بازسازی پویای دیاگرام ورونوی با موانع متحرک در گرید

سعید آسعیدی*
گروه علوم کامپیوتر، دانشکده علوم ریاضی، دانشگاه کاشان، کاشان، ایران
و علی محدث
گروه علوم کامپیوتر، دانشکده ریاضی و علوم کامپیوتر، دانشگاه امیرکبیر، تهران، ایران

چکیده. دیاگرام ورونوی ازجمله مفاهیم کاربردی در شاخه‌های مختلف علوم کامپیوتر است. یکی از کاربردهای مهم دیاگرام ورونوی در علم رباتیک جهت طراحی مسیرهای ایمن در محیط با وجود موانع برای حرکت ربات‌ها است. تاکنون الگوریتم‌های مختلفی جهت ساخت و بازسازی دیاگرام ورونوی در شرایط مختلف از جمله پس از افزودن و حذف یک مانع در محیط ارائه شده است. ما در این مقاله قصد داریم به مسأله طراحی ایمن‌ترین مسیر برای حرکت ربات‌ها با وجود موانع متحرک بپردازیم. در این مقاله سه راهکار جهت بازسازی پویای دیاگرام ورونوی در صورت حرکت یک مانع در محیط ارائه کرده، و راهکارهای ارائه شده را نیز از لحاظ کارایی با یکدیگر مقایسه می‌کنیم.

واژه‌های کلیدی: هندسه محاسباتی، دیاگرام ورونوی، موانع متحرک، بازسازی پویا.
طبقه‌بندی موضوعی [۲۰۱۰]: 68U05, 68W01.

۱. پیش‌گفتار

دیاگرام ورونوی یکی از مفاهیم اساسی در هندسه محاسباتی است که در زمینه‌های مختلف مانند رباتیک، شبکه‌های کامپیوتری، جغرافیا و اقلیم شناسی و ... کاربرد دارد. فرض کنید مجموعه‌ای از نقاط در صفحه داده شده است، ناحیه‌ی ورونوی یک نقطه عبارت است از مجموعه نقاطی از صفحه که به آن نقطه نزدیک‌ترین هستند. دیاگرام ورونوی نیز نشان دهنده‌ی مرزهای نواحی ورونوی برای این مجموعه از نقاط است. مفهوم دیاگرام ورونوی بیش از یک قرن قدمت داشته و در سال ۱۸۵۰ برای اولین بار ارائه شده است [۱]. در سال ۱۹۸۷ آقای Fortune الگوریتمی از مرتبه زمانی $O(nlogn)$ برای محاسبه‌ی دیاگرام ورونوی ارائه داد [۳].

در زمینه‌ی رباتیک استفاده‌های زیادی از دیاگرام ورونوی شده است. در [۵، ۲] نمونه‌هایی از کاربرد ورونوی در طراحی مسیر و برنامه‌ریزی حرکت ربات‌ها ارائه شده است. در بررسی رفتار و حرکت ربات، معمولاً فضای کار ربات به صورت یک گرید در نظر گرفته می‌شود که موانع موجود در مسیر یک ربات در سلول‌های گرید به عنوان سایت‌ها در نظر گرفته می‌شوند. به این ترتیب ایمن‌ترین مسیر که از موانع بیشترین فاصله‌های ممکن را داشته باشد توسط دیاگرام ورونوی بدست خواهد آمد [۲، ۴]. در [۴] تغییرات دیاگرام ورونوی به صورت پویا پس از حذف و اضافه شدن یک مانع در یک گرید مورد بررسی قرار گرفته است. در [۷] دیاگرام ورونوی توسعه یافته بر روی 'گریدهای شش ضلعی به منظور مسیریابی یک عامل هوشمند محاسبه شده است.

با تغییر محیط و مشاهدات جدید، دیاگرام ورونوی به صورت پویا تغییر کرده و دیاگرام جدید از دیاگرام قبلی بدست می‌آید. تغییرات در محیط ممکن است مشاهده‌ی یک مانع جدید در گرید و یا حذف یک مانع توسط عوامل خارجی باشد [۴] . همچنین، حرکت مانع در گرید به صورت پویا می‌تواند باعث تغییر دیاگرام ورونوی شود. برای محاسبه‌ی دیاگرام ورونوی برای موانع متحرک در فضاها و شرایط مختلف مطالعاتی انجام شده است. به عنوان مثال، در [۸] دیاگرام ورونوی موانع دیسکی متحرک، و در [۶] دیاگرام ورونوی نقاط متحرک در صفحه و فضا تحت شرایط محدود مورد بررسی قرار گرفته است. ما در این مقاله به بررسی نحوه تغییر دیاگرام ورونوی با حرکت یک مانع در گرید، و بازسازی دیاگرام ورونوی در این شرایط می‌پردازیم.

ما در بخش ۲ به ارائه مفاهیم اولیه و در بخش ۳ به بررسی راه‌کارهایی جهت بازسازی دیاگرام ورونوی با وجود یک مانع متحرک می‌پردازیم. در بخش ۴ راه‌کارهای ارائه شده از لحاظ کارائی مورد ارزیابی قرار می‌گیرند.

۲. مفاهیم اولیه

ما در این مقاله مانند [۴] محیط کار ربات را به صورت گرید در نظر می‌گیریم. از ساختمان داده‌ای به نام $dist_i$ برای ذخیره‌ی فاصله‌ی iامین سلول گرید تا نزدیک‌ترین مانع به آن سلول استفاده می‌کنیم و این نزدیک‌ترین مانع در ساختمان داده‌ای به نام $obst_i$ ذخیره می‌شود. به عنوان مثال اگر $dist_۵ = ۳$ و $obst_۵ = ۹$ باشد، به این معنی است که نزدیک‌ترین مانع به سلول شماره ۵ در سلول شماره ۹ قرار دارد و فاصله این دو سلول برابر ۳ است. بنابراین اگر مانعی در سلول شماره i قرار داشته باشد، برای آن سلول $dist_i = ۰$ و $obst_i = i$ خواهد بود. در ادامه از عبارت "مقدار سلول i" برای $dist_i$ استفاده می‌کنیم.

* سخنران

مقدار هر سلول کمترین فاصله سلول‌ها تا موانع را نشان می‌دهد و با استفاده از آن نواحی ورونوی موانع محاسبه خواهند شد. تعاریف زیر سلول‌های داخلی، مرزی و خارجی ناحیه‌ی ورونوی را معرفی می‌کنند:

تعریف ۲.۱. سلول x متعلق به ناحیه‌ی ورونوی مانع O را سلول داخلی ناحیه‌ی ورونوی مانع O گوییم اگر تمام همسایگان x موجود و متعلق به ناحیه‌ی ورونوی مانع O باشند.

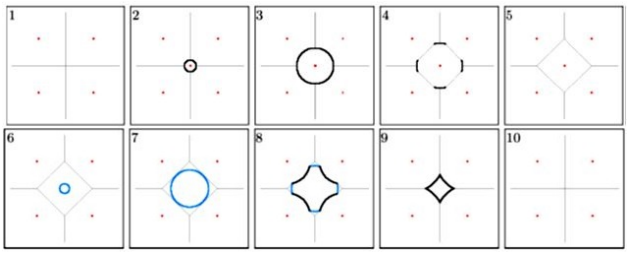
تعریف ۲.۲. سلول x متعلق به ناحیه‌ی ورونوی مانع O را سلول مرزی ناحیه‌ی ورونوی مانع O گوییم اگر x سلول داخلی ناحیه‌ی ورونوی مانع O نباشد.

تعریف ۲.۳. سلول x متعلق به گرید را سلول خارجی ناحیه‌ی ورونوی مانع O گوییم اگر x متعلق به ناحیه‌ی ورونوی مانع O نباشد.

از آنجایی که وضعیت گرید ممکن است به صورت پویا تغییر کند و مانعی اضافه یا حذف شود، مقادیر ساختمان داده‌ی سلول‌های گرید نیز با هر تغییر باید ویرایش شوند. برای ویرایش این مقادیر از الگوریتم دایجسترا و ساختمان داده‌ی دیگری به نام $dist_i^{new}$ استفاده شده است به طوری که $dist_i^{new}$ مقدار ویرایش شده فاصله برای سلول شماره i را ذخیره می‌کند و الگوریتم دایجسترا این مقدار را به صورت زیر حساب می‌کند [۴]:

$$dist_i^{new} = \min_{n \in adj(i)} \{dist_i, distance(i,n) + dist_n^{new}\} \quad (۱)$$

در رابطه (۱)، $adj(i)$ بیانگر شماره سلول‌های همسایه با سلول شماره i است که می‌تواند ۴ یا ۸ سلول همسایه درنظر گرفته شود. همچنین $distance(i,n)$ بیانگر فاصله‌ی سلول شماره n از سلول i است که می‌توان آن را فاصله‌ی اقلیدسی در نظر گرفت. رابطه (۱) یک روال سلسله مراتبی را در ویرایش مقادیر کمترین فاصله‌ها ارائه می‌دهد. هنگامی که مانعی به گرید اضافه می‌شود، یک موج، که آن را موج کاهنده می‌نامیم، از سلول مانع شروع می‌شود و مقدار سلول‌ها را کاهش می‌دهد و تا سلول‌هایی که مقدارشان کاهش نیابد پیش می‌رود و منتشر می‌شود. این سلول‌هایی که موج کاهنده در آنها متوقف می‌شود مرز ناحیه‌ی ورونوی جدید را ایجاد می‌کنند. این موج با دستور $dist_o^{new} = 0$ شروع و با افزودن این سلول به صف اولویت منتشر می‌یابد. برای پیاده سازی این موج از ساختمان داده صف اولویت استفاده شده است. هر سلول که مقدار جدیدی اخذ می‌کند، به این صف افزوده شده تا باعث پیشبرد موج در گرید گردد. سطر اول در شکل ۱ انتشار موج کاهنده به ازای اضافه شدن یک مانع را نشان می‌دهد.



شکل ۱. سطر بالایی: مراحل انتشار موج کاهنده با افزودن یک مانع در مرکز. سطر پایینی: مراحل انتشار موج افزاینده (شماره‌های ۶،۷،۸) و انتشار موج کاهنده (شماره‌های ۸،۹،۱۰) با حذف یک مانع از مرکز [۴].

هنگامی که مانعی از گرید حذف می‌شود، دو موج ایجاد می‌شود تا نواحی ورونوی جدید بدست آید. موج اول را موج افزاینده می‌نامیم و از سلولی که مانع از آن حذف شده شروع می‌شود. این موج با دستور $dist_o^{new} = \infty$ شروع می‌شود و با قرار دادن این سلول به صف اولویت منتشر می‌شود. موج افزاینده باعث افزایش مقدار سلول‌های همسایه می‌شود و مقادیری که برای سلول‌ها بدست می‌آورد ناصحیح است! در ابتدا سلول حذف شده مقدار بینهایت اخذ می‌کند. سپس برای هر سلول همسایه که مانع مربوط به آن همان مانع حذف شده باشد، مقدار سلول را -که به دلیل همسایگی با مانع حذف شده مقدار کوچکی بود- بینهایت قرار می‌دهیم و این باعث می‌شود مقدار جدید این سلول طبق رابطه (۱) از همسایگانش بدست آید که البته مقدار ناصحیحی خواهد بود. به همین ترتیب موج افزاینده پیش می‌رود تا جایی که به سلول‌هایی برسیم که متعلق به ناحیه‌ی ورونوی مانع حذف شده نباشند. این سلول‌ها در مرز ناحیه‌ی ورونوی مانع حذف شده قرار دارند. در این لحظه موج دوم بوجود می‌آید که یک موج کاهنده است. این موج از مقادیر صحیح سلولهای نواحی دیگر استفاده کرده و مقادیر صحیح سلول‌های ناحیه‌ی حذف شده را بازمحاسبه می‌کند. سطر دوم در شکل ۱ این دو موج را نشان می‌دهد [۴].

۳. راهکارهای بازساخت دیاگرام ورونوی حساس به حرکت مانع

فرض کنیم مانع مورد نظر از سلول شماره i به سلول شماره j حرکت کند به‌طوری‌که سلول j یکی از ۸ همسایه‌ی سلول i باشد. در این حالت مانع از سلول شماره i حذف شده و در سلول شماره j اضافه شده است. بنابراین برای پیاده‌سازی حرکت مانع از سلول i به

سلول j کافیست دستور حذف مانع از سلول i و افزودن مانع در سلول j را اجرا کنیم. این ساده‌ترین و اولین راهکار پیاده‌سازی حرکت مانع است.

در ادامه دو راهکار دیگر برای بازساخت دیاگرام ورونوی پس از جابجایی مانع ارائه خواهیم کرد. راهکار دوم بر اساس عمل شیفت مقادیر سلول‌های گرید پیاده‌سازی شده است و راهکار سوم براساس انتشار یک موج یگانه برای بازساخت دیاگرام ورونوی مطرح می‌شود. کارایی این روش‌ها در بخش ۴ مورد بررسی قرار گرفته است.

۱.۳. **بازسازی دیاگرام ورونوی بر اساس شیفت دادن سلول‌ها.** در حرکت یک مانع، اگر مقادیر سلول‌های گرید قبل از حرکت مانع را با جدول A و مقادیر سلول‌ها بعد از حرکت را با جدول B در نظر بگیریم، با مقایسه‌ی ناحیه ورونوی مانع حرکت کرده در جداول A و B مشاهده خواهیم کرد که تقریباً اکثر سلول‌های جدول B از شیفت دادن سلول‌های A در جهت حرکت بدست می‌آیند. این سلول‌ها، که مقادیرشان توسط عمل شیفت سلول‌های A بدست می‌آید، را سلول‌های داخلی ناحیه‌ی ورونوی مانع حرکت کرده نامگذاری کرده‌ایم.

بنابراین، با حرکت یک مانع، مقادیر جدید سلول‌های داخلی آن مانع با عمل شیفت سلول‌های گرید در جهت حرکت محاسبه می‌شوند و نیازی به انتشار موج و قرار دادن این سلول‌ها در صف اولویت نیست. تنها کافیست مقادیر جدید را برای سلول‌های مرزی ناحیه‌ی ورونوی مانع حرکت کرده محاسبه کنیم. برای بدست آوردن مقادیر جدید سلول‌های مرزی، مقدار این سلول‌ها را بینهایت قرار می‌دهیم تا بتوانیم به ترتیب فاصله آنها از مانع، مقدار صحیح آنها را به کمک همسایگانش و طبق رابطه (۱) بدست آوریم. با محاسبه‌ی مقدار صحیح هر سلول، آن سلول را با وزن مقدار جدیدی که اخذ کرده به صف اولویت اضافه می‌کنیم تا انتشار برای آن سلول صورت گیرد.

انتشار موج برای سلول‌های مرزی سه نتیجه‌ی مهم دارد. اول، وزن سلول‌های مرزی ویرایش شده و مقدار صحیح اخذ می‌کنند. دوم، برخی از سلول‌های خارجی ناحیه‌ی ورونوی در طی مراحل انتشار و با توجه به مقادیر جدید بدست آمده برای سلول‌های مرزی، متعلق به ناحیه‌ی ورونوی مانع خواهند شد. به عبارت دیگر، طی عمل انتشار در سلول‌های مرزی، سلول‌های جدید ناحیه‌ی ورونوی مانع حرکت کرده مشخص می‌شوند. سوم، برخی از سلول‌های مرزی در طی مراحل انتشار با توجه به فاصله گرفتن مانع از آنها، تعلق خود را به ناحیه‌ی ورونوی مانع از دست می‌دهند و به ناحیه‌ی ورونوی مانع مجاور متعلق خواهند شد.

بنابراین، با حرکت یک مانع، الگوریتم بازسازی دیاگرام ورونوی براساس شیفت دادن سلول‌ها را به صورت زیر مطرح می‌کنیم:

(۱) تمام سلول‌های ناحیه‌ی ورونوی مانع حرکت کرده را به سمت جهت حرکت شیفت می‌دهیم. این کار با کپی مقدار سلول‌های مجاور در خلاف جهت حرکت مانع انجام می‌شود.
(۲) مقدار سلول‌های مرزی ناحیه‌ی ورونوی مانع حرکت کرده را بینهایت قرار می‌دهیم.
(۳) به ترتیب فاصله از مانع به ازای هرسلول دارای مقدار بینهایت به شماره‌ی i:
(آ) مقدار صحیح سلول i را به کمک همسایگانش طبق رابطه (۱) حساب می‌کنیم و به عنوان مقدار جدید در آن قرار می‌دهیم.
(ب) سلول i را با مقدار جدیدش در صف اولویت قرار می‌دهیم.

در مرحله ۱ از الگوریتم مقدار سلول‌های داخلی مشخص می‌شود و در قسمت ۳آ مقدار سلول‌های مرزی مشخص می‌شود و در قسمت ۳ب مرز جدید مشخص می‌شود.

۲.۳. **بازسازی دیاگرام ورونوی بر اساس انتشار موج کاهنده–افزاینده.** به منظور حذف یک مانع، با بینهایت قرار دادن مقدار سلول مانع حذف شده، مقدار سلول‌های اطراف تغییر می‌کند و به این صورت یک موج افزاینده صورت می‌گیرد تا به مرز نواحی ورونوی دیگر برسد. سپس با استفاده از مقادیر سلول‌های ورونوی نواحی همسایه، مقدار صحیح سلول‌های تغییر یافته بدست می‌آید. به همین ترتیب سلول‌های همسایه ویرایش می‌شوند و مقادیر صحیح اخذ می‌کنند و به این صورت یک موج کاهنده صورت می‌گیرد تا جایی که سلول مانع حذف شده دارای وزن بینهایت مقدار صحیح خود را اخذ کند.

اما در حرکت یک مانع نیازی به بینهایت قرار دادن مقدار سلول مانع نیست. در عمل حذف، از آنجایی که مقدار سلول مانع بعد از حذف مشخص نیست از دو انتشار استفاده می‌شود تا مقدار صحیح بدست آید. اما در عمل حرکت یک مانع، مقادیر هر دو سلول مبدأ و مقصد حرکت مشخص است. سلول مقصد مقدار صفر و سلول مبدأ مقداری معادل با فاصله آن تا مقصد اخذ می‌کند و هر دو مقدار صحیح است. بنابراین، تنها کافیست این دو سلول را با وزن مقادیر جدید به صف اضافه کنیم و با انجام تنها یک موج انتشار کاهنده–افزاینده سلول‌های دیگر را ویرایش کنیم. شبه‌کد ارائه شده در قسمت الف از شکل ۲ نشان دهنده‌ی حرکت مانع می‌باشد.

در انتشار موج کاهنده گفته شده در بخش ۲، در صورتی که مقدار جدید محاسبه شده برای یک سلول در طول انتشار کمتر از مقدار آن سلول باشد، آن مقدار کمتر جایگزین مقدار سلول می‌شود و سلول ویرایش شده با وزن مقدار جدید خود جهت پیشبرد موج انتشار به صف اولویت اضافه خواهد شد. اما در این بخش، در دو حالت مقدار یک سلول در طول انتشار کاهنده–افزاینده تغییر می‌کند. اول اینکه

مقدار جدید برای آن سلول کاهش یابد و دوم اینکه آن سلول نامعتبر باشد. سلول متعلق به ناحیه‌ی ورونوی مانع حرکت کرده نامعتبر گفته می‌شود اگر فاصله سلول از سلول مبدأ مانع کمتر یا مساوی فاصله سلول از سلول مقصد مانع باشد.

معتبر بودن هر سلول پس از حرکت یک مانع، در حین انتشار موج کاهنده–افزاینده، زمانی که موج به یک سلول می‌رسد، بررسی خواهد شد. برای پیاده سازی این روش باید عمل بررسی اعتبار یک سلول در شبه‌کد انتشار کاهنده–افزاینده قرار داده شود. بنابراین شبه‌کد مربوط به انتشار کاهنده–افزاینده در قسمت ب از شکل ۲ ارائه می‌گردد و بررسی اعتبار یک سلول در قسمت ج از شکل ۲ نمایش داده می‌شود.

```
Move_obstacle(i, j , O)
{
1       dist_j^new = 0
2       dist_i^new = distance(i, j)
3       obst_j = O
4       obst_i = j
5       Insert(OPEN, j, dist_j^new)
6       Insert(OPEN, i, dist_i^new)
}
```

الف

```
ProcessLower_Raise(s)
{
1  Foreach n ∈ Adj(s)
2    if (tie_n = s)
3       insert(TIES, n)
4    if(dist_n^new > dist_s^new)
5       d' = distance(n, s) + dist_s^new
6       if(d' < dist_n^new || Is_Invalid(n) )
7             valid[n] = true
8             dist_n^new = d'
9             parent_n = s
10            obst_n = obst_s
11            Insert(OPEN, n, dist_n^new)
}
```

ب

```
bool Is_Invalid(n)
{
1  If (Valid[n] == true)
2     Return false
3  If(dist_index(n, i) < dist_index(n, j))
4     Return true
5  else   Return false
}

float dist_index(cell a, cell b)
{
1       i_1 = row_index(a)
2       j_1 = column_index(a)
3       i_2 = row_index(b)
4       j_2 = column_index(b)
5       Return √((i_1 − i_2)^2 + (j_1 − j_2)^2)
}
```

ج

شکل ۲. الف: شبه‌کد راهکار سوم برای پیاده‌سازی حرکت مانع O از سلول i به سلول j. ب: شبه‌کد انتشار موج کاهنده–افزاینده. ج: بررسی معتبر بودن یک سلول

از ساختمان داده‌ی $valid$ جهت بررسی معتبر شدن سلول‌های نامعتبر استفاده می‌کنیم. یک سلول نامعتبر پس از ویرایش شدن و اخذ مقدار صحیح معتبر خواهد شد. روال $Is_Invalid$ معتبر بودن یک سلول را به کمک تابع فاصله و ساختمان داده‌ی $valid$ بررسی می‌کند. در خط شماره ۶ از روال مربوط به انتشار موج کاهنده–افزاینده در قسمت ب از شکل ۲، اگر مقدار جدید بدست آمده کمتر از مقدار قبلی سلول باشد یا اگر سلول نامعتبر باشد، مقدار سلول ویرایش می‌شود. این سلول معتبر شده و به صف اولویت اضافه می‌شود.

بنابراین، با جابجایی مانع از سلول i به سلول j، این سلول‌ها با مقادیر جدید خود به صف اولویت اضافه شده و باعث انتشار موج کاهنده–افزاینده می‌شوند. طبق الگوریتم، در طول انتشار مقدار سلول‌های نامعتبر افزایش می‌یابد و سلول‌هایی که مقدار جدیدشان کمتر می‌شود کاهش می‌یابند. موج انتشار تا مرزهای ناحیه‌ی ورونوی پیش می‌رود و مرزهای جدید مانع حرکت کرده را ایجاد می‌کند.

۴. بررسی و مقایسه‌ی راهکارها

از آنجایی که در الگوریتم‌های ارائه شده عملیات و مقایسه‌های انجام شده روی سلول‌ها مبتنی بر ورود آنها در صف اولویت است، ما در این مقاله برای بررسی کارایی و مقایسه‌ی راهکارها از لحاظ پیچیدگی زمانی و فضایی تنها به بررسی تعداد سلول‌های اضافه شده به صف اولویت می‌پردازیم. همچنین در هر سه راهکار از درنظر گرفتن سلول‌هایی که از ناحیه‌ی ورونوی همسایه به ناحیه‌ی ورونوی مانع حرکت کرده افزوده می‌شوند صرف نظر می‌کنیم.

در راهکار اول به سه موج انتشار برای بازسازی دیاگرام ورونوی نیاز است. دو موج جهت حذف مانع از مبدأ و یک موج جهت افزودن مانع در مقصد. در هر موج تقریباً تمام سلول‌های ناحیه‌ی ورونوی مانع حرکت کرده در صف اولویت قرار می‌گیرند و باعث پیشروی موج می‌شوند. بنابراین اگر ناحیه‌ی ورونوی مانع حرکت کرده دارای n سلول باشد، برای بازسازی دیاگرام ورونوی به تعداد $3n$ سلول به صف اولویت اضافه می‌شود. البته در این راهکار لزومی ندارد سلول مقصد مجاور سلول مبدأ باشد.

ما در راهکار دوم با استفاده از عمل شیفت دادن سلول‌ها، مقدار جدید سلول‌های داخلی را بدست می‌آوریم و با قرار دادن سلول‌های مرزی در صف اولویت مرز جدید ناحیه‌ی ورونوی مانع را محاسبه می‌کنیم. این راهکار برای موانعی مفید است که ناحیه‌ی ورونوی آنها دارای سلول‌های داخلی زیادی باشد. زیرا بدون انجام محاسبات و انتشار موج و تنها با استفاده از شیفت سلول همسایه مقدار آنها محاسبه می‌شود. از طرف دیگر، با حرکت یک خانه‌ای مانع درگرید، مرز ناحیه‌ی ورونوی مانع، پیشرفت و پسرفت زیادی در نواحی دیگر ندارد. بنابراین با محاسبه‌ی مقدار جدید یک سلول مرزی و قرار دادن آن در صف اولویت موج مربوط به آن سلول مرزی پیشروی زیادی در اطراف آن سلول نخواهد داشت. همانطور که گفته شد از این پیشروی ناچیز در نواحی همسایه در تمامی راهکارها صرف نظر می‌کنیم.

بنابراین در راهکار دوم اگر فرض کنیم ناحیه‌ی ورونوی مانع حرکت کرده دارای n سلول باشد که k سلول داخلی و m سلول مرزی است، در این صورت مقدار k سلول به صورت مستقیم بدست می‌آید و با قرار دادن m سلول در صف اولویت دیاگرام ورونوی بازسازی خواهد شد. این تعداد در مقایسه با تعداد $3n$ در راهکار اول کمتر است بخصوص زمانی که مقدار m نسبت به k کوچک باشد.

البته در این راهکار هرچند که برای محاسبه‌ی سلول‌های داخلی نیازی به انتشار موج و عملیات مربوط به آن نیست، اما نباید از عملیات لازم برای عمل شیفت سلول‌ها صرف نظر کرد. همچنین از دیگر مواردی که در این راهکار باید درنظر گرفته شود زمان لازم برای مرتب سازی سلول‌های مرزی به ترتیب فاصله از مانع است.

در راهکار سوم با قرار دادن دو سلول مبدأ و مقصد در صف اولویت و انتشار تنها یک موج، دیاگرام ورونوی نهایی را بازسازی می‌کنیم. در طول انتشار اگر مقدار یک سلول کاهش یابد یا اگر یک سلول نامعتبر باشد مقدار جدید آن محاسبه می‌شود و با مقدار جدید به صف اولویت اضافه می‌شود. بنابراین یک موج تا انتهای ناحیه‌ی ورونوی مقادیر جدید سلول‌ها را محاسبه می‌کند. اگر ناحیه‌ی ورونوی مانع حرکت کرده دارای n سلول باشد، تمامی n سلول به صف اولویت رفته و مقدار جدید آنها محاسبه می‌شود. البته همانند راهکارهای قبل از درنظر گرفتن سلول‌های افزوده شده به ناحیه‌ی ورونوی صرف نظر می‌کنیم.

اگر فرض کنیم ناحیه‌ی ورونوی مانع حرکت کرده دارای n سلول باشد که k سلول داخلی و m سلول مرزی است، زمانی که مقدار m بیشتر از مقدار k باشد با توجه به سرباری که عملیات شیفت و مرتب سازی در راهکار دوم دارد ممکن است راهکار سوم بهتر و سریعتر به نتیجه برسد. راهکار سوم با انتشار تنها یک موج مقدار سلول‌ها را ویرایش می‌کند و عملیات سرباری مانند راهکار دوم ندارد. همانطور که گفته شد نامعتبر بودن یک سلول نیز در حین انتشار موج بررسی می‌شود و هزینه‌ی عملیاتی به الگوریتم اضافه نمی‌کند.

بنابراین هر دو راهکار دوم و سوم برای بازسازی دیاگرام ورونوی در حرکت یک مانع کارایی بسیار بالاتری نسبت به راهکار اول دارند و می‌توان با استفاده از این راهکارها دیاگرام ورونوی یک مانع را پس از حرکت در زمان مورد قبول بازسازی کرد.

۵. نتیجه‌گیری

دیاگرام ورونوی یکی از مفاهیم مهم و کاربردی در هندسه محاسباتی است. محاسبه برخط و پویای دیاگرام ورونوی کاربردهای زیادی در زمینه‌های مختلف مانند رباتیک دارد. ما در این مقاله به بازسازی پویای دیاگرام ورونوی پس از حرکت مانع در محیط گریدی پرداختیم و به این منظور، الگوریتم‌هایی ارائه نمودیم. الگوریتم اول با حذف مانع از مبدأ و افزودن مانع در مقصد نواحی ورونوی موانع را به‌روزرسانی می‌کند. الگوریتم دوم با شیفت نواحی ورونوی در جهت حرکت به بازسازی دیاگرام ورونوی می‌پردازد و راهکار سوم با انتشار یک موج کاهنده–افزاینده، ناحیه ورونوی مانع حرکت کرده را بازسازی می‌کند. هر کدام از این الگوریتم‌های بازسازی پویای دیاگرام ورونوی در شرایط خاصی بهتر از دیگری عمل می‌کنند.

مراجع

1. Franz Aurenhammer, *Voronoi diagrams—a survey of a fundamental geometric data structure*, ACM Computing Surveys (CSUR) **23** (1991), no. 3, 345–405.
2. Facundo Benavides, Gonzalo Tejera, Martín Pedemonte, and Serrana Casella, *Real path planning based on genetic algorithm and voronoi diagrams*, IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, 2011 IEEE, IEEE, 2011, pp. 1–6.
3. Steven Fortune, *A sweepline algorithm for voronoi diagrams*, Algorithmica **2** (1987), no. 1-4, 153.
4. Nidhi Kalra, Dave Ferguson, and Anthony Stentz, *Incremental reconstruction of generalized voronoi diagrams on grids*, Robotics and Autonomous Systems **57** (2009), no. 2, 123–128.
5. Shahed Shojaeipour, Sallehuddin Mohamed Haris, Khalil Khalili, and Ali Shojaeipour, *Motion planning for mobile robot navigation using combine quad-tree decomposition and voronoi diagrams*, 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), vol. 1, IEEE, 2010, pp. 90–93.
6. Amit Weisman, L Paul Chew, and Klara Kedem, *Voronoi diagrams of moving points in the plane and of lines in space: tight bounds for simple configurations*, Information processing letters **92** (2004), no. 5, 245–251.
7. Fen Tang, Xiong You, Xin Zhang, and Kunwei Li, *Hexagon-based generalized voronoi diagrams generation for path planning of intelligent agents*, Mathematical Problems in Engineering **2020** (2020).
8. Chanyoung Song, Jehyun Cha, Mokwon Lee, and Deok-Soo Kim, *Dynamic voronoi diagram for moving disks*, IEEE Transactions on Visualization and Computer Graphics (2019).

پست الکترونیکی: asaeedi@kashanu.ac.ir
پست الکترونیکی: mohades@aut.ac.ir