An Approximate Computation of the Dominant Region Diagram for the Real-Time Analysis of Group Behaviors

Ryota Nakanishi, Junya Maeno, Kazuhito Murakami, and Tadashi Naruse

Graduate School of Information Science and Technology,
Aichi Prefectural University, Nagakute-cho, Aichi, 480-1198 Japan

Abstract. This paper describes a method for a real-time calculation of a dominant region diagram (simply, a dominant region). The dominant region is proposed to analyze the features of group behaviors. It draws spheres of influence and is used to analyze a teamwork in the team sports such as soccer and handball. In RoboCup Soccer, particularly in small size league(SSL), the dominant region takes an important role to analyze the current situation in the game, and it is useful for evaluating the suitability of the current strategy. Another advantage of its real-time calculation is that it makes possible to predict a success or failure of passing. To let it work in a real environment, a real-time calculation of the dominant region is necessary. However, it takes 10 to 40 seconds to calculate the dominant region of the SSL's field by using the algorithm proposed in [3]. Therefore, this paper proposes a real-time calculation algorithm of the dominant region. The proposing algorithm compute an approximate dominant region. The basic idea is (1) to make a reachable polygonal region for each time $t_1, t_2, ..., t_n$, and (2) to synthesize it incrementally. Experimental result shows that this algorithm achieves about 1/1000 times shorter in computation time and 90% or more approximate accuracy compared with the algorithm proposed in [3]. Moreover, this technique can predict the success or failure of passing in 95% accuracy.

1 Introduction

In RoboCup Soccer, the cooperative plays such as passing and shooting are the important basic skills. Particularly in RoboCup Small Size League (SSL), high level cooperative plays are developed so far. Since the strategies based on them are growing year after year [1], it is important to analyze the actions of opponent team in real time and then to change team's strategy dynamically in order to overcome the opponent. For such analysis, the *voronoi diagram* [2] and the *dominant region diagram* [3] are useful. They are used to analyze the sphere of influence. The voronoi diagram divides the region based on the distance between robots, while the dominant region diagram divides the region based on the arrival time of robots. It is considered that the dominant region diagram shows an adequate sphere of influence under the dynamically changing environment such as a soccer game.

J. Baltes et al. (Eds.): RoboCup 2009, LNAI 5949, pp. 228–239, 2010.

In the SSL, the dominant region diagram has been used for arranging teammate robots to perform the cooperative play such as passing and shooting [4][5]. However, the existing algorithm takes much time to compute the dominant region diagram, the use of the algorithm is restricted to the case that the computation time can keep, i.e. a typical case is a restart of play. If the dominant region diagram can be computed in real time, we can apply it any time.

In this paper, we propose an algorithm that computes the dominant region diagram in real time. In the SSL, it is required to compute the dominant region diagram within 5 *msec*. So, we put this time to be our present goal. Proposed algorithm is an approximate computation of the dominant region diagram so that we discuss the computation time and the approximation accuracy through the experiment. It is shown that proposed algorithm achieves 1/1000 times shorter in computing time compared with the algorithm proposed in literature [3] and over 90% accuracy. Moreover, 5 *msec* computation time can be possible under the parallel computers. We also show that the dominant region diagram is useful for the prediction of success for passing.

2 Dominant Region Diagram

Our main purpose is to discuss a real-time computation of the dominant region diagram. At first, we briefly describe it and compare it with the voronoi diagram in this section.

2.1 Computation of Dominant Region

A dominant region of an agent[1] is defined as "a region where the agent can reach faster than any other agents". A dominant region diagram, simply a dominant region, shows the dominant region of every agent [3]. The dominant region diagram is one of the generalized voronoi diagrams. Though the dominant region diagram is an n dimensional diagram in general, we discuss a two dimensional diagram here because we consider a soccer field.

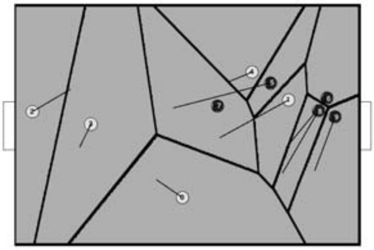
The dominant region is calculated as follows. Assume that an agent i is at the point $\mathbf{P}^i(= (P_x^i, P_y^i))$ and is moving at a velocity $\mathbf{v}^i(= (v_x^i, v_y^i))$. Assume also that the agent can move to any direction and its maximum acceleration is $\mathbf{a}_\theta^i(= (a_{\theta x}^i, a_{\theta y}^i))$ for a θ-direction. The position that the agent will be after t seconds is given by[2],

$$\begin{pmatrix} x_\theta^i \\ y_\theta^i \end{pmatrix} = \begin{pmatrix} \frac{1}{2}a_{\theta x}^i t^2 + v_x^i t + P_x^i \\ \frac{1}{2}a_{\theta y}^i t^2 + v_y^i t + P_y^i \end{pmatrix}. \tag{1}$$

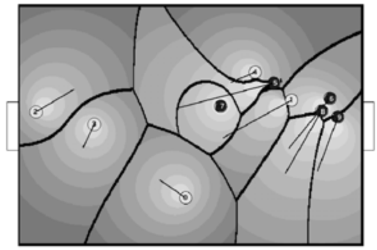
For given t, the set of above points makes a closed curve with respect to θ. Conversely, for given point $\mathbf{x} = (x, y)$, we can compute the time which each

[1] We call a considering object (such as a player) an agent.

[2] These equation do not consider the maximum velocity of the agent. If the maximum velocity must be considered, the equations should be replaced to the non-accelerated motion equations after reaching the maximum velocity.



(a)Voronoi diagram



(b)Dominant region diagram

Fig. 1. Voronoi diagram vs. dominant region diagram

agent takes[3]. Therefore, for each point $\mathbf{x}$ in a region (or a soccer field), we can get the dominant region by computing the following equation,

$$I_x = \underset{i}{\text{argmin}}\{t^i(\mathbf{x})\}, \tag{2}$$

where, I_x is an agent's number which comes at first to the point $\mathbf{x}$.

Preliminary experiment using the algorithm proposed in [3] shows that the computation time takes 10 to 40 seconds when the soccer field is digitized by 610×420 grid points.

2.2 Comparisons between Voronoi Diagram and Dominant Region Diagram

The voronoi diagram divides the region based on the distance between agents while the dominant region divides the region based on the arrival time of agents. The voronoi diagram is used to analyze the spheres of influence and is shown that it is useful in RoboCup Soccer Simulation [6]. However, we think that arrival time should be considered when analyzing the sphere of influence in RoboCup Soccer, since the robots are moving in various speeds. Figure 1 shows an example of the voronoi and dominant region diagram of a scene in the game. In the figure above, small circles are agents and the straight line originated from each agent is a current velocity vector of the agent. Note that the shape of the border lines of the regions are quite different between two diagrams. The dominant region becomes a powerful tool when deciding strategy/tactics under the consideration of the motion model of the robots.

3 Approximated Dominant Region

To achieve a real-time computation of the dominant region, where the real time means a few milliseconds here, we propose an *approximated dominant region*. It

[3] If more than one arrival time are obtained at point $\mathbf{x}$ for the agent i, the minimal arrival time is taken.

(a)Acceleration vectors (b)Reachable polygonal region

Fig. 2. Acceleration vectors and reachable polygonal region

can be obtained as a union of *reachable polygonal regions*. A reachable polygonal region is a polygon which is uniquely calculated when the motion model and time are given.

3.1 Motion Model of Robots

In this paper, we define a motion model as a set of maximum acceleration vectors of a robot. Figure 2(a) shows an example of a motion model. Each maximum acceleration vector shows that the robot can move to that direction with the given maximum acceleration. This is an example of an omni-directional robot. Eight vectors are given. The number of vectors depends on the accuracy of obtaining the dominant region.

3.2 Computation of Reachable Polygonal Region

The reachable polygonal region is a region that is included in the polygon made by connecting the points, where each point is given as a point that an agent arrives at after t seconds when it moves toward the given direction of maximum acceleration vector in maximum acceleration. Eq. (1) is used to compute the point. Figure 2(b) shows an example of reachable polygonal region (shaded area) after 1 second passed when the acceleration vectors of figure 2(a) is given. We assume the reachable polygonal region is convex[4]. The reachable polygonal region is calculated by the following algorithm.

[Reachable polygonal region]
Step 1 Give a motion model of each agent (figure 2(a)).
Step 2 Give time t. Calculate each arrival point (x^i_θ, y^i_θ) according to the equation (1) using the corresponding maximum acceleration vector in Step1.
Step 3 Connect points calculated in Step2 (figure 2(b)).

3.3 Calculation of Approximated Dominant Region

The approximated dominant region is obtained from the reachable polygonal regions for every agent. When some of reachable polygonal regions are overlapped,

[4] If it is concave, we consider a convex hull of it.

(a)Overlapped reachable polygonal regions (b)Divided reachable polygonal regions

Fig. 3. Division of two overlapped Reachable polygonal regions

we have to decide which agent, the point belongs to the overlapped region. Figure 3(a) shows two overlapped reachable polygonal regions (A_1, A_2) of two agents. In this case, it is natural to divide overlapped region into two by the line connecting the points of intersection of two polygons. Figure 3(b) shows a result for the reachable polygonal regions. However, since the number of points of intersection between two polygons (with n vertices) varies from 0 to $2n$, we have to clarify the method of division for each case. Moreover, we need to consider the method of division when many reachable polygonal regions are overlapped. We describe these methods in the following algorithm. We call this an algorithm of the approximated dominant region.

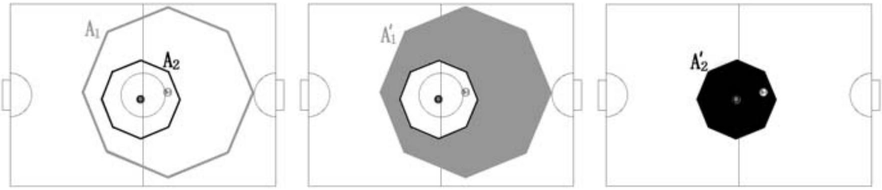
[**Approximated dominant region**]

Step 1 For given time t, make a reachable polygonal region of each agent. (Figure 2(b)).

Step 2 For two reachable polygonal regions, if they are overlapped, divide the overlapped region in the following way. Generally, a number of points of intersection between two polygons with n vertices varies from 0 to $2n$. If a vertex of one polygon is on the other polygon, move the vertex infinitesimally to the direction where the number of points of intersection does not increase. (There is no side effect with respect to this movement.) Therefore, the number of points of intersection is even. We show the way to divide in case of 0, 2 and $2k$ intersections.

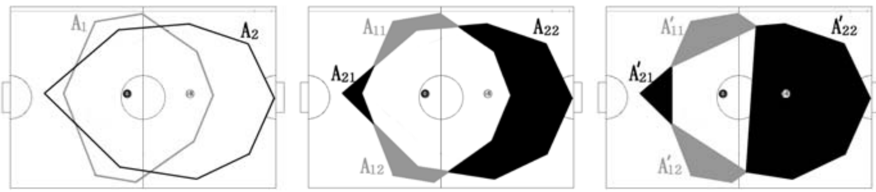
1. No points of intersection: There are two cases.
   (a) Disjoint: As two reachable polygonal regions are disjoint, there is no need to divide.
   (b) Properly included: One includes the other. Figure 4(a) shows an example($A_1 \supset A_2$). In this case, $A_1 - A_2$ is a dominant region of agent 1 (Fig. 4(b)) and A_2 is a dominant region in agent 2 (Fig. 4(c))[5].
2. 2 points of intersection: The overlapped regions of A_1 and A_2 is divided into two region by the line connecting the points of intersection between two polygons to create dominant regions A'_1 and A'_2 (Figure 3).

[5] This is not correct definition, but we adopt this to perform the real time computation.



(a)Overlapped reachable polygonal regions(A_1 and A_2) (b)A_1's divided region(gray area) (c)A_2's divided region(black area)

Fig. 4. Division of overlapped reachable polygonal regions(one-contains - other case)



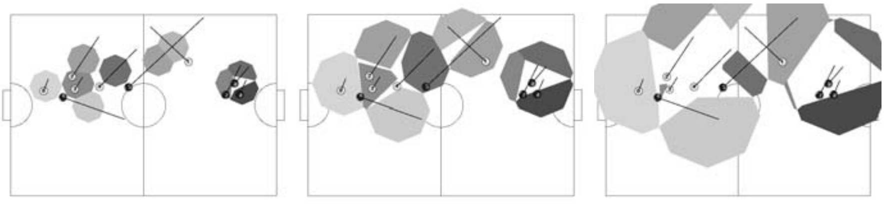
(a)Overlapped reachable polygonal regions (b)Symmetric difference (c)Divided reachable polygonal regions

Fig. 5. Division of overlapped reachable polygonal regions(4 intersecting case)

3. $2k$ points of intersection: Let A_1 and A_2 be two reachable polygonal regions and I be a set of points of intersection between the polygons of A_1 and A_2. Then, compute $A_1 - A_2$ and $A_2 - A_1$. Figure 5(a) shows an example. In this example, there are 4 points of intersection. Figure 5(b) shows a difference between two regions, where $A_1 - A_2 (= \{A_{11}, A_{12}\})$ is shaded in grey and $A_2 - A_1 (= \{A_{21}, A_{22}\})$ is shaded in black. Make convex hulls of subregions. Figure 5(c) shows the result $(A_{11}^{'}, A_{12}^{'}, A_{21}^{'}, A_{22}^{'})$. Thus, we have *partial dominant regions* ($A_1' = A_{11}^{'} \cup A_{12}^{'}$ and $A_2' = A_{21}^{'} \cup A_{22}^{'}$) of the agent 1 and 2. A white area in the overlapped region in figure 5(c) doesn't belong to either of two partial dominant regions.

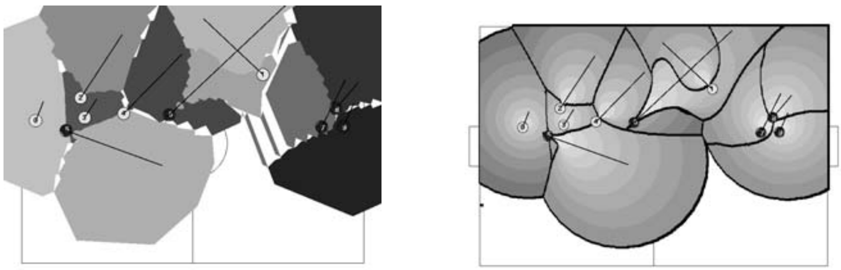
Step 3 If n reachable polygonal regions $(A_1, A_2, \cdots, A_n)$ are overlapped, we process as follows. First, for A_1 and A_2, we take partial dominant regions A_1' and A_2' by using the procedure in step 2. Replace A_1 and A_2 with A_1' and A_2'. Then, for A_1 and A_3, and A_2 and A_3, do the same computation. Repeat this until A_n is computed. As a result, we get new reachable polygonal regions $(A_1, A_2, \cdots, A_n)$ where any two A_is are disjoint. These are the partial dominant regions of agents at given time t. Figure 6 shows three examples of partial dominant regions of 10 agents at time t= 0.5, 0.7 and 0.9 seconds.

Step 4 Synthesize the partial dominant regions incrementally. For given times $t_1, t_2, \cdots, t_n$ $(t_1 < t_2 \cdots < t_n)$, compute the partial dominant regions. Let them be $B_1, B_2, \cdots B_n$. Then, compute $B_1 + (B_2 - B_1) + \cdots + (B_n - \sum_{i=1}^{n-1} B_i)$.



(a)$t = 0.5[sec]$ (b)$t = 0.7[sec]$ (c)$t = 0.9[sec]$

Fig. 6. Synthesis of reachable polygonal regions



(a)Approximated dominant region diagram (b)Precise dominant region diagram

Fig. 7. Approximated dominant region diagram vs. dominant region diagram

This makes an approximated dominant region diagram. Figure 7(a) shows an example constructed from the examples shown in figure 6, but using 10 partial dominant regions computed by every 0.1 seconds.

3.4 Feature of Proposed Algorithm

The proposed algorithm computes the approximated dominant region and it makes the great reduction of the computation time. However, there are some small regions that are not in the approximated dominant region, but in some of the dominant regions of agents. We allowed to remain such small regions because we strongly pursue the real-time computation. We evaluate the negative effect of such small regions through the evaluation experiments.

4 Experimental Evaluation for Algorithm of Approximated Dominant Region

In this section, we apply our algorithm to the SSL and evaluate how well it works.

Table 1. Computation time and accuracy of proposed algorithm

acc. vectors	arrival-time steps	computation time[$msec$]	accuracy[%]
8	10	17.5	91.8
8	20	33.8	92.9
16	10	31.4	95.3
64	200	2.4×10^3	99.9
2048	400	5.0×10^5	100

Table 2. Computation time on various computers (parameters: max acc. vectors: 8, arrival-time steps: 10)

	CPU	proposed method(A)	existing method(B)	rate(B/A)
1	3.16GHz	17.5[msec]	13.3[sec]	760
2	3.2GHz	19.3[msec]	24.2[sec]	1254
3	2.2GHz	38.2[msec]	40.5[sec]	1060
4	2.2GHz	38.3[msec]	40.4[sec]	1055

4.1 Experiment

In the SSL, since the ball moves very fast, the standard processing cycle is 60 processings per second. One processing includes an image processing, a decision making, action planning, command sending and so on. Therefore, the allowed time for the computation of the dominant region is at most 5 milli-seconds[6]. Our purpose is to make the computation of the approximated dominant region within 5 msec.

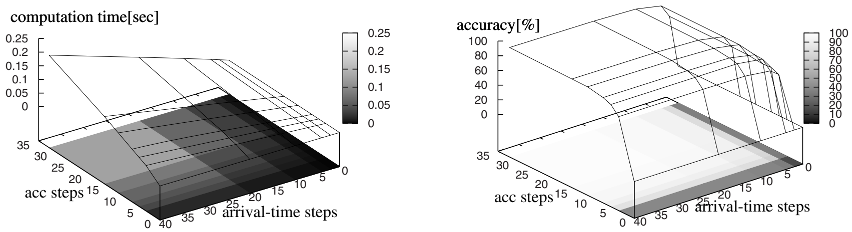
4.2 Experimental Result

We digitize the SSL's field into 610 × 420 grid points (1 grid represents the area of about 1 cm^2) and, for 10 agents (5 teammates and 5 opponents), compute the approximated dominant region that can arrive within 1 second. The reason why we set arrival time to 1 sec is that almost all of the whole field can be covered by the dominant region as shown in Fig. 7. We measure the computation time and the accuracy of the approximated dominant region. We define the accuracy by the following equation,

$$\texttt{Accuracy}[\%] = \frac{\text{Total grid points that } \mathrm{I}_\mathrm{x}^\mathrm{d} \text{ and } \mathrm{I}_\mathrm{x}^\mathrm{a} \text{ coincide}}{\text{All grid points}} * 100 \qquad (3)$$

where, I_x^d and I_x^a are given by Eq. (2) for the precise dominant region and the approximated dominant region, respectively.

[6] This time constraint is sufficient when our algorithm will be applied for the other leagues in RoboCup soccer and human soccer.



(a)Computation time vs. acceleration vector and arrival-time step pairs

(b)Accuracy vs. acceleration vector and arrival-time step pairs

Fig. 8. Computation time and accuracy

We used the computer with Xeon X5460 as CPU, 8GB main memory and FreeBSD operating system for this experiment. We measured the computation time by running the program in a single thread.

The computation time and the accuracy of the approximated dominant region depend on the number of maximum acceleration vectors and the number of partial dominant regions (i.e. the number of time-divisions). We measured the computation time and the accuracy for the various values by using two parameters above. Figure 8 shows the results of the measure. Table 1 shows the computation time and the accuracy for some typical values of the parameters. The resulting approximated dominant region of the first row of the Table 1 is shown in Figure 7(a).

In comparison with Fig. 7(b), it is considered that Fig. 7(a) is a good approximation of the dominant region diagram. The accuracy is ranging from 91.8% to 100% from Table 1. These numbers show that our algorithm gives a good approximated dominant region.

Table 2 shows the computation time of the approximated dominant region diagram measured on the various computers. (The parameters on this experiment are fixed as the number of maximum acceleration vectors takes 8 and the number of time-divisions takes 10.) From the table, it is shown that the computation time can be reduced about 1/1000 times shorter compared with the algorithm shown in [3] and the accuracy keeps a little more over 90%. In addition, it is shown that, from Fig. 8(a), the computation time increases in proportion to the number of maximum acceleration vectors and the number of time-divisions, and from Fig. 8(b), the accuracy goes up rapidly to be 90% according to the increase of the number of maximum acceleration vectors and/or time-divisions, and then still slightly increases.

4.3 Discussion

From Table 1, the approximated dominant region with parameters of 8 maximum acceleration vectors and 10 time-divisions achieves the accuracy of 92%. However, its computation time takes 17.5 *msec*. It is a little bit far from our

Table 3. Computation time necessary to make and to synthesize reachable polygonal regions

acc. vectors	8	16	24	32
average computation time [$msec$]	1.71	3.10	4.37	5.46
standard deviation[$msec$]	0.016	0.017	0.014	0.014

goal, which is computation time to be 5 $msec$. To achieve this goal, we discuss a parallel computation here. Another issue whether the accuracy of 92% is enough in our purpose will be discussed in the next section.

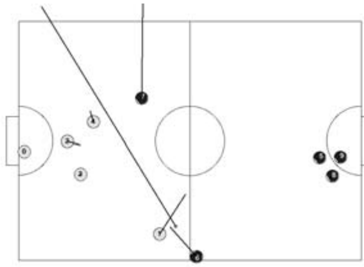
In our algorithm, we create an approximated dominant region by synthesizing the partial dominant regions incrementally. Each partial dominant region can calculate independently and its computation time is almost equal for all the partial regions. The latter is supported by the fact that the computation time doubles when the partial dominant regions double. (See Fig. 8(a).) And also Table 3 shows an average time to compute a partial dominant region. For synthesis of 10 partial dominant regions, it takes $0.5msec$ on average. Therefore, it is expected in parallel computation that the computation time will be about $1.71 + 0.5 + \alpha$ $msec$ when 10 partial dominant regions are synthesized, where α is an overhead of the parallel computation and is considered as a constant. In the multi-core parallel computer, the α is small enough. We consider that it is possible to make the computation time within 5 $msec$, which is our goal.

5 Prediction of Success for Passing

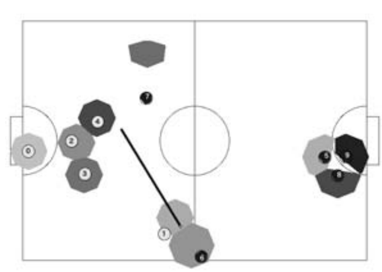
One application of the dominant region is the prediction of success for passing. In this section, we introduce a dominant region of a ball. The approximated dominant region takes a significant role to predict the success for passing. If we can predict the success of passing accurately in real time, we can choose a defense or an offense strategy appropriately.

5.1 Approximated Dominant Region of Ball

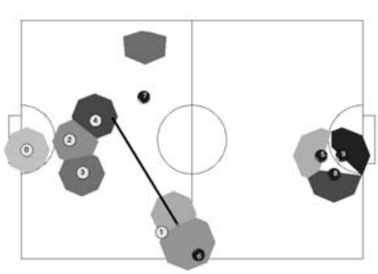
We consider a dominant region of a ball. The motion of the kicked ball on the SSL's field can be considered as a uniform decelerated motion, since the ball receives the force by the friction of the field only, and the friction is constant over the field. In addition, the ball moves on a straight line unless it meets with an object. Thus, the dominant region of a ball is defined as a line segment that the ball does not meet with any agents. By using the following way, it is possible to find an agent who can get the ball first: 1) compute a partial dominant region for time t_i and draw the position of ball at time t_i on it. 2) If the ball is in the dominant region of an agent, then the agent can get the ball. if not, repeat computation for next time t_{i+1} until the ball is in the dominant region of an agent. By this way, we can predict the agent who gets the ball first, and also we can find a dominant region of a ball.



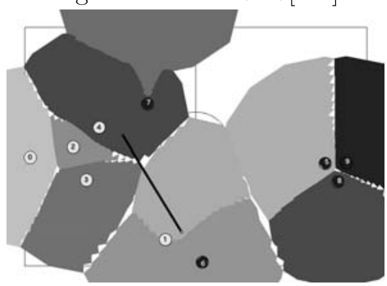
(a)An example of soccer game (at present)



(b) A synthesized partial dominant region until $t = 0.45[sec]$



(c) A synthesized partial dominant region until $t = 0.5[sec]$



(d)Approximated dominant region diagram with ball ($t = 1sec$)

Fig. 9. Example of approximated dominant region diagram with ball

Figure 9 shows an example. Fig. 9 (a) is a current situation of the game. The ball is at the lower part of the left side from the center line. The lines in front of the agents and the ball show the velocity of them. Figs. 9 (b) and (c) are the synthesized partial dominant region until $t = 0.45sec$ and $t = 0.5sec$, respectively. In Fig. 9 (b), there is no agent who can get the ball, but in Fig. 9 (c), the agent No. 4 can get the ball, since the ball is in the dominant region of the agent 4. Fig. 9 (d) shows the approximated dominant region until $t = 1sec$ and the ball's dominant region. To make this diagram, it takes 0.5 $msec$ more time than the computation time of the diagram without the ball.

5.2 Discussion

In this experiment, we use the approximated dominant region with parameters of 8 maximum acceleration vectors and 20 time-divisions for 1 second interval[7]. We used the logged data of the third-place match in 2007 RoboCup competition to analyze the prediction of success for passing. By using the proposed algorithm, we predict the robot who gets the ball first. 60 passings are predicted correctly out of 63 total passings in the game (95% accuracy), i.e. the predicted agent and the agent that gets the ball in the game coincide.

[7] It is possible to obtain the approximated dominant region within 5 $msec$ under the parallel computation environment even if these parameter values are used.

In the results, 3 passings are failed to predict correct agents. The detailed analysis shows that the cause of mis-prediction is not due to the accuracy of the approximated dominant region but due to the strategy of the team. That is, the mis-predicted agent acts to achieve an other goal like moving the goal area to defend the goal by the team's strategy instead of getting the ball. Therefore, we think the approximated dominant region is very useful to judge the prediction of success for passing as well as to evaluate the team's strategy.

6 Conclusion

In this paper, we discussed the real-time computation of the dominant region. We proposed an approximated dominant region algorithm which can calculate an approximated dominant region in real time with accuracy over 90 % under the parallel computation environment.

Moreover, we proposed an approximated dominant region including a ball's dominant region. Experimental results show that it predicts the agent who will get the ball correctly over the accuracy of 95%. This also shows that the approximated dominant region is useful for the analysis of the team's strategy.

In this paper, we also discussed the application of the algorithm to the RoboCup, but it is possible to apply the algorithm to the other objects by establishing the moving model of the agents correctly. In this case, it is necessary to choose appropriate values of parameters, because the calculation cost depends on the required accuracy of approximation.

The future problems are to reduce the computation time further with keeping the approximation accuracy, and to exploit new applicable fields of the algorithm, not just for sports games.

References

1. Murakami, K., Hibino, S., Kodama, Y., Iida, T., Kato, K., Naruse, T.: Cooperative Soccer Play by Real Small-Size Robot. In: Polani, D., Browning, B., Bonarini, A., Yoshida, K. (eds.) RoboCup 2003. LNCS (LNAI), vol. 3020, pp. 410–421. Springer, Heidelberg (2004)
2. Preparata, F.P., Shamos, M.I.: Computational Geometry. Springer, Heidelberg (1988)
3. Taki, T., Hasegawa, J.: Dominant Region: A Basic Feature for Group Motion Analysis and Its Application to Teamwork Evaluation in Soccer Games. In: Proc. SPIE Conference on Videometrics VI, January 1999, vol. 3641, pp. 48–57 (1999)
4. Nakanishi, R., Bruce, J., Murakami, K., Naruse, T.: Cooperative 3-robot passing and shooting in the RoboCup Small Size League. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006. LNCS (LNAI), vol. 4434, pp. 418–425. Springer, Heidelberg (2007)
5. Nakanishi, R., Murakami, K., Naruse, T.: Dynamic Positioning Method Based on Dominant Region Diagram to Realize Successful Cooperative Play. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007. LNCS (LNAI), vol. 5001, pp. 488–495. Springer, Heidelberg (2008)
6. Dashti, H.T., et al.: Dynamic Positioning Based on Voronoi Cells (DPVC). In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, pp. 219–229. Springer, Heidelberg (2006)