Pergamon

Computers Math. Applic. Vol. 36, No. 3, pp. 11–20, 1998

Printed in Great Britain
0898-1221/98 $19.00 + 0.00

PII: S0898-1221(98)00125-4

Maximal Empty Cuboids Among Points and Blocks

S. C. NANDY AND B. B. BHATTACHARYA
Indian Statistical Institute
Calcutta 700 035, India
<nandysc><bhargab>@isical.ac.in

(Received October 1996; revised and accepted March 1998)

Abstract—Given a three-dimensional *box* containing n points, we consider the problem of identifying all *Maximal Empty Isothetic Cuboids (MEC)*, i.e., all 3D empty parallelepiped bounded by six isothetic rectangular faces. It is shown that the total number of MECs is bounded by $O(n^3)$ in the worst case. An output-sensitive algorithm, based on plane-sweep paradigm, is proposed for generating all the MECs present on the floor. The algorithm runs in $O(C + n^2 \log n)$ time in the worst case, and requires $O(n)$ space, where C is the number of MECs present inside the box. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords—Computational geometry, Algorithm, Complexity.

1. INTRODUCTION

Recognition of all Maximal-Empty-Rectangles (MERs) is a well-studied problem in computational geometry and was introduced in [1]. Given a set of n points on a two-dimensional rectangular floor, the objective is to locate all isothetic MERs and/or the largest area MER. The time complexity of the algorithm proposed in [1] is $O(\min(n^2, R \log n))$, where R is the number of existing MERs which could be $O(n^2)$ in the worst case. This is further improved in [2] to $O(R + n \log n)$. The algorithms in [2,3] locate the largest empty rectangle among a point set without inspecting all MERs, in time $O(n \log^3 n)$ and $O(n \log^2 n)$, respectively. The MER problem is generalized for a set of isothetic solid obstacles [5] and also for nonisothetic obstacles [6].

In this paper, we extend the MER problem into three dimensions. Given a set of n points inside a box, the objective is to report all maximal empty isothetic cuboids (MECs), i.e., all empty isothetic parallelepiped. Using the classical plane-sweep paradigm, the proposed algorithm reports all the MECs in $O(C+n^2 \log n)$ time and $O(n)$ space, where C is the total number of MECs present inside the box. It is also shown that C is $O(n^3)$ in the worst case. The largest (volume) MEC can be reported in $O(n^{8/3} \log^3 n)$ time without generating all MECs. The problem of locating empty cuboids finds its application in VLSI design automation, 3D graphics, operations research, database, etc.

2. FORMULATION OF THE PROBLEM

Consider a box, i.e., a three-dimensional region bounded by six isothetic rectangular faces. The top and bottom faces of the box will be referred to as the *roof* and *floor* of the box, respectively. The south-west corner of the floor is treated as the origin, and the edges of the box incident on

Typeset by $\mathcal{A}_\mathcal{M}\mathcal{S}$-TEX

the origin are assumed to be the axes of reference. Let $P = \{p_i(x_i, y_i, z_i),\ i = 1, 2, \ldots n\}$ denote the set of points placed randomly inside the box.

A *cuboid* is an isothetic parallelepiped, i.e., a three-dimensional region bounded by six axis-parallel rectangular faces. The three pairs of opposite faces of a cuboid are called (north, south), (east, west), and (top, bottom).

DEFINITION. *A cuboid λ is said to be a maximal empty cuboid (MEC) if*

(i) *λ does not contain any point in P, and*

(ii) *there exists no other empty cuboid that contains λ.*

An MEC is represented by a six-tuple $[(n, s), (e, w), (t, b)]$, where n, s, e, w, t, b are the points or the faces of the box through which the north, south, east, west, top, and bottom faces of the cuboid pass, respectively. If the plane corresponds to a face of the bounding box, its parameter is dropped and the face specification of the concerned face is attached.

Let p be an arbitrary point in P. The plane that passes through the point p and is parallel to the X-Y plane, is said to be a horizontal plane and is denoted by $H(p)$.

In order to visualize how the MECs are formed, let us consider projections of all the points on the floor. Let P^0 denote the set of projections of all the points in P, where $p_i^0 \in P^0$ is the projection of $p_i \in P$.

DEFINITION. *A rectangle $R[(n, s), (e, w)]$ on the floor is said to be valid if each of the four sides of R either coincides with a bounding side of the floor, or touches a member of P^0. The degree $\delta(R)$ of a valid rectangle is the number of sides of the rectangle that pass through some point(s) of P^0.*

Now consider a valid rectangle $R[(n, s), (e, w)]$. Several MECs may now exist with the rectangle R as their horizontal cross-sections, as stated in the following cases.

Case 1. $\delta(R) = 0$, i.e., R becomes identical to the floor of the box. In this case, there exist $(n + 1)$ MECs; they are obtained by slicing the box horizontally through the plane $H(p_i)$ for each point $p_i \in P$.

Case 2. $\delta(R) \neq 0$. In this case, the following subcases may arise.

Case 2.1. The rectangle R is empty. Correspondingly, there exists exactly one MEC whose top (bottom) face touches the roof (floor) of the box, respectively.

Case 2.2. The rectangle R is nonempty. Let $\eta \subseteq P^0$ be the set of projected points that fall inside the rectangle. Let the z coordinates of the points in P through whose projections the north, south, east, and west sides of R pass, be z_n, z_s, z_e, and z_w, respectively. If any side of R coincides with the boundary of the floor, the corresponding z value is undefined.

Let $z_{\min} = \min\{z_n, z_s, z_e, z_w\}$ and $z_{\max} = \max\{z_n, z_s, z_e, z_w\}$. The undefined z values, if any, do not contribute to the $z_{\min}$ or $z_{\max}$. Here again two subcases may arise.

Case 2.2.1. There does not exist a point $p_\alpha \in \eta$ such that $z_{\min} < z_\alpha < z_{\max}$. In this case, the aforesaid rectangle corresponds to exactly one MEC. To decide the pair of points through which the top and bottom faces of the MEC pass, let η_1 and η_2 be the two disjoint subsets of η ($\eta_1 \bigcup \eta_2 = \eta$) such that $z_\alpha \leq z_{\min}$, $\forall\ p_\alpha \in \eta_1$ and $z_\alpha \geq z_{\max}$, $\forall\ p_\alpha \in \eta_2$. Let p_b be the point in η_1 such that $z_b \geq z_\alpha$, $\forall\ p_\alpha \in \eta_1$. Similarly, let p_t be the point in η_2 such that $z_t \leq z_\alpha$, $\forall\ p_\alpha \in \eta_2$. Then the top (bottom) face will pass through p_t (p_b). In particular, if $\eta_1 = \phi$, the bottom face of the MEC will coincide with the floor of the box. If $\eta_2 = \phi$, the top face of the MEC will coincide with the roof of the box.

Case 2.2.2. There exists at least one point $p_\alpha \in \eta$ such that $z_{\min} < z_\alpha < z_{\max}$. In this case, there does not exist any MEC whose horizontal cross section is R.

It is easy to verify that if $\delta(R) = 1$, then $z_{\min} = z_{\max}$, and hence, the Case 2.2.2 will not arise. Thus, a corresponding MEC will always exist. From the above discussions, it is now easy to prove the following result.

THEOREM 1. *Every MEC whose horizontal cross-section R is not the entire floor, must satisfy either Cases 2.1 or 2.2.1, and every valid rectangle (other than the entire floor) with the projected points that satisfies either Cases 2.1 or 2.2.1, must form exactly one MEC.* ∎

EXAMPLE. Consider a scenario as demonstrated in Figure 1. All the points inside the box are projected on the floor of the box. The position of a point as shown in the figure reflects its x- and y-coordinates, and the label attached to it denotes the z-coordinate. Now, consider the following cases:

(i) the rectangle described by $R_1[(p_7, p_2), (p_9, p_4)]$ cannot form an MEC;
(ii) the rectangle described by $R_2[(p_7, p_2), (p_8, p_4)]$ forms an MEC whose top coincides with the roof of the box, and the bottom face passes through the point p_1;
(iii) the rectangle described by $R_3[(p_7, p_2), (p_8, p_5)]$ forms an MEC whose top and bottom faces pass through the points p_4 and p_1, respectively;
(iv) the rectangle described by $R_4[(p_7, p_1), (p_8, p_4)]$ forms an MEC whose top coincides with the roof, and the bottom coincides with the floor of the box;
(v) the rectangle described by $R_5[(p_8, p_1), (p_9, p_5)]$ forms an MEC whose top face passes through p_4, and bottom face coincides with the floor of the box.

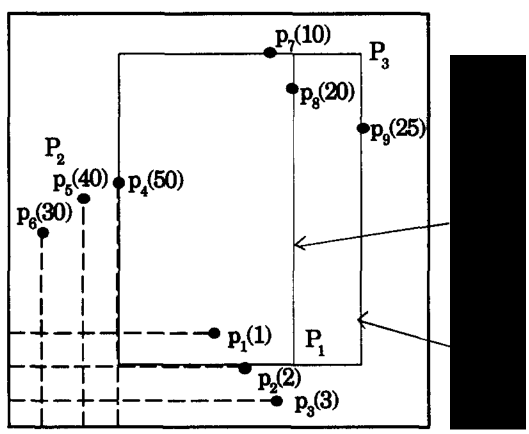


Figure 1. Horizontal cross-sections of MECs.

The above discussions lead to the fact that a loose upper bound on the number of MECs is $O(n^4)$, where n is the number of point obstacles inside the box. We will now show that the bound can be improved to $O(n^3)$, and present an example where the bound is achieved.

Bounds on the Number of MECs

In the earlier section, we have got a trivial $O(n^4)$ upper bound on the number of MECs. To make the bound tighter and to formulate the algorithm for generating all the MECs, let us introduce the following classification of MECs.

DEFINITION. *An MEC is said to be of type-A if its top face touches the roof of the bounding box. An MEC which is not of type-A is called a type-B MEC.*

Let us now concentrate on the type-A MECs. Let $p_i(x_i, y_i, z_i) \in P$ and $P^* \subset P$ be a set of points such that $z^* \geq z_i$, for all $p^*(x^*, y^*, z^*) \in P^*$. Project the members of P^* along with p_i on $H(p_i)$. Now each MER on $H(p_i)$ will serve as the horizontal cross-section of a unique MEC. The top faces of all these MECs touch the roof of the box, and the bottom face of each of them will be determined by sweeping the corresponding MER downwards until it hits a point or the floor of the box for the first time.

It is easy to observe that the number of MERs touching the point p_i on $H(p_i)$ is $O(m)$, where $m = |P^*|$. This leads to the following theorem.

THEOREM 2. *The total number of type-A MECs is $O(n^2)$ in the worst case.*

COROLLARY 2.1. *The number of MECs whose top (bottom) faces coincide with the roof (floor) of the box is $O(n^2)$.*

THEOREM 3. *The total number of type-B MECs is $O(n^3)$ in the worst case.*

PROOF. Consider each point $p_i \in P$ and the set of MECs whose top face pass through p_i. Consider $H(p_i)$ as the roof and find all the type-A MECs whose top face pass through $H(p_i)$, with the points lying below p_i. By Theorem 2, there are $O(n^2)$ such MECs in the worst case. Among these set of MECs, the desired set of type-B MECs are those whose top faces contain p_i. Thus, in the worst case, the number of type-B MECs each of whose top face passes through p_i is $O(n^2)$. Accumulating the set of type-B MECs for all the points in P, we get $O(n^3)$ type-B MECs in the worst case. ∎

Figure 1 shows an instance where the total number of MECs is $O(n^3)$. Here, the given set P of n points is divided into three subsets, say P_1, P_2, and P_3, as shown in the figure, such that each subset contains at least $\lfloor n/3 \rfloor$ points. Draw horizontal (vertical) lines from each member of the subset P_1 (P_2). These lines intersect at $O(n^2)$ points. Let this set of intersection points be denoted as U. Two consecutive points in the subset P_3 create a set of $O(n)$ corner points. Let this set be denoted as V. One can now construct $O(n^3)$ rectangles whose south-west (north-east) corner coincides with a member of $U(V)$. By properly assigning the z coordinates of the points in P, each of these rectangles can be made to correspond to the horizontal cross-section of an MEC as shown in Figure 1.

3. LOCATION OF TYPE-A MECS

To locate the set of type-A MECs with the point $p_i (\in P)$ touching the bottom face, consider the horizontal plane $H(p_i)$ and project all the points that lie above $H(p_i)$ (excluding p_i), on $H(p_i)$. Let us denote the projection of the point p_j by p_j^*. Now, the following observation is immediate.

LEMMA 1. *Let R denote an arbitrary maximal-empty-rectangle (MER) on $H(p_i)$ considering all the projected points as obstacles (excepting p_i itself). Then there exists a type-A MEC whose horizontal cross-section matches with R and whose bottom face passes through p_i iff R encloses the point p_i.*

In order to obtain such type-A MECs, we draw two isothetic orthogonal lines through p_i to partition $H(p_i)$ into four quadrants; let them meet the four boundaries of $H(p_i)$ at p_n^*, p_s^*, p_e^*, and p_w^*, respectively as shown in Figure 2. Let Q_θ be the set of points in the θ^{th} quadrant, $\theta = 1, 2, 3, 4$. In each quadrant θ, we define a set of points $\text{STAIR}_\theta(p_i)$ around the point p_i. We now give the definition for $\theta = 1$ only.

DEFINITION. *The set of points $\text{STAIR}_1(p_i) \subseteq Q_1$ is said to form a maximal-closest-stair in the first quadrant around the point p_i, if*

(i) *the points on $\text{STAIR}_1(p_i) = \{p_n^*, p_1^*, p_2^*, \dots, p_k^*, p_e^*\}$ are linearly ordered with respect to their increasing x coordinates;*
(ii) *for any two consecutive points p_j^* and p_{j+1}^* on $\text{STAIR}_1(p_i)$, $y_j^* > y_{j+1}^*$;*
(iii) *the largest area isothetic polygon bounded by the edges (p_e^*, p_i), (p_i, p_n^*), and the staircase path through the points in $\text{STAIR}_1(p_i)$ is empty; and*
(iv) *no other point from the set Q_1 can be added to $\text{STAIR}_1(p_i)$, satisfying Conditions (i)–(iii).*

The maximal-closest-stairs in the other three quadrants, are defined similarly. Needless to say, the maximal-closest-stairs in all the four quadrants are unique. An example with four stairs is shown in Figure 2. The concatenation of these four staircase paths creates an isothetic Orthoconvex Polygon (OP) [4] with a nonempty kernel.

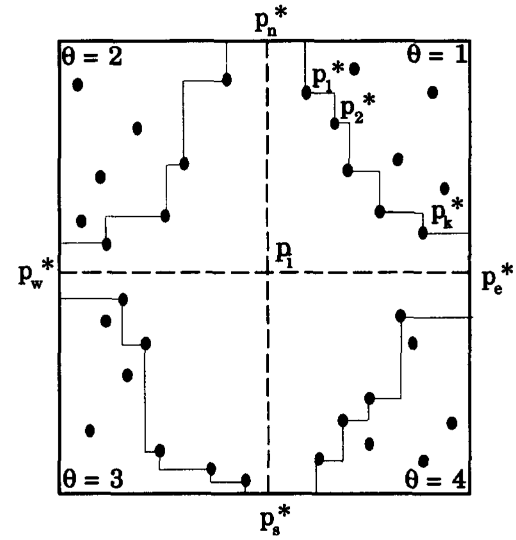


Figure 2. Maximal-closest-stairs around p_i.

THEOREM 4. *Every MER inside the OP corresponds to the horizontal cross-section of a unique type-A MEC whose bottom face touches the point p_i, and conversely, the horizontal cross-section of every type-A MEC whose bottom face touches p_i, corresponds to a unique MER inside the OP.*

PROOF. Follows from Lemma 1. ∎

Thus, the problem of locating MECs now boils down to recognizing all MERs within an orthoconvex polygon.

It is easy to observe that the north (respectively, south) side of every MER within an OP touches a point of $\{\mathrm{STAIR}_1(p_i) \bigcup \mathrm{STAIR}_2(p_i)\}$ (respectively, $\{\mathrm{STAIR}_3(p_i) \bigcup \mathrm{STAIR}_4(p_i)\}$). In our algorithm, we shall consider each point of $\{\mathrm{STAIR}_1(p_i) \bigcup \mathrm{STAIR}_2(p_i)\}$ individually, and report all the MERs touching it.

Considering a MER whose north side touches $p_j(x_j, y_j) \in \mathrm{STAIR}_1(p_i)$. It is now easy to recognize the *feasible set of points* in $\{\mathrm{STAIR}_3(p_i) \bigcup \mathrm{STAIR}_4(p_i)\}$ that can appear on its south side. In this context, let us define the nearest-neighbor of a point $p_j \in P$, denoted by $\mathrm{NN}(p_j)$, in its vertically opposite quadrant as follows.

DEFINITION. *Let p_j be any point in $STAIR_\theta(p_i)$, $\theta = 1,2,3,4$, such that $p_j \neq \{p_n^*, p_s^*, p_e^*, p_w^*\}$.*

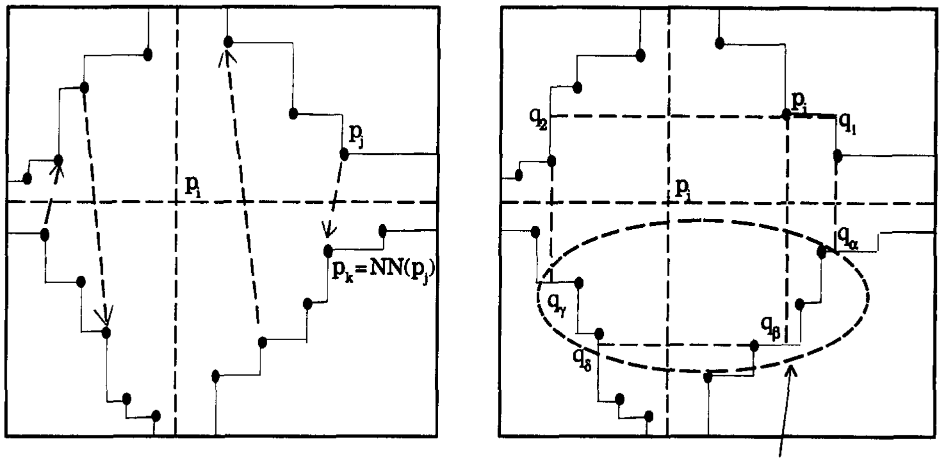
If $p_j \in \mathrm{STAIR}_1(p_i)$ (respectively, $\mathrm{STAIR}_4(p_i)$), its nearest-neighbor $\mathrm{NN}(p_j) = p_k(x_k, y_k)$, such that, $p_k \in \mathrm{STAIR}_4(p_i)$ (respectively, $\mathrm{STAIR}_1(p_i)$) and $x_k = \min\{x_\alpha \mid (x_j - x_\alpha) > 0\}$.

Similarly, if $p_j \in \mathrm{STAIR}_2(p_i)$ (respectively, $\mathrm{STAIR}_3(p_i)$), its nearest-neighbor $\mathrm{NN}(p_j) = p_k(x_k, y_k)$, such that, $p_k \in \mathrm{STAIR}_3(p_i)$ (respectively, $\mathrm{STAIR}_2(p_i)$) and $x_k = \min\{x_\beta \mid (x_\beta - x_j) > 0\}$.

Also, $\mathrm{NN}(p_n^*) = p_s^*$; $\mathrm{NN}(p_s^*) = p_n^*$, $\mathrm{NN}(p_e^*)$, and $\mathrm{NN}(p_w^*)$ are undefined.

Figure 3a shows an illustration of nearest-neighbors. It is easy to determine the nearest neighbors of all the points in a stair by merging its members with those of the stair in vertically opposite quadrant. The feasible set of points in $\{\mathrm{STAIR}_3(p_i) \bigcup \mathrm{STAIR}_4(p_i)\}$ for a given point p_j in $\mathrm{STAIR}_1(p_i)$ can now be easily obtained.

Let the horizontal line through p_j meet the staircase paths in the first and second quadrants at q_1 and q_2, respectively. The vertical line from q_1 (p_j) meets the staircase path on the fourth



(a) Illustration of nearest neighbors. (b) Feasible set of points.

Figure 3. Illustration of nearest neighbors and their use in determining feasible set of points.

quadrant at $q_\alpha(x_\alpha, y_\alpha)$ and $q_\beta(x_\beta, y_\beta)$, respectively. Now draw a vertical line from q_2 and a horizontal line from q_β, that meet the staircase path in the third quadrant at $q_\gamma(x_\gamma, y_\gamma)$ and $q_\delta(x_\delta, y_\delta)$, respectively. Now all the points $p_k(x_k, y_k) \in \mathrm{STAIR}_3(p_i)$ that satisfy $x_\gamma < x_k < x_\delta$, and also all the points $p_\ell(x_\ell, y_\ell) \in \mathrm{STAIR}_4(p_i)$ that satisfy $x_\beta < x_\ell < x_\alpha$, are the *feasible set* of the point p_j (see Figure 3b). In other words, each of them may lie on the south side to form an MER whose north side touches p_j. The feasible set of points can easily be obtained with the help of nearest-neighbor pointers.

An MER is uniquely defined whenever its two opposite sides are fixed, and hence, the points touching its east and west sides are also uniquely defined, and can easily be recognized.

Similarly, for a point $p_j \in \mathrm{STAIR}_2(p_i)$ defining the north side of an MER, the feasible set of points on which the south side may lie, can easily be identified.

The type-A MECs whose top (bottom) faces touch the roof (floor) of the box are obtained by projecting all the points on the floor, and then locating the MERs with the projected points as obstacles.

The algorithm is based on the plane-sweep paradigm, i.e., the points are processed in decreasing order of their z-coordinates. A dynamically managed AVL-tree T will be used in the algorithm. The tree T contains all the points that are processed so far, ordered with respect to their x coordinates. This tree is required to recognize the maximal-closest-stairs in the four quadrants while processing p_i. The point p_i is inserted in T when its processing is over, and the sweep is advanced to hit the next point. The stairs are stored as doubly-linked lists. In addition, each point of a stair has an associated pointer pointing to its nearest neighbor.

Complexity

For each point p_i, recognition of maximal-closest-stair require $O(m)$ time, where m is the number of points lying above $H(p_i)$. The merging time required for getting the nearest-neighbors is $O(m)$ in the worst case. Reporting of all MECs with p_i on its bottom face, requires $O(A_i + n)$, where A_i is the number of type-A MEC reported at this stage. Last, the time required to insert the point p_i in the data structure is $O(\log n)$. Thus, the overall complexity of recognizing all type-A MECs whose bottom face touch a point is $O(A' + n^2)$ in the worst case, A' is the total number of such type-A MECs. Location of all the type-A MECs whose bottom face touch the floor of the box requires $O(A'' + n \log n)$ time [5], A'' is the number of such type-A MECs. Thus, the overall time complexity of the algorithm is $O(A + n^2)$, where $A(= A' + A'')$. The space complexity is clearly $O(n)$.

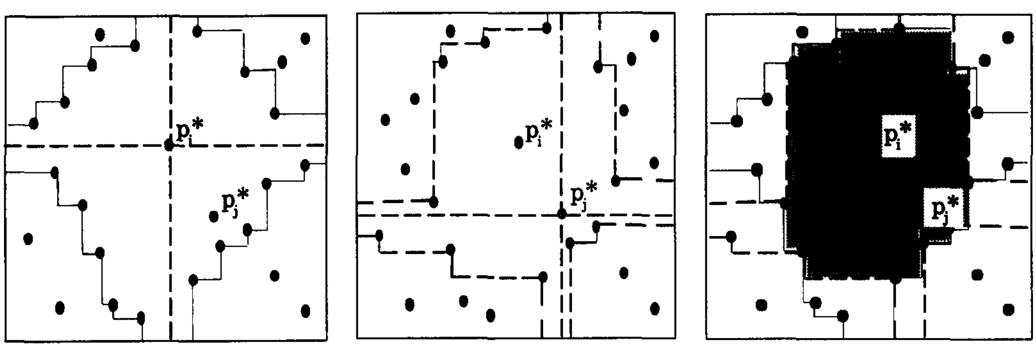
4. LOCATION OF TYPE-B MECS

A type-B MEC is one whose top face touches a point in P. Here, we will assume that no two points in P have same z coordinate. Let $p_i(x_i, y_i, z_i)$ and $p_j(x_j, y_j, z_j)$ be any two points in P such that $z_i > z_j$.

LEMMA 2. *Let λ denote the cuboid whose diagonal is the line segment (p_i, p_j). Then there exists an MEC whose top (bottom) face touches $p_i(p_j)$ if and only if the cuboid λ is empty.*

PROOF. Obvious. ∎

As before, we consider the points in P in the decreasing order of their z coordinates. While processing a point p_i, all the MECs with top face passing through it, are recognized. This processing is referred to as primary processing of p_i. For each point p_i under primary processing, we consider the points below $H(p_i)$ in decreasing order of their z coordinates. For each point p_j $(z_j < z_i)$, all the MECs whose top (bottom) faces pass through p_i (p_j) are recognized. This processing is called secondary processing of p_j with respect to p_i. To do this, we proceed as follows. Let p_i is under primary processing and p_j is under secondary processing.



(a) Staircase around p_i^*. (b) Staircase around p_j^*. (c) Feasible zone around (p_i^*, p_j^*).

Figure 4. Recognition of type-B MECs.

Consider the projections of all points lying within $H(p_i)$ and $H(p_j)$ on $H(p_j)$. Let us denote the projection of a point p_k on $H(p_j)$ by p_k^*. By Lemma 2, if the rectangle drawn on $H(p_j)$ with p_i^* and p_j^* as diagonally opposite corners, contains any (projected) point, no MEC exists with p_i (p_j) on its top (bottom) face, otherwise at least one MEC is possible. In order to get these MECs, draw a pair of isothetic orthogonal lines through p_i^* that partition $H(p_j)$ into four quadrants. The four maximal-closest-stairs around the point p_i^* are obtained among the projected points (excluding p_i^* and p_j^*). The concatenation of these four stairs form an orthoconvex polygon (OP) as before. All the MERs inside the orthoconvex polygon enclose the point p_i^*. Similarly, we construct another orthoconvex polygon around the point p_j^*; all the MERs inside it will enclose the point p_j^*. It is easy to observe that the intersection of these two orthoconvex polygons is also orthoconvex, and all the MERs inside it will enclose both the points p_i^* and p_j^*. We call this region as the *feasible zone* around the pair (p_i^*, p_j^*) (see Figure 4). The following theorem now becomes immediate.

THEOREM 5. *Each MER inside the feasible zone (disregarding the presence of p_i^*, p_j^*) corresponds to the horizontal cross-section of a unique type-B MEC whose top (bottom) face passes through p_i (p_j), and conversely, for every type-B MEC whose top (bottom) face passes through p_i (p_j), its horizontal cross-section matches with an MER inside the feasible zone.* ∎

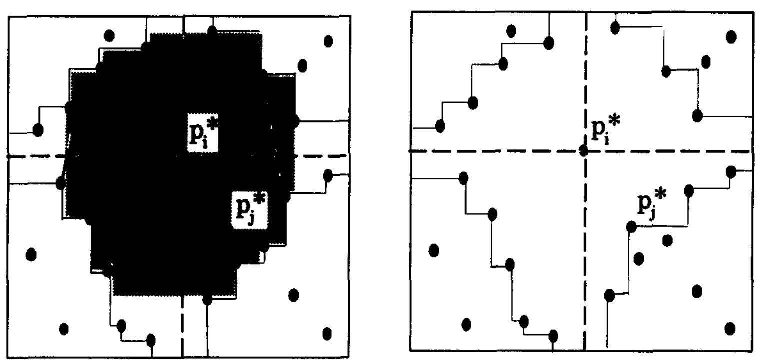
In our algorithm, instead of constructing the orthoconvex polygon around p_j^* each time, we exploit the geometry of the maximal-closest-stairs around p_i^* and the position of p_j^*, to get the feasible zone. The method is illustrated below.

The points on the maximal-closest-stairs in the four quadrants are stored in four AVL-trees ordered with respect to their y-coordinates. Each member (p_k) of a maximal-closest-stair points to its nearest neighbor NN(p_k) in the vertically opposite quadrant to which it belongs.

The secondary processing of p_j involves two situations depending on whether the point p_j^* lies within the OP around p_i^*. If the point p_j^* is outside, no MEC is possible with top (bottom) face touching p_i (p_j); and hence, we ignore it and proceed. Otherwise, the MECs with top (bottom) faces passing through p_i (p_j) are to be reported, which again involves two steps:

(i) finding the feasible zone around the pair (p_i, p_j) and subsequently, the MERs and the corresponding MECs,

(ii) updating STAIR$_\theta$ by inserting p_j without constructing the maximal-closest-stairs around p_i^* afresh for the secondary processing of the subsequent points. The search proceeds with the updated OP for the secondary processing of the next point.



(a) Location of the feasible zone. (b) Updating maximal-stairs around p_i^*.

Figure 5.

Location of the Feasible Zone

Let θ_H, θ_V, and θ_D represent the quadrants that are adjacent to quadrant θ in the horizontal, vertical, and diagonal directions, respectively, where p_j lies in quadrant θ and $y_j \in [y_a, y_b]$. The feasible zone around the pair (p_i^*, p_j^*) is a subregion of OP and can be determined as follows.

(i) Traverse the AVL-tree corresponding to STAIR$_\theta$ in-order from p_a^* towards p_b^* to get a point, say p_c^*, such that its x-coordinate just crosses that of p_j (Figure 5a). Then all the points of STAIR$_\theta$ between p_a^* to p_c^* form the stair of the feasible zone in quadrant θ.

(ii) Let p_d^* denote NN(p_a^*) in the quadrant θ_V. Proceed from p_d^* along the STAIR of θ_V to locate the point p_e^* such that its x-coordinate just crosses that of p_j. Then all the vertices of the STAIR in θ_V between p_d^* and p_e^* form the stair of the feasible zone.

(iii) Let p_f^* and p_g^* be two points in quadrant θ_H such that all the points in the STAIR of θ_H belong to the vertical interval formed by the y-coordinates of p_j^* and p_c^*. These points are obtained by searching the AVL-tree corresponding to the STAIR of θ_H. Consider the next point of p_g^*, i.e., p_h^*. The set of points in this stair between p_f^* and p_h^* forms the boundary of the feasible zone.

(iv) Finally, the portion of the stair between the two points, say p_k^* and p_ℓ^*, in the quadrant θ_D which fall inside the rectangle inscribing the feasible zone, is the desired portion of the stair in that quadrant. The point p_k^* is actually NN(p_h^*), and p_ℓ^* is determined by locating the position of the y-coordinate of p_e^* in the STAIR of the quadrant θ_D.

The illustration of the feasible zone at the time of (secondary) processing p_j is shown in Figure 5a. Now, all MERs inside the feasible zone is to be obtained as in the case of type-A MECs, to get all type-B MECs whose top (bottom) faces touch p_i (p_j).

Updating STAIR$_\theta$

STAIR$_\theta$ can be updated by deleting all the points from p_b^* to p_c^*, both inclusive, and then inserting p_j^* as shown in Figure 5b. The nearest neighbor of the point p_j^* will be the point p_e^* in quadrant θ_V.

Complexity

Let p_i be a point that is under primary processing, and the point p_j is under secondary processing, and p_j^* falls in quadrant θ. The time complexity of the different steps of the secondary processing of p_j are as follows.

- The inclusion of p_j^* inside the current OP can be tested in $O(\log n)$ time from the AVL-tree of STAIR$_\theta$.
- If p_j^* lies inside OP, then the feasible zone around (p_i^*, p_j^*) can be constructed in $(\mu_j + O(\log n))$ time, where μ_j is the number of physical points currently on the four stairs of the feasible zone. Clearly, there exists at least μ_j nearest-neighbor-pointers, and if NN$(p_j^*) = p_k^*$, then there exists a distinct MER where two horizontal sides touch p_j^* and p_k^*. Also, for any physical point p_j, NN$(p_j^*) = p_k^*$ implies NN$(p_k^*) \neq p_j^*$. Therefore, at least μ_j type-B MECs will be reported in this step.
- The reporting of type-B MECs require B_j^i time, where B_j^i $(\geq \mu_j)$ is the number of type-B MECs reported with top (bottom) face passing through p_i (p_j).
- The time required for updating STAIR$_\theta$ requires μ_j time in the worst case.

So, the total time complexity of primary processing of the point p_i is $O(B^i + n \log n)$, where B^i $(= \Sigma_{j=i+1}^n B_j^i)$ is the number of type-B MECs with top face passing through p_i. Thus, aggregating for all the points in P, the time complexity of our algorithm is $O(B + n^2 \log n)$, where B is the total number of type-B MECs reported. Hence, the overall time complexity becomes $O(C + n^2 \log n)$, where $C = A + B$.

If the location of only the largest MEC is of interest, it is not necessary to inspect all the MECs present inside the box. Authors in [7] suggested an online algorithm for finding the largest MER among point obstacles, after any insertion or deletion of a point on the floor. The amortized time complexity of their algorithm is $O(n^{2/3} \log^3 n)$ per update operation. In 3D scenario, each iteration of secondary processing inserts a new point on the sweeping plane, and only the largest MER is to be obtained. Therefore, the time complexity of locating the *largest* MEC can further be reduced to $O(n^{8/3} \log^3 n)$, keeping the space complexity unaltered.

5. MECS AMONG NONOVERLAPPING BLOCKS

We now discuss the method of recognizing all the MECs among a set of nonoverlapping isothetic solid cuboids (blocks) arbitrarily placed inside the box. The algorithm is similar to the earlier one; the only difference lies in the fact that the shape of the isothetic polygon defining the feasible zone around a (a pair of) solid block(s), is not orthoconvex. However, it can be easily constructed as observed below.

Let $S = \{S_1, S_2, \dots, S_n\}$ be the set of solid blocks placed inside the box which will be treated as obstacles. The objective is to locate all the MECs inside the box. Let us consider a block S_i, and the horizontal cross-sections of all the solid blocks that are strictly above S_i, and project them on the horizontal plane $H(S_i)$. If the union of these projected rectangles completely covers the top face of S_i, then there exists no MEC whose top face touches the roof, and bottom face touches the top face of S_i; otherwise at least one such MEC exists. We now discuss the method of recognizing these MECs.

Draw four isothetic lines on $H(S_i)$ along the boundaries of the top face of S_i. Let these lines intersect at four points a, b, c, and d, and divide $H(S_i)$ into nine parts $H_1, H_2, \dots, H_9$, shown by dotted lines in Figure 6. Consider an intersection point, say a, and find the maximal-closest-stairs

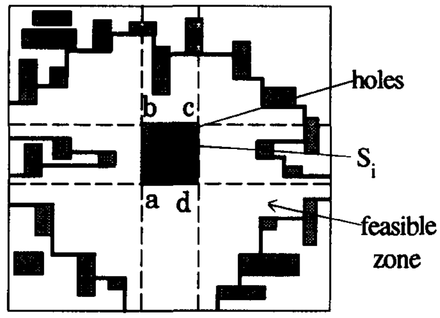


Figure 6. Location of type-A MECs among solid 3D blocks.

in the three quadrants around it (see Figure 6). The quadrant containing S_i is ignored. Similarly for all other intersection points, the maximal-closest-stairs are determined. The isothetic closed polygon bounded by these maximal-closest-stairs form the feasible zone for locating the type-A MECs, whose bottom face pass through the top of S_i. The projections that fall on the rectangle describing the top of S_i, create the holes inside the polygon. Thus, the problem reduces to location of all the MERs inside the feasible zone which is an isothetic polygon with a number of holes ($IPH(S_i)$). The time complexity of locating all the MERs inside the feasible zone is $O(C_i + n_i)$, where C_i is the number of MERs reported and n_i is the number of solid blocks whose top faces lie above $H(S_i)$.

To recognize the type-B MECs with top (bottom) face passing through the bottom (top) face of S_i (S_j), one has to consider the projections of all the blocks that lie completely or partially between the horizontal planes $H(S_i)$ and $H(S_j)$. The feasible zone in this case will be the isothetic polygon formed by the intersection of $IPH(S_i)$ and $IPH(S_j)$, and can be obtained using the neighbor pointers as before. Each MER in the feasible zone return an MEC whose top (bottom) face touches the bottom (top) of S_i (S_j). The time and space complexities remain invariant.

6. CONCLUSION

In this paper, we have presented an output-sensitive algorithm of time complexity $O(C + n^2 \log n)$ and space complexity $O(n)$, for locating all maximal empty cuboids (MECs) inside a box containing n point obstacles, where the number (C) of reported MECs, may be $O(n^3)$ in the worst case. A minor modification in the data structure can also handle the same problem among a set of isothetic 3D polyhedral obstacles, retaining the worst case time and space complexities invariant. Location of MECs finds its applications in 3D graphics, operations research, database, and 3D VLSI chips.

REFERENCES

1. A. Naamad, D.T. Lee and W.L. Hsu, On the maximum empty rectangle problem, *Discrete Applied Mathematics* **8**, 267–277, (1984).
2. B. Chazelle, R.L. Drysdale and D.T. Lee, Computing the largest empty rectangle, *SIAM Journal on Computing* **15**, 300–315, (1986).
3. A. Aggarwal and S. Suri, Fast algorithm for computing the largest empty rectangle, In *Proc. Third ACM Symposium on Computational Geometry*, pp. 278–290, (1987).
4. G.J.E. Rawlins and D. Wood, Orthoconvexity and its generalizations, In *Computational Morphology*, (Edited by G. Toussaint), pp. 137–152, Elsevier Science B. V., (1988).
5. S.C. Nandy, B.B. Bhattacharya and S. Ray, Efficient algorithms for identifying all maximal isothetic empty rectangles in VLSI layout design, In *Proc. FSTTCS-10*, Volume 472, pp. 255–269, Lecture Notes in Computer Science, Springer-Verlag, Berlin, (1990).
6. S.C. Nandy, A. Sinha and B.B. Bhattacharya, Location of the largest empty rectangle among arbitrary obstacles, In *Proc. FSTTCS-14*, Volume 880, pp. 159–170, Lecture Notes in Computer Science, Springer-Verlag, Berlin, (1994).
7. D. Dobkkin and S. Suri, Dynamically computing the maxima of decomposable functions with applications, In *Proc. 29^th FOCS*, pp. 488–493, (1989).
8. M. Orlowski, A new algorithm for the largest empty rectangle problem, *Algorithmica* **5**, 65–73, (1990).