Computer-Aided Design and Applications
© 2009 CAD Solutions, LLC
http://www.cadanda.com

Reconstruction of 3D Models from Measurements Obtained by a Moving Sensor

Miri Weiss-Cohen[1], Alina Bondarenko[2] and Yoram Halevi[3]

[1]ORT Braude College of Engineering, miri@braude.ac.il
[2]Rafael Advanced Defense Systems Ltd., alinab@rafael.co.il
[3]Technion—Israel Institute of Technology, yoramh@technion.ac.il

ABSTRACT

A comprehensive method for automatically constructing a 3D solid model from orthographic views obtained by a moving sensor is suggested. The views are constructed from sensors, e.g. distancemeters or cameras, in an automatic fashion using estimation techniques. The raw distance measurements are processed via a filter that generates estimates of the part dimensions and position. The fact that the outcome of the estimation (measurement) process is a set of explicit contour equations is suitable for the second step, which is creating a 3D model from the orthographic views through a graph theoretic approach. The process is implemented by using variational geometry representation and graph theoretic tools that enable us to construct a composite graph representing the 3D object. Results of the composite graph are translated into a boundary representation object.

Keywords: geometric modeling, graph theory, Kalman filtering, parameter estimation.
DOI: 10.3722/cadaps.2009.103-114

1. INTRODUCTION

The work deals with creating a 3D geometrical model of an object based on automatically generated 2D orthogonal views. Theoretically, not only can one reconstruct the full model directly from measurements, but moreover, both the object and the sensor may be moving at the time the readings are taken. However, a typical scenario is a stationary object and measurements taken from orthogonal directions. The measurements can be made by an in-plane laser, an orthogonal laser or a camera.

The main idea when creating the views is that the object consists of primitives with known shapes but with unknown parameters, such as a cylinder with unknown dimensions etc. Other unknowns are the object position and orientation. Such problems are of interest for quite some time and different approaches were suggested e.g. [4], [5], [9], [21], [24]. The problem bears also some similarities to simultaneous location and mapping (SLAM) [8]. The methodology of the identification used in this paper follows the one in [2].

With the raw measurements provided by any type of sensor, the construction of the object in the pre-specified shape space becomes a non-linear estimation problem. This is one of the most addressed problems in the literature with numerous approaches ranging from standard Least Squares, through Gradient Weighted Least Squares [1], to the more robust M-estimators [12]. The most common approach for estimation from a sequence of measurements is using a Kalman Filter, which is the optimal estimator for linear problems, on a linearized model about the prior estimate. This is the Extended Kalman Filter (EKF) [18]. The Iterative Extended Kalman Filter (IEKF) [13] uses repeated linearizations to increase accuracy. The recent Noise Updated Iterative Extended Kalman Filter (IEKF)

[2] goes one step further and uses the identified noise in the linearization. This is most effective in cases where the noise effect is strongly non-linear.

A number of approaches have been developed over the past decades to interpret user-supplied orthographic views. The two main reconstruction approaches are the wireframe – B-rep bottom-up approach [16-17], and the volume-solid oriented approach [3-4], [8], [17-18].

The stage of constructing a wireframe model and translating it into a 3D solid model in a bottom-up approach has caused many errors and raised various research issues. Spatial configurations involving lines and planes containing primitives in an automatic way is detailed in [10]. The automatic procedure for 3D reconstruction mimics trained human experts, which is achieved by combining elements of variational geometry, and graph theoretic methods [14,20]. In particular, the main novelty in the approach is its use of understanding the nature of 2D engineering drawings. This understanding is translated into an actual algorithm by means of topological relations and dimensional scheme analysis [14, 23].

2. CREATING THE 2D VIEWS

We assume that the measurement is planar. One possibility is an in-plane laser measurement shown schematically in Fig. 1 that is taken at several levels, which is geometrically expressed as planes intersecting the object. At each measurement instant the sensor is in a known position and orientation, up to small uncertainty, which is modeled as noise. Then the distance to the contour of the object is measured, again with a certain noise. In [3] the most general situation in which the situation is 3D and the object is moving as well was considered, but here we assume that the problem is planar and that the object is stationary, but in an unknown position. Treating the position of the object as unknown is appealing from a practical point of view because it eliminates the need for registration and\or initialization.

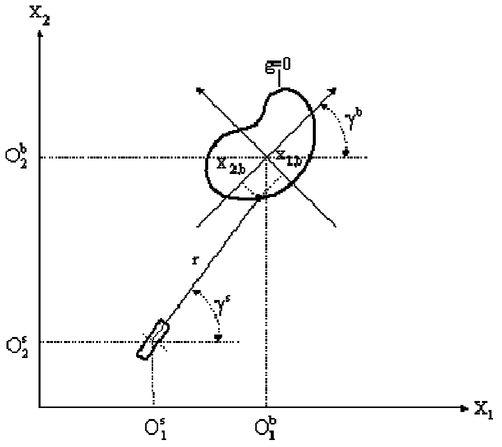

Fig. 1: General setting of the object and the sensor.

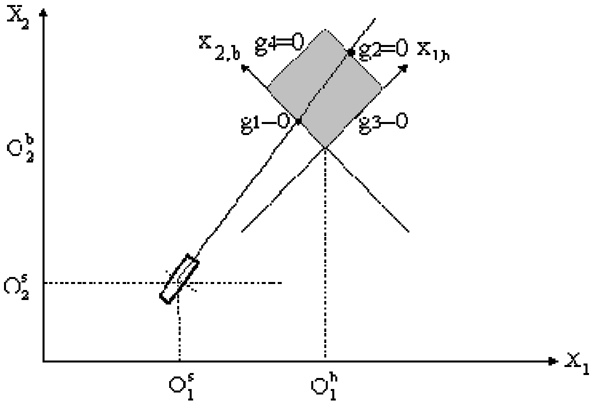

Fig. 2: A rectangle represented by four functions.

The general shape of the object, or parts of the object, is assumed to be known, but not its parameters. This assumption is in line with many real life cases where the object is made of standard elements such as cylinder, plane phase, box ect. In body coordinates (see Fig. 1), the contour is given by

$$g(x_1^b, x_2^b, \theta) = 0 \ , \qquad (2.1)$$

where θ is a vector of parameters. For example, in a circle this vector includes the two coordinates of the center and the radius, and in an ellipse it includes the two coordinates of the center and the two radii. g may be only piecewise continuous or a vector of several functions, e.g. for multi-facet objects, as is the case in Fig. 2. The four functions in this case are

$$g_1 = x_2^b, \quad g_2 = x_2^b - B, \quad g_3 = x_1^b, \quad g_4 = x_1^b - A, \qquad \theta = (O_1^b, O_2^b, A, B)\ . \tag{2.2}$$

Alternatively, the four lines can be described by the single function that is their product,

$$g = x_2^b \cdot (x_2^b - B) \cdot x_1^b \cdot (x_1^b - A)\ . \tag{2.3}$$

The advantage of this form is the automatic calculations without the need for logical operations that determine which facet is active. Dealing with simpler functions, on the other hand, is better from a statistical point of view.

To relate the local (2.1) to the actual measurement, a coordinate transformation is required. While this can be done by means of standard geometry, the derivation is more structured and pseudo-linear when homogeneous coordinates are used. The body coordinates system and the global coordinates system are related as

$$\begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix} = T_b R_b \begin{bmatrix} x_{1,b} \\ x_{2,b} \\ 1 \end{bmatrix}, \tag{2.4}$$

where $R_b(\gamma^b)$ and $T_b(O_1^b)$ are the rotation and translation transformation matrices from the body coordinates, respectively. On the other hand, a point on the contour, from the sensor's point of view, is given by

$$\begin{bmatrix} X_1 \\ X_2 \\ 1 \end{bmatrix} = T_s R_s \begin{bmatrix} r \\ 0 \\ 1 \end{bmatrix}, \tag{2.5}$$

where $R_s(\gamma^s)$ and $T_s(O_1^s, O_2^s)$ are the transformations from the sensor coordinates. Combining the two relationships we have

$$\begin{bmatrix} x_{1,b} \\ x_{2,b} \\ 1 \end{bmatrix} = R_b^{-1} T_b^{-1} T_s R_s \begin{bmatrix} r \\ 0 \\ 1 \end{bmatrix}. \tag{2.6}$$

Substitution into the contour (2.1) gives the k^{th} measurement reading of a point on the contour, in terms of the measured quantities as

$$g(x^s, r, \theta) = 0\,, \tag{2.7}$$

where

$$x^s = \begin{bmatrix} O_1^s & O_2^s & \gamma^s \end{bmatrix}^T\ . \tag{2.8}$$

So far the derivation has been purely geometric, assuming perfectly accurate measurements. In reality each of the measured quantities contains noise. The k^{th} measurement point is described by

$$\underbrace{\begin{bmatrix} x_k^s \\ r_k \end{bmatrix}}_{true} = \underbrace{\begin{bmatrix} z_k^s \\ \overline{r}_k \end{bmatrix}}_{measured} - \underbrace{\begin{bmatrix} v_k^s \\ v_k^r \end{bmatrix}}_{noise} = z_k - v_k\ , \tag{2.9}$$

where z_k are the actual measurements and the elements of v_k represent the corresponding noises. With this notation, (2.7) becomes

$$y_k = g(z_k, \theta, v_k) = 0\ . \tag{2.10}$$

Eqn. (2.10) is an implicit measurement where the artificial output is always zero but the actual measurements, in particular the distance *r*, appear as coefficients. The equation is nonlinear in its measurements, and consequently, in the noise as well. Since there is uncertainty about the sensor position and the orientation, these quantities, which in general change from one measurement to another, need to be estimated in addition to the object parameters. Define the vector

$$\tilde{x}_k = \begin{bmatrix} x_k^s \\ \theta \end{bmatrix} . \tag{2.11}$$

The overall measurement is, therefore,

$$\begin{bmatrix} z_k^s \\ 0 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} I & o \end{bmatrix} \tilde{x}_k + v^s \\ g(\tilde{x}_k, z_k, v_k) \end{bmatrix} . \tag{2.12}$$

Or generically

$$\tilde{y}_k = H(\tilde{x}_k, v_k) . \tag{2.13}$$

Notice that although only the parameters θ are of interest, they cannot be separated from the rest of $\tilde{x}_k$; one has to estimate x_k^s as well. The implicit measurement is non-linear. One way to overcome this is to use the extended Kalman filter (EKF), which uses a linearized version of the measurement. For the sake of brevity, we present only a general statement of the estimation scheme; the details can be found in [2].

$$\hat{\tilde{x}}_k = \overline{\tilde{x}}_k + K_k(\tilde{y}_k - H(\overline{\tilde{x}}_k, 0)) \tag{2.14}$$

$$K_k = K_k(\tilde{C}_k, \tilde{D}_k) , \tag{2.15}$$

where $\tilde{C}_k$ and $\tilde{D}_k$ are the coefficients of the state and the noise after linearization, which are given by

$$\tilde{C}_k = \begin{bmatrix} I & 0 \\ \dfrac{\partial g}{\partial x^s} & \dfrac{\partial g(\tilde{x})}{\partial \theta} \end{bmatrix}_{|\tilde{x}=\overline{\tilde{x}}_k} , \quad \tilde{D}_k = \begin{bmatrix} I & 0 \\ 0 & \dfrac{\partial g}{\partial r} \end{bmatrix}_{|\tilde{x}=\overline{\tilde{x}}_k} . \tag{2.16a,b}$$

Since the EKF is based on linearization about the *a priori* estimation $\overline{\tilde{x}}_k$, a natural extension is a recursive procedure with $\hat{\tilde{x}}_k$ replacing $\overline{\tilde{x}}_k$ and so on. This is the iterative extended Kalman filter (IEKF) [13]. A further extension is given in [2] with the introduction of the noise updated iterative extended Kalman filter (NUIEKF). The key idea is that better estimation of the state variables can be obtained if the measurement noise is updated iteratively as well. In general terms, the estimation in the *i*th iteration of the k^{th} time step is given by

$$\begin{aligned} \hat{\tilde{x}}_{k,i+1} &= \overline{\tilde{x}}_k + \mathrm{K}_{1,\mathrm{k,i}}\left(y_k - H(\hat{\tilde{x}}_{k,i}, \hat{v}_{k,i}) \text{-} \tilde{C}_{k,i}\,(\overline{\tilde{x}}_k - \hat{\tilde{x}}_{k,i}) + \tilde{D}_{k,i}\hat{v}_{k,i}\right) \\ \hat{v}_{k,i+1} &= \mathrm{K}_{2,\mathrm{k,i}}\left(y_k - H(\hat{\tilde{x}}_{k,i}, \hat{v}_{k,i}) \text{-} \tilde{C}_{k,i}\,(\overline{\tilde{x}}_k - \hat{\tilde{x}}_{k,i}) + \tilde{D}_{k,i}\hat{v}_{k,i}\right) \end{aligned} , \tag{2.17}$$

where the gains $\mathrm{K}_{1,\mathrm{k,i}}$ and $\mathrm{K}_{2,\mathrm{k,i}}$ are calculated based on the iterative linearization $\tilde{C}_{k,i}, \tilde{D}_{k,i}$.

Overhead measurements are somewhat simpler as they directly produce points that are, apart from the noise, on the contour $g(x_1^b, x_2^b, \theta) = 0$. The estimation process then follows along the same lines as in (2.8)–(2.10) but with (2.10) as the only measurement, i.e. $\tilde{x}_k = \theta$ and Eqns. (2.12) and (2.16) are reduced to their lower parts.

The situation is somewhat different when the measurement is made by a single, or repeated, camera picture. The details of that case are omitted for the sake of brevity and we will only mention that edge detection methods are used to define points that, apart from noise, lie on the contour. The situation then is similar to taking overhead laser measurements, and the estimation problem is that of fitting unknown parameters into a known function.

At the end of the measurement process, regardless of its technology, we have two or three orthogonal views. In the next section we outline the procedure for creating a 3D object out of these views.

3. TOPOLOGICAL RELATIONS AND DIMENSIONING ANALYSIS USING VARIATIONAL GEOMETRY

The input at this stage is a 2D view, which goes through a constraint evaluation process resulting in a 2D view constraint set. Dimensions define geometric constraints, such as distance between two points, distance between a point and a bar, and an angle between two bars. Spatial relations define topological constraints such as tangency, parallelism, and perpendicularity [23]. The constraints extracted from each 2D view represent relations among explicit and implicit characteristics. Each dimension is formulated as a constraint. There are two kinds of constraints: one, defined by a single equation, and the second, a compound constraint, which requires two or more equations.

Each constraint equation is a function of points in the geometric dimension scheme. Eqn. i, denoted f_i, is formulated as follows:

$$f_i = \{x_1, y_1, x_2, y_2, \ldots, x_n, y_n\}, \tag{3.1}$$

where n denotes the number of points constraints by the geometric entity.

For the complete 2D view, a set of constraints, denoted as *F*, is given as follows:

$$F = \{f_1, f_2, \ldots, f_m\}. \tag{3.2}$$

As an example, a distance from a point to a line is presented. To constrain the distance D between a point P_a and a line P_bP_c, two vectors must be defined: a unit vector $\hat{U}$ from P_b to P_c and a vector $\bar{V}$ from P_b to P_a; the distance D is a cross product, $\hat{U} \times \bar{V}$.

$$\hat{U} = \frac{x_c - x_b}{\left|P_bP_c\right|}\hat{i} + \frac{y_c - y_b}{\left|P_bP_c\right|}\hat{j} = U_x\hat{i} + U_y\hat{j} \tag{3.3}$$

and

$$\bar{V} = \left(x_b - x_a\right)\hat{i} + \left(y_b - y_a\right)\hat{j}, \tag{3.4}$$

where i and j are unit vectors in the *x* and *y* directions, respectively.

The point-to-line constraint is formalized as

$$f_1 = U_x\left(y_b - y_a\right) - U_y\left(x_b - x_a\right) - D = 0. \tag{3.5}$$

The system constructs a knowledge base of variational geometry rules for constraining the dimensioning scheme and the relations between the geometry sites in the view [15,23]. Constraining the dimensioning scheme is done by positioning a selected point, called the anchor point, at the origin in order to prevent solid body translation. All points are defined relative to this anchor point. To prevent solid body rotation, a bar is defined to be horizontal, i.e., parallel to the *x* axis. For the dimensioning and the constraint set to be valid, the Jacobian constraint matrix should meet two requirements. First, the number of constraints must be equal to twice the number of vertices, and second, the rank of the matrix must equal the number of constraints. Meeting these requirements indicates that the matrix is non-singular and hence there is neither redundancy nor lack of dimensions and definitions of the constraints. The constraint set F{Front}, for the front view in Fig. 3 is formalized in (3.6).

4. GRAPH REPRESENTATION AND DEPENDENCIES MATCHING 2D VIEW CONSTRAINT SET

The graph representation of a constraint expresses the relationship and connections among parameters. Moreover, a graph representation of a constraint set is a declarative structure that expresses the existence of relations among the parameters of more than one constraint. The motivation of this conversion is to find the minimal set of relations that fully represent the 2D view. The 2D pair representation provides a compact quantitative and qualitative mapping of the relations among the parameters for each view. The process of converting the constraint set into a 2D pair graph goes through the following stages:

- Representing the constraint set as a full undirected graph. The nodes represent the parameters and the arcs represent the existence of the constraint between the parameters. The arcs are labeled according to the constraint.
- Obtaining a bipartite graph from the complete undirected graph and finding a maximum matching for the bipartite graph. This is done by applying a matching process yielding a mach set.
- Transforming the complete undirected graph into a minimal graph using the dependencies found in the previous step. Using the results of the matching process, the undirected graph is transformed into a minimal graph by the following rules: For a matching pair (v,fi) the undirected graph is modified such that all arcs labeled other than fi and incident on v are removed.

- Converting the minimal graph into a pair graph is done by combining two single nodes into a one-pair node. For nodes labeled xi and yi, for example, i=j, a new pair node labeled $\{x_i,y_i\}$ is defined. All other arcs are removed.

4.1 Graph Representation Example

The following example illustrates the construction of the minimal undirected graph process from two 2D dimensioned orthographic views.

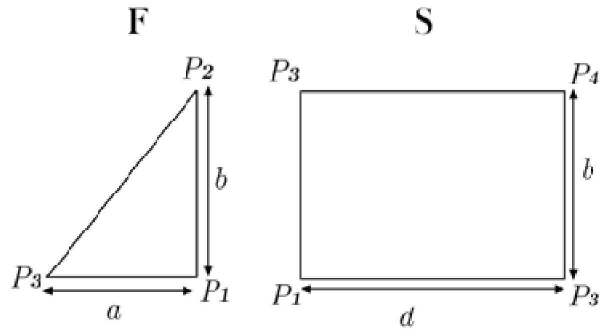

Fig. 3: Front view (F) and side view (S) of a triangular prism

A triangular prism is described in Fig. 3 by two orthographic views: the front view (F) and the side view (S).

In this example we used the following parameters for the vertices found in the two 2D views. For the front view we defined the three vertices $P_1(x_1,y_1)$, $P_2(x_2,y_2)$, $P_3(x_3,y_3)$ and for the side view we defined the four vertices $P_1(x_1,z_1)$, $P_2(x_2,z_2)$, $P_3(x_3,z_3)$, $P_4(x_4,z_4)$. Two rules from the variational geometry rule base described in Section 3 were used:

- Rule #1 – Euclidian distance between two points, and;
- Rule #2 – Perpendicularity.

To prevent solid body translation, the anchor point was fixed as the origin (0,0). All the points were defined relative to this anchor point. To prevent solid body rotation, we chose a particular bar to be horizontal, i.e., parallel to the horizontal axis.

For the front view of Fig. 5 we formulated the following six constraints:

$$f_1 : (x_2 - x_1)^2 + (y_2 - y_1)^2 - b^2 = 0$$

$$f_2 : (x_6 - x_1)^2 + (y_6 - x_1)^2 - a^2 = 0$$

$$f_3 : (x_1 - x_2)(x_6 - x_1) + (y_1 - y_2)(y_6 - y_1) = 0$$

$$f_4 : x_1 = d = 0$$

$$f_5 : x_1 = e = 0$$

$$f_6 : (y_2 - y_1) = 0 \tag{4.1}$$

From this set of equations we obtained a 6x6 Jacobian matrix. The rank of the matrix was calculated and the dimensioning was found to be proper and equal twice the number of vertices.

For the side view, eight other constraints were formulated as follows:

$$f_1 : (y_2 - y_1)^2 + (z_2 - z_1)^2 - d^2 = 0$$

$$f_2 : (y_8 - y_1)^2 + (z_8 - z_1)^2 - b^2 = 0$$

$$f_3 : (y_4 - y_8)^2 + (z_4 - z_8)^2 - d^2 = 0$$

$$f_4 : (y_4 - y_2)^2 + (z_4 - z_2)^2 - b^2 = 0$$

$$f_6 : (y_1 - y_2)(y_8 - y_1) + (z_1 - z_2)(z_8 - z_1) = 0$$

$$f_8 : y_1 = e = 0$$
$$f_8 : z_1 = a = 0$$
$$f_8 : (z_2 - z_1) = 0 \tag{4.2}$$

As before, we calculated an 8x8 Jacobian matrix from the set of equations. The rank of this matrix was found to be 8, indicating proper dimensioning and constraint definition of the side view as well.

Fig. 4 describes the complete undirected graph constructed from the set of constraints in (4.1). The graph was constructed by the method described in [20]. For example, the edge between x_3 and x_3 is labeled f_3 because the parameters are connected through the formulation of constraint f_3. Another example is the edge connecting x_1 and y_2, which is labeled f_1 and f_3, indicating that these parameters are connected by the two constraints, f_1 and f_3.

Fig. 5 is a bipartite graph, which was also obtained from the set of constraints in (4.1). The match set found from the maximum procedure is indicated by the thick lines connecting the parameters and the constraints. It consists of the edges: $M = \{(x_1, f_4), (y_1, f_5), (x_2, f_1), (y_2, f_6), (x_3, f_2), (y_3, f_3)\}$. Fig. 6 represents the minimal graph for the front view, obtained from the complete undirected graph of Fig. 4 and the corresponding bipartite graph of Fig. 5.

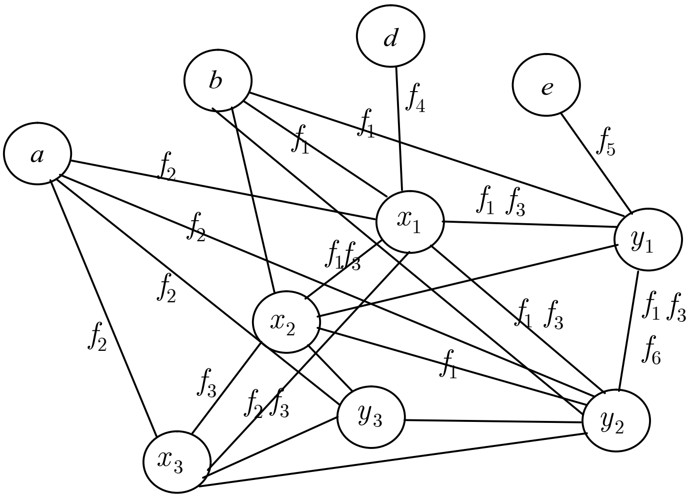

Fig. 4: Full undirected graph for the constraint set in 4.1

5. 3D RECONSTRUCTION APPROACH

Once the 2D pair graph for each orthographic view is obtained, a 3D object can be reconstructed. This process consists of three major stages [20]:

1. Initial matching
2. Complete matching
3. Graph-to-object conversion

The processes of initial matching and complete matching use the set of separate 2D pair graphs for each 2D view. These graphs are analyzed for matching and merging conditions. Fig. 9 schematically illustrates the starting point, in which each 2D view has an independent pair graph representation (left). The right scheme illustrates a complete graph generated by adding new edges (the dashed lines) that connect vertices in the various 2D pair graphs.

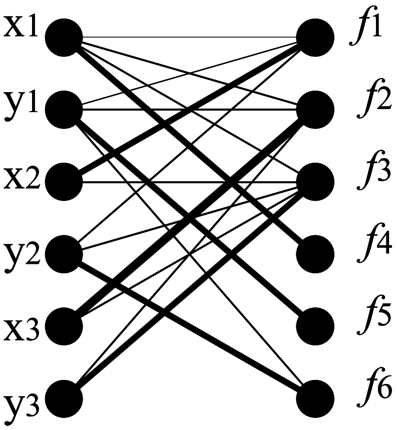

Fig. 5: Bipartite graph for the front view.

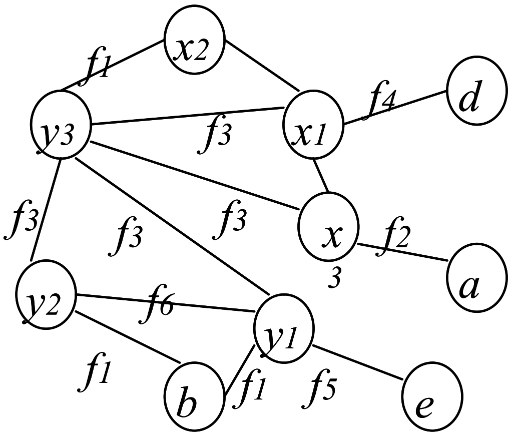

Fig. 6: Minimal graph for the front view.

Fig. 7 represents the pair graph constructed from the minimal graph. Fig. 8 describes the pair graph for the side view, [22].

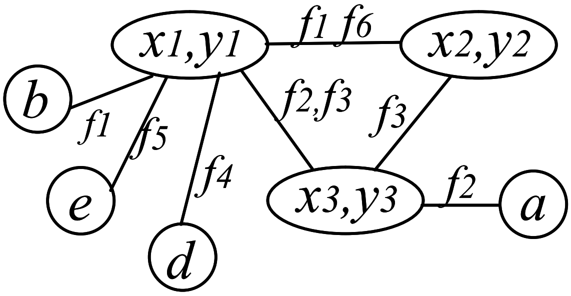

Fig. 7: Pair graph for the Front View.

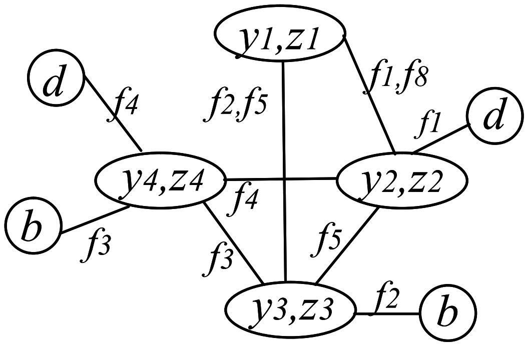

Fig. 8: Pair graph for the Side View.

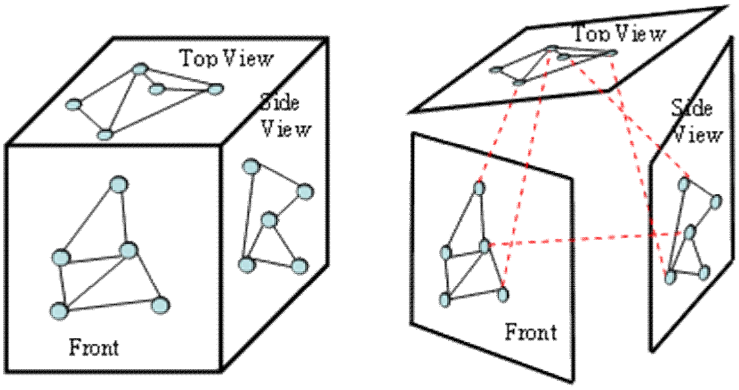

Fig. 9: A schematic illustration of the composite 3D graph representation [20].

Matching the first 3D vertex is based on searching each one of the 2D projections for different plane representations of the vertex. These vertices from the projections are matched simultaneously in the representation graphs.

Fig. 10 represents the initial matching for the triangular prism in Fig. 4. The figure shows only a partial initial matching of the composite graph. As indicated by the brown circle, the initial matching in the graph represents the coincidence of p_1 on the front view and p_1 on the side view. This is a representation of the same first 3D vertex in two different 2D views.

The initial 3D vertex serves as a starting point for the matching completion. Each link represents both the connecting related parameters and the nature of the link (e.g., parallelism, dimensions, etc.). The procedure of complete matching starts from the initial point on the initial composite graph and proceeds by searching the connected edges of the initial 3D vertex, while checking two criteria: connectivity and constraint type. Edges having the constraint type of Euclidian distance between two points are chosen first, followed by a matching criterion of topological connections represented in the graph or equality of coordinate values in the relevant planes. The complete matching process yields a composite graph, where all 3D vertices in the 3D object are represented by the triplet of two nodes and an edge linking them.

The complete composite graph is converted into a 3D object by translating the links, connections and constraint types into geometry and topology. The 3D object is represented by boundary representation (B-rep). This process is based on using graph theory tools, search models and heuristics for retrieving information from the composite graph and then using the high level understanding of the 2D views. The Chinese Postman Problem (CPP) algorithm is used for arc routing of all vertices in the graph and traversing all edges by the criterion of minimal total distance. A short introduction to the stages of the algorithm, which is used for the 3D reconstruction process, is presented. The postman, prior to starting his route, must pick up the mail from the post office. He then delivers the mail along each block on his route, and finally returns to the post office. To make the job easier, he does it with minimal walking.

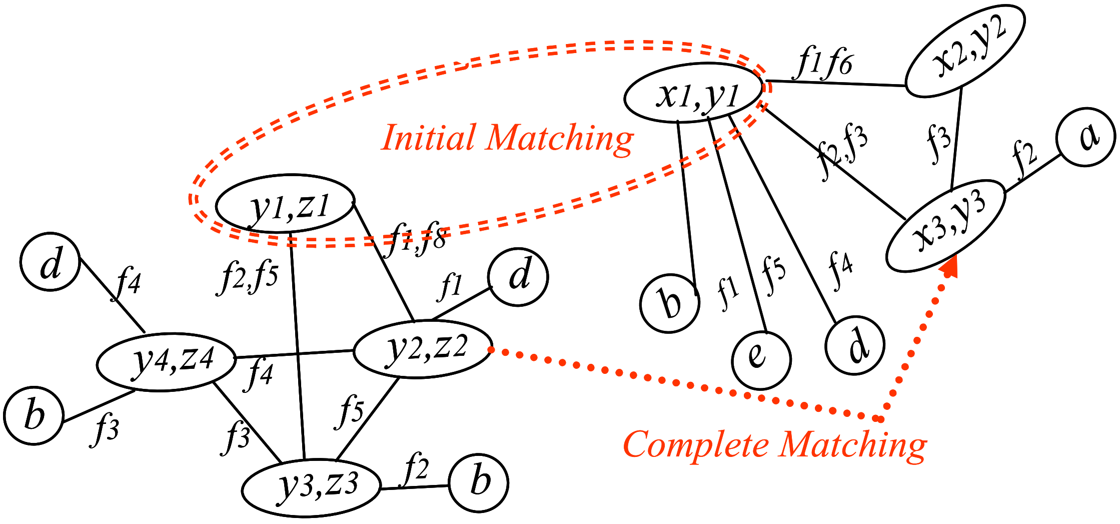

Fig. 10: Partial composite graph for the top and side views.

Given a graph $G=(V,E)$, which consists of a set of nodes $V=\{v_1,v_2...\}$ and a set of edges $E=\{e1,e2,...\}$, this problem is equivalent to finding a cycle in G that traverses each edge at least once in minimal total distance.

Definition: Any cycle in a graph that crosses each edge exactly once is an Euler tour.

Definition: An undirected graph is a Euler graph if and only if all vertices have even degrees.

Constructing Euler graphs and Euler tours is detailed in [8]. Solving the CPP for any undirected Euler graph is described below:

A Euler tour maintains a list of vertices $Lx(1), Lx(2),..., Lx(k)$ to the next visit, when vertex x is reached the k^{th} time.

Step 1: Select any vertex x_0. Let $v = v_0$ and $kx = 0$ for all vertices x. All edges are labeled "unused".

Step 2: Randomly select an unused edge incident to vertex v. Mark this edge "used". Let y be the vertex at the other end of the edge. Set $kv = kv + 1$ and $Lv(kv) = y$. If the vertex y has any incident, unused edges, go to Step 3; otherwise vertex y must be v_0. In this case, go to Step 4.

Step 3: Set $v = y$ and return to Step 2.

Step 4: Let v_0 be a vertex that has at least one used edge and one unused edge incident to it. Set $v = v_0$ and return to Step 2. If no such vertex exists, go to Step 5.

Step 5: To construct the tour, start at v_0. The first time vertex x is reached, leave it by going to vertex $Lx(kx)$. Set $kx = kx - 1$ and continue, each time going from vertex x to vertex $Lx(kx)$.

Implementation of the CPP assists in defining tours in which each triplet (two nodes and a connecting edge) is translated into the geometry of the 3D object, depending on the constraint type and value. This procedure is composed of the following steps: The first tour is defined by starting from the initial matching, a 3D vertex represented in the composite graph and the contour loop defined in the high level understanding. This step has the following sub-processes:

- Choose the initial 3D vertex representation for the selected view and label it "selected node".
- Check the edges connected to the "selected node" for a tuple (edge or node) so that the node represents a vertex on the contour loop. If more than one node is found, select one randomly.
- Traversing to the node found in the selected view, check the composite graph for connecting edges in the remaining views.
- Translate the vertex node representation and relevant constraint type represented by the edge to the geometry and topology. In cases where more than one constraint is represented by the edge, each of them is checked for feasibility by checking the loop content value found in the high level understanding stage.
- Implement sub-processes 2-4 until the first "selected node" is reached for the second time.

Stopping condition: Translation from the 3D composite graph is completed when all loops found have been covered.

The 3D boundary models indirectly represent a solid through a representation of its bounding surface. Boundary representation models are a collection of faces representing a solid object. The 3D object reconstruction from three engineering drawings representing orthographic views yields a boundary representation of a solid. Fig. 11 is the geometrical example of the results of the 3D reconstruction of two simple examples, Figure 12 is an example of a complex 3D reconstruction. The full detailed process of the 3D reconstruction, and more examples, are found in [20].

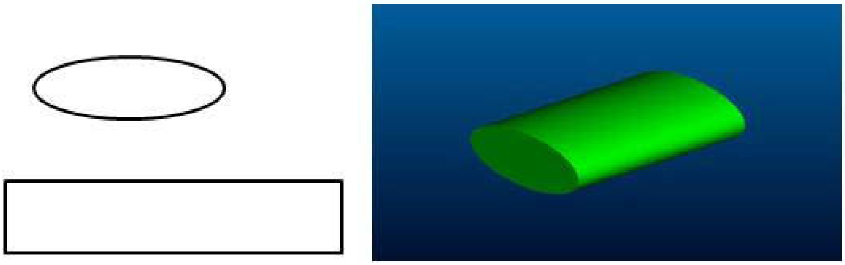

Fig. 11: A simple examples of the 3D reconstruction process.

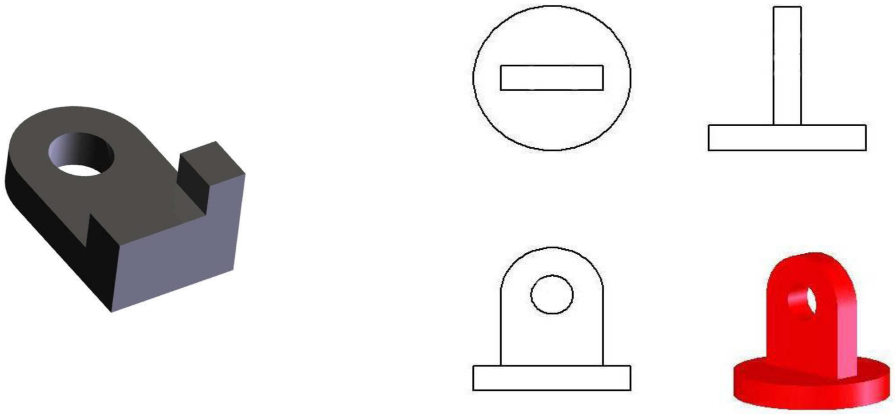

Fig. 12: Two Simple examples of a 3D reconstruction.

6. CONCLUSION

A comprehensive method for automatically constructing a 3D solid model from orthographic views obtained by a moving sensor was reported and described. It is assumed that the shape of the object body, actually sub-objects of the entire body, is known but its parameters are not. Also, the position of the body need not be known in advance. The raw distance measurements are processed via a filter that generates estimates of the part dimensions and position. Since the formulation leads to implicit measurement equations, standard extended Kalman filter techniques usually fail to converge to accurate values. A new method, called a Noise Updated Iterative Extended Kalman Filter, was developed and used.

Solving the problem of 3D object reconstruction from 2D dimensioned orthographic views combines elements drawn from the experienced human mind. The process is implemented by using variational geometry representation and graph theoretic tools to construct a composite graph representing the 3D object. In the stage we translate the graph into an explicit geometric model. These elements are amenable to automation, and the complex procedure described in this work, therefore, serves as a means for reliable and accurate 3D reconstruction of solids.

The fact that the outcome of the estimation (measurement) process is a set of explicit contour equations is suitable for the second step, which is creating a 3D model from the orthographic views through a graph theoretic approach. In previous applications these equations had to be built from the views as a preliminary step. The match between the output of the Kalman filtering approach for 2D reconstruction, and the starting point for the graph theoretic approach for 3D reconstruction is the key advantage of the integrated approach.

7. REFERENCES

[1] Beck, J. V.; Arnold, K. J.: Parameter Estimation in Engineering and Science, Wiley Series in Probability and Mathematical Statistics. J. Wiley, New York. 1997.

[2] Bondarenko, A.; Halevi, Y.; Shpitalni, M.: Object identification and tracking via noise updated iterative extended Kalman filter, 7th Biennial ASME Conference on Engineering Systems Design and Analysis (ESDA). 2004: Manchester, UK, 7p.

[3] Chen, K.-Z.; Feng, X.-A.: Holo-extraction of information from paper drawing for 3D reconstruction, Computer Aided Design, 34(9), 2002, 665-677.

[4] Çiçek, A.; Güles, M.: Reconstruction of 3D models from 2D orthographic views using solid extrusion and revolution, Journal of Materials Processing Technology, 152(3), 2004, 291-298.

[5] Cippola, R.; Blake, A.; Surface shape from the deformation of apparent contours, International Journal of Computer Vision, 9(2), 1992, 83-112.

[6] De Geeter, J.; Van Brussel, H.; De Schutter, J.: A smoothly constrained Kalman filter, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(10), 1997, 1171-1177.

[7] Dori, D.; Weiss, M.: A scheme for 3D object reconstruction from dimensioned orthographic views, Engineering Applications of Artificial Intelligence, 9(1), 1996, 52-64.

[8] Durrant-Whyte, H.; Bailey, T: Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. Robotics and Automation Magazine 13, 2006, 99–110.

[9] Faugeras, O. D.: Three Dimensional Vision, A Geometric Viewpoint, MIT Press, Boston, 1993.

[10] Gao, X.-S.; Hoffmann,C. M.; Yang, W.-Q.: Solving spatial basic geometric constraint configurations with locus intersection, Computer-Aided Design 36(2), 2004,111-122.

[11] Geng, W.; Wang, J.: Embedding visual cognition in 3D reconstruction from multi-view engineering drawing, Computer Aided Design, 34(4), 2002, 321-336.

[12] Huber, P. J.: Robust Statistics, John Wiley & Sons, New York, 1981.

[13] Jazwinsky, A. M.: Stochastic Processes and Filtering Theory, Academic, New York, 1970.

[14] Langbein, F. C.; Marshall, A. D.; Martin, R. R.: Choosing Consistent Constraints for Beautification of Reverse Engineered Geometric Models, Computer Aided Design 36 (3), 2004, 261-278.

[15] Lin, V. C.; Light, R. A.; Gossard, D. C.: Variational geometry in computer aided design, Computer Graphics, 14 1981, 171-177.

[16] Liu, S.; Hu, S.; Chen, Y.; Sun, J.: Reconstruction of curved solids from engineering drawing, Computer-Aided Design, 33, 2001, 1059-1072.

[17] Martson, R. E.; Kuo, M. H.: Reconstruction of 3D object from three orthographic projections using decision chaining method, IAPR MVA'94, Workshop on Machine Vision Applications 1994; Japan.

[18] Mendel, J. M.: Lessons in Digital Estimation Theory, Prentice-Hall, Englewoods Cliff, NJ, 1987.

[19] Shi-Xia, L.; Shi-Min, H.: Reconstruction of curved solids from engineering drawings, Computer-Aided Design, 33, 2001, 1059-1072.

[20] Shin, B.-S.; Shin, Y.-G.: Fast 3D solid model reconstruction from orthographic views, Computer-Aided Design, 30(1), 1998, 63-76.

[21] Shotton J.; Blake, A.; Cipolla, R.: Multiscale Categorical Object Recognition Using Contour Fragments, IEEE Trans. Pattern Anal. Mach. Intell, 30(7), 2008, 1270-1281.

[22] Shum, S. P.; Lau, W. S.; Yuen, M. F.; Yu, K. M.: Solid reconstruction from orthographic opaque views using incremental extrusion, Computers & Graphics, 21(6), 1997, 787-900.

[23] Weiss-Cohen, M.: Reconstruction of solids models from orthographic views using knowledge retrieval and composite graphs, Computer-Aided Design and Applications, 45, 2007, 159-167.

[24] Zhang, Z.: Parameter estimation techniques: A tutorial with application to conic fitting, Image and Vision Computing Journal, 15(1), 1997, 59-76.