www.cadanda.com Computer-Aided Design AND APPLICATIONS

A Linear Algorithm for Pseudo Recognition in 3D Reconstruction

Qi Li[1,2,3,4] and Hui Zhang[1,3,4]

[1]School of Software, Tsinghua University
[2]Department of Computer Science and Technology, Tsinghua University
[3]Key Laboratory for Information System Security, Ministry of Education
[4]Tsinghua National Laboratory for Information Science and Technology
selmalq@gmail.com, huizhang@tsinghua.edu.cn

ABSTRACT

Pseudo recognition is essential to reconstruct correct 3D models from engineering drawings. This paper proposes a novel linear algorithm to recognize pseudo elements based on the existence of non-manifold edges. First, mini face loops are defined to construct correct face loops and avoid overlapped loops, which previous loop construction methods would always result in when pseudo elements exist. Then, suspect faces whose all boundary edges are non-manifold edges, are defined to reduce the searching space of the algorithm for pseudo recognition, thus the algorithm can achieve linear time complexity. The algorithm is applicable for wireframe models including planes and basic conicoid surfaces. Experimental results are provided to demonstrate the algorithm.

Keywords: 3D reconstruction, pseudo elements, mini face loop, suspect face.
DOI: 10.3722/cadaps.2011.827-839

1 INTRODUCTION

3D solid reconstruction from engineering drawings is an important research topic in Computer Aided Design. In the research area of reconstruction, boundary representation (B-Rep) model is extensively used to describe a 3D solid, as it can give a complete boundary description and topology structure information, and it is also really convenient and unique to handle complex objects. Flow diagrams of reconstruction algorithm for B-Rep model can be summarized as follows:

- Get orthographic projection relationship among three engineering views.
- Construct 3D points and edges of the object.
- Search for faces of the object.

The reconstruction method described above is the reverse process of orthographic projection. In orthographic projection, several 3D edges may be projected to an identical line in engineering drawings; while in the procedure of reconstruction, several lines from different views would be combined together to generate a 3D edge. However, redundant combination of lines would result in pseudo elements, which can prevent the reconstruction algorithm from searching the correct faces of the object. Therefore, pseudo recognition is a really important topic in 3D reconstruction.

Ever since Idesawa [1] proposed the algorithm of reconstruction from three orthographic views in 1973 first, many researchers have tried to find a method to reconstruct a correct wireframe model by recognizing all pseudo elements with high efficiency. Maikowsky and Wesley [2, 3] was the first to propose a method of body loops searching for recognizing all pseudo elements in 1980, which is based on topology information such as Moebius Rule and characteristics of two-manifold objects. The method was then improved by Gu *et al* [4] in 1986. Nevertheless, the method of body loops searching is inefficient, as it takes too much time in constructing virtual blocks.

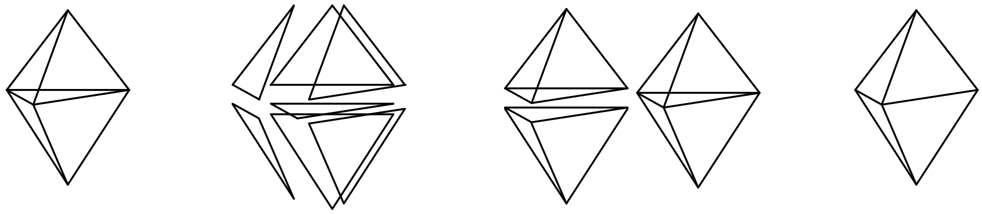
In 1988, Lequette [5] proposed Local condition (Moebius Rule) and Global condition to detect pseudo elements and generalized some decision rules to combine candidate faces to obtain the correct wireframe model. In 1989, Gujar and Nagendra [6] proposed an algorithm based on geometry information i.e. 2D dashed lines in engineering drawings to detect a specified type of pseudo elements. And Yan et al. [7] proposed a method for pseudo recognition based on the degree of vertices and collineation of edges by analyzing topology structure in 1994. In 1996, You and Yang [8] proposed a divide-and-conquer algorithm to remove non-coexistent faces and non-manifold edges to get the correct solid model. However, these pseudo recognition methods can only handle a small range of pseudo elements.

In 1998, Kuo [9, 10] proposed a new method of decision-chaining which can remove all pseudo elements based on 9 rules, summarized by Moebius Rule and characteristics of engineering drawings. This method is much better than the method of body loops searching in efficiency. Nevertheless, the method of decision-chaining has a limitation of combinatorial explosion for some objects with a large number of non-manifold edges. After that, Liu et al. [11] proposed a detailed method based on geometry information to recognize part pseudo elements in 2002, and Ye et al. [12] improved the method in 2006.

The existing algorithms for pseudo recognition can be classified in two categories:

- **By geometry information.** Geometry information such as depth information and 2D dashed lines in engineering drawings can be used to recognize pseudo elements. The efficiency of this kind of algorithm is high. However, they can only handle a small range of pseudo elements.
- **By topology information.** Topology information such as Moebius Rule and characteristics of a two-manifold object can be used to recognize all the pseudo elements in reconstruction. Unfortunately, since this kind of algorithm requires searching large quantities of face loops of the object, the efficiency is relatively low.

Existing methods usually take geometry information first to recognize part pseudo elements, then search face loops of the object and take topology information to remove the rest pseudo elements. According to the above methods, the second step for completely pseudo recognition can be generalized to two major methods:



(a) (b) (c) (d)

Fig. 1: Body loops searching: (a) wireframe; (b) virtual faces; (c) virtual blocks; (d) object.

- **Body Loops Searching.**
  This method is based on the algorithm proposed by Markowsky and Wesley [2, 3], and then improved by Gu et al. [4]. The process of the method is shown in Fig. 1. First, the method searches the wireframe for all virtual faces (see Fig. 1(b)) to construct all possible virtual blocks (see Fig .1(c)) with introducing cutting vertices and cutting edges. A virtual block is completely constructed if all the edges of the block are shared by two virtual faces. Also all the virtual blocks are completely constructed if all the virtual faces are selected twice. Then, the method constructs a decision-tree to assign *solid* or *hole* state to each virtual block. When all the virtual

blocks are assigned, an object is obtained. However, not all assignments can yield the desired wireframe unless the result is in accordance with the characteristics of two-manifold object. The method would construct dozens of virtual blocks for complex objects, while most of these virtual blocks would be discard after assignment. Hence, the efficiency of the algorithm is relatively low.

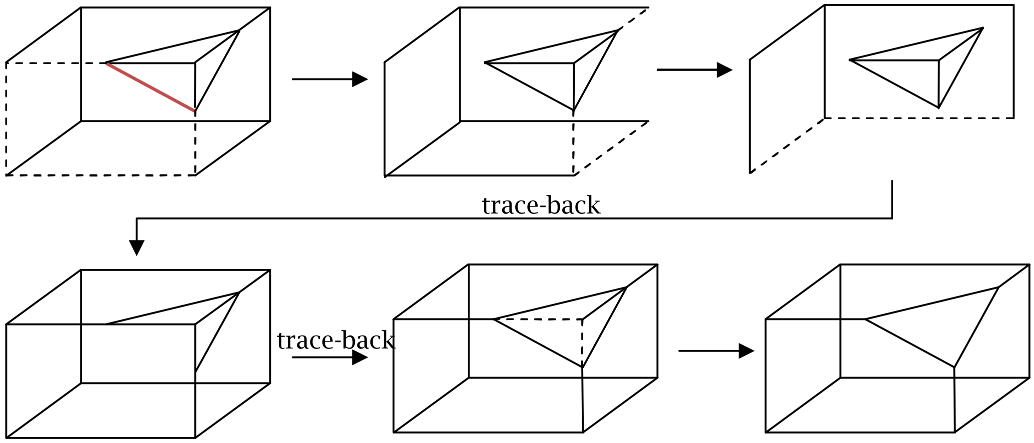


Fig. 2: Face loops combining.

- **Face Loops Combining.**
  This method is based on the algorithm proposed by Kuo [9, 10]. The method traces candidate faces first, and then uses decision-chaining method to search correct combination of candidate faces. The process of the method is shown in Fig. 2. First, a non-manifold edge is chosen, whose adjacent faces number is not equal to 2. Two of its adjacent faces are assumed to be true faces (the pseudo faces in assumption is shown as dashed lines in Fig. 2) and then all the other elements of the wireframe should be examined based on 9 rules, until another non-manifold edge is examined and another assumption should be made to continue the decision-chaining method. In the process of examining, if some rules cannot be satisfied, then one of the previous assumptions is false. Trace back to the previous assumption and make another assumption to examine the 9 rules until a correct wireframe is completely constructed. For some complex objects with a large number of non-manifold edges, the method would result in the problem of combinatorial explosion, though the efficiency of the method is better than body loops searching. Hence, the time complexity of the method of face loop combining is high.

In general, these existing pseudo recognition methods remain many problems, which can be summarized as follows:

- Not all pseudo elements can be recognized. Many pseudo points, pseudo edges and pseudo faces cannot be robustly detected for complex solid models.
- The efficiency of pseudo recognition is rather low. The existing reconstruction methods usually take much time in recognizing pseudo elements and searching for the correct solid. Therefore, they are often hard to be used in practice.

In this paper, we propose a novel linear algorithm based on the existence of non-manifold edges for completely pseudo recognition. The efficiency of the algorithm is significantly high in reconstruction and the correctness can be also ensured in our experiments.

The rest of the paper is organized as follows. Section 2 gives some theorems and definitions to explain the algorithm. Section 3 describes the details of the algorithm, which can be divided as three parts: construction of mini face loops, suspect faces examining and true faces searching. Section 4 analyzes the time complexity of the algorithm and compares our method with decision-chaining method by Kuo [9, 10]. The result of the algorithm and several examples are provided in Section 5 to illustrate the performance of our algorithm. Section 6 is the conclusion of the paper.

2 THEOREM AND DEFINITION

Vertices, edges and faces of wireframe model, are all connected by topology features. To ensure the machinability of the object, the object should be a two-manifold solid. Some theorems and definitions are summarized as follows:

Lemma 1 (Moebius Rule): In a two-manifold solid, any edge is shared exactly by two non-coplanar faces, and its direction is reverse in different faces (see Fig. 3).

Definition 1: In a wireframe model, an edge is called a non-manifold edge if the number of its adjacent faces is not 2.

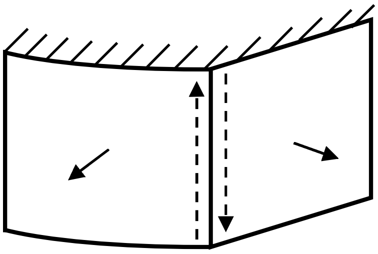


Fig. 3: Moebius rule.

Each face in wireframe model has its own face direction which is represented by its normal vector. And each face is represented by a set of boundary edges called outer face loop, whose direction is the same as the face's normal vector by Thumb rule. Also a face may have some inner face loops within outer face loop, and they are in the opposite direction of the face. Each edge needs to be exactly shared by two non-coplanar faces no matter the edge belongs to outer face loop or inner face loop, and the edge's direction is reverse in the two faces. Thus, the existence of non-manifold edges means the existence of pseudo elements in wireframe model. A correct wireframe model can be reconstructed after removing all non-manifold edges.

Lemma 2: Two faces of a two-manifold solid should not have any intersecting line except their boundary edges.

This lemma implies that two intersecting faces cannot exist in wireframe at the same time. Therefore, at least one of these two faces should be a pseudo face.

The decision-chaining method by Kuo [9, 10] is based on nine rules concluded from Lemma 1 and Lemma 2. But these rules have the limitations of redundancy and disorder. Here we propose six basic rules which are ordered by priority in our pseudo recognition algorithm to remove all pseudo elements and finally construct the correct wireframe model.

Theorem 1: six rules in our pseudo recognition algorithm are ordered by priority as follows:

- **Rule 1** Projection Completeness: Each line in engineering drawing should be reflected to a 3D edge in solid model.
- **Rule 2** Intersecting Faces: If a wireframe model has two intersecting faces, at least one of them is a pseudo face.
- **Rule 3** Non-manifold Edge: An edge which is shared by two coplanar faces is a pseudo edge.
- **Rule 4** Non-manifold Edge: If an edge is only shared by two faces, both of the faces should be pseudo faces or true faces.
- **Rule 5** Non-manifold Edge: If an edge is only shared by one face, the edge and the face are both pseudo elements.
- **Rule 6** Two-manifold Edge: If the number of an edge's adjacent faces is more than 2, at most two of these faces are true faces, and the rest are all pseudo elements.

The first rule ensures the completeness of orthographic projection which is the basis of reconstruction. The second rule is concluded from the examining of practical models, and intersecting faces should be removed first. The following three rules are equal in priority. They can be used in pseudo recognition algorithm at the same time to detect pseudo elements of adjacent faces. And the last rule is based on the characteristics of a two-manifold object, which ensures the removal of all pseudo elements, as it can be used to remove all non-manifold edges in wireframes.

3 LINEAR ALGORITHM FOR PSEUDO RECOGNITION

The algorithm for pseudo recognition is based on geometry and topology information to remove all pseudo elements in wireframe model and construct the correct wireframe model in reconstruction.

Before stepping into the algorithm, we adopt several pseudo recognition methods based on geometry information to reduce the searching space of face loops. According to the methods described in Section 1, cutting vertices and cutting edges are introduced into pathological edges and faces to obtain the correct wireframes (Gu et al. [1]). And depth information is taken to remove some redundant edges (You and Yang [8]). Also, topology information such as the degree of vertices is taken to remove part of pathological elements (Yan et al. [7]).

Our algorithm for pseudo recognition can be divided into three stages:

- Construction of mini face loops.
- Suspect faces examining.
- True faces searching.

The first stage construct face loops from wireframes to provide elements of searching spaces for pseudo recognition algorithm. The stage of "suspect faces examining" recognizes pseudo elements along with the suspect faces which stand a good chance that the pseudo elements would be. Thus most pseudo elements can be detected and result in a marked reduction of searching spaces to the next stage. And the stage of "true faces searching" can recognize the rest pseudo elements, as the combination of confirmed true faces can exclude the non-manifold edges and thus removal of all non-manifold edges guarantee the removal of all pseudo elements. Hence our algorithm can obtain a correct wireframe model.

3.1 Construction of Mini Face Loops

In a correct wireframe model, the relationship among face loops on the same plane is either nestification or separation. When pseudo elements exist, face loops would be overlapped with each other and general face loop construction algorithms such as turn-to-the-left-most method (Gujar and Nagendra [6]) and Face-Loop Generation algorithm (Yan et al. [7]) cannot construct correct face loops of the object. Therefore, we propose a new definition called mini face loop to solve this problem.

Definition 2: The face loop which is composed of all the boundary edges of a connected region on a plane is called mini face loop.

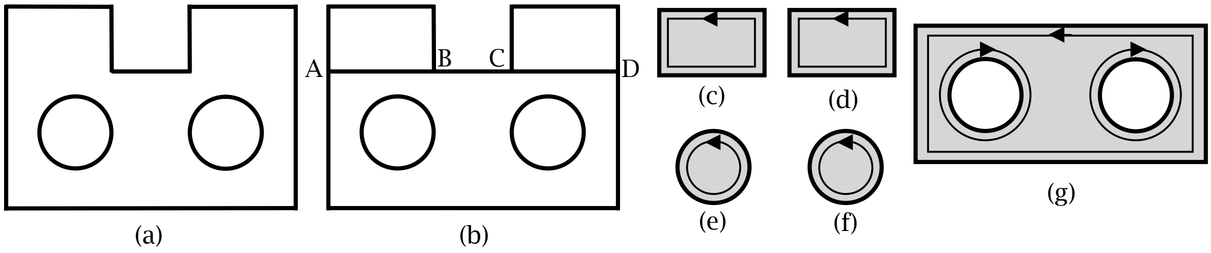


Fig. 4: (a) Correct plane wireframe, (b) plane wireframe with pseudo edges, (c, d, e, f, g) mini face loops.

Taking Fig. 4 as an example, when the plane wireframe has pseudo edges AB and CD, the original three separated face loops in Fig. 4(a) turn into some overlapped face loops in Fig. 4(b). However, previous loop construction methods cannot divide these loops properly, and they would even miss some face loops such as (e) and (f) in Fig. 4. In our algorithm, all overlapped face loops can be divided into five basic face loops shown in Fig. 4(c, d, e, f, g), which are composed of boundary edges of connected regions on the plane, called mini face loops. The directions of mini face loops in Fig. 4 are all in accordance with Thumb rule, and the normal vector of the plane is outward.

The algorithm for construction of mini face loops can be briefly summarized as follows:

- **Step 1**: Construct planar faces and sort the adjacent half edges of all vertices by the angle on each plane.
- **Step 2**: Get a non-constructed half edge and select its end vertex's next adjacent half edge according to the sorted list in Step 1 until the loop is closed. We can obtain all initial base loops shown in Fig. 5.
- **Step 3**: Compute the normal vector of each base loop and compare it with the normal vector of the face. If they are the same, we call the loop as positive loop shown Fig. 5(a). Otherwise, we call it the negative loop shown Fig. 5(b).
- **Step 4**: Delete the maximum outer loop in negative loops, and count the number of the rest negative loops. If the number is zero, all the positive loops are mini faces loops; else all the rest negative loops are inner loops. We should find its corresponding outer loop in the positive loop list and merge them together (when existing more than one inner loop corresponding to the same outer loop, we merge all of them together). Thus we obtain all mini face loops.

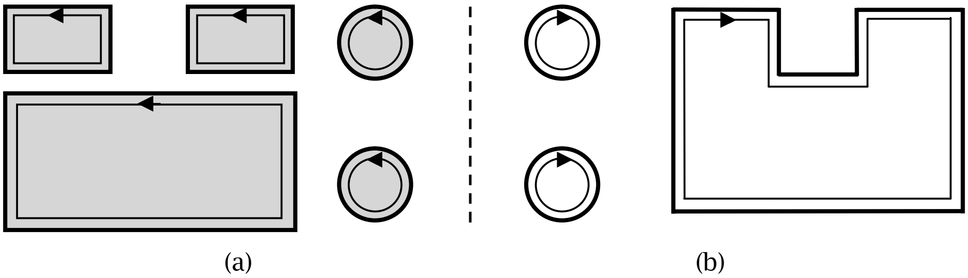


Fig. 5: Basic face loops: (a) positive face loops; (b) negative face loops.

3.2 Suspect Faces Examining

3.2.1 Processing Intersecting Faces

According to the priority of the six rules in Theorem 1, we should handle the intersecting faces in wireframe models first.

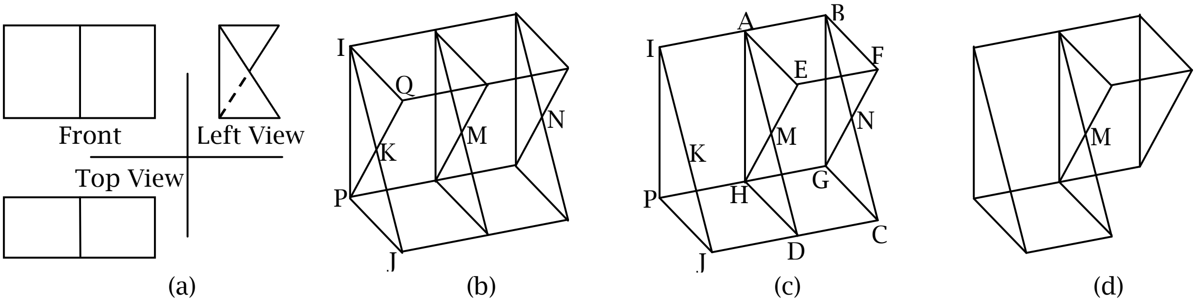


Fig. 6: (a) Three orthographic views; (b) incorrect wireframe model; (c) modified wireframe model; (d) correct wireframe model.

The wireframe model is preprocessed by partly pseudo recognition methods in Fig. 6. Cutting vertices K, M, N are introduced in Fig. 6(b), and depth information, i.e. 2D dashed line shown in the left view in Fig. 6(a) is used to remove part of pseudo edges in Fig. 6(c). But the wireframe still has two intersecting faces ABCD and EFGH after the construction of mini face loops. Both of the two faces cannot exist at the same time. We examine these faces by Rule 1 in Theorem 1 to ensure the completeness of orthographic projection in engineering views. If the face EFGH is a pseudo face, it will be deleted along with the face ABEF, which will result in missing projection lines of edges AE, BF, ME and NF in engineering views in Fig. 6(a). Therefore, the face EFGH is a true face, and face ABCD is a pseudo face. We can get the correct wireframe model shown in Fig. 6(d).

3.2.2 The Algorithm of Suspect Faces Examining

According to Lemma 1 (Moebius Rule), every edge can only be adjacent to two non-coplanar faces in a correct wireframe model. If the adjacent faces number of the edge is less than 2, the edge and its adjacent face are both pseudo elements; if the adjacent faces number of the edge is more than 2, only two of these faces can be true faces (If the edge itself is a pseudo edge, at most two coplanar faces are true faces, and the rest are all pseudo faces). Hence, the existence of non-manifold edges means the existence of pseudo elements. When the boundary edges of a face (including outer face loop and inner face loop) are all non-manifold edges, the face has a high probability to be a pseudo face. We define suspect face as follows:

Definition 3: In the wireframe model, if all the boundary edges of a face are non-manifold edges, the face is called a suspect face.

In pseudo recognition, searching suspect faces instead of all face loops could reduce the searching space and obviously improve the efficiency of reconstruction. The suspect faces examining algorithm is described as follows:

Algorithm 1: ***Suspect Faces Examining***

Input: Wireframe with no intersecting faces, Mini Face Loops List

Output: *Flag*(l_i) as the state of each face loop l_i, (*UNKNOWN*: initial state of all face loops; *SUSPECT*: state for suspect face loops; *TRUE*: state for true face loops; *PSEUDO*: state for pseudo face loops)

```
For (each mini face loop l_i in Mini Face Loops List)
    If (Flag(l_i)≠UNKNOWN), continue;
    Set Flag(l_i)= SUSPECT;
    For (each loop edge e_j on mini face loop l_i)
            Let AdjFaceNum = the adjacent faces number of e_j;
            If (AdjFaceNum=1), call Process Pseudo Face (l_i), break;
            If (AdjFaceNum=2), set Flag(l_i)=UNKNOWN, break;
            If (AdjFaceNum>2), continue;
    If (Flag(l_i)≠SUSPECT), continue;
    Examine the Projection Completeness(Rule 1), supposing l_i is deleted from the wireframe
            Let Projection = the result of the examination;
            If (Projection=TRUE), call Process True Face (l_i);
            Else, call Process Pseudo Face (l_i);
```

Subroutine 1: ***Process Pseudo Face (Mini Face Loop*** l_i***)***

```
Set Flag(l_i)=PSEUDO;
For (each loop edge e_j on mini face loop l_i)
    Delete l_i form the adjacent faces list of e_j;
    Let AdjFaceNum = the adjacent faces number of e_j;
    If (AdjFaceNum=0), delete e_j;
```

6. If (*AdjFaceNum=1*), call *Process Pseudo Face (the adjacent face of e_j)*;
7. If (*AdjFaceNum=2* and the two faces of e_j are coplanar), merge them together as a new face;

Subroutine 2: ***Process True Face (Mini Face Loop l_i)***

1. Set *Flag(l_i)=TRUE*;
2. For (each loop edge e_j on mini face loop l_i)
3. Let *AdjFaceNum* = the adjacent faces number of e_j;
4. If (*AdjFaceNum=2*)
5. Let l_k = the adjacent face of e_j besides l_i;
6. If (l_i and l_k are coplanar), merge them as a new face l_t, call *Process True Face (l_t)*, break;
7. Else if (*Flag(l_k)=UNKNOWN*), call *Process True Face (l_k)*;
8. Else find another true face l_k form the adjacent faces list of e_j, if there exists
9. For (each adjacent face l_t of loop edge e_j besides l_i and l_k), call *Process Pseudo Face (l_t)*;
10. If (l_i and l_k are coplanar), merge them as a new face l_t, set *Flag(l_t)=TRUE*;

According to Definition 3, there are 10 suspect faces in the incorrect wireframe model in Fig. 7(b), e.g. CBIJ, CDKJ, COPJ, JPI, JPK, JPG, GJK, GJI, GPK and GPI. Compared with the correct wireframe model shown in Fig. 7(c), we can see that 4 of them are true faces, the rest 6 faces are pseudo face, and the other 4 pseudo faces COB, COD, POBI and PODK are not suspect faces.

We examine the 4 true faces by our algorithm. If we delete face CBIJ then BI would be also deleted, and if we merge the coplanar adjacent faces BIPO and BIGHA of BI, then the edge CJ would be a broken line projection in left engineering view, which is not equal to the left view in Fig. 7(a); if we delete BI with its all adjacent faces, then CJ is not a broken line projection while the deletion of BIGHA would result in the deletion of HGFK, and edges HG and KF are projection lost in left view, thus the face CBIJ cannot be deleted and it's a true face. Also we can examine the other 3 faces CJDK, CJI, CJK to be true faces by our algorithm.

The rest 6 probable pseudo faces will not result in projection lost if they are deleted. And after the deletion of faces PKG and PKJ, the non-suspect face PODK would also be deleted. In the same way, the non-suspect faces POBI, COB and COD all can be detected as pseudo faces. Thus, after the "Suspect Faces Examining" algorithm, all pseudo elements are deleted, and a correct wireframe model has been obtained before stepping into the algorithm "True Faces Searching". The efficiency of reconstruction is improved obviously.

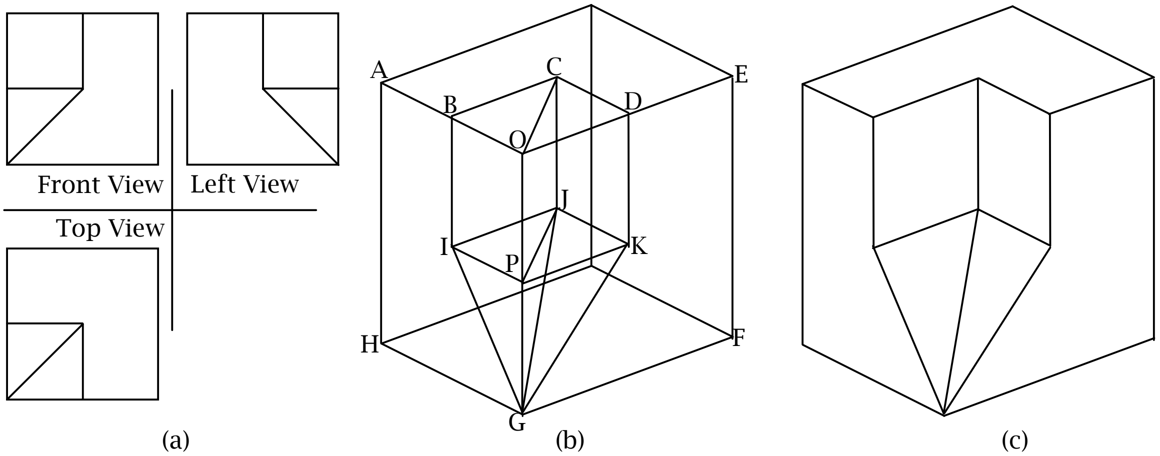


Fig. 7: Example of pseudo recognition algorithm: (a) three orthographic views, (b) incorrect wireframe model, (c) correct solid model.

3.3 True Faces Searching

After the above "Suspect Faces Examining" algorithm, pseudo elements of most models can all be detected. While for some complex models such as Wireframe Model 1 in Fig. 8, we should step into the algorithm “True Faces Searching” for further pseudo recognition. The algorithm is described as follows:

Algorithm 2: ***True Faces Searching***

Input: Wireframe with no intersecting faces, Mini Face Loop List, *Flag*(l_i) as the state of each face loop l_i

Output: Manifold Wireframe

1. For (each mini face loop l_i in *Mini Face Loops List*)
2. If (*Flag*(l_i)≠*UNKNOWN*), continue;
3. Examine the *Projection Completeness(Rule 1)*, supposing l_i is deleted from the wireframe
4. Let *Projection* = the result of the examination;
5. If (*Projection=TRUE*), call *Process True Face* (l_i);
6. Else move l_i to the end of *Mini Face Loops List;*

After Algorithm 1 and Algorithm 2, all the pseudo elements of the wireframe models in our database can be recognized and deleted. As our algorithm is based on the existence of non-manifold edges to recognize pseudo elements, the searching space is quite small and therefore the efficiency is sufficient high.

4 THE ANALYSIS OF ALGORITHM COMPLEXITY

Our paper proposes a linear algorithm for pseudo recognition which can be divided into two stages: (1) Suspect Faces Examining; (2) True Faces Searching. We analyze their complexity and compare the algorithm with the previous decision-chaining method by Kuo [9, 10].

According to Algorithm 1 (Suspect Faces Examining) and Algorithm 2 (True Faces Searching), we can get the time complexity of each step. Given L as the number of mini face loops in the wireframe model, S as the number of suspect face loops, P as the number of pseudo face loops, T as the number of true face loops, E_i as the loop edges number of a certain face loop i, and E as the number of total 3D edges of the wireframe model.

- **Check Suspect Faces**
  In Algorithm 1, we should check each mini face loop and the number of adjacent faces of its boundary edges to determine whether it is a suspect face or not. The time complexity is:

$$\sum_{i=1}^{L} E_i$$

- **Check Projection Completeness**
  For each suspect face loop, we examine the projection lines of its boundary edges to check it is a true face or not. The total time complexity is:

$$\sum_{i=1}^{S} E_i$$

- **Process Pseudo Faces**
  This subroutine process pseudo faces after examination of projection completeness. The adjacent faces number of all loop edges of the input pseudo face should be examined to search other pseudo faces based on topological adjacency. The time complexity is:

$$\sum_{i=1}^{P} E_i$$

- **Process True Faces**
  Similar to the subroutine of process pseudo faces, this subroutine should also examine all loop edges of the input true face based on topological adjacency to search other true faces. The time complexity is:

$$\sum_{i=1}^{T} E_i$$

- **Search True Faces**
  In Algorithm 2, we should examine each non-determined face loop and then examine its projection completeness to check whether it is a true face or not. The time complexity is:

$$\sum_{i=1}^{L-P-T} E_i$$

As the loop edges of suspect faces are all non-manifold edges and the adjacent faces number of manifold edges is 2, given D_i as the adjacent faces number of a certain non-manifold edge *i*, and D as the maximum value of D_i, N as the number of non-manifold edges, we can get the total time complexity of our algorithm as:

$$\begin{aligned}
&\sum_{i=1}^{L} E_i + \sum_{i=1}^{S} E_i + \sum_{i=1}^{P} E_i + \sum_{i=1}^{T} E_i + \sum_{i=1}^{L-P-T} E_i \\
&= 2\sum_{i=1}^{L} E_i + \sum_{i=1}^{S} E_i = 2\sum_{i=1}^{E} D_i + \sum_{i=1}^{N} D_i \\
&= 2(\sum_{i=1}^{N} D_i + \sum_{i=1}^{E-N} 2) + \sum_{i=1}^{N} D_i \\
&= 3\sum_{i=1}^{N} D_i + 4(E-N) < 3\sum_{i=1}^{N} D + 4(E-N) \\
&= 3DN + 4(E-N) = O(DN+E)
\end{aligned}$$

As the value of D is a constant number, out algorithm for pseudo recognition can achieve a linear time complexity. Compared our algorithm with the decision-chaining method by Kuo [9, 10], whose time complexity is analyzed in the work [9], we can obtain the conclusion in the table below:

Algorithm	*Complexity*	*Time Complexity*	*Explanation*
Probable Pseudo Face	$3\sum_{i=1}^{N} D_i + 4(E-N)$	$O(DN+E)$	E is the number of edges
			N is the number of non-manifold edges
Decision-Chaining Method	$\prod_{i=1}^{N} \frac{D_i(D_i-1)}{2}$	$O((\frac{D^2}{2})^N)$	D is a constant number and D>2
			D_i is the adjacent faces number of a certain non-manifold edge *i*

Tab. 1: The comparison of complexity of our algorithm and the Decision-Chaining method.

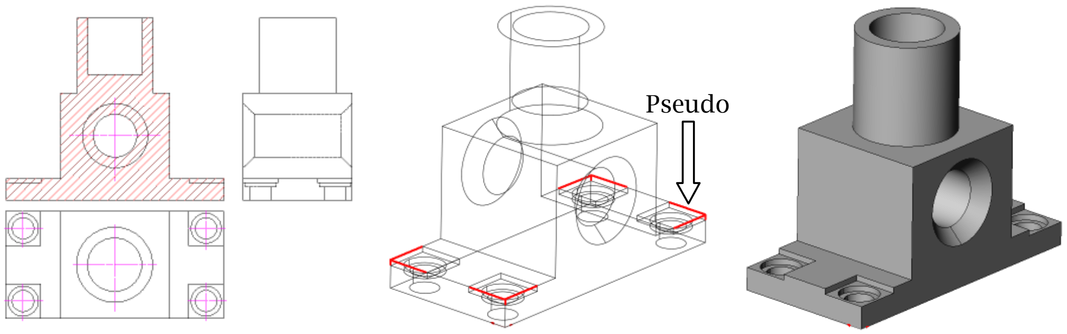
As the Decision-Chaining Method [9] has to check each pair of adjacent faces of all non-manifold faces, the complexity is rather high. In contrast, our algorithm can achieve a linear time complexity. Therefore, we can obtain the correctness and efficiency in 3D reconstruction at the same time.

5 RESULTS

The linear algorithm for pseudo recognition in our paper are implemented in the system of GEMS, which is a reconstruction system developed by our laboratory. We test all practical engineering drawings in our database (more than 100 models), and obtain correct results. The algorithm is executed on the environment of windows 7 OS, 2.80GHz CPU, 3.24GB Memory. The table and figures below list the information of three models with running time, pseudo elements information and so on.

Model	*Running Time(ms)*	*Vertices*	*Edges*	*Mini Face Loops*	*Pseudo Vertices*	*Pseudo Edges*	*Pseudo Faces*
1	172	69	110	56	4	12	28
2	390	112	169	62	0	13	31
3	125	66	102	37	0	4	21

Tab. 2: Results of testing models.



(a) (b) (c)

Fig. 8: Wireframe model 1: (a) three orthographic views; (b) wireframe model; (c) solid model.

As the redundant combination of projection lines in engineering view, this model has four obvious pseudo vertices, as shown in Fig. 8(b). The adjacent edges of these pseudo vertices are not non-manifold edges as the number of their adjacent faces is exactly 2. Thus there is no probable pseudo face in the wireframe and it will not step into the Algorithm 1 (Suspect Faces Examining). But the existing pseudo edges will result in its adjacent edges as non-manifold edges, and two of the non-manifold edges' adjacent faces can be detected as true faces by examining projection completeness, so these pseudo elements can be detected in Algorithm 2 (True Faces Searching).

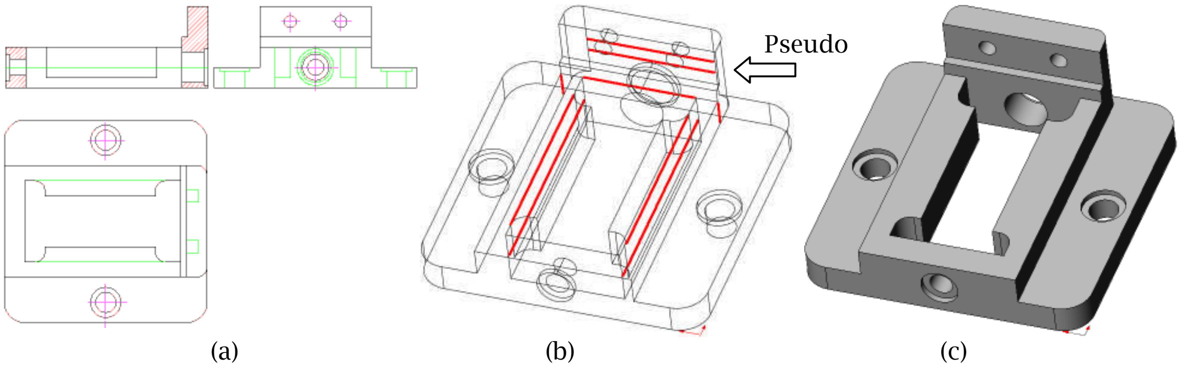


Fig. 9: Wireframe model 2: (a) three orthographic views; (b) wireframe model; (c) solid model.

In Fig. 9(b), the wireframe model has many pseudo edges and pseudo faces which are shown as red lines, because of redundant combination of projection lines in three views. These pseudo elements will result in many probable pseudo faces and non-manifold edges which can all be detected by Algorithm 1 (Suspect Faces Examining) and Algorithm 2 (True Faces Searching). Thus we can obtain the correct solid model shown in Fig. 9(c).

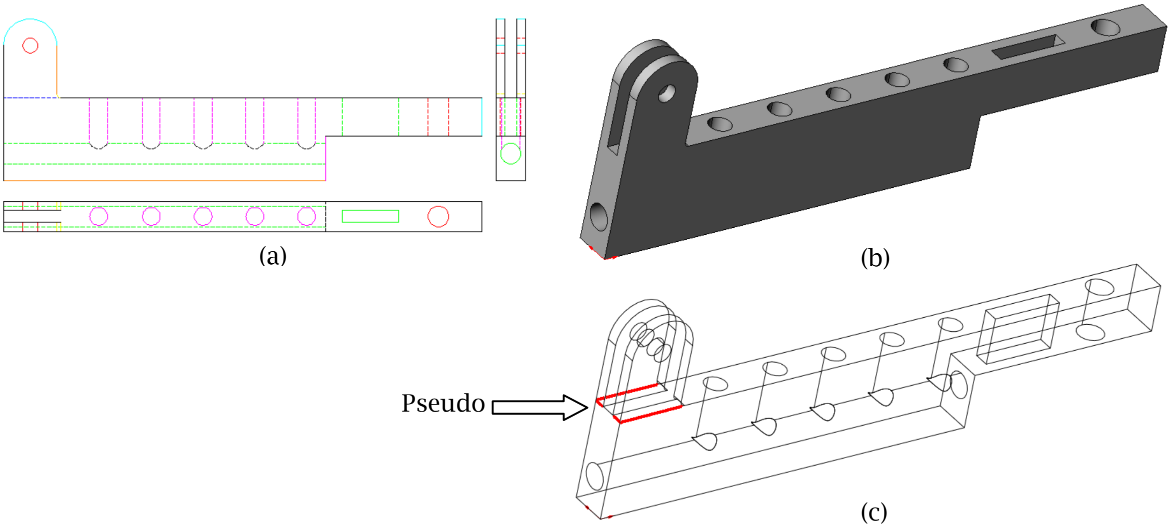


Fig. 10: Wireframe model 3: (a) three orthographic views; (b) solid model; (c) wireframe model.

In Fig. 10(c), the wireframe model has 4 obvious pseudo edges shown as red lines, which separate the original faces of the wireframes and result in 21 pseudo faces. After the construction of mini face loops, the linear algorithm for pseudo recognition can recognize all the pseudo elements in Algorithm 1 (Suspect Faces Examining), and obtain the correct solid model shown in Fig. 10(b).

6 CONCLUSION

Pseudo recognition is essential in 3D reconstruction. However, the previous methods cannot achieve efficiency and correctness at the same time. In this paper, we have presented a linear algorithm to robustly solve the problem of pseudo elements for wireframe models with planes and basic conicoid surfaces. Instead of searching all face loops, we focus on reducing the searching space of suspect face loops. Our algorithm is significantly efficient compared with previous methods, as well as the correctness of our method can be verified by practical engineering wireframe models. The time complexity of the algorithm is linear which is analyzed in the paper and it is proportional to the number of edges in the wireframe model.

7 ACKNOWLEDGEMENTS

This work was supported by the 973 Program of China (Grant No. 2010CB328001) and the National Nature Science Foundation of China (Grant No. 60903106 and No. 61035002).

REFERENCES

[1] Idesawa, M.: A System to Generate a Solid Figure from Three View, Bulletin of the JSME, 16(92), 1973, 216–225.

[2] Markowsky, G.; Wesley, M. A.: Fleshing out Wireframes, IBM Journal of Research and Development, 24(5), 1980, 582–597. doi:10.1147/rd.245.0582

[3] Wesley, M. A.; Markowsky, G.: Fleshing Out Projection, IBM Journal of Research and Development, 25(6), 1981, 934–953. doi:10.1147/rd.256.0934

[4] Gu, K.; Tang,Z.; Sun, J.: Reconstruction of 3D solid objects from orthographic projections, Computer Graphics Forum, 5(4), 1986, 317–324. doi:10.1111/j.1467-8659.1986.tb00318.x

[5] Lequette, R.: Automatic construction of curvilinear solids from wireframe views, Computer-Aided Design, 20(4), 1988, 171–179. doi:10.1016/0010-4485(88)90273-4

[6] Gujar, U. G.; Nagendra, I. V.: Construction of 3D Solid Objects from Orthographic Views, Computers and Graphics, 13(4), 1989, 505–521. doi:10.1016/0097-8493(89)90012-5

[7] Yan, Q. W.; Chen, C. L.; Tang, Z. S.: Efficient Algorithm for the Reconstruction of 3D Objects from Orthographic Projections, Computer-Aided Design, 26(9), 1994, 699–717. doi:10.1016/0010-4485(94)90020-5

[8] You, C. F.; Yang, S. S.: Reconstruction of Curvilinear Manifold Objects from Orthographic Views, Computers and Graphics, 20(2), 1996, 275–293. doi:10.1016/0097-8493(95)00130-1

[9] Kuo, M. H.: Reconstruction of quadric surface solids from three-view engineering drawings, Computer-Aided Design, 30(7), 1998, 517–527. doi:10.1016/S0010-4485(98)00006-2

[10] Kuo, M. H.: Automatic extraction of quadric surface from wire-frame models, Computers and Graphics, 25(1), 2001, 109–119. doi:10.1016/S0097-8493(00)00111-4

[11] Liu, S. X.; Hu, S. M.; Sun, J. G.: Two Accelerating Techniques for 3D Reconstruction, Journal of Computer Science and Technology, 17(3), 2002, 362–368. doi:10.1007/BF02947315

[12] Ye, N.; Zou, B.; Wang, T.: Pseudo Element Removing in 3D Reconstruction form 2D Orthographic Views, Journal of Computer-Aided Design & Computer Graphics, 18(11), 2006, 1730-1734.