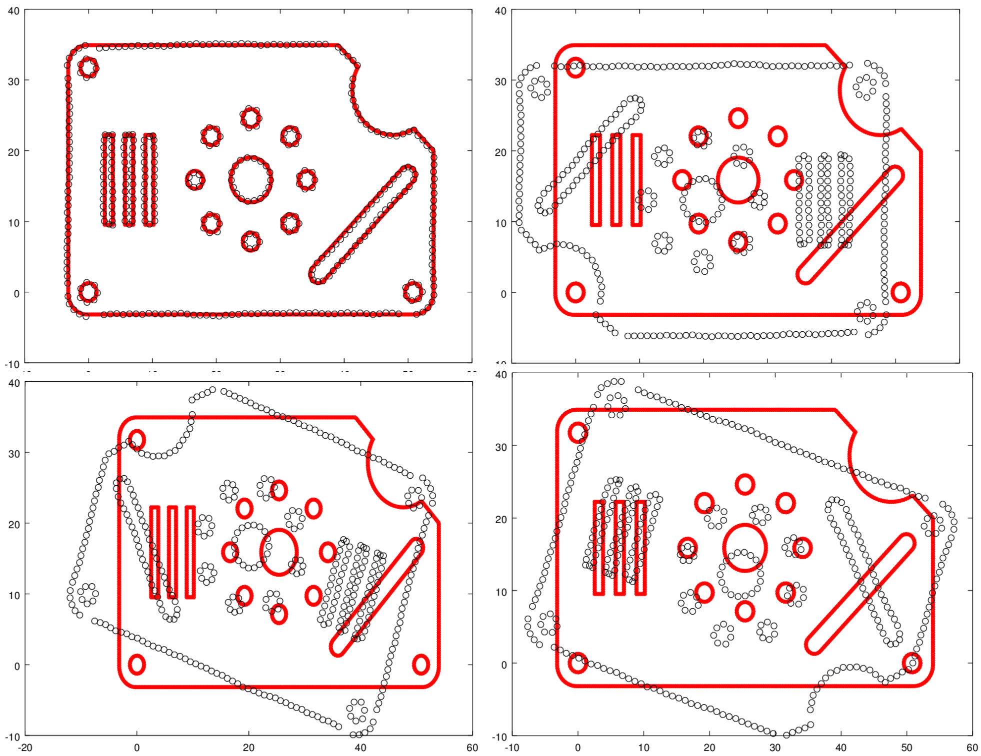

Fig. 5. Different local frame orientations for the first example

transformed CAM data w.r.t. Algorithm 2.1 are shown on top left on Fig. 5 and Fig. 6, respectively. Both scanned specimens have no manufacturing defects.

For the first example we see (almost) perfect alignment of the two KL local frames. This is due to the good specifics of both specimen and its scanning (no defects and close-to-uniform CAM sample). Moreover, even for small $\varepsilon$, the fuzzy CAD data incorporates almost all of the CAM points, making the orientation check in step. 7 of the algorithm straightforward (see Fig. 5).

This is not the case with the second example. There, even though the CAM sample is 20 times bigger than the one in the first example, some of the scanned data was untrustworthy and erased during the denoising process that preceded our work. This resulted into several CAD regions for which no scanning information is available (see Fig. 4). The latter polarizes the CAM point sample and affects its local axes. In turn, the outcome of Algorithm 2.1 is not the optimal transformation matrix. Furthermore, the CAD model is almost a square, thus has

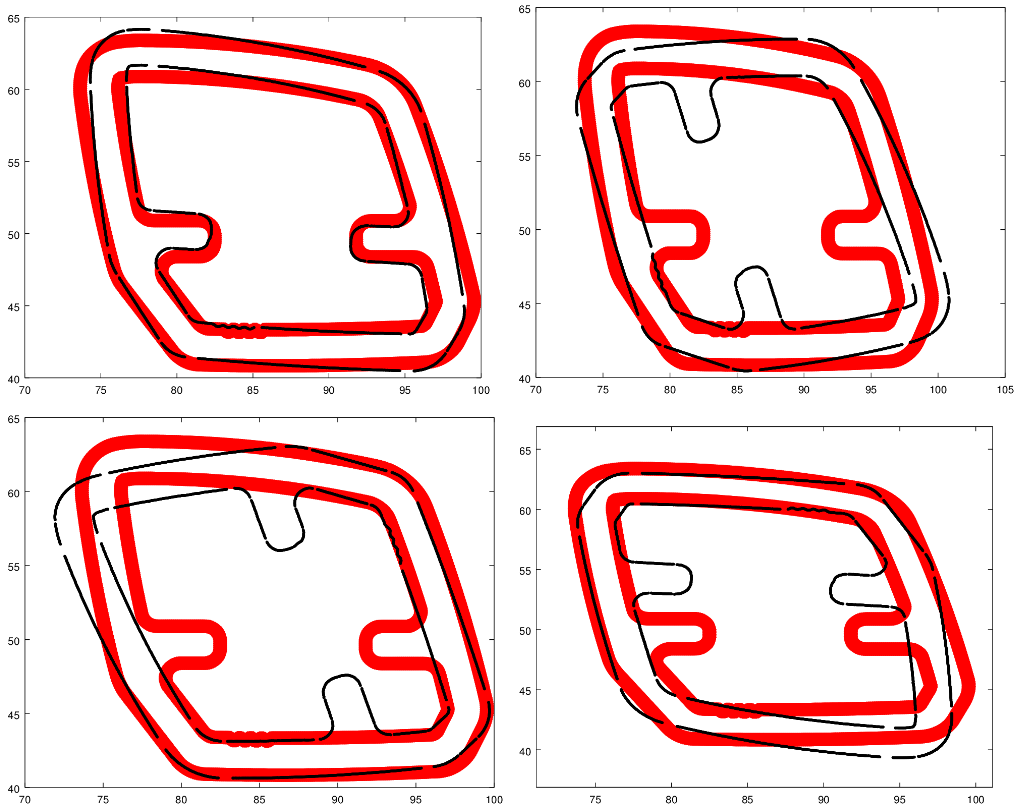


Fig. 6. Different local frame orientations for the second example

plenty of symmetries and the range of the cost function $F$ in (1) is small. Combining the two problems, we witness a very hard orientation check (see Fig. 6). For each of the axes flips and for small $\varepsilon$ the fuzzed CAD data contains some portions of the CAM points and misses the two "bumps". Thus, the number of CAM points outside the fuzzed region does not differ significantly among the different orientations. In such a case, choosing the correct frame orientation is not secure, and we increase $\varepsilon$ until a clear winner appears, namely until the "bumps" are captured by one of the candidates. The latter is indeed the correct orientation.

## 4. Future work

Algorithm 2.1 is just a preliminary step of the desired Best-Fit algorithm. As seen from the numerical experiments, it provides satisfactory transformation matrix only for non-deformed parts/details with uniformly sampled CAM data.

Moreover, among the desired algorithmic properties, listed in the introduction, we cover only points 4-5, because the Karhunen-Loève transform is applicable also in 3D, and Least Squares estimates are maximum-likelihood ones of Gaussian noise. Clearly, such an approach cannot deal with point 1. In order to address the remaining issues, further modifications of $\mathcal{T}$ are necessary.

The biggest benefit of Algorithm 2.1 is that it structures $\bar{\mathcal{M}}_{sc} := \mathcal{T}(\mathbf{sc\_data} + \vec{\mathbf{d}})$ in a sense that most of the scanned points away from the endpoints of CAD primitives can now be assigned to their corresponding primitive! We assume that the barycenters $G^{CAD}$ and $G^{sc}$ are (almost) correct. This is witnessed in both of the examples in Section 3. Moreover, Georgi Evtimov wrote a LISP function, that computes the barycenter of the continuous CAD model, and when compared to the barycenter of even random point samples (that still capture the detail geometry) we saw that the latter approximates well the former. Therefore, mainly the axes directions of the CAM local frame (but not its origin) are affected by the quality of both the detail and its scanning, so we search for another rotation matrix

$$\mathcal{R}_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \qquad \theta \in (-\pi/2, \pi/2),$$

to deal with that. Finally, the optimal rotation matrix will be

$$\mathcal{T}_{final} = R_\theta \mathcal{T},$$

while the best coordinate transform is

$$\begin{pmatrix} x \\ y \end{pmatrix} \to \mathcal{T}_{final} \begin{pmatrix} x + d_x \\ y + d_y \end{pmatrix}.$$

Let $\mathcal{M}_i \subset \bar{\mathcal{M}}_{sc}$ be the points that clearly belong to the CAD primitive $P_i$. Then, we can break our Best-Fit problem into $N$ smaller and simpler ones, where $N$ is the number of different CAD primitives in the model. Since the primitives are either line or arc segments, there are only 2 types of optimization problems that appear. Those $N$ processes are independent and small-scale, thus they can be computed in parallel (point 10) and very efficiently (point 9). Different optimization criteria can be used (as long as the corresponding optimization problem can be numerically solved on line and arc segments!), thus point 7 is also covered. For each $i = 1, \ldots, N$, given $\mathcal{M}_i$ and (uniform sample of) $P_i$, the local Best-Fit problem will produce as output a rotation angle $\theta_i$, i.e.,

$$(\mathcal{M}_i, P_i) \xrightarrow{Local\ BestFit} \theta_i, \qquad i = 1, \ldots, N.$$

The global angle $\theta$ can be a convex combination of the local ones

$$\theta = \sum_{i=1}^{N} \omega_i \theta_i, \qquad \omega_i \geq 0, \quad \sum_{i=1}^{N} \omega_i = 1,$$

which addresses point 2. We can project $\mathcal{T}_{final}$ onto any subgroup of $SO(2)$ (those, considered admissible by the user), which solves point 3.

Let us consider two different primitives $P_i$ and $P_j$ that are both line segments. Assuming that $\mathcal{M}_i$ and $\mathcal{M}_j$ are indeed scanned samples of $P_i$ and $P_j$, independently of the optimization criteria the angles $\theta_i = \angle(\ell_i, P_i)$ and $\theta_j = \angle(\ell_j, P_j)$ should be equal, unless the samples are noisy or the detail is defected. Here, $\ell_i$ and $\ell_j$ are the optimal lines for $\mathcal{M}_i$ and $\mathcal{M}_j$, respectively. Moreover, this equality is not affected by the accuracy of the barycenter computations, because the quantities are invariant under translation. Thus, the distribution of the set $\{\theta_i \mid P_i - \text{line segment}\}$ provides us with information about defects and/or data discrepancies (point 8).

When $P_i$ is an arc segment, assigning the angle $\theta_i$ is not a priori clear. Furthermore, Best-Fit problems on circles are usually much more complicated than the ones on lines. For example, even the Least-Squares fit, which is a linear problem in the latter setting, has no closed form solution in the former one and there is no direct algorithm for it (see [4]). However, in our case we have additional information from the CAD model that we incorporate into the optimization problem as constraints. In particular, the center $O_i$ and the radius $R_i$ of the arc $P_i$ are given, while the distance $r_i := |G^{CAD} O_i|$ can be computed. Therefore, we search for a point $\bar{O}_i$ on the circle $C(G^{CAD}, r_i)$ for which the circle $C(\bar{O}_i, R_i)$ minimizes the least-squares functional. In other words, we restricted a 3D optimization problem (the unknowns are the coordinates of $\bar{O}_i$ and the radius $\bar{R}_i$ of the optimal circle) to a 1D one, where the only unknown is the angle $\theta_i = \angle \bar{O}_i G^{CAD} O_i$. The restricted problem remains nonlinear, but analyzing 1D functions is a much easier task, that, at least numerically, can be efficiently performed.

The time frame, needed for the execution of such ambitious work plan, is far beyond the one of the workshop. However, if EngView Systems are interested in further collaboration, this might be an interesting and fruitful project for both industry and science.

**Acknowledgements.** Geno Nikolov was partially supported by the Sofia University Research Fund through Grant 75/2015. Velislav Bodurov was partially supported by the Sofia University "St. Kliment Ohridski" under contract No. 08/26.03.2015.

## References


[1] D. Dimov, G. Gluhchev, N. Nikolov, I. Burov, E. Kalcheva, S. Bonchev, S. Milanov, B. Kiossev. The Computer System STEMB for Verification of Replicas of the Bulgarian State Emblem. *Cybernetics and Information Technologies*, 1(1):95–106, Bulgarian Academy of Sciences, Sofia, 2001.

[2] D. Jiménez, G. Petrova. On Matching Point Configurations. In *Constructive Theory of Functions*, 141–154, Prof. Martin Drinov Academic Publishing House, Sofia, 2014.

[3] R. D. Dony. Karhunen-Loève Transform. In *The Transform and Data Compression Handbook*, (K. R. Rao and P. C. Yip, Eds.), Boca Raton, CRC Press LLC, 2001.

[4] N. Chernov, C. Lesort. Least Squares Fitting on Circles. *Journal of Mathematical Imaging and Vision*, 23(3):239–252, 2005.