Appendix A. Source Code

```
static Arrangement atinstant (Polyhedron ph, Kernel::FT z) {
  // Build the AABB Tree
  Tree tree(faces(ph).first, faces(ph).second, ph);
  // Define the intersection plane
  Plane pl(Point(0, 0, z), Vector(0, 0, 1));
  list<Plane_intersection> pis;
  // Compute all intersections
  tree.all_intersections(pl, std::back_inserter(pis));
  Arrangement arr;
  // Iterate over all intersection segments
  for (list<Plane_intersection>::iterator it = pis.begin(); it != pis.end(); ++it) {
    Segment *s = boost::get<Segment>(&((*it)->first));
    if (s == NULL || s->is_degenerate()) continue;
    // Push the segments into the Arrangement
    CGAL::insert(arr, Segment_2(Point_2(s->source().x(), s->source().y()),
                                Point_2(s->target().x(), s->target().y())));
  }

  return arr;
}
```

Listing 2: Implementation of atinstant

```
MBOOL mpointinside(Polyhedron ph, vector<Segment> segs) {
  MBOOL mbool;
  // Create the AABB Tree
  Tree tree(faces(ph).first, faces(ph).second, ph);
  // Define the point-inside-polyhedron tester
  Point_inside inside_tester(tree);

  // Iterate over all segments
  for (unsigned int i = 0; i < segs.size(); i++) {
    list<Segment_intersection> is;
    // Determine the intersection points between the segment and the polyhedra surfaces
    tree.all_intersections(segs[i], std::back_inserter(is));
    set<Kernel::FT> events;
    for (list<Segment_intersection>::iterator it = is.begin(); it != is.end(); ++it) {
      Segment_intersection si = *it;
      // Put the intersection points into an ordered set
      if (Point *p = boost::get<Point>(&(si->first)))
        events.insert(p->z());
    }
    // Include the z coordinate of the segment's end point
    events.insert(segs[i].target().z());

    Kernel::FT prev = segs[i].source().z();
    // Check, if the moving point starts inside or outside the polyhedra/movingregion
    bool inside = inside_tester(segs[i].source()) == CGAL::ON_BOUNDED_SIDE;
    for (set<Kernel::FT>::iterator it = events.begin(); it != events.end(); it++) {
      Kernel::FT start = prev, end = *it;
      mbool.append(start, end, inside);
      // The next unit switches from inside to outside or vice versa
      inside = !inside;
      prev = *it;
    }
  }

  return mbool;
}
```

Listing 3: Implementation of inside

```
Polyhedron union (Polyhedron& p1, Polyhedron& p2) {
   // Convert the polyhedra into Nef polyhedra
   Nef_polyhedron np1(p1), np2(p2);

   // Perform the set operation "union"
   Nef_polyhedron np12 = np1 + np2;

   // Convert the result back to normal polyhedra
   Polyhedron p12;
   np12.convert_to_polyhedron(p12);

   return p12;
}
```

Listing 4: Implementation of Union

```
vector<Polygon_with_holes_2> traversedarea (Polyhedron p) {
  vector<Polygon_2> ii;
  vector<Polygon_with_holes_2> oi;
```

```
  // Iterate over the polyhedra's faces
  for (Facet_iterator s = p.facets_begin();
         s != polyhedron.facets_end(); ++s) {
    Halfedge_facet_circulator h = s->facet_begin(), he(h);

    Polygon_2 polygon;
    // Iterate over the surface vertices
    do {
      Point3d p1 = h->vertex()->point();
      polygon.insert(polygon.vertices_end(),
                                  Point_2(p1.x(), p1.y()));
    } while (++h != he);
    if (polygon.orientation() == CGAL::NEGATIVE)
      polygon.reverse_orientation();
    // Store the projected polygon
    ii.push_back(polygon);
  }
  // Perform a union over all projected polygons
  CGAL::join(ii.begin(), ii.end(), std::back_inserter(oi));

  return oi;
}
```

Listing 5: Implementation of Traversedarea

```
PMRegion p1 = PMRegion::fromRList(file2rlist("pmreg1"));
PMRegion p2 = PMRegion::fromRList(file2rlist("pmreg2"));

PMRegion p12 = p1 + p2;
cout << p12.atinstant(2000000).ToString() << endl;
cout << p12.atinstant(3000000).ToString() << endl;
```

Listing 6: Union and Atinstant

```
PMRegion p1 = PMRegion::fromRList(file2rlist("pmreg1"));
PMRegion p2 = PMRegion::fromRList(file2rlist("pmreg2"));
RList mpoint = file2rlist("mpoint");

PMRegion p12 = p1 + p2;
cout << p12.mpointinside(mpoint).rl.ToString() << endl;
```

Listing 7: Union and inside

Appendix B. Comparison results

Excerpt of 171 results out of 1500 experiments.

	Union		Intersection		Difference	
Edges	Surfaces	Msegs	Surfaces	Msegs	Surfaces	Msegs
3	36	173	12	53	28	137
4	52	320	20	102	32	159
5	64	404	24	152	48	370
6	80	616	32	253	56	416
7	128	1873	64	789	104	1614
8	108	1115	44	460	80	795
9	148	1998	84	1170	116	1445
10	180	2591	92	1244	136	1655
20	384	11063	224	6151	296	8307

	Union		Intersection		Difference	
Edges	Surfaces	Msegs	Surfaces	Msegs	Surfaces	Msegs
30	872	52924	648	36349	752	37818
40	1118	94575	790	52925	950	74549
50	1570	156312	1178	104497	1384	120031
60	1904	204056	1516	136561	1690	193361
70	2290	320548	1738	220234	2020	225593
80	3444	624845	2900	433731	3156	486251
90	3378	641316	2714	403962	3056	459003
100	3934	834937	3102	474953	3496	591302
110	4636	1134325	3844	666806	4248	739733
120	5382	1374130	4550	955699	4974	995553
130	6088	1534595	5264	1035303	5716	1355041
140	6338	1723105	5294	1066225	5812	1645757
150	7246	2136067	6050	1329170	6600	1728401
160	8724	2916373	7452	1720965	8102	2607695
170	9358	3244573	8170	2132675	8774	3258062
180	10270	4173900	8922	2370450	9512	2999232
190	10096	3920560	8736	2582716	9394	3337221
200	10646	4160542	9166	2723629	9930	3461964
210	11090	4645132	9338	2741731	10210	3699898
220	11854	5131776	10310	3342738	11094	4251991
230	14534	6958515	12918	4343332	13790	5794770
240	13522	6685733	11794	3833048	12706	5312496
250	17834	9929729	16038	6274667	17034	8176708
260	16888	9051495	14760	5866261	15768	7478533
270	16968	9221903	14944	5962029	15954	7652273
280	16906	9398076	14670	5937942	15846	7743203
290	19500	11704008	17504	7679939	18506	10139911
300	20950	13666337	18810	7889873	19956	11286724
310	22056	14403176	19708	8783888	20920	11596127
320	21816	14376503	19440	8950900	20618	11558344
330	22896	15144704	20508	9937500	21800	12674696
340	25494	18098499	23026	11572565	24272	14927378
350	24474	17628916	21694	10897825	23272	14426866
360	26606	19853569	23938	12688064	25218	16233260
370	28564	22093625	25860	14547931	27206	18424915
380	27872	21711450	25288	14515806	26474	17736226
390	29204	23441049	26536	15046492	27928	19966855
400	29540	24452072	26644	14860393	28100	20257758
410	32314	26855265	29234	17939257	30836	23225357
420	30148	25054376	26912	16324104	28570	20718133
430	34078	29395466	30846	19323819	32408	24575042
440	36518	33016581	33282	21311360	34932	28171755
450	34968	31903677	31408	20108630	33230	26562427
460	38806	37478424	35394	23024682	37122	30870207
470	37438	35883042	33874	22664238	35712	29746512
480	40006	38312754	36574	26179680	38116	33333168
490	39696	39272774	36248	25106447	37914	32766028
500	42580	43568499	39128	28704312	40996	36361609