MAX-PLANCK-INSTITUT FÜR INFORMATIK

Maintaining the Visibility Map of Spheres while Moving the Viewpoint on a Circle at Infinity

Hans-Peter Lenhof Michiel Smid

MPI–I–92–102 January 1992

mpi INFORMATIK

Im Stadtwald
66123 Saarbrücken
Germany

Maintaining the Visibility Map of Spheres while Moving the Viewpoint on a Circle at Infinity

Hans-Peter Lenhof Michiel Smid

MPI–I–92–102 January 1992

"Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministers für Forschung und Technologie (Betreuungskennzeichen ITS 9103) gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor."

Maintaining the Visibility Map of Spheres while Moving the Viewpoint on a Circle at Infinity

Hans-Peter Lenhof Michiel Smid*

Max-Planck-Institut für Informatik

W-6600 Saarbrücken, Germany

January 14, 1992

Abstract

We investigate 3D visibility problems for scenes that consist of n non-intersecting spheres. The viewing point v moves on a flightpath that is part of a "circle at infinity" given by a plane P and a range of angles $\{\alpha(t)|t \in [0:1]\} \subset [0:2\pi]$. At "time" t, the lines of sight are parallel to the ray $r(t)$ in the plane P, which starts in the origin of P and represents the angle $\alpha(t)$ (orthographic views of the scene). We describe algorithms that compute the visibility graph at the start of the flight, all time parameters t at which the topology of the scene changes, and the corresponding topology changes. We present an algorithm with running time $O((n+k+p)\log n)$, where n is the number of spheres in the scene; p is the number of transparent topology changes (the number of different scene topologies visible along the flightpath, assuming that all spheres are transparent); and k denotes the number of vertices (conflicts) which are in the (transparent) visibility graph at the start and do not disappear during the flight.

1 Introduction

In this paper we investigate a dynamic 3D visibility problem, where the viewing position moves on a circular arc around the origin. We consider a scene, that consists of n non-intersecting spheres $s_1, \cdots, s_n$. The flightpath f is a part of a "circle at infinity" given by a plane P and a range of angles. The range of angles $\{\alpha(t)|t \in [0:1]\} \subset [0:2\pi]$ is parametrized by a "time" parameter $t \in [0:1]$. Here, $\alpha(t)$ is a monotonically increasing function. At time t, the lines of sight are parallel to the ray $r(t)$ in the plane P, which starts in the origin of P and represents the angle $\alpha(t)$ (see Figure 1).

We describe an algorithm which computes the visibility graph at the start of the flightpath and time parameters t, at which the topology of the scene changes, together with the corresponding topology changes. The algorithm has a running time of $O((n+k+p)\log n)$, where p is the number of transparent topology changes (the number of different scene topologies visible along the flightpath, assuming that all spheres are transparent); and k denotes the number of intersections (conflicts) which are in the transparent visibility graph at the

*This author was supported by the ESPRIT II Basic Research Actions Program, under contract No. 3075 (project ALCOM).

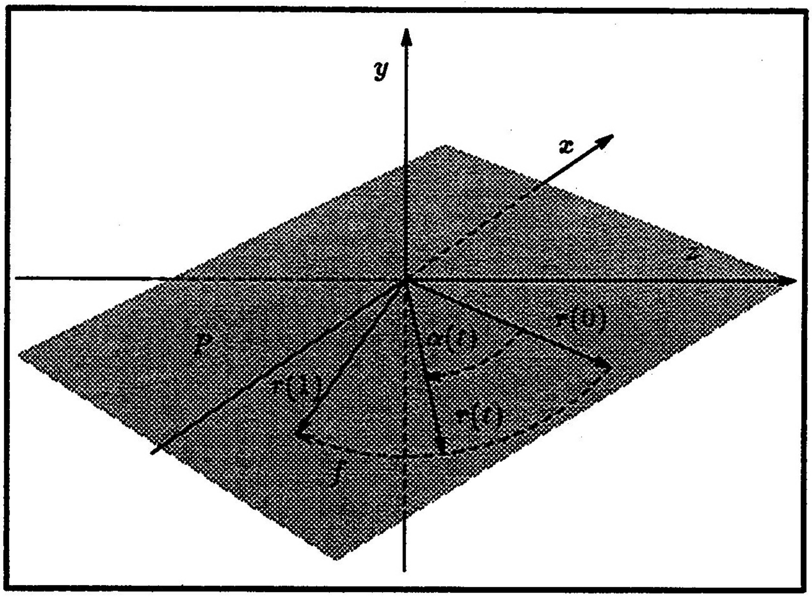

Figure 1: Flightpath f, the plane P and the range of angles $[\alpha(0) : \alpha(1)]$.

start of the flightpath and do not disappear during the flight. Note that $0 \leq k \leq n^2$ and $0 \leq p \leq n^3$. In the transparent visibility graph we store not only the visible, but also the hidden edges and vertices.

Visibility problems of this kind arise in the field of molecular graphics, when, for example, molecules are rotated. In this application the atoms of the molecules are represented by spheres, the so called van der Waals hulls of the atoms.

A short overview about similar problems concerning polygonal scenes can be found in [1]. In that paper, Bern et al. present algorithms having running times $O((n^2+p)\log n)$, $O(n^2 \log n + p)$ and $O(n^2 + p \log n)$, where n is the total number of edges of the polygonal objects in the scene. For terrains, they obtain a running time of $O((n+l)\,\lambda_3(n)\log n)$, where l is the number of opaque topology changes. For terrains and vertical flightpaths, they get a running time of $O(n\,\lambda_4(n)\log n)$. The latter result was also obtained by Cole and Sharir [2].

Other results and methods can be found in [6] and [4]. In [6], Plantinga and Dyer use "aspect graphs" to solve visibility problems. Mulmuley [4] gives algorithms for hidden surface removal for a polygonal scene with respect to a moving viewpoint. His first algorithm preprocesses the set A of scene polygons in time $O(n_r(A)\log n + n(\log n)^2)$ and builds a cylindrical partition of $\mathbf{R}^3$ of size $O(n_r(A)+n)$. Here, $n_r(A)$ denotes the number of "regular crossings", which is smaller than n^2. Given any viewpoint $v \in \mathbf{R}^3$, the scene visible from v can be generated in $O((\log n)^2 + l\log n)$ time, where l is the size of the "fictitious" scene. Mulmuley's second algorithm computes all k_s "semi-opaque" topological changes between successive scenes in case the viewpoint is moving on a linear flightpath. The algorithm takes $O(k_s \log n + n^2\alpha(n)\log n)$ time, where n is the number of edges of the polygons in the scene and $\alpha(n)$ is the inverse Ackermann function.

As far as we know, the present paper is the first one that considers these problems for spheres.

In Section 2 we show what kind of events change the topology of the scene. We describe in Section 3, how the spheres (resp. their centers) move in the projection plane during the flight. In Section 4 we give the algorithm for computing the events that cause changes in the topology of the scene and analyze its running time. Section 5 contains some concluding remarks.

2 Transparent topology changes

We consider a scene in 3-space, that consists of n non-intersecting spheres $s_1, \cdots, s_n$ with centers $M_1, \cdots, M_n$. The flightpath f is a "circle at infinity" given by a plane P and a range of angles $\{\alpha(t) | t \in [0:1]\} \subset [0:2\pi]$. The orbit f is parametrized by the time parameter $t \in [0:1]$. At time t, the scene is projected on a plane $\mathcal{P}_t$, which is orthogonal to the ray $r(t)$. This ray $r(t)$, which represents the angle $\alpha(t)$, is contained in the plane P and starts at the origin of P. We assume wlog that (1) the origin of the object space is always projected in the origin of $\mathcal{P}_t$ and (2) the intersection of P and $\mathcal{P}_t$ is the x-axis of plane $\mathcal{P}_t$. We investigate the topology changes in the planar graph G_t, which represents the projection of the scene in $\mathcal{P}_t$ at time t. The vertices of G_t represent all visible and non-visible intersection points of the circles in $\mathcal{P}_t$. These circles are the images of the spheres under the projection in $\mathcal{P}_t$. The image c_i^t of the sphere s_i in plane $\mathcal{P}_t$ is called the *circle of* s_i at time t. The edges of the graph G_t represent the circular arcs in $\mathcal{P}_t$. A transparent topology change occurs at time t, if the graph G_t changes, that is, for all sufficiently small $\epsilon > 0$, $G_{t-\epsilon}$ and $G_{t+\epsilon}$ are non-isomorphic graphs.

Lemma 1 *A transparent topology change occurs at time t if and only if*

(1) there are two circles c_i^t and c_j^t, which touch at time t and do not touch at times $t-\epsilon$ and $t+\epsilon$ for all sufficiently small $\epsilon > 0$, or

(2) there are three circles c_i^t, c_j^t and c_k^t, which intersect in one point at time t, or

(3) there are two circles c_i^t and c_j^t, which are identical at time t.

The proof of Lemma 1 is obvious. We refer to topology changes being of type (1), (2) or (3), respectively, according to their classification in Lemma 1.

Instead of moving the viewing point, we can also rotate the scene around a rotation axis and keep the viewing point fixed. We transform the whole scene in $O(n)$ time, such that (1) the viewing point is at $z = +\infty$, i.e., the projection plane $\wp$ is parallel to the xy-plane, (2) the rotation axis is the y-axis, (3) the origin of the object space is projected in the origin of the projection plane and (4) the xz-plane of the object space is projected in the x-axis of the projection plane $\wp$.

Let $A := \alpha(1) - \alpha(0)$, i.e., A is the total angle by which the spheres are rotated. Instead of considering the original problem, we investigate the transformed rotation problem. We compute the graph G_0 at the start of the rotation and all topology changes in the projection plane $\wp$ that occur when the spheres are rotated by the angle A around the y-axis.

3 Conflicts between circles

For each $i \in \{1, \cdots, n\}$ we denote the radius of s_i by r_i and the center of s_i at time t by $M_i(t) = (x_i(t), y_i(t), z_i(t))$. Then, $m_i(t) := (x_i(t), y_i(t))$ is the image of $M_i(t)$ under the projection at time t.

In order to compute all topology changes of type (1), we determine all conflicts between circles in the projection plane. To be more precise, we say that two spheres s_i and s_j *conflict* at time t, if $c_i^t \cap c_j^t \neq \emptyset$, or c_i^t is contained in the interior of c_j^t (or vice versa). Two spheres s_i and s_j have a conflict, if there is a time parameter $t \in [0:1]$, such that s_i and s_j conflict at time t.

We determine all pairs s_i, s_j of spheres that have a conflict. For each such pair, we determine all time parameters t at which they conflict and the corresponding parametrized curves

$$c_{ij}^l(t) := \begin{cases} \mathit{undef} & \text{if } c_i^t \cap c_j^t = \emptyset \text{ or } c_i^t \cap c_j^t = c_i^t \\ c_i^t \cap c_j^t & \text{if } c_i^t \cap c_j^t \text{ is one single point} \\ p_l & \text{if } c_i^t \cap c_j^t = \{p_l, p_r\} \text{ and} \\ & p_l \text{ lies to the left of } \overrightarrow{m_i(t)m_j(t)} \end{cases}$$

$$c_{ij}^r(t) := \begin{cases} \mathit{undef} & \text{if } c_i^t \cap c_j^t = \emptyset \text{ or } c_i^t \cap c_j^t = c_i^t \\ c_i^t \cap c_j^t & \text{if } c_i^t \cap c_j^t \text{ is one single point} \\ p_r & \text{if } c_i^t \cap c_j^t = \{p_l, p_r\} \text{ and} \\ & p_r \text{ lies to the right of } \overrightarrow{m_i(t)m_j(t)}. \end{cases}$$

The curves $c_{ij}^l(t)$ and $c_{ij}^r(t)$ describe the movements of the intersection points of c_i^t and c_j^t during the rotation.

If $A \leq 2\pi$, the set of all time parameters t at which s_i and s_j have a conflict consists of at most three subintervals of $[0:1]$. This set of parameters and the corresponding curves can be computed in constant time for each pair s_i, s_j. (See below.)

How can we determine all pairs s_i, s_j of spheres having a conflict? The circles c_i^t and c_j^t have a conflict if and only if $|m_i(t) - m_j(t)| \leq r_i + r_j$. Thus, we have to consider the orbits of the centers in the projection plane. Since $y_i(t) = y_i$ and $y_j(t) = y_j$ for all t, the orbits $O_i = \cup_{0 \leq t \leq 1} m_i(t)$ and $O_j = \cup_{0 \leq t \leq 1} m_j(t)$ are line segments that are parallel to the x-axis in the projection plane. In order to determine if the corresponding circles have a conflict, we have to determine

$$\min_{0 \leq t \leq 1} |m_i(t) - m_j(t)|,$$

the minimal distance between $m_i(t)$ and $m_j(t)$ over all times. If this minimum is less than or equal to $r_i + r_j$, then c_i^t and c_j^t have a conflict.

Since $y_i(t) - y_j(t) = y_i - y_j$ is constant and

$$\min_{0 \leq t \leq 1} |m_i(t) - m_j(t)| = \sqrt{(y_i - y_j)^2 + (\min_{0 \leq t \leq 1} |x_i(t) - x_j(t)|)^2},$$

it suffices to consider

$$\min_{0 \leq t \leq 1} |x_i(t) - x_j(t)|,$$

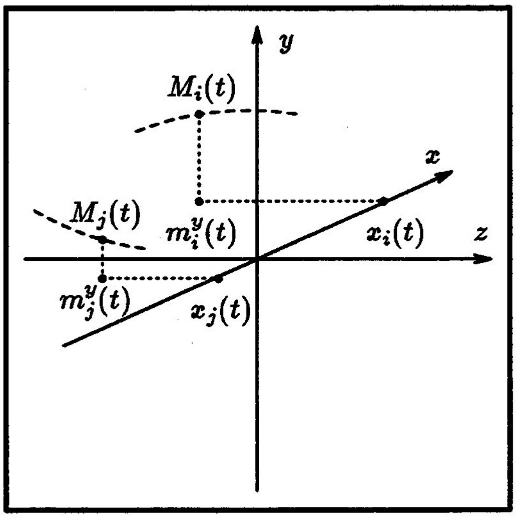

Figure 2: The orbits of $m_i^y(t)$ and $m_j^y(t)$ in the xz-plane.

the minimal distance of the x-coordinates of $m_i(t)$ and $m_j(t)$ at any time.

We consider the orbits of $m_i^y(t) := (x_i(t), z_i(t))$ and $m_j^y(t) := (x_j(t), z_j(t))$ of the centers $M_i(t)$ and $M_j(t)$ under the parallel projection in the xz-plane. (See Figure 2.)

Let $d_i = |m_i^y(0)|$, $d_j = |m_j^y(0)|$, $\phi_i = \arccos(x_i(0)/d_i)$ and $\phi_j = \arccos(x_j(0)/d_j)$. We consider the x-coordinates

$$x_i(t) = x_i(\alpha(t)) = d_i \cos(\alpha(t) + \phi_i) \text{ and } x_j(t) = x_j(\alpha(t)) = d_j \cos(\alpha(t) + \phi_j)$$

as a function of the angle $\alpha(t)$ and define $d_{ij}(\alpha(t))$ as the difference of the x-coordinates of $m_i(t)$ and $m_j(t)$. We have

$$\begin{aligned} d_{ij}(\alpha(t)) &= x_i(\alpha(t)) - x_j(\alpha(t)) = d_i \cos(\alpha(t) + \phi_i) - d_j \cos(\alpha(t) + \phi_j) \\ &= d_i \cos(\alpha(t)) \cos(\phi_i) - d_i \sin(\alpha(t)) \sin(\phi_i) \\ &\quad - d_j \cos(\alpha(t)) \cos(\phi_j) + d_j \sin(\alpha(t)) \sin(\phi_j) \\ &= [d_i \cos(\phi_i) - d_j \cos(\phi_j)] \cos(\alpha(t)) - [d_i \sin(\phi_i) - d_j \sin(\phi_j)] \sin(\alpha(t)) \\ &= \sqrt{(d_i \cos(\phi_i) - d_j \cos(\phi_j))^2 + (d_i \sin(\phi_i) - d_j \sin(\phi_j))^2} \cdot \sin(\alpha(t) + \varphi), \quad (1) \end{aligned}$$

where

$$\tan(\varphi) = -\frac{d_i \cos(\phi_i) - d_j \cos(\phi_j)}{d_i \sin(\phi_i) - d_j \sin(\phi_j)}.$$

We discuss $|d_{ij}(\alpha(t))|$ as a function of $\alpha(t)$, where $0 \leq \alpha(t) < 2\pi$. The function has its minimum (which is zero) for $\alpha(t) = -\varphi$ and $\alpha(t) = -\varphi + \pi$, and its maximum for $\alpha(t) = -\varphi + \pi/2$ and $\alpha(t) = -\varphi + 3\pi/2$. The function $|d_{ij}(\alpha(t))|$ increases from $\alpha(t) = -\varphi$ to $\alpha(t) = -\varphi + \pi/2$, and from $\alpha(t) = -\varphi + \pi$ to $\alpha(t) = -\varphi + 3\pi/2$. It decreases from $\alpha(t) = -\varphi + \pi/2$ to $\alpha(t) = -\varphi + \pi$, and from $\alpha(t) = -\varphi + 3\pi/2$ to $\alpha(t) = -\varphi + 2\pi = -\varphi$.

The following lemma is an immediate consequence of this monotonicity property.

Lemma 2 *The value of* $\min_{0 \leq t \leq 1} |d_{ij}(\alpha(t))|$ *is equal to* $|d_{ij}(\alpha(0))|$, $|d_{ij}(\alpha(1))|$ *or* 0.

It follows from Lemma 2, that we can find all conflicts between circles in the projection plane in the following way: First, we compute all conflicts in the graphs G_0 and G_1. Then we search all pairs s_i, s_j of spheres with

$$\min_{0 \le t \le 1} |d_{ij}(\alpha(t))| = 0 \text{ and } |y_i - y_j| \le r_i + r_j.$$

Lemma 3 *Suppose the rotation angle $A = \alpha(1) - \alpha(0)$ satisfies $A \le \pi$. Then, $\min_{0 \le t \le 1} |d_{ij}(\alpha(t))| = 0$, if and only if one of the two following conditions holds:*

(1) $x_i(0) \le x_j(0)$ and $x_i(1) \ge x_j(1)$

(2) $x_i(0) \ge x_j(0)$ and $x_i(1) \le x_j(1)$.

Proof: From Equation (1), we know that the relative order of the x-coordinates of $m_i(t)$ and $m_j(t)$ changes only at the (at most 2) moments when these x-coordinates become equal. □

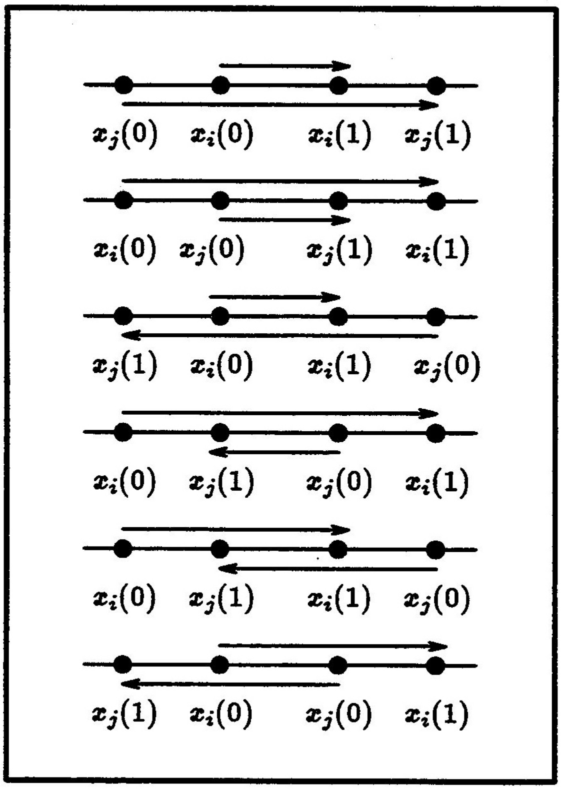

Figure 3: Situations where $\min_{0 \le t \le 1} |d_{ij}(\alpha(t))| = 0$.

In Figure 3, we characterize the different situations where $\min_{0 \le t \le 1} |d_{ij}(\alpha(t))| = 0$. We assume that the four numbers are different and consider the case that $x_i(0) < x_i(1)$. The case $x_i(0) > x_i(1)$ delivers symmetric events.

Lemma 3 tells us that $\min_{0 \le t \le 1} |d_{ij}(\alpha(t))| = 0$ if and only if one of the following conditions holds

(a) $x_i(0) = x_i(1)$ and
$x_i(0)$ is contained in the interval with endpoints $x_j(0), x_j(1)$,

(b) $x_i(0) < x_i(1)$ and
$[(x_j(0) \le x_j(1)$ and $([x_i(0) : x_i(1)]$ is contained in, or contains $[x_j(0) : x_j(1)]))$ or
$(x_j(0) > x_j(1)$ and
$(x_i(0) \in [x_j(1) : x_j(0)]$ or $x_i(1) \in [x_j(1) : x_j(0)]$ or $[x_j(1) : x_j(0)] \subset [x_i(0) : x_i(1)]))]$,

(c) $x_i(0) > x_i(1)$ and
$[(x_j(0) > x_j(1)$ and $([x_i(1) : x_i(0)]$ is contained in, or contains $[x_j(1) : x_j(0)]))$ or
$(x_j(0) \le x_j(1)$ and
$(x_i(0) \in [x_j(0) : x_j(1)]$ or $x_i(1) \in [x_j(0) : x_j(1)]$ or $[x_j(0) : x_j(1)] \subset [x_i(1) : x_i(0)]))]$.

(Note that these three cases are mutually exclusive.) Thus, the problem to compute for a given s_i all s_j with $\min_{0 \le t \le 1} |d_{ij}(\alpha(t))| = 0$ is reduced to a few simple interval queries (note that points are degenerate intervals). These interval queries can be answered efficiently using data structures that are based on priority search trees.

Theorem 1 **([3])** *Let S be a set of n intervals on the real line. There exists a dynamic data structure $T(S)$ storing S, such that we can insert intervals into S and delete intervals from S in time $O(\log n)$. The data structure has size $O(n)$. Given a query interval we can enumerate*

- *all s intervals $[a : b] \in S$ containing the query interval in time $O(\log n + s)$*
- *all s intervals $[a : b] \in S$ contained in the query interval in time $O(\log n + s)$.*

See McCreight [3] for a proof of Theorem 1. The structure $T(S)$ is called the *ic-structure* of S. Note that $T(S)$ is based on two priority search trees.

4 Determining all topology changes

In this section we give the algorithm that computes all topology changes in the scene, when the spheres in the object space are rotated by an angle $A \le \pi$. The algorithm consists of three steps.

STEP 1: We compute all conflicts at $t = 0$ and $t = 1$. This can be done using a slightly modified version of the standard plane sweep algorithm. (See Nurmi [5].)

For every pair s_i and s_j of spheres, which have a conflict in G_0 or G_1, we test, whether and when a topology change of type (1) or (3) arises during the rotation. We store all these topology changes in an event queue Q_e sorted in order of increasing time parameter t.

STEP 2: We search for all pairs s_i, s_j such that

$$\min_{0 \le t \le 1} |d_{ij}(\alpha(t))| = 0 \text{ and } |y_i - y_j| \le r_i + r_j.$$

Note that we can find pairs of spheres, which we have already detected in STEP 1. We consider the points $m_i^- := (x_i(0), y_i - r_i)$ and $m_i^+ := (x_i(0), y_i + r_i), 1 \le i \le n$. We sort these points in order of increasing y-coordinates. We do a sweep from $y = -\infty$ to $y = \infty$. Each time, we reach one of these points, the sweep line stops. We now explain what has to be done when we stop at point p.

Assume $p = m_i^-$. At this moment, the set S_{lr} of intervals $[x_j(0) : x_j(1)]$ (movement from left to right) with $y_j - r_j \le y_i - r_i \le y_j + r_j$ is stored in an *ic*-structure $T(S_{lr})$ and the set S_{rl} of intervals $[x_j(1) : x_j(0)]$ (movement from right to left) with $y_j - r_j \le y_i - r_i \le y_j + r_j$ is stored in an *ic*-structure $T(S_{rl})$.

(a) If $x_i(0) = x_i(1)$, we search for all intervals in $T(S_{lr})$ and $T(S_{rl})$ which contain $x_i(0)$. Then, we insert the interval $[x_i(0) : x_i(1)]$ in the tree $T(S_{lr})$.

(b) If $x_i(0) < x_i(1)$, we search for all intervals $[x_j(0) : x_j(1)]$ in $T(S_{lr})$ which are contained in, or contain the interval $[x_i(0) : x_i(1)]$, and for all intervals $[x_j(1) : x_j(0)]$ in $T(S_{rl})$ which contain $x_i(0)$ or $x_i(1)$, or are contained in $[x_i(0) : x_i(1)]$. Then, we insert the interval $[x_i(0) : x_i(1)]$ in $T(S_{lr})$.

(c) If $x_i(0) > x_i(1)$, we search for all intervals $[x_j(0) : x_j(1)]$ in $T(S_{lr})$ which contain $x_i(0)$ or $x_i(1)$, or are contained in $[x_i(1) : x_i(0)]$, and for all intervals $[x_j(1) : x_j(0)]$ in $T(S_{rl})$ which are contained in, or contain $[x_i(1) : x_i(0)]$. Then, we insert the interval $[x_i(1) : x_i(0)]$ in $T(S_{rl})$.

For every pair s_i, s_j we find, we test whether we have already detected it in STEP 1. If this is not the case, we compute when topology changes of type (1) and (3) take place for this pair. All these topology changes are stored in the event queue Q_e.

Assume $p = m_i^+$. Then, we delete the interval with endpoints $x_i(0), x_i(1)$ from the corresponding *ic*-structure. More precisely, if $x_i(0) \le x_i(1)$, we delete the interval $[x_i(0) : x_i(1)]$ from $T(S_{lr})$. If $x_i(1) < x_i(0)$, then we remove $[x_i(1) : x_i(0)]$ from $T(S_{rl})$.

STEP 3: Finally we determine all topology changes of type (2). For each sphere s_i which has at least one conflict, we do a sweep from $t = 0$ to $t = 1$. During this sweep we only consider the set C_i of spheres which have a conflict with s_i. This set C_i is obtained from Q_e.

During the sweep, we maintain a balanced binary search tree B, which contains the intersection curves $c_{ij}^*(t), * \in \{l, r\}$ defined at time t (i.e., $c_{ij}^*(t)$ is not equal to *undef*). At time t, the points $c_{ij}^*(t)$ in the current version of the tree B are sorted in order of increasing angle $\delta_{ij}^*(t) := \arccos((x_{coord}(c_{ij}^*(t)) - x_i(t))/r_i)$. In the event queue Q_s of the sweep we store all time parameters t when (1) a curve $c_{ij}^*(t)$ changes from undefined to defined or vice versa, (2) $c_i^t \cap c_j^t$ becomes equal to c_i^t for some sphere $s_j \in C_i$, or (3) $c_{ij}^a(t) = c_{ik}^b(t), a, b \in \{l, r\}$, for $j \neq k$.

When we stop during the sweep, in order to insert or delete a curve $c_{ij}^*(t)$ from the tree B or to swap two curves' positions, we have to compute all intersection points of the new neighbors in B. This can be done in constant time for every pair of new neighbors $c_{ij}^a(t), c_{ik}^b(t), a, b \in \{l, r\}$. The equations of the intersection points, which can be found in the appendix, imply that there are only a constant number of parameters t, where $c_{ij}^a(t) = c_{ik}^b(t)$. If $c_{ij}^a(t) = c_{ik}^b(t)$, then we have found a parameter t where three circles c_i^t, c_j^t, c_k^t intersect in one point. If $c_i^t = c_j^t$, then every point $c_{ik}^*(t)$ with $k \neq j$, which is in B at time t, is a point, where three circles c_i^t, c_j^t, c_k^t intersect. All these parameters t and the corresponding topology changes are stored in the event queue Q_e.

As mentioned above, we carry out this procedure for every sphere $s_i, i = 1, \cdots, n$, which has at least one conflict with another sphere.

We now analyze the running time of the algorithm. Let k be the number of conflicts between the circles $c_1^0, \cdots, c_n^0$ in the graph G_0, which do not disappear during the rotation. Let p_1 be the number of transparent topology changes which we find in STEP 1. The integer p_2 is defined as the number of transparent topology changes we find in STEP 2, which we have not found in STEP 1. Finally, p_3 is the number of transparent topology changes of type (2). Then, $p := p_1 + p_2 + p_3$ is the total number of topology changes during the rotation.

The conflicts in the graphs G_0 and G_1 can be computed in $O((n + k + p_1) \log n)$ time using the plane sweep algorithm of [5]. Note that we determine both visible and occluded intersections. The sweep in STEP 2 can be done in $O(n \log n + k + p_1 + p_2)$ time. Since intersections between two curves $c_{ij}^a(t), c_{ik}^b(t), a, b \in \{l, r\}$ can be computed in constant time, the entire STEP 3 can be done in $O((k + p_1 + p_2 + p_3) \log n)$ time. Thus, we get a running time of $O((n + k + p) \log n)$ for the entire algorithm.

Since an arbitrary angle $A \leq 2\pi$ can be partitioned into 2 angles that are at most π, we have obtained the following result:

Theorem 2 *Let S be a scene, which consists of n non-intersecting spheres. All transparent topology changes for a flightpath, which is a part of a "circle at infinity" given by a plane P and a range of angles, can be computed in time $O((n+k+p) \log n)$. Here, p is the number of transparent topology changes and k is the number of transparent conflicts at the start, which do not disappear during the flight.*

5 Concluding remarks

We have given an algorithm for maintaining the visibility map of a collection of non-intersecting spheres when the viewpoint moves on a circle at infinity. The algorithm has a running time of $O((n + k + p) \log n)$.

Of course, one open problem is to improve our time bound. First, it would be interesting to find an algorithm that spends only constant time for each topology change, instead of $O(\log n)$.

Second, it would be interesting to develop algorithms whose running times depend on the number of *opaque* topology changes, instead of the number of transparent changes.

Finally, it is not known whether our techniques can be applied to similar problems for other objects besides spheres.

References

[1] M. Bern, D. Dobkin, D. Eppstein and R. Grossmann. *Visibility with a Moving Point of View.* Proc. 1st Annual ACM-SIAM Symp. on Discrete Algorithms 1990, pp. 107-117.

[2] R. Cole and M. Sharir. *Visibility Problems for Polyhedral Terrains.* J. of Symbolic Computation, vol. 7 (1989), pp. 11-30.

[3] E. M. McCreight. *Priority Search Trees.* SIAM Journal of Computing, vol. 14 (1985), pp. 257-276.

[4] K. Mulmuley. *Hidden Surface Removal with Respect to a Moving Viewpoint.* Proc. 23th Annual ACM Symposium on Theory of Computing (1991), pp. 512-522.

[5] O. Nurmi. *A Fast Line-Sweep Algorithm For Hidden Line Elimination.* BIT, vol. 25 (1985), pp. 466-472.

[6] W. H. Plantinga and C. D. Dyer. *An Algorithm for Constructing the Aspect Graph.* Proc. 27th IEEE Foundations of Comp. Science (1986), pp. 123-131.

Appendix

We describe a way to solve the intersection equations, which arise in STEP 3 of the algorithm. We consider the sphere s_i and two other spheres s_j, s_k, which have a conflict with s_i. Let

$$\begin{aligned} c_i^t &: (X - x_i(t))^2 + Y^2 = r_i^2 \\ c_j^t &: (X - x_j(t))^2 + (Y - y_j)^2 = r_j^2 \\ c_k^t &: (X - x_k(t))^2 + (Y - y_k)^2 = r_k^2 \end{aligned}$$

be the equations of the corresponding circles in the projection plane. We assume for simplicity that $y_i = 0$. First of all, we compute two lines $l_{ij}(t)$ and $l_{ik}(t)$: The line $l_{ij}(t)$ (resp. $l_{ik}(t)$) passes through the intersection points $c_i^t \cap c_j^t$ (resp. $c_i^t \cap c_k^t$). The line $l_{ij}(t)$ (resp. $l_{ik}(t)$) is orthogonal to the line through the centers $m_i(t)$ and $m_j(t)$ (resp. $m_i(t)$ and $m_k(t)$). Since

$$\begin{aligned} l_{ij}(t) &: 2(x_j(t) - x_i(t))X + (x_i(t))^2 - (x_j(t))^2 + 2y_jY - y_j^2 - r_i^2 + r_j^2 = 0 \\ l_{ik}(t) &: 2(x_k(t) - x_i(t))X + (x_i(t))^2 - (x_k(t))^2 + 2y_kY - y_k^2 - r_i^2 + r_k^2 = 0, \end{aligned}$$

the intersection point $(x(t), y(t)) = l_{ij}(t) \cap l_{ik}(t)$ has coordinates

$$x(t) = -\frac{x_i(t)^2 - r_i^2 - x_j(t)^2 - \frac{y_j x_i(t)^2}{y_k} + \frac{y_j r_i^2}{y_k} + \frac{y_j x_k(t)^2}{y_k} + y_k y_j - \frac{y_j r_k^2}{y_k} - y_j^2 + r_j^2}{2x_j(t) - 2x_i(t) + \frac{2y_j x_i(t)}{y_k} - \frac{2y_j x_k(t)}{y_k}},$$

$$y(t) = -\frac{x_i(t)^2 - 2x(t)x_i(t) - r_i^2 + 2x(t)x_k(t) - x_k(t)^2 - y_k^2 + r_k^2}{2y_k}.$$

If the circles c_i^t, c_j^t, c_k^t intersect in one point at time t, then $(x(t), y(t))$ has to be a point on the circle c_i^t. In order to compute all these time parameters t, we replace X and Y in the equation of c_i^t by $x(t)$ and $y(t)$. We multiply this equation by $den(t) = y_k(x_i(t) - x_j(t)) - y_j(x_i(t) - x_k(t))$ and obtain the following polynomial equation in $x_i(t), x_j(t), x_k(t)$. (This polynomial was obtained using MAPLE.)

$$\begin{aligned} 0 = {} & -2y_j^2y_k^2r_j^2 - 2r_i^4y_ky_j - 2y_j^3r_i^2y_k - 2r_i^2y_k^3y_j - 2r_i^2y_k^2r_j^2 + 2y_j^3r_k^2y_k + 4r_i^2y_k^2y_j^2 + 2y_jy_k^3r_j^2 - \\ & 2y_j^2y_k^2r_k^2 + (y_k^4 - 2y_k^2r_i^2 - 2r_k^2r_j^2 + r_j^4 + 2y_ky_jr_k^2 - 2y_j^3y_k + 2y_k^2y_j^2 + r_k^4 + 4y_ky_jr_i^2 - 2y_j^2r_j^2 - 2y_j^2r_i^2 - \\ & 2r_k^2y_k^2 + 2y_jr_j^2y_k - 2y_k^3y_j + y_j^4)x_i(t)^2 + (2y_jr_j^2y_k - 2y_j^2r_k^2 + y_j^4 - 2y_ky_jr_i^2 - 2y_j^3y_k + r_i^4 - 2y_j^2r_j^2 + \\ & r_j^4 - 2r_i^2r_j^2 + 2y_k^2y_j^2)x_k(t)^2 - 2y_j^2r_i^2r_k^2 + (2r_k^2 - 2r_i^2 - 2y_k^2)x_k(t)x_j(t)^3 + (4r_k^2 - 2r_j^2 + 2y_j^2 - \\ & 2r_i^2 + 2y_k^2 - 2y_ky_j)x_j(t)^2x_i(t)^2 + (4y_k^3y_j - 2r_i^2r_j^2 - 2y_k^2y_j^2 - 2y_j^2r_k^2 + 2r_k^2r_j^2 + 2r_k^2r_i^2 - 4y_ky_jr_i^2 - \\ & 4y_ky_jr_k^2 - 2y_k^4 + 2y_k^2r_i^2 + 4r_k^2y_k^2 + 2y_j^2r_i^2 + 2y_k^2r_j^2 - 2r_k^4)x_i(t)x_j(t) + (8y_ky_jr_i^2 + 2r_k^2r_i^2 - 2r_i^4 + \\ & 2r_i^2r_j^2 + 2y_j^2r_k^2 - 2y_k^2y_j^2 - 2r_k^2r_j^2 - 2y_j^2r_i^2 - 2y_k^2r_i^2 + 2y_k^2r_j^2)x_k(t)x_j(t) + (y_j^2 - 2y_ky_j + y_k^2)x_i(t)^4 + \\ & (2y_k^2 + 4r_j^2 - 2r_k^2 + 2y_j^2 - 2y_ky_j - 2r_i^2)x_k(t)^2x_i(t)^2 + (-2r_i^2 - 2y_j^2 + 2r_j^2)x_j(t)x_k(t)^3 + (4r_i^2 - \\ & 2r_j^2 + 2y_j^2 - 2r_k^2 - 2y_ky_j + 2y_k^2)x_j(t)^2x_k(t)^2 + (2r_i^2 - 2r_k^2 - 2y_k^2)x_i(t)x_j(t)^3 + (-2y_j^2 - 2r_j^2 + \\ & 2r_i^2)x_i(t)x_k(t)^3 + (4r_k^2 - 2r_j^2 - 2r_i^2 - 4y_k^2 + 2y_j^2 + 4y_ky_j)x_j(t)x_k(t)^2x_i(t) + (4r_i^2 + 2y_k^2 - 2r_j^2 + \\ & 2y_j^2 - 8y_ky_j - 2r_k^2)x_j(t)x_i(t)^2x_k(t) + (4r_j^2 + 2y_k^2 - 2r_k^2 - 4y_j^2 - 2r_i^2 + 4y_ky_j)x_j(t)^2x_i(t)x_k(t) + \\ & (-2y_k^2 - 2y_j^2 + 4y_ky_j + 2r_j^2 - 2r_k^2)x_j(t)x_i(t)^3 + (-2y_k^2 - 2r_j^2 - 2y_j^2 + 4y_ky_j + 2r_k^2)x_k(t)x_i(t)^3 + \\ & (2y_j^2r_i^2 + 4y_j^3y_k + 2y_k^2r_i^2 - 2r_k^2r_i^2 - 4y_jr_j^2y_k + 2r_i^2r_j^2 - 2y_k^2r_j^2 - 4y_ky_jr_i^2 + 4y_j^2r_j^2 - 2y_j^4 + 2y_j^2r_k^2 + \\ & 2r_k^2r_j^2 - 2y_k^2y_j^2 - 2r_j^4)x_i(t)x_k(t) + (y_k^4 + 2y_k^2y_j^2 + r_i^4 - 2y_k^3y_j - 2y_k^2r_j^2 - 2y_ky_jr_i^2 + 2y_ky_jr_k^2 - 2r_k^2y_k^2 + \end{aligned}$$

$r_k^4 - 2r_k^2r_i^2)x_j(t)^2 + r_i^4y_k^2 + y_j^2r_i^4 + y_j^2y_k^4 - 2y_j^3y_k^3 + y_j^2r_k^4 + y_j^4y_k^2 + r_j^4y_k^2 + 2r_i^2y_ky_jr_k^2 + 2y_jr_i^2r_j^2y_k -$
$2y_jr_k^2r_j^2y_k - 2x_i(t)x_j(t)^4x_k(t) + 2x_i(t)x_j(t)^3x_k(t)^2 + 2x_i(t)^3x_j(t)^2x_k(t) + 2x_k(t)^3x_i(t)x_j(t)^2 -$
$2x_k(t)^4x_i(t)x_j(t) + 2x_k(t)^2x_i(t)^3x_j(t) + 2x_i(t)^2x_j(t)^3x_k(t) - 6x_k(t)^2x_i(t)^2x_j(t)^2 -$
$2x_i(t)^4x_j(t)x_k(t) + 2x_k(t)^3x_i(t)^2x_j(t) + x_j(t)^4y_k^2 + y_j^2x_k(t)^4 + x_k(t)^4x_i(t)^2 - 2x_k(t)^3x_i(t)^3 +$
$x_i(t)^2x_j(t)^4 - 2x_i(t)^3x_j(t)^3 + x_i(t)^4x_j(t)^2 + x_k(t)^2x_i(t)^4 + x_k(t)^4x_j(t)^2 - 2x_k(t)^3x_j(t)^3 +$
$x_k(t)^2x_j(t)^4.$

(Note that we also have to consider the case $den(t) = 0$. We have $den(t) = 0$ if and only if $m_i(t), m_j(t), m_k(t)$ lie on one line. Thus, if $den(t) = 0$ and the three circles intersect in one point at time t, then the lines $l_{ij}(t)$ and $l_{ik}(t)$ are identical. This computation is similar to the one that follows, but shorter. Therefore, we do not carry out this computation.)

In the above equation, we replace $x_*(t)$ by $a_* \cos(\alpha(t)) + b_* \sin(\alpha(t)), * \in \{i, j, k\}$. Then we replace $\cos(\alpha(t))$ by C and $\sin(\alpha(t))$ by S:

$0 = (-2b_k^3b_j^3 - 2b_ib_j^4b_k - 2b_i^4b_jb_k - 2b_k^4b_ib_j + 2b_k^3b_ib_j^2 + 2b_ib_j^3b_k^2 - 6b_k^2b_i^2b_j^2 - 2b_k^3b_i^3 + 2b_i^3b_j^2b_k +$
$2b_k^3b_i^2b_j + 2b_i^2b_j^3b_k - 2b_i^3b_j^3 + b_i^2b_j^4 + b_k^2b_i^4 + b_k^4b_i^2 + b_k^2b_j^4 + b_i^4b_j^2 + 2b_k^2b_i^3b_j + b_k^4b_j^2)S^6$
$+(4b_k^2a_jb_j^3 - 2a_ib_j^4b_k + 2a_kb_kb_j^4 - 6b_k^3a_jb_j^2 - 6a_kb_k^2b_j^3 + 4a_kb_k^3b_j^2 + 2b_k^4a_jb_j + 2b_i^3b_j^2a_k + 4b_i^3a_jb_jb_k +$
$6a_ib_i^2b_j^2b_k + 4b_ib_j^3a_kb_k + 6b_ia_jb_j^2b_k^2 + 2a_ib_j^3b_k^2 + 2b_i^4a_jb_j + 4a_ib_i^3b_j^2 - 6b_i^3a_jb_j^2 - 6a_ib_i^2b_j^3 + 4b_i^2a_jb_j^3 +$
$2a_ib_ib_j^4 + 6a_kb_k^2b_i^2b_j + 4b_k^3a_ib_ib_j + 2b_k^3b_i^2a_j - 8a_kb_k^3b_ib_j - 2b_k^4b_ia_j - 2b_k^4a_ib_j - 2b_i^4b_ja_k - 2b_i^4a_jb_k -$
$8a_ib_i^3b_jb_k - 12a_kb_kb_i^2b_j^2 - 12b_k^2b_i^2a_jb_j - 12b_k^2a_ib_ib_j^2 + 2b_k^3a_ib_j^2 + 4b_k^3b_ia_jb_j + 4b_k^2a_ib_i^3 + 6a_kb_k^2b_ib_j^2 +$
$2a_kb_kb_i^4 - 6b_k^3a_ib_i^2 - 6a_kb_k^2b_i^3 + 2b_i^2b_j^3a_k + 2b_k^4a_ib_i + 4a_kb_k^3b_i^2 + 6b_i^2a_jb_j^2b_k + 4a_ib_ib_j^3b_k + 2b_k^2b_i^3a_j +$
$6b_k^2a_ib_i^2b_j + 4a_kb_kb_i^3b_j - 2b_ib_j^4a_k - 8b_ia_jb_j^3b_k)S^5C$
$+(-8a_ia_jb_j^3b_k - 2a_ib_j^4a_k + 6b_k^2a_j^2b_j^2 + 8a_kb_ka_jb_j^3 + a_k^2b_j^4 - 6b_k^3a_j^2b_j - 18a_kb_k^2a_jb_j^2 - 6a_k^2b_kb_j^3 + b_k^4a_j^2 +$
$8a_kb_k^3a_jb_j + 4b_i^3a_jb_ja_k + 6a_k^2b_k^2b_j^2 + 2b_i^3a_j^2b_k + 6a_ib_i^2b_j^2a_k + 6a_i^2b_ib_j^2b_k + 2b_ib_j^3a_k^2 + 4a_ib_j^3a_kb_k +$
$6a_ia_jb_j^2b_k^2 + 8a_ib_i^3a_jb_j + b_i^4a_j^2 + 6a_i^2b_i^2b_j^2 - 6b_i^3a_j^2b_j - 18a_ib_i^2a_jb_j^2 - 6a_i^2b_ib_j^3 + a_i^2b_j^4 + 6b_i^2a_j^2b_j^2 +$
$8a_ib_ia_jb_j^3 + 6a_kb_k^2b_i^2a_j + 4b_k^3a_ib_ia_j + 12a_kb_k^2a_ib_ib_j + 6a_k^2b_kb_i^2b_j + 2b_k^3a_i^2b_j - 8a_kb_k^3b_ia_j - 8a_kb_k^3a_ib_j -$
$12a_k^2b_k^2b_ib_j - 2b_k^4a_ia_j - 8a_ib_i^3a_jb_k - 8a_ib_i^3b_ja_k - 12a_i^2b_i^2b_jb_k - 2b_i^4a_ja_k - 6b_k^2b_i^2a_j^2 - 6b_k^2a_i^2b_j^2 -$
$24b_k^2a_ib_ia_jb_j - 24a_kb_kb_i^2a_jb_j - 24a_kb_ka_ib_ib_j^2 - 6a_k^2b_i^2b_j^2 + 2b_k^3b_ia_j^2 + 6a_kb_k^2a_ib_j^2 + 4b_k^3a_ia_jb_j +$
$6b_k^2a_i^2b_i^2 + 8a_kb_ka_ib_i^3 + 12a_kb_k^2b_ia_jb_j + 6a_k^2b_kb_ib_j^2 - 6b_k^3a_i^2b_i - 18a_kb_k^2a_ib_i^2 - 6a_k^2b_kb_i^3 + a_k^2b_i^4 +$
$6b_i^2a_jb_j^2a_k + 8a_kb_k^3a_ib_i + 6b_i^2a_j^2b_jb_k + 12a_ib_ia_jb_j^2b_k + 6a_k^2b_k^2b_i^2 + 4a_ib_ib_j^3a_k + 2a_i^2b_j^3b_k + b_k^4a_i^2 +$
$6b_k^2a_ib_i^2a_j + 6b_k^2a_i^2b_ib_j + 4a_kb_kb_i^3a_j + 12a_kb_ka_ib_i^2b_j + 12a_ib_i^2a_jb_jb_k + 12b_ia_jb_j^2a_kb_k + 6b_ia_j^2b_jb_k^2 +$
$2a_k^2b_i^3b_j - 12b_ia_j^2b_j^2b_k - 8b_ia_jb_j^3a_k)S^4C^2$
$+(-8a_ia_jb_j^3a_k - 12a_ia_j^2b_j^2b_k - 18a_k^2b_ka_jb_j^2 - 2a_k^3b_j^3 + 4b_k^2a_j^3b_j + 12a_kb_ka_j^2b_j^2 + 4a_k^2a_jb_j^3 - 2b_k^3a_j^3 -$
$18a_kb_k^2a_j^2b_j + 4a_ib_i^3a_j^2 + 4a_kb_k^3a_j^2 + 12a_k^2b_k^2a_jb_j + 4a_k^3b_kb_j^2 + 6a_ib_i^2a_j^2b_k + 6a_i^2b_ib_j^2a_k + 2a_i^3b_j^2b_k +$
$6b_ia_jb_j^2a_k^2 + 2b_ia_j^3b_k^2 + 2a_ib_j^3a_k^2 + 6a_ia_j^2b_jb_k^2 + 12a_i^2b_i^2a_jb_j + 4a_i^3b_ib_j^2 - 2b_i^3a_j^3 - 18a_ib_i^2a_j^2b_j -$
$18a_i^2b_ia_jb_j^2 - 2a_i^3b_j^3 + 4b_i^2a_j^3b_j + 12a_ib_ia_j^2b_j^2 + 4a_i^2a_jb_j^3 - 12a_k^2b_k^2b_ia_j + 12a_kb_k^2a_ib_ia_j + 6a_kb_k^2a_i^2b_j +$
$2b_k^3a_i^2a_j + 6a_k^2b_kb_i^2a_j + 12a_k^2b_ka_ib_ib_j + 2a_k^3b_i^2b_j - 8a_kb_k^3a_ia_j - 12a_k^2b_k^2a_ib_j - 8a_k^3b_kb_ib_j - 8a_ib_i^3a_ja_k -$
$12a_i^2b_i^2b_ja_k - 12a_i^2b_i^2a_jb_k - 8a_i^3b_ib_jb_k - 12b_k^2a_ib_ia_j^2 - 12b_k^2a_i^2a_jb_j - 48a_kb_ka_ib_ia_jb_j - 12a_kb_kb_i^2a_j^2 -$
$12a_kb_ka_i^2b_j^2 - 12a_k^2b_i^2a_jb_j - 12a_k^2a_ib_ib_j^2 + 2b_k^3a_ia_j^2 + 4b_k^2a_i^3b_i + 12a_kb_ka_i^2b_i^2 + 6a_kb_k^2b_ia_j^2 + 12a_kb_k^2a_ia_jb_j +$
$6a_k^2b_ka_ib_j^2 + 4a_k^2a_ib_i^3 - 18a_kb_k^2a_i^2b_i + 12a_k^2b_kb_ia_jb_j + 6b_i^2a_j^2b_ja_k - 18a_k^2b_ka_ib_i^2 + 2a_k^3b_ib_j^2 + 12a_k^2b_k^2a_ib_i -$
$2b_k^3a_i^3 + 2b_i^2a_j^3b_k + 12a_ib_ia_jb_j^2a_k - 2a_k^3b_i^3 + 12a_ib_ia_j^2b_jb_k + 4a_kb_k^3a_i^2 + 6a_i^2a_jb_j^2b_k + 2a_i^2b_j^3a_k +$
$6b_k^2a_i^2b_ia_j + 4a_k^3b_kb_i^2 + 2b_k^2a_i^3b_j + 12a_kb_ka_ib_i^2a_j + 12a_kb_ka_i^2b_ib_j + 2b_i^3a_j^2a_k + 12a_ib_i^2a_jb_ja_k +$
$12a_i^2b_ia_jb_jb_k + 12b_ia_j^2b_ja_kb_k + 12a_ia_jb_j^2a_kb_k + 2a_k^2b_i^3a_j + 6a_k^2a_ib_i^2b_j - 12b_ia_j^2b_j^2a_k - 8b_ia_j^3b_jb_k)S^3C^3$
$+(-8b_ia_j^3b_ja_k - 2b_ia_j^4b_k - 12a_ia_j^2b_j^2a_k - 8a_ia_j^3b_jb_k - 18a_k^2b_ka_j^2b_j + 8a_kb_ka_j^3b_j + b_k^2a_j^4 + 6a_k^2a_j^2b_j^2 -$
$6a_kb_k^2a_j^3 + 6a_k^2a_i^2b_ib_j - 6a_k^3a_jb_j^2 + 6a_k^2b_k^2a_j^2 + 8a_k^3b_ka_jb_j + a_k^4b_j^2 + 6a_ib_i^2a_j^2a_k + 6a_i^2b_ia_j^2b_k + 2a_i^3b_j^2a_k +$

$4a_i^3a_jb_jb_k+6b_ia_j^2b_ja_k^2+4b_ia_j^3a_kb_k+6a_ia_jb_j^2a_k^2+6a_i^2b_i^2a_j^2+8a_i^3b_ia_jb_j+a_i^4b_j^2+2a_ia_j^3b_k^2-6a_ib_i^2a_j^3-$
$18a_i^2b_ia_j^2b_j-6a_i^3a_jb_j^2+b_i^2a_j^4+8a_ib_ia_j^3b_j+6a_i^2a_j^2b_j^2+6a_kb_k^2a_i^2a_j+12a_k^2b_ka_ib_ia_j+6a_k^2b_ka_i^2b_j-$
$12a_k^2b_k^2a_ia_j-8a_k^3b_kb_ia_j-8a_k^3b_ka_ib_j-2a_k^4b_ib_j-12a_i^2b_i^2a_ja_k-8a_i^3b_ib_ja_k-8a_i^3b_ia_jb_k-2a_i^4b_jb_k-$
$6b_k^2a_i^2a_j^2-24a_kb_ka_i^2a_jb_j-24a_kb_ka_ib_ia_j^2-6a_k^2b_i^2a_j^2-24a_k^2a_ib_ia_jb_j-6a_k^2a_i^2b_j^2+b_k^2a_i^4+6a_kb_k^2a_ia_j^2+$
$8a_kb_ka_i^3b_i+6a_k^2a_i^2b_i^2+6a_k^2b_kb_ia_j^2+12a_k^2b_ka_ia_jb_j-6a_kb_k^2a_i^3-18a_k^2b_ka_i^2b_i-6a_k^3a_ib_i^2+4a_k^3b_ia_jb_j+$
$2a_k^3a_ib_j^2+8a_k^3b_ka_ib_i+4a_ib_ia_j^3b_k+2b_i^2a_j^3a_k+6a_k^2b_k^2a_i^2+12a_ib_ia_j^2b_ja_k+6a_i^2a_j^2b_jb_k+6a_i^2a_jb_j^2a_k+$
$a_k^4b_i^2+2b_k^2a_i^3a_j+4a_kb_ka_i^3b_j+12a_kb_ka_i^2b_ia_j+12a_i^2b_ia_jb_ja_k+12a_ia_j^2b_ja_kb_k+2a_k^3b_i^2a_j+4a_k^3a_ib_ib_j+$
$6a_k^2a_ib_i^2a_j)S^2C^4$
$+(-2b_ia_j^4a_k-8a_ia_j^3b_ja_k+4a_k^3a_ib_ia_j-6a_k^2b_ka_j^3+2a_kb_ka_j^4+4a_k^2a_j^3b_j-6a_k^3a_j^2b_j+6a_k^2a_i^2b_ia_j+$
$4a_k^3b_ka_j^2+2a_k^4a_jb_j+6a_i^2b_ia_j^2a_k+2a_i^3a_j^2b_k+4a_i^3a_jb_ja_k+2b_ia_j^3a_k^2+6a_ia_j^2b_ja_k^2+4a_i^3b_ia_j^2+2a_i^4a_jb_j+$
$4a_ia_j^3a_kb_k-2a_ia_j^4b_k-6a_i^2b_ia_j^3-6a_i^3a_j^2b_j+2a_ib_ia_j^4+4a_i^2a_j^3b_j+6a_k^2b_ka_i^2a_j-8a_k^3b_ka_ia_j-2a_k^4b_ia_j-$
$8a_i^3b_ia_ja_k-2a_i^4a_jb_k-2a_i^4b_ja_k-12a_kb_ka_i^2a_j^2-12a_k^2a_ib_ia_j^2-12a_k^2a_i^2a_jb_j+2a_kb_ka_i^4+4a_k^2a_i^3b_i-$
$6a_k^2b_ka_i^3+6a_k^2b_ka_ia_j^2+2a_k^3b_ia_j^2-6a_k^3a_i^2b_i+4a_k^3a_ia_jb_j-2a_k^4a_ib_j+4a_ib_ia_j^3a_k+6a_i^2a_j^2b_ja_k+$
$4a_k^3b_ka_i^2+2a_k^4a_ib_i+4a_kb_ka_i^3a_j+2a_i^2a_j^3b_k+2a_k^2a_i^3b_j+2a_k^3a_i^2b_j)SC^5$
$+(a_k^2a_i^4-2a_ia_j^4a_k+2a_k^3a_ia_j^2+2a_ia_j^3a_k^2+a_k^4a_j^2+a_i^2a_j^4+a_k^2a_j^4+2a_k^3a_i^2a_j-2a_k^4a_ia_j-2a_i^3a_j^3+$
$2a_i^3a_j^2a_k+a_i^4a_j^2+2a_k^2a_i^3a_j+2a_i^2a_j^3a_k-2a_k^3a_j^3-2a_k^3a_i^3-2a_i^4a_ja_k+a_k^4a_i^2-6a_k^2a_i^2a_j^2)C^6$
$+(2y_j^2b_jb_i^2b_k-2r_j^2b_jb_i^2b_k+2y_k^2b_jb_i^2b_k+2y_j^2b_jb_k^2b_i+4r_i^2b_jb_i^2b_k-8y_ky_jb_jb_i^2b_k-2r_k^2b_jb_i^2b_k+$
$4r_k^2b_jb_k^2b_i+y_j^2b_k^4+y_j^2b_i^4+y_k^2b_i^4+4y_ky_jb_j^2b_ib_k+4y_ky_jb_jb_k^2b_i-2y_ky_jb_i^4-4y_k^2b_jb_k^2b_i-2r_i^2b_jb_k^2b_i-$
$2r_j^2b_jb_k^2b_i+2y_k^2b_j^2b_k^2+2y_j^2b_j^2b_k^2-2r_k^2b_j^2b_k^2-2y_ky_jb_j^2b_k^2+4r_i^2b_j^2b_k^2-2r_i^2b_k^2b_i^2-2r_j^2b_j^2b_k^2+2r_j^2b_jb_k^3-$
$2y_j^2b_jb_k^3-2r_i^2b_jb_k^3-2r_i^2b_kb_j^3-2y_k^2b_kb_j^3+2r_k^2b_kb_j^3+2y_j^2b_k^2b_i^2-2r_k^2b_k^2b_i^2-2y_ky_jb_k^2b_i^2+4r_j^2b_k^2b_i^2-$
$4y_j^2b_j^2b_ib_k+2y_k^2b_k^2b_i^2+2r_i^2b_ib_k^3-2r_j^2b_ib_k^3+4y_ky_jb_kb_i^3+2r_k^2b_kb_i^3-2y_j^2b_ib_k^3-2y_j^2b_kb_i^3-2r_k^2b_j^2b_ib_k-$
$2r_j^2b_kb_i^3-2y_k^2b_kb_i^3+2y_k^2b_j^2b_ib_k+y_k^2b_j^4-2y_ky_jb_j^2b_i^2+2y_k^2b_j^2b_i^2-2r_i^2b_j^2b_i^2+4r_j^2b_j^2b_ib_k+2y_j^2b_j^2b_i^2-$
$2r_i^2b_j^2b_ib_k+4r_k^2b_j^2b_i^2-2r_j^2b_j^2b_i^2-2y_k^2b_ib_j^3+2r_i^2b_ib_j^3-2r_k^2b_ib_j^3-2r_k^2b_jb_i^3+2r_j^2b_jb_i^3+4y_ky_jb_jb_i^3-$
$2y_j^2b_jb_i^3-2y_k^2b_jb_i^3)S^4$
$+(8y_ky_ja_jb_jb_ib_k-2r_j^2b_jb_i^2a_k-4r_j^2b_ja_ib_ib_k+4y_ky_ja_jb_k^2b_i-2r_j^2a_jb_i^2b_k+2y_k^2a_jb_i^2b_k+4y_k^2b_ja_ib_ib_k+$
$2y_k^2b_jb_i^2a_k+4r_i^2a_jb_i^2b_k+8r_i^2b_ja_ib_ib_k+4r_i^2b_jb_i^2a_k+4y_j^2b_ja_kb_kb_i-8y_ky_ja_jb_i^2b_k-8y_k^2b_ja_kb_kb_i-$
$16y_ky_jb_ja_ib_ib_k-8y_ky_jb_jb_i^2a_k+4y_j^2b_ja_ib_ib_k+2y_j^2b_jb_i^2a_k+2y_j^2a_jb_i^2b_k-2y_k^2a_kb_i^3+4r_j^2b_j^2b_ia_k-$
$4r_k^2b_ja_ib_ib_k-2r_k^2b_jb_i^2a_k-4r_i^2b_ja_kb_kb_i+8r_k^2b_ja_kb_kb_i-4r_j^2b_ja_kb_kb_i+4y_j^2a_kb_k^3+4y_k^2a_jb_j^3-$
$8y_ky_ja_ib_i^3+4r_j^2b_j^2a_ib_k-2r_k^2a_jb_i^2b_k-8y_j^2a_jb_jb_ib_k+2y_j^2b_jb_k^2a_i+4y_k^2a_ib_i^3+2y_j^2a_jb_k^2b_i+4y_j^2a_ib_i^3-$
$4y_k^2b_jb_k^2a_i-4y_k^2a_jb_k^2b_i-2r_i^2b_jb_k^2a_i-2r_i^2a_jb_k^2b_i-4r_k^2a_jb_jb_ib_k+4r_k^2b_jb_k^2a_i+4r_k^2a_jb_k^2b_i-2r_j^2b_jb_k^2a_i+$
$4y_k^2a_jb_jb_ib_k-2r_j^2a_jb_k^2b_i+4y_k^2a_jb_jb_k^2+4y_k^2b_j^2a_kb_k-4y_ky_jb_j^2a_kb_k-4r_k^2a_jb_jb_k^2-4r_k^2b_j^2a_kb_k+$
$4y_j^2a_jb_jb_k^2+4y_j^2b_j^2a_kb_k-4y_ky_ja_jb_jb_k^2+8r_i^2b_j^2a_kb_k-4r_i^2b_k^2a_ib_i+8r_i^2a_jb_jb_k^2-4r_j^2a_jb_jb_k^2-4r_j^2b_j^2a_kb_k+$
$6r_j^2b_ja_kb_k^2+2r_j^2a_jb_k^3-6y_j^2b_ja_kb_k^2-2y_j^2a_jb_k^3-4y_j^2b_j^2b_ia_k-4y_j^2b_j^2a_ib_k-2r_i^2a_jb_k^3-6r_i^2b_ja_kb_k^2-$
$2r_k^2b_j^2b_ia_k-2r_k^2b_j^2a_ib_k-2y_k^2a_kb_j^3-6r_i^2b_ka_jb_j^2-2r_i^2a_kb_j^3-6y_k^2b_ka_jb_j^2+6r_k^2b_ka_jb_j^2+2r_k^2a_kb_j^3-$
$4r_i^2a_kb_kb_i^2-4y_ky_jb_k^2a_ib_i+4y_j^2a_kb_kb_i^2+4y_j^2b_k^2a_ib_i-4r_k^2b_k^2a_ib_i-4y_ky_ja_kb_kb_i^2+8r_j^2b_k^2a_ib_i+$
$8r_j^2a_kb_kb_i^2-4r_k^2a_kb_kb_i^2+4y_k^2a_kb_kb_i^2+4y_k^2b_k^2a_ib_i+6r_i^2b_ia_kb_k^2+2r_i^2a_ib_k^3-2r_j^2a_ib_k^3-6r_j^2b_ia_kb_k^2-$
$6y_j^2b_ia_kb_k^2+6r_k^2b_ka_ib_i^2-2y_j^2a_ib_k^3+2r_k^2a_kb_i^3+12y_ky_jb_ka_ib_i^2+4y_ky_ja_kb_i^3+2y_k^2b_j^2b_ia_k-2y_j^2a_kb_i^3-$
$6y_j^2b_ka_ib_i^2-6y_k^2b_ka_ib_i^2-2r_j^2a_kb_i^3-6r_j^2b_ka_ib_i^2+8r_j^2a_jb_jb_ib_k+4y_ky_jb_j^2b_ia_k+4y_ky_jb_j^2a_ib_k+$
$2y_k^2b_j^2a_ib_k-4y_ky_jb_j^2a_ib_i-4y_ky_ja_jb_jb_i^2-2r_i^2b_j^2b_ia_k-2r_i^2b_j^2a_ib_k+4y_k^2b_j^2a_ib_i-4r_i^2b_j^2a_ib_i-4r_i^2a_jb_jb_i^2+$
$4y_k^2a_jb_jb_i^2-4r_i^2a_jb_jb_ib_k+4y_j^2a_jb_jb_i^2+4y_j^2b_j^2a_ib_i+8r_k^2b_j^2a_ib_i+8r_k^2a_jb_jb_i^2-4r_j^2b_j^2a_ib_i-6y_k^2b_ia_jb_j^2-$
$2y_k^2a_ib_j^3+2r_i^2a_ib_j^3+6r_i^2b_ia_jb_j^2-4r_j^2a_jb_jb_i^2-2r_k^2a_ib_j^3-6r_k^2b_ia_jb_j^2-2r_k^2a_jb_i^3-6r_k^2b_ja_ib_i^2+2r_j^2a_jb_i^3+$
$6r_j^2b_ja_ib_i^2+4y_ky_ja_jb_i^3+12y_ky_jb_ja_ib_i^2+4y_ky_jb_jb_k^2a_i-6y_j^2b_ja_ib_i^2-6y_k^2b_ja_ib_i^2-2y_j^2a_jb_i^3-2y_k^2a_jb_i^3+$
$8y_ky_jb_ja_kb_kb_i)CS^3$
$+(-4r_j^2b_ja_kb_ka_i-4r_j^2a_ja_kb_kb_i+8y_ky_ja_jb_jb_ia_k+8y_ky_ja_jb_ja_ib_k+8y_ky_ja_ja_kb_kb_i-2r_j^2a_jb_i^2a_k-$

$2r_j^2b_ja_i^2b_k - 4r_j^2b_ja_ib_ia_k - 4r_j^2a_ja_ib_ib_k + 2y_k^2a_jb_i^2a_k + 2y_k^2b_ja_i^2b_k + 4y_ky_ja_jb_k^2a_i + 4y_k^2a_ja_ib_ib_k +$

$4y_k^2b_ja_ib_ia_k + 4r_i^2b_ja_i^2b_k + 4r_i^2a_jb_i^2a_k + 8r_i^2b_ja_ib_ia_k - 4r_k^2a_ja_ib_ib_k - 2r_k^2a_jb_i^2a_k + 4y_j^2b_ja_kb_ka_i +$

$4y_j^2a_ja_kb_kb_i - 8y_ky_jb_ja_i^2b_k - 16y_ky_jb_ja_ib_ia_k + 2y_j^2a_jb_k^2a_i - 8y_ky_ja_jb_i^2a_k - 8y_k^2b_ja_kb_ka_i -$

$8y_k^2a_ja_kb_kb_i - 16y_ky_ja_ja_ib_ib_k + 2y_j^2a_jb_i^2a_k + 4y_j^2b_ja_ib_ia_k - 2r_k^2b_ja_i^2b_k - 4r_k^2b_ja_ib_ia_k - 4r_i^2b_ja_kb_ka_i -$

$4r_i^2a_ja_kb_kb_i + 8r_k^2b_ja_kb_ka_i + 8r_k^2a_ja_kb_kb_i + 6y_j^2a_k^2b_k^2 + 6y_k^2a_j^2b_j^2 - 12y_ky_ja_i^2b_i^2 + 4r_j^2b_j^2a_ia_k +$

$4r_j^2a_j^2b_ib_k - 2r_i^2a_j^2b_ib_k + 2y_j^2b_ja_i^2b_k - 8y_j^2a_jb_jb_ia_k + 6y_k^2a_i^2b_i^2 + 2y_j^2b_ja_k^2b_i - 8y_j^2a_jb_ja_ib_k - 4y_k^2b_ja_k^2b_i +$

$6y_j^2a_i^2b_i^2 - 4y_k^2a_jb_k^2a_i - 2r_i^2b_ja_k^2b_i - 2r_i^2a_jb_k^2a_i + 4r_k^2b_ja_k^2b_i + 4r_k^2a_jb_k^2a_i + 16r_i^2a_jb_ja_kb_k - 4r_k^2a_jb_jb_ia_k -$

$4r_k^2a_jb_ja_ib_k - 2r_j^2b_ja_k^2b_i - 2r_j^2a_jb_k^2a_i + 4y_k^2a_jb_jb_ia_k + 4y_k^2a_jb_ja_ib_k + 2y_k^2b_j^2a_k^2 - 2y_ky_jb_j^2a_k^2 -$

$2y_ky_ja_j^2b_k^2 - 8y_ky_ja_jb_ja_kb_k - 8r_k^2a_jb_ja_kb_k - 2r_k^2b_j^2a_k^2 - 2r_k^2a_j^2b_k^2 + 2y_j^2b_j^2a_k^2 + 8y_j^2a_jb_ja_kb_k +$

$2y_k^2a_j^2b_k^2 + 8y_k^2a_jb_ja_kb_k + 2y_j^2a_j^2b_k^2 - 2r_j^2a_j^2b_k^2 - 2r_j^2b_j^2a_k^2 + 4r_i^2b_j^2a_k^2 + 4r_i^2a_j^2b_k^2 - 8r_j^2a_jb_ja_kb_k +$

$6r_j^2b_ja_k^2b_k + 6r_j^2a_ja_kb_k^2 - 6y_j^2a_ja_kb_k^2 - 6y_j^2b_ja_k^2b_k - 6r_i^2b_ja_k^2b_k - 6r_i^2a_ja_kb_k^2 - 4y_j^2b_j^2a_ia_k - 4y_j^2a_j^2b_ib_k -$

$6y_k^2a_ka_jb_j^2 - 6r_i^2b_ka_j^2b_j - 6r_i^2a_ka_jb_j^2 - 6y_k^2b_ka_j^2b_j + 6r_k^2b_ka_j^2b_j - 2r_i^2b_k^2a_i^2 + 6r_k^2a_ka_jb_j^2 - 8r_i^2a_kb_ka_ib_i -$

$2y_ky_jb_k^2a_i^2 - 2y_ky_ja_k^2b_i^2 - 8y_ky_ja_kb_ka_ib_i + 8y_j^2a_kb_ka_ib_i + 2y_j^2b_k^2a_i^2 + 2y_j^2a_k^2b_i^2 - 2r_k^2b_k^2a_i^2 - 2r_i^2a_k^2b_i^2 -$

$8r_k^2a_kb_ka_ib_i - 2r_k^2b_j^2a_ia_k - 2r_k^2a_j^2b_ib_k - 2r_k^2a_k^2b_i^2 + 8y_k^2a_kb_ka_ib_i + 2y_k^2b_k^2a_i^2 + 4r_j^2a_k^2b_i^2 + 16r_j^2a_kb_ka_ib_i +$

$4r_j^2b_k^2a_i^2 + 6r_i^2b_ia_k^2b_k + 2y_k^2a_k^2b_i^2 + 6r_i^2a_ia_kb_k^2 - 6r_j^2b_ia_k^2b_k - 6r_j^2a_ia_kb_k^2 - 6y_j^2b_ia_k^2b_k - 6y_j^2a_ia_kb_k^2 +$

$6r_k^2a_ka_ib_i^2 + 6r_k^2b_ka_i^2b_i + 12y_ky_ja_ka_ib_i^2 + 12y_ky_jb_ka_i^2b_i - 6y_j^2b_ka_i^2b_i - 6y_k^2b_ka_i^2b_i - 6r_j^2a_ka_ib_i^2 -$

$6y_k^2a_ka_ib_i^2 - 6r_j^2b_ka_i^2b_i - 6y_j^2a_ka_ib_i^2 + 8r_j^2a_jb_jb_ia_k + 2y_k^2b_j^2a_ia_k + 2y_k^2a_j^2b_ib_k + 4y_ky_jb_j^2a_ia_k -$

$2y_ky_jb_j^2a_i^2 + 4y_ky_ja_j^2b_ib_k - 2y_ky_ja_j^2b_i^2 - 8y_ky_ja_jb_ja_ib_i + 2y_k^2b_j^2a_i^2 + 2y_k^2a_j^2b_i^2 + 8y_k^2a_jb_ja_ib_i -$

$2r_i^2b_j^2a_i^2 + 2y_j^2b_j^2a_i^2 + 8r_i^2a_ja_ib_ib_k + 8r_j^2a_jb_ja_ib_k - 2r_i^2a_j^2b_i^2 - 8r_i^2a_jb_ja_ib_i - 4r_i^2a_jb_jb_ia_k + 2y_j^2a_j^2b_i^2 +$

$8y_j^2a_jb_ja_ib_i - 2r_i^2b_j^2a_ia_k - 4r_i^2a_jb_ja_ib_k + 4r_k^2b_j^2a_i^2 + 4r_k^2a_j^2b_i^2 + 16r_k^2a_jb_ja_ib_i - 2r_j^2b_j^2a_i^2 - 6y_k^2a_ia_jb_j^2 +$

$6r_i^2a_ia_jb_j^2 + 6r_i^2b_ia_j^2b_j - 6y_k^2b_ia_j^2b_j - 2r_j^2a_j^2b_i^2 - 8r_j^2a_jb_ja_ib_i - 6r_k^2a_ia_jb_j^2 - 6r_k^2b_ia_j^2b_j - 6r_k^2a_ja_ib_i^2 -$

$6r_k^2b_ja_i^2b_i + 6r_j^2b_ja_i^2b_i + 6r_j^2a_ja_ib_i^2 + 12y_ky_ja_ja_ib_i^2 + 12y_ky_jb_ja_i^2b_i - 6y_j^2a_ja_ib_i^2 + 4y_j^2a_ja_ib_ib_k -$

$6y_j^2b_ja_i^2b_i - 6y_k^2a_ja_ib_i^2 - 6y_k^2b_ja_i^2b_i + 8y_ky_jb_ja_kb_ka_i + 4y_ky_jb_ja_k^2b_i)S^2C^2$

$+(8y_ky_ja_ja_kb_ka_i - 4r_j^2a_ja_kb_ka_i + 4y_ky_jb_ja_k^2a_i + 2y_j^2a_ja_i^2b_k + 4y_j^2a_ja_ib_ia_k - 2r_j^2b_ja_i^2a_k - 2r_j^2a_ja_i^2b_k -$

$4r_j^2a_ja_ib_ia_k + 2y_k^2b_ja_i^2a_k + 4y_k^2a_ja_ib_ia_k + 2y_k^2a_ja_i^2b_k + 4y_ky_ja_ja_k^2b_i + 4r_i^2b_ja_i^2a_k + 4y_j^2a_i^3b_i +$

$4r_i^2a_ja_i^2b_k - 2r_k^2a_ja_i^2b_k - 4r_k^2a_ja_ib_ia_k + 4y_j^2a_ja_kb_ka_i - 8y_ky_jb_ja_i^2a_k - 8y_ky_ja_ja_i^2b_k - 16y_ky_ja_ja_ib_ia_k -$

$8y_k^2a_ja_kb_ka_i - 4r_i^2a_ja_kb_ka_i + 8r_k^2a_ja_kb_ka_i + 4y_j^2a_k^3b_k - 4r_i^2a_jb_ja_ia_k + 4y_k^2a_j^3b_j - 8y_ky_ja_i^3b_i +$

$4r_j^2a_j^2b_ia_k + 4r_j^2a_j^2a_ib_k + 2y_j^2b_ja_k^2a_i + 2y_j^2a_ja_k^2b_i + 4y_k^2a_i^3b_i - 4y_k^2b_ja_k^2a_i - 8y_j^2a_jb_ja_ia_k - 4y_k^2a_ja_k^2b_i -$

$2r_i^2b_ja_k^2a_i - 2r_i^2a_ja_k^2b_i + 4r_k^2b_ja_k^2a_i + 4r_k^2a_ja_k^2b_i + 8r_i^2a_jb_ja_k^2 - 4r_k^2a_jb_ja_ia_k - 2r_j^2b_ja_k^2a_i - 2r_i^2a_j^2b_ia_k +$

$4y_k^2a_j^2a_kb_k - 4y_ky_ja_jb_ja_k^2 - 2r_i^2a_j^2a_ib_k - 4r_k^2a_j^2a_kb_k - 4r_k^2a_jb_ja_k^2 - 4y_ky_ja_j^2a_kb_k + 4y_k^2a_jb_ja_k^2 +$

$4y_j^2a_j^2a_kb_k + 4y_j^2a_jb_ja_k^2 - 4r_j^2a_j^2a_kb_k - 4r_j^2a_jb_ja_k^2 + 8r_i^2a_j^2a_kb_k + 6r_j^2a_ja_k^2b_k + 2r_j^2b_ja_k^3 - 2y_j^2b_ja_k^3 -$

$2r_i^2b_ja_k^3 - 6r_i^2a_ja_k^2b_k - 6y_j^2a_ja_k^2b_k - 4y_j^2a_j^2b_ia_k - 4y_j^2a_j^2a_ib_k - 2y_k^2b_ka_j^3 - 6r_i^2a_ka_j^2b_j - 2r_i^2b_ka_j^3 -$

$6y_k^2a_ka_j^2b_j + 6r_k^2a_ka_j^2b_j + 2r_k^2b_ka_j^3 - 4y_ky_ja_k^2a_ib_i - 4y_ky_ja_kb_ka_i^2 + 4y_j^2a_kb_ka_i^2 + 4y_j^2a_k^2a_ib_i -$

$4r_i^2a_k^2a_ib_i - 4r_i^2a_kb_ka_i^2 - 4r_k^2a_kb_ka_i^2 - 4r_k^2a_k^2a_ib_i + 4y_k^2a_kb_ka_i^2 + 8r_j^2a_k^2a_ib_i + 8r_j^2a_kb_ka_i^2 + 4y_k^2a_k^2a_ib_i +$

$6r_i^2a_ia_k^2b_k + 2r_i^2b_ia_k^3 - 2r_j^2b_ia_k^3 - 6r_j^2a_ia_k^2b_k - 6y_j^2a_ia_k^2b_k + 6r_k^2a_ka_i^2b_i + 2r_k^2b_ka_i^3 - 2y_j^2b_ia_k^3 +$

$4y_ky_jb_ka_i^3 - 2r_k^2a_j^2b_ia_k - 2r_k^2a_j^2a_ib_k - 2y_j^2b_ka_i^3 + 12y_ky_ja_ka_i^2b_i + 4y_k^2a_jb_ja_ia_k - 6r_j^2a_ka_i^2b_i -$

$6y_k^2a_ka_i^2b_i - 2y_k^2b_ka_i^3 - 2r_j^2b_ka_i^3 - 6y_j^2a_ka_i^2b_i + 2y_k^2a_j^2b_ia_k - 2r_k^2b_ja_i^2a_k + 2y_k^2a_j^2a_ib_k + 4y_ky_ja_j^2b_ia_k -$

$4y_ky_ja_j^2a_ib_i - 4y_ky_ja_jb_ja_i^2 + 4y_k^2a_j^2a_ib_i + 4y_k^2a_jb_ja_i^2 + 4y_ky_ja_j^2a_ib_k + 8r_i^2a_ja_ib_ia_k - 4r_i^2a_j^2a_ib_i -$

$4r_i^2a_jb_ja_i^2 + 4y_j^2a_jb_ja_i^2 + 4y_j^2a_j^2a_ib_i - 2r_j^2a_ja_k^2b_i + 8r_j^2a_jb_ja_ia_k + 8r_k^2a_j^2a_ib_i + 8r_k^2a_jb_ja_i^2 - 4r_j^2a_j^2a_ib_i -$

$2y_k^2b_ia_j^3 + 2r_i^2b_ia_j^3 + 6r_i^2a_ia_j^2b_j - 6y_k^2a_ia_j^2b_j - 4r_j^2a_jb_ja_i^2 - 6r_k^2a_ia_j^2b_j - 2r_k^2b_ia_j^3 - 2r_k^2b_ja_i^3 + 2r_j^2b_ja_i^3 -$

$6r_k^2a_ja_i^2b_i + 2y_j^2b_ja_i^2a_k + 6r_j^2a_ja_i^2b_i + 4y_ky_jb_ja_i^3 - 6y_j^2a_ja_i^2b_i - 2y_j^2b_ja_i^3 + 12y_ky_ja_ja_i^2b_i - 2y_k^2b_ja_i^3 -$

$6y_k^2a_ja_i^2b_i + 8y_ky_ja_jb_ja_ia_k)SC^3$

$+(2y_j^2a_ja_i^2a_k - 2r_j^2a_ja_i^2a_k - 2y_ky_ja_i^4 + 2y_k^2a_ja_i^2a_k - 2r_j^2a_ja_k^2a_i - 2r_k^2a_ja_i^2a_k + 2y_k^2a_j^2a_ia_k - 2y_j^2a_ia_k^3 +$

$4r_i^2a_ja_i^2a_k - 8y_ky_ja_ja_i^2a_k + y_j^2a_i^4 + y_k^2a_i^4 + 4y_ky_ja_ja_k^2a_i + 2y_j^2a_ja_k^2a_i - 4y_k^2a_ja_k^2a_i - 2r_i^2a_ja_k^2a_i +$

$$4r_k^2a_ja_k^2a_i+2y_k^2a_j^2a_k^2-2y_ky_ja_j^2a_k^2-2r_k^2a_j^2a_k^2+2y_j^2a_j^2a_k^2+2r_k^2a_ka_j^3-2r_j^2a_j^2a_k^2+4r_i^2a_j^2a_k^2+2r_j^2a_ja_k^3-$$
$$2r_i^2a_ja_k^3-2y_j^2a_ja_k^3-2y_k^2a_ka_j^3-2r_i^2a_ka_j^3-2r_i^2a_k^2a_i^2+2y_j^2a_k^2a_i^2-2y_ky_ja_k^2a_i^2-2r_k^2a_k^2a_i^2-2r_i^2a_j^2a_ia_k-$$
$$4y_j^2a_j^2a_ia_k+4r_j^2a_k^2a_i^2+2y_k^2a_k^2a_i^2+2r_i^2a_ia_k^3-2r_j^2a_ia_k^3+2r_k^2a_ka_i^3+4y_ky_ja_ka_i^3-2r_j^2a_ka_i^3-2y_k^2a_ka_i^3-$$
$$2y_j^2a_ka_i^3+y_k^2a_j^4+y_j^2a_k^4-2r_k^2a_j^2a_ia_k+4r_j^2a_j^2a_ia_k-2y_ky_ja_j^2a_i^2+4y_ky_ja_j^2a_ia_k+2y_k^2a_j^2a_i^2-2r_i^2a_j^2a_i^2+$$
$$2y_j^2a_j^2a_i^2+4r_k^2a_j^2a_i^2-2y_k^2a_ia_j^3+2r_i^2a_ia_j^3-2r_k^2a_ia_j^3-2r_k^2a_ja_i^3-2y_k^2a_ja_i^3+2r_j^2a_ja_i^3+4y_ky_ja_ja_i^3-$$
$$2y_j^2a_ja_i^3-2r_j^2a_j^2a_i^2)C^4$$
$$+(r_k^4b_i^2+r_j^4b_i^2+2y_k^2y_j^2b_k^2-2r_i^2r_j^2b_k^2-2y_j^2r_j^2b_k^2-2y_j^3y_kb_k^2-2y_ky_jr_i^2b_k^2+y_k^4b_i^2+r_k^4b_j^2+r_i^4b_j^2+$$
$$2y_jr_j^2y_kb_k^2-2y_j^2r_k^2b_k^2+y_k^4b_j^2-2y_k^3y_jb_i^2-2r_k^2y_k^2b_i^2-2y_j^2r_i^2b_i^2-2y_j^2r_j^2b_i^2+y_j^4b_k^2+4y_ky_jr_i^2b_i^2+2y_k^2y_j^2b_i^2+$$
$$2y_ky_jr_k^2b_i^2-2y_j^3y_kb_i^2-2r_k^2r_j^2b_i^2-2y_k^2r_i^2b_i^2-2r_k^2r_i^2b_j^2-2r_k^2y_k^2b_j^2+2y_ky_jr_k^2b_j^2-2y_ky_jr_i^2b_j^2-2y_k^2r_j^2b_j^2-$$
$$2y_k^3y_jb_j^2+2y_k^2y_j^2b_j^2+r_i^4b_k^2-2y_j^4b_ib_k-2y_k^2r_i^2b_kb_j+2y_k^2r_j^2b_kb_j-2r_k^2r_j^2b_kb_j-2y_j^2r_i^2b_kb_j-2y_k^2y_j^2b_kb_j+$$
$$2y_j^2r_k^2b_kb_j-2r_i^4b_kb_j+2r_i^2r_j^2b_kb_j+8y_ky_jr_i^2b_kb_j-4y_jr_j^2y_kb_ib_k+r_j^4b_k^2+2y_k^2r_i^2b_ib_k-4y_ky_jr_i^2b_ib_k+$$
$$4y_j^2r_j^2b_ib_k+2r_k^2r_j^2b_ib_k-2r_k^2r_i^2b_ib_k-2y_k^2r_j^2b_ib_k-2r_j^4b_ib_k+y_j^4b_i^2-2y_k^2y_j^2b_ib_k+4y_j^3y_kb_ib_k+2r_i^2r_j^2b_ib_k+$$
$$2y_j^2r_k^2b_ib_k+2y_j^2r_i^2b_ib_k+2r_k^2r_i^2b_kb_j+2y_jr_j^2y_kb_i^2-4y_ky_jr_k^2b_ib_j-4y_ky_jr_i^2b_ib_j+2r_k^2r_j^2b_ib_j+2r_k^2r_i^2b_ib_j-$$
$$2r_k^4b_ib_j+2y_k^2r_i^2b_ib_j+2y_j^2r_i^2b_ib_j+2y_k^2r_j^2b_ib_j-2y_j^2r_k^2b_ib_j-2y_k^2y_j^2b_ib_j-2r_i^2r_j^2b_ib_j+4y_k^3y_jb_ib_j+$$
$$4r_k^2y_k^2b_ib_j-2y_k^4b_ib_j)S^2$$
$$+(-2y_k^4b_ia_j-2y_j^4b_ia_k+4y_k^2y_j^2a_kb_k-4r_i^2r_j^2a_kb_k+2r_j^4a_kb_k-4y_j^2r_j^2a_kb_k+2r_i^4a_kb_k-4y_j^3y_ka_kb_k-$$
$$4y_ky_jr_i^2a_kb_k+2y_j^4a_kb_k+4y_jr_j^2y_ka_kb_k-4y_j^2r_k^2a_kb_k+2y_j^4a_ib_i-4y_k^3y_ja_ib_i-4r_k^2y_k^2a_ib_i-4y_j^2r_i^2a_ib_i-$$
$$4y_j^2r_j^2a_ib_i+2r_k^4a_ib_i+4y_k^2y_j^2a_ib_i-4y_j^3y_ka_ib_i+2r_j^4a_ib_i-4r_k^2r_j^2a_ib_i+2y_k^4a_ib_i-4y_k^2r_i^2a_ib_i-4r_k^2r_i^2a_jb_j+$$
$$2r_k^4a_jb_j-4r_k^2y_k^2a_jb_j-4y_k^2r_j^2a_jb_j-4y_k^3y_ja_jb_j+2r_i^4a_jb_j+4y_k^2y_j^2a_jb_j+2y_k^4a_jb_j-2y_j^4a_ib_k+$$
$$2y_k^2r_j^2b_ka_j-2y_k^2r_i^2b_ka_j-2y_j^2r_i^2b_ka_j-2r_k^2r_j^2a_kb_j-2y_k^2y_j^2b_ka_j-2r_k^2r_j^2b_ka_j-2y_j^2r_i^2a_kb_j-2y_k^2r_i^2a_kb_j+$$
$$2y_k^2r_j^2a_kb_j+2y_j^2r_k^2a_kb_j+2y_j^2r_k^2b_ka_j-2y_k^2y_j^2a_kb_j-2r_i^4a_kb_j-2r_i^4b_ka_j+2r_i^2r_j^2a_kb_j+2r_i^2r_j^2b_ka_j+$$
$$2r_k^2r_i^2a_kb_j+2r_k^2r_i^2b_ka_j+8y_ky_jr_i^2a_kb_j+8y_ky_jr_i^2b_ka_j+2y_k^2r_i^2b_ia_k-4y_ky_jr_i^2a_ib_k-4y_ky_jr_i^2b_ia_k+$$
$$2y_k^2r_i^2a_ib_k+4y_j^2r_j^2b_ia_k+2r_k^2r_j^2b_ia_k-2y_k^2r_j^2a_ib_k-2y_k^2r_j^2b_ia_k-4y_jr_j^2y_ka_ib_k-2r_j^4a_ib_k-4y_jr_j^2y_kb_ia_k+$$
$$4y_j^2r_j^2a_ib_k-2r_k^2r_i^2a_ib_k-2r_k^2r_i^2b_ia_k-2r_j^4b_ia_k+2r_k^2r_j^2a_ib_k-2y_k^2y_j^2b_ia_k+2y_j^2r_k^2a_ib_k+2y_j^2r_k^2b_ia_k+$$
$$4y_j^3y_ka_ib_k+2r_i^2r_j^2a_ib_k+2r_i^2r_j^2b_ia_k+4y_j^3y_kb_ia_k-2y_k^2y_j^2a_ib_k+2y_j^2r_i^2a_ib_k+2y_j^2r_i^2b_ia_k+4y_jr_j^2y_ka_ib_i+$$
$$8y_ky_jr_i^2a_ib_i+4y_ky_jr_k^2a_ib_i-2r_k^4b_ia_j+4y_ky_jr_k^2a_jb_j-4y_ky_jr_i^2a_jb_j-4y_ky_jr_k^2b_ia_j-4y_ky_jr_i^2b_ia_j+$$
$$2r_k^2r_i^2b_ia_j-2r_k^4a_ib_j+2r_k^2r_j^2b_ia_j+2r_k^2r_i^2a_ib_j-4y_ky_jr_i^2a_ib_j-4y_ky_jr_k^2a_ib_j+2y_k^2r_i^2a_ib_j+2y_k^2r_i^2b_ia_j+$$
$$2y_j^2r_i^2a_ib_j+2y_j^2r_i^2b_ia_j+2y_k^2r_j^2a_ib_j+2y_k^2r_j^2b_ia_j+2r_k^2r_j^2a_ib_j+4r_k^2y_k^2a_ib_j-2y_j^2r_k^2a_ib_j-2y_k^2y_j^2a_ib_j+$$
$$4y_k^3y_jb_ia_j-2r_i^2r_j^2a_ib_j+4y_k^3y_ja_ib_j-2r_i^2r_j^2b_ia_j-2y_k^2y_j^2b_ia_j-2y_j^2r_k^2b_ia_j+4r_k^2y_k^2b_ia_j-2y_k^4a_ib_j)CS$$
$$+(y_j^4a_i^2-2y_k^2r_i^2a_i^2+r_k^4a_i^2+2y_k^2y_j^2a_k^2-2r_i^2r_j^2a_k^2-2y_j^2r_j^2a_k^2-2y_j^3y_ka_k^2+r_j^4a_i^2+r_k^4a_j^2-2y_ky_jr_i^2a_k^2+$$
$$r_i^4a_j^2-2y_j^2r_k^2a_k^2+y_k^4a_j^2+y_j^4a_k^2+8y_ky_jr_i^2a_ka_j-2y_k^3y_ja_i^2+2y_jr_j^2y_ka_i^2-2r_k^2y_k^2a_i^2-2y_j^2r_i^2a_i^2-$$
$$2y_j^2r_j^2a_i^2+4y_ky_jr_i^2a_i^2+2y_k^2y_j^2a_i^2+2y_ky_jr_k^2a_i^2-2y_j^3y_ka_i^2-2r_k^2r_j^2a_i^2-2r_k^2r_i^2a_j^2+2y_ky_jr_k^2a_j^2-$$
$$2r_k^2y_k^2a_j^2-2y_ky_jr_i^2a_j^2-2y_k^2r_j^2a_j^2-2y_k^3y_ja_j^2+2y_k^2y_j^2a_j^2+y_k^4a_i^2+r_i^4a_k^2+2r_k^2r_i^2a_ka_j-2y_j^4a_ia_k+$$
$$2y_k^2r_j^2a_ka_j-2y_k^2r_i^2a_ka_j-2y_j^2r_i^2a_ka_j-2r_k^2r_j^2a_ka_j-2y_k^2y_j^2a_ka_j+2y_j^2r_k^2a_ka_j+r_j^4a_k^2-2r_i^4a_ka_j+$$
$$2r_i^2r_j^2a_ka_j+4y_j^2r_j^2a_ia_k+2y_k^2r_i^2a_ia_k-4y_jr_j^2y_ka_ia_k-2r_k^2r_i^2a_ia_k-4y_ky_jr_i^2a_ia_k-2r_j^4a_ia_k-$$
$$2y_k^2r_j^2a_ia_k+2r_k^2r_j^2a_ia_k+2y_j^2r_k^2a_ia_k+4y_j^3y_ka_ia_k-2y_k^2y_j^2a_ia_k+2r_i^2r_j^2a_ia_k+2y_j^2r_i^2a_ia_k-2r_k^4a_ia_j-$$
$$4y_ky_jr_k^2a_ia_j-4y_ky_jr_i^2a_ia_j+2y_jr_j^2y_ka_k^2+2r_k^2r_i^2a_ia_j+2y_k^2r_i^2a_ia_j+2y_k^2r_j^2a_ia_j+2r_k^2r_j^2a_ia_j+$$
$$4r_k^2y_k^2a_ia_j-2y_j^2r_k^2a_ia_j-2y_k^2y_j^2a_ia_j-2r_i^2r_j^2a_ia_j+2y_j^2r_i^2a_ia_j-2y_k^4a_ia_j+4y_k^3y_ja_ia_j)C^2$$
$$-2y_j^2r_i^2r_k^2-2y_j^2y_k^2r_j^2-2r_i^4y_ky_j-2y_j^3r_i^2y_k-2r_i^2y_k^3y_j-2r_i^2y_k^2r_j^2+2y_j^3r_k^2y_k+4r_i^2y_k^2y_j^2+r_i^4y_k^2+$$
$$y_j^2r_i^4+y_j^2y_k^4-2y_j^3y_k^3+y_j^2r_k^4+y_j^4y_k^2+r_j^4y_k^2+2r_i^2y_ky_jr_k^2+2y_jr_i^2r_j^2y_k-2y_jr_k^2r_j^2y_k+2y_jy_k^3r_j^2-$$
$$2y_j^2y_k^2r_k^2$$

Let e_{qr} be the coefficient of $C^q S^r$ in the above polynomial. Using this notation, we can write the above equation in the form:

$$\sum_{\substack{0 \leq q,r \leq 6 \\ q+r \leq 6}} e_{qr} C^q S^r = 0.$$

The following statement is valid for all coefficient $e_{qr} \neq 0$:

$$q \equiv 0 \bmod 2 \iff r \equiv 0 \bmod 2.$$

Hence we can write the above equation in the form

$$\sum_{\substack{q,r \equiv 0 \bmod 2 \\ 0 \leq q+r \leq 6}} e_{qr} C^q S^r = \sum_{\substack{q,r \equiv 1 \bmod 2 \\ 0 \leq q+r \leq 6}} -e_{qr} C^q S^r.$$

We denote the polynomial on the left side by $P_e(C,S)$ and the polynomial on the right side by $P_o(C,S)$. The fact $P_e(C,S) = P_o(C,S)$ implies $(P_e(C,S))^2 = (P_o(C,S))^2$. Note that all solutions of the first equation are also solutions of the second equation. Now we solve this second equation. Consider the equation

$$0 = (P_e(C,S))^2 - (P_o(C,S))^2 := \sum_{\substack{0 \leq q,r \leq 12 \\ q+r \leq 12}} e'_{qr} C^q S^r.$$

The coefficients e'_{qr} with $q \equiv 1 \bmod 2$ or $r \equiv 1 \bmod 2$ are 0. We replace S^2 by $1 - C^2$ (note that $\sin(\alpha(t))^2 = 1 - \cos(\alpha(t))^2$) and obtain a polynomial $P(C) = \sum_{q=0}^{6} \bar{e}_q C^{2q}$ in one variable. Replacing C^2 by $\bar{c}$ gives $P(\bar{c}) = \sum_{q=0}^{6} \bar{e}_q \bar{c}^q$. Then, we compute the at most 6 real-valued solutions of the equation $P(\bar{c}) = 0$. For each positive solution s we compute the (at most two) square roots of s. For each square root r with $-1 \leq r \leq 1$ we determine the (at most 2) parameters t with $r = \cos(\alpha(t))$. Finally, we determine whether or not these values of t are solutions of the original equation.

To summarize, we have shown that the number of parameters t such that the three circles c_i^t, c_j^t and c_k^t intersect in one point at time t, is a constant (≤ 24). Note that usually, this number is much smaller than 24.