یک الگوریتم هیبرید برای ساده‌سازی سرزمین

فهیمه دباغی زرندی و محمد قدسی

چکیده: با پیشرفت تکنولوژی و مجهزشدن دوربین‌های تصویربرداری، دقت تصاویر موجود افزایش یافته است. بالارفتن دقت تصاویر نقش مهمی در کیفیت تجزیه و تحلیل آنها دارد اما این دقت که به واسطه افزایش نقاط موجود در تصویر و بالارفتن حجم اطلاعاتی آن حاصل شده است، مشکلات بسیاری را در خصوص نگهداری و سرعت پردازش تصاویر به وجود آورده و به همین دلیل مسأله ساده‌سازی سرزمین مطرح شده است (معمولاً یک سرزمین به صورت مجموعه‌ای از n نقطه در فضای سه‌بعدی تعریف می‌شود). هدف مسأله ساده‌سازی این است که تعدادی از نقاط یک سرزمین حذف شود به نحوی که خطای سرزمین پس از ساده‌سازی، بیشتر از میزان تعیین‌شده نباشد. خطای ساده‌سازی به دو صورت تعریف می‌شود، یکی این که پس از ساده‌سازی، m نقطه با حداقل خطا در سرزمین وجود داشته باشد $(m \le n)$ یا این که حداکثر خطا پس از ساده‌سازی به ازای کمترین تعداد نقاط، ε باشد $(\varepsilon > ۰)$. این مسأله در حوزه مسایل ان پی- سخت قرار دارد.

در این مقاله، یک الگوریتم هیبرید برای ساده‌سازی سرزمین مطرح شده است که در سه مرحله ساده‌سازی را انجام می‌دهد. ابتدا سرزمین مربوط بر اساس یکی از روش‌های خوشه‌بندی به تعدادی خوشه تقسیم می‌شود، سپس هر خوشه بر اساس یک الگوریتم ساده‌سازی به صورت مجزا ساده می‌شود و در نهایت خوشه‌های ساده‌شده با هم ادغام می‌شوند. این الگوریتم از نظر زمان اجرا در رده مسایل $O(n^۲\sqrt{n})$ قرار دارد. در انتهای مقاله، الگوریتم مطرح‌شده روی سرزمین‌های واقعی مورد آزمایش قرار گرفته و نتایج با استفاده از معیارهای موجود تحلیل شده است.

کلید واژه: **سرزمین، ساده‌سازی، سطح نامنظم مثلث‌بندی‌شده، خوشه‌بندی، هیبرید، الگوریتم‌های حریصانه.**

۱- مقدمه

با توجه به استفاده فراوان از داده‌های جغرافیایی در پردازش‌های آماری در حوزه‌هایی از قبیل سیستم‌های اطلاعات جغرافیایی، گرافیک‌های کامپیوتری، پردازش تصویر و غیره مسأله ساده‌سازی سطوح مطرح شده است. دلیل تمایل به ساده‌سازی این است که وجود اطلاعات با جزئیات بسیار بالا باعث پایین‌آمدن سرعت محاسبات در پردازش‌ها شده است، البته وجود دوربین‌های ماهواره‌ای با دقت بالا باعث می‌شود که دقت پردازش‌ها بالا رود ولی در کاربردهای بسیار متداول ترجیح می‌دهند که اندکی دقت را از دست دهند ولی در ازای آن سرعت بالاتری کسب کنند. در ساده‌سازی هدف این است که تعدادی از نقاط کم‌اهمیت و بسیار جزئی سطح حذف شود ولی این ساده‌سازی باید به نحوی صورت گیرد که دقت سطح در یک حد قابل قبول باشد. یکی از انواع عمومی در مسایل ساده‌سازی، مسأله ساده‌سازی سرزمین است که با توجه به دقت مد نظر به صورت‌های متفاوتی تعریف می‌شود که دو مورد از آن را در ادامه بیان می‌کنیم.

یک مجموعه S از n نقطه در فضای سه‌بعدی (هر نقطه با تابع $z = f(x,y)$ مشخص می‌شود) و یک پارامتر $\varepsilon > ۰$ داده می‌شود. در خروجی الگوریتم ساده‌سازی یک سرزمین ساده‌شده با تقریب ε ایجاد می‌شود که شامل یک مجموعه S' از m نقطه است. m، حداقل نقاط ممکن به ازای خطای داده‌شده می‌باشد. نقاط خروجی با یک تابع $g(x,y)$ نمایش داده می‌شوند که رابطه زیر برای آنها برقرار است

$$\left|f(x_p,y_p)-g(x_p,y_p)\right| \le \varepsilon \quad (x_p,y_p,z_p)\in S \qquad (۱)$$

بنابراین خطایی که در اینجا برای ساده‌سازی در نظر گرفته‌ایم برابر با حداکثر فاصله عمودی نقاط حذف‌شده از سرزمین ساده‌شده است. بر این اساس باید نقاطی را از سرزمین حذف کرد که حداکثر فاصله آنها از سطح ساده‌شده ε باشد. به این نکته نیز باید توجه کرد که در حالت بهینه باید حداکثر نقاط ممکن به ازای خطای ε از سرزمین اولیه حذف شود و به عبارت دیگر، به ازای خطای ε باید حدقل تعداد نقاط را در خروجی داشته باشیم. این تعریف از ساده‌سازی بیشتر در الگوریتم‌های تقریبی [۱] و ... بیان شده که در بخش ۳ شرح داده شده‌اند.

یک مجموعه S از n نقطه در فضای سه‌بعدی (هر نقطه با تابع $z = f(x,y)$ مشخص می‌شود) و یک عدد $m<n$ داده می‌شود. در خروجی یک سرزمین ساده‌شده شامل مجموعه S' از m نقطه با حداقل خطا خواهیم داشت. در اینجا هدف این است که تعداد نقاط یک سرزمین از n نقطه به m نقطه کاهش داده شود با این توجه که حداکثر خطای ممکن نقاط حذف‌شده باید مینیمم باشد. این تعریف از ساده‌سازی بیشتر در الگوریتم‌های مکاشفه‌ای [۲] و ... بیان شده است که در بخش ۳ شرح داده شده‌اند.

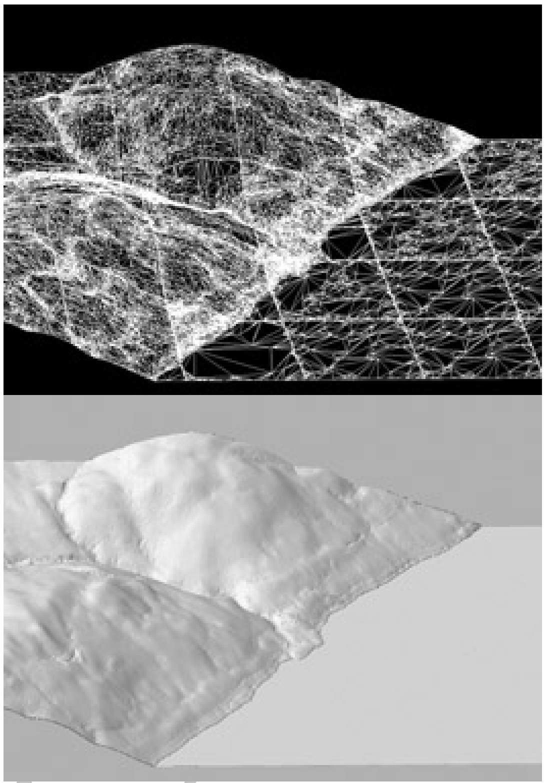
هر کدام از تعریف‌های بیان‌شده در بالا می‌توانند برای ساده‌سازی سرزمین به کار برده شوند و در اکثر کاربردهای گرافیکی و پردازش تصویر اصلی‌ترین معیار برای ساده‌سازی، خطای سرزمین می‌باشد. همان طور که می‌دانیم در تعریف اول همیشه خطای ساده‌سازی مشخص است و پس از ساده‌سازی ممکن است تعداد نقاط زیادی حذف نگردد ولی مطمئن هستیم خطا از میزان مشخص‌شده بیشتر نمی‌شود، ولی در معیار دوم مهم این است که تعداد نقاط خواسته‌شده از متن حذف گردند و هیچ تضمینی در میزان خطای ایجادشده وجود ندارد. بنابراین در جاهایی که خطای سرزمین پس از ساده‌سازی مهم است، بهتر است از معیار اول استفاده کرد ولی اکثر الگوریتم‌هایی که با توجه به معیار دوم طراحی می‌شوند راحت‌تر و کاربردی‌تر هستند.

در ادامه به معرفی چندین مفهوم اولیه در ساده‌سازی سرزمین می‌پردازیم که برای ادامه کار مورد استفاده قرار می‌گیرد. در بخش ۳ کارهای انجام شده در حوزه ساده‌سازی سرزمین با بیان دسته‌بندی روش‌ها مطرح می‌گردد و در نهایت، الگوریتم مطرح‌شده به طور مفصل بیان می‌شود و با استفاده از چندین سرزمین مورد آزمایش قرار می‌گیرد و نتایج تحلیل می‌شوند.

این مقاله در تاریخ ۱ آذر ماه ۱۳۹۰ دریافت و در تاریخ ۶ مهر ماه ۱۳۹۲ بازنگری شد.

فهیمه دباغی زرندی، دانشکده فنی و مهندسی، دانشگاه ولی عصر (عج)، رفسنجان، کرمان، (email: f.dabaghi@vru.ac.ir).

محمد قدسی، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف، تهران، (email: ghodsi@sharif.ir).



شکل ۱: یک سرزمین و تین حاصل پس از مثلث‌بندی آن.

۲- مفاهیم اولیه

در ابتدای کار بهتر است سرزمین را تعریف کرد زیرا ساده‌سازی روی سرزمین تعریف می‌شود. سرزمین[1] یک سطح در فضای سه‌بعدی است که با مجموعه نقاط آن مشخص می‌شود و هر نقطه p با سه مشخصه (x, y, z) بیان می‌گردد. سرزمین دارای خاصیت ارتفاع است و با یک تابع $f(x, y)$ می‌توان آن را به صورت $z_p = f(x_p, y_p)$ نشان داد.

۲-۱ سطح نامنظم مثلث‌بندی شده (تین)

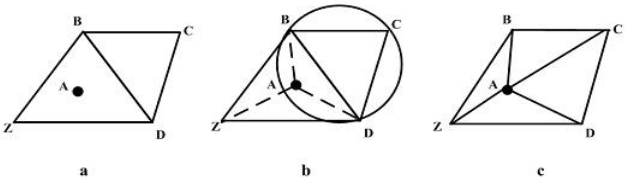
همان طور که گفته شد سرزمین را با استفاده از مجموعه نقاط آن می‌توان شناسایی کرد ولی برای این که تحلیل، نگهداری و پردازش یک سرزمین راحت‌تر انجام شود، آن را به سطح نامنظم مثلث‌بندی شده[2] (تین) تبدیل می‌کنند. اگر همه نقاط یک سرزمین را با استفاده از مثلث‌های مجزا مثلث‌بندی کنیم، سرزمین به تین تبدیل می‌شود و در تین رئوس هر مثلث نقاط سرزمین هستند. در شکل ۱ تصویر یک سرزمین و تین حاصل پس از مثلث‌بندی این سرزمین نشان داده شده است.

این مثلث‌بندی نباید به صورت دلخواه انجام شود، بلکه مثلث‌ها باید تا حد ممکن به مثلث متساوی‌الاضلاع شبیه باشند تا از ایجاد مثلث‌هایی با زاویه‌های تند جلوگیری شود. مشهورترین روش برای مثلث‌بندی نقاط یک سرزمین، مثلث‌بندی Delaunay است. جزئیات این روش در [۳] بیان شده و در ادامه به طور خلاصه این مثلث‌بندی شرح داده می‌شود.

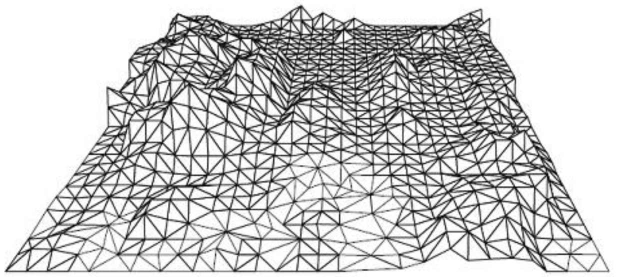
در روش Delaunay مجموعه نقاط به صورتی مثلث‌بندی می‌شوند که دایره محیطی حاصل از سه رأس هیچ مثلثی، شامل نقطه‌ای از مجموعه نقاط سطح نشود. به عبارت دیگر در این مدل مثلث‌بندی، هدف این است که مینیمم زاویه موجود بین مثلث‌ها ماکسیمم شود. روش Delaunay

1. Terrain

2. Triangulated Irregular Network



شکل ۲: درج یک نقطه در مثلث‌بندی Delaunay.



شکل ۳: نمونه‌ای از یک مثلث‌بندی Delaunay.

از یک مثلث ابتدایی شروع می‌کند و نقاط به صورت ترتیبی به تین اضافه می‌شوند. بعد از اضافه‌کردن هر نقطه باید مثلث‌بندی موجود را با وجود این نقطه به هنگام کرد. به طور مثال در شکل ۲ می‌خواهیم رأس A را اضافه کنیم، بنابراین ابتدا از همه نقاط مثلثی که A در آن قرار دارد به A وصل می‌کنیم. حال باید لبه BD را چک کنیم، به این صورت که دایره محیطی مثلث BCD را رسم می‌کنیم و اگر A را شامل شود نتیجه می‌گیریم که BCD مثلث Delaunay نیست. بنابراین لبه BD را حذف می‌کنیم و به جای آن لبه AC را درج می‌کنیم. به همین صورت تمام نقاط سرزمین را مثلث‌بندی می‌کنیم. در شکل ۳ یک نمونه از تین که توسط الگوریتم Delaunay مثلث‌بندی شده، قابل مشاهده است.

در [۴] و [۵] یک مدل دیگر از مثلث‌بندی Delaunay به نام مثلث‌بندی Delaunay با مرتبه بالاتر بیان شده است. این روش به این صورت است که در سطح مثلث‌بندی شده، دایره محیطی هر مثلث می‌تواند حداکثر k نقطه از نقاط دیگر را در بر گیرد که به این مثلث‌بندی Delaunay با مرتبه k گویند.

۲-۲ ساده‌سازی سطح

مسأله ساده‌سازی یک از مهم‌ترین مباحث در سیستم‌های اطلاعات جغرافیایی[3] (GIS)، کاربردهای پردازش تصویر، گرافیک کامپیوتری و ... است. تعداد نقاط موجود در یک سطح واقعی بسیار زیاد است و به میلیون‌ها نقطه می‌رسد زیرا اطلاعات هر سطح واقعی با استفاده از دوربین‌هایی با کیفیت بالا تهیه می‌شود، این سایز بزرگ برای پردازش‌های تصویر و محاسبات آماری مشکل به وجود می‌آورد و سرعت پردازش‌ها را پایین می‌آورد. بنابراین هدف ساده‌سازی این است که تعدادی از این نقاط را حذف کند تا سطح موجود ساده‌تر شود ولی نباید این ساده‌سازی به نحوی انجام گیرد که دقت پردازش‌ها خیلی پایین بیاید و باید در یک حد مقبول خطا داشته باشیم.

ورودی یک مسأله ساده‌سازی سرزمین، یک سرزمین جغرافیایی است که شامل یک مجموعه S با n نقطه است. معیار دیگری که در اینجا برای ساده‌سازی به عنوان ورودی مسأله می‌تواند تعریف شود، دو نوع می‌باشد: نوع اول یک عدد m است که این عدد مشخص می‌کند کاربر می‌خواهد تعداد نقاط خروجی بعد از ساده‌سازی چه تعداد باشد. نوع دوم

3. Geometric Information Systems

می‌تواند ε باشد که یک مقدار مثبت بزرگ‌تر از صفر می‌باشد و مشخص می‌کند حداکثر خطای سطح ساده‌شده نسبت به سطح اصلی ε باشد. خطاهایی که برای نقاط در ساده‌سازی استفاده می‌شوند گوناگون هستند و با توجه به کاربرد مربوط مشخص می‌شوند، ولی در حالت عمومی می‌توان این خطا را بر اساس فاصله عمودی هر نقطه نسبت به سطح ساده‌شده در نظر گرفت، به این صورت که اگر نقطه‌ای از سطح اصلی حذف نشود خطای آن صفر است ولی اگر یک نقطه بعد از ساده‌سازی از سطح حذف شود خطای آن به صورت فاصله عمودی آن نقطه از سطح ساده‌شده به دست می‌آید. در خروجی ساده‌سازی سرزمین یک مجموعه S' از نقاط باقیمانده داده می‌شود. در اکثر کاربردها سرزمین خروجی به صورت یک تین گزارش می‌شود.

در جایی که ساده‌سازی تین مد نظر است در ورودی یک مجموعه S از n نقطه داده می‌شود که همه این نقاط مثلث‌بندی شده‌اند و برای نمایش هر مثلث از یک سه تایی استفاده می‌شود که بیان‌کننده سه رأس مثلث است، بنابراین در ورودی یک مجموعه Δ از n' مثلث داریم که در خروجی نیز مجموعه نقاط و مثلث‌های ساده‌شده را خواهیم داشت.

۲-۳ تبدیل مسأله به فضای دوبعدی xy

همان طور که بیان شد ورودی مسایل ساده‌سازی، یک سرزمین سه‌بعدی خواهد بود. کارکردن در فضای سه‌بعدی خیلی ملموس نیست و اگر بتوان فضای سه‌بعدی را به فضای دوبعدی تبدیل کرد، کارکردن در این حوزه ملموس‌تر خواهد شد. البته این تبدیل باید به نحوی صورت گیرد که تغییری در خروجی و دقت الگوریتم حاصل نشود. برای این تبدیل به تعریف مثلث معتبر نیاز داریم که بعد از بیان آن به بررسی کاهش فضای سه‌بعدی به دوبعدی می‌پردازیم.

معتبربودن یک مثلث در سطح xy: یک مثلث ABC در سطح xy معتبر است اگر فاصله همه نقاط داخل آن مثلث در فضای سه‌بعدی از صفحه حاصل از سه نقطه A، B و C حداکثر ε باشد.

ابتدا همه نقاط در فضای سه‌بعدی $\Re$ را روی سطح xy تصویر می‌کنیم، بنابراین مجموعه S' در سطح xy با n نقطه تصویرشده از مجموعه S به دست می‌آید، سپس تمام مثلث‌هایی که می‌توان از سه نقطه اختیاری در S' تشکیل داد را در نظر می‌گیریم و هر کدام که معتبر نبودند را حذف کرده، از بین مثلث‌های معتبر نیز آنهایی که حداقل یک نقطه را پوشش می‌دهند (حداقل یک نقطه به جز رئوس) را در مجموعه T قرار می‌دهیم. بنابراین مجموعه T تمام مثلث‌های معتبر موجود در سطح xy را شامل می‌شود.

مشخص است که تعداد کل مثلث‌هایی که می‌توان با n نقطه در فضای xy تشکیل داد $O(n^3)$ می‌باشد (اگر هیچ مثلث معتبری با این محدودیت وجود نداشت، مشکلی در الگوریتم ایجاد نمی‌شود زیرا در آن صورت مجبوریم همه نقاط را در ساده‌سازی در نظر بگیریم)، برای این که معتبربودن هر مثلث را چک کنیم نیاز به $O(n)$ زمان داریم بنابراین تبدیل مسأله به فضای دوبعدی نیاز به $O(n^4)$ زمان دارد.

همان طور که مشخص است بعد از حذف مثلث‌های غیر معتبر هر کدام از مثلث‌های مجموعه T که انتخاب شود شرط حداکثر خطای ε را برای نقاط موجود در S ارضا می‌کند. حال باید مجموعه‌ای از مثلث‌ها را انتخاب کنیم که از هم مجزا باشند و تعداد نقاطی که پوشش می‌دهند حداکثر باشد، پس از به دست آوردن این مجموعه مثلث می‌توان تمام نقاط داخل همه مثلث‌ها را حذف کرد و فقط نقاطی که در رئوس مثلث‌ها قرار می‌گیرند را نگهداری کرد. در ضمن باید نقاطی که توسط مثلث‌ها پوشش داده نمی‌شوند را به صورت مجزا نگهداری کرد.

۳- کارهای انجام‌شده

تا کنون کارهای زیادی بر روی ساده‌سازی سرزمین و تین انجام شده است و الگوریتم‌های ارائه‌شده را می‌توان از نظر دقت و سرعت به دسته‌های زیر تقسیم کرد:

- الگوریتم‌های دقیق
- الگوریتم‌های مکاشفه‌ای
- الگوریتم‌های تقریبی

در ادامه توضیح مختصری روی هر نوع از این الگوریتم‌ها داده می‌شود.

۱-۳ الگوریتم‌های دقیق

الگوریتم‌های دقیق بهینه‌ترین جواب ممکن را در خروجی می‌دهند. همان طور که قبلاً بیان شد، دو معیار می‌تواند در ورودی یک الگوریتم ساده‌سازی در نظر گرفته شود که الگوریتم‌های دقیق را بر اساس این دو معیار شرح می‌دهیم:

- **معیار تعداد نقاط پس از ساده‌سازی:** اگر در ورودی یک عدد m داده شود و از الگوریتم خواسته شود سرزمین ورودی که شامل n نقطه است را به نحوی ساده کند که در خروجی m نقطه داشته باشیم، در الگوریتم‌های دقیق خروجی با m نقطه داده می‌شود که کمترین مقدار خطا را داشته باشد.
- **معیار حداکثر خطا:** اگر در ورودی حداکثر خطای ε داده شود، الگوریتم دقیق ساده‌سازی را به نحوی انجام می‌دهد که خروجی حداقل نقاط با حداکثر خطای ε را به دست آورد.

آگاروال و سوری در سال ۱۹۹۴ [1] اثبات کردند که مسأله ساده‌سازی در حالت بهینه، در ردیف الگوریتم‌های ان پی- سخت[1] قرار دارد. بنابراین به دست آوردن الگوریتم دقیق برای ساده‌سازی سرزمین دارای مرتبه زمانی بالایی است که با وجود داده‌های انبوه در ورودی، الگوریتم دقیق نمی‌تواند در کاربردها به کار گرفته شود. بنابراین الگوریتم‌های دقیق از نظر دقت در بالاترین مرتبه ولی از نظر سرعت در پایین‌ترین مرتبه قرار دارند.

۲-۳ الگوریتم‌های مکاشفه‌ای

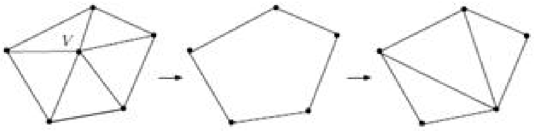
الگوریتم‌های مکاشفه‌ای[2] غالباً به صورت حریصانه عمل می‌کنند. در این مدل الگوریتم‌ها سعی بر این است که بهترین سرعت را برای خروجی ارائه دهند ولی هیچ تضمینی بر کیفیت و دقت خروجی حاصل از ساده‌سازی نمی‌دهند، در کاربردهایی با داده‌های بسیار بالا این الگوریتم‌ها اغلب مناسب می‌باشند.

روش‌هایی که تاکنون برای ساده‌سازی تین به صورت مکاشفه‌ای بیان شده‌اند، دارای قالب یکسانی هستند و تفاوت اغلب آنها در معیار خطای انتخابی برای ساده‌سازی است. در کل می‌توان اغلب الگوریتم‌های مکاشفه‌ای را به دو دسته زیر تقسیم‌بندی کرد:

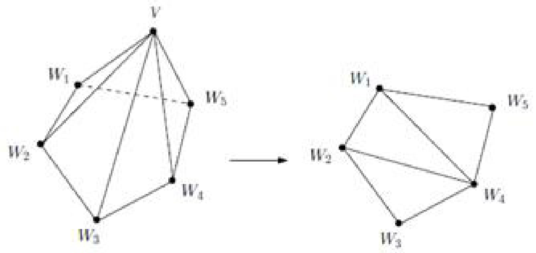
- روش‌های افزایشی[3]
- روش‌های کاهشی[4]

در ادامه به توصیف این دو روش می‌پردازیم.

1. NP-Hard
2. Heuristic
3. Refinement
4. Decimation



شکل ۴: مرحله‌ای از یک الگوریتم کاهشی با استفاده از حذف نقطه.



شکل ۵: مثلث‌بندی مجدد یک حفره.

۳-۲-۱ روش‌های افزایشی

در این مدل از یک مثلث‌بندی کلی شروع می‌کنند و به صورت حریصانه نقاط را به سطح اضافه می‌کنند. به این صورت که می‌توان ابتدا با استفاده از نقاط مرزی سرزمین، آن را به دو مثلث تبدیل کرد سپس در هر مرحله نقطه یا نقاط مهمی را به آن اضافه می‌کنیم و سطح را مجدداً با نقطه یا نقاط جدید مثلث‌بندی می‌کنیم. معیاری که برای مهم‌بودن یک نقطه در نظر گرفته می‌شود، می‌توان معیار خطای آن نقطه باشد به این صورت که فاصله عمودی هر نقطه در سطح اصلی را نسبت به سطح ساده‌شده موجود محاسبه کرده و آن نقطه‌ای که بیشترین فاصله را دارد انتخاب کرده و به سطح اضافه می‌کنند.

یک الگوریتم افزایشی کارا الگوریتم اسکیپ[1] در سال ۱۹۹۵ توسط مایکل گارلند [۲] بیان شده است. این الگوریتم به صورت حریصانه نقاط را در تین درج می‌کند که درج حریصانه شامل دو نوع ترتیبی و موازی است. در نوع ترتیبی در هر گام فقط یک نقطه اضافه می‌شود ولی در درج حریصانه موازی در هر گام ممکن است چند نقطه اضافه شود. در این مقاله چهار الگوریتم درج حریصانه ترتیبی شرح داده شده است:

۱) یک درج حریصانه ترتیبی با مثلث‌بندی Delaunay انجام می‌گردد. این الگوریتم خیلی کند است.

۲) در اینجا الگوریتم قبل را به نحوی بهبود می‌دهند تا پیاده‌سازی آن کاراتر شود. به این ترتیب که تغییرات مثلث‌بندی را به صورت محلی اعمال می‌کنند و بنابراین محاسبات مجدد اضافی حذف می‌شود.

۳) در این الگوریتم ساختارهای داده بهینه‌ای برای انتخاب سریع نقاط در نظر گرفته می‌شود.

۴) در اینجا از همان الگوریتم ۳ استفاده می‌شود ولی به جای مثلث‌بندی Delaunay از مثلث‌بندی وابسته به داده استفاده می‌شود که تقریب کیفیتی بالاتری نسبت به الگوریتم‌های قبل ارائه می‌دهد.

هزینه الگوریتم‌های گفته‌شده را می‌توان به صورتی که در جدول ۱ نشان داده شده، مقایسه کرد.

۳-۲-۲ روش‌های کاهشی

در این روش ابتدا سرزمین اصلی به صورت کامل مانند شکل ۱ مثلث‌بندی می‌شود، سپس در هر مرحله نقاطی از این تین با استفاده از معیار خطای در نظر گرفته شده حذف می‌شود. برای انتخاب نقاط حذفی روش‌های مختلفی وجود دارد که در زیر مهم‌ترین آنها را بررسی می‌کنیم.

1. Scape

جدول ۱: خلاصه پیچیدگی محاسباتی الگوریتم‌های اسکیپ.

الگوریتم	بدترین حالت	مورد انتظار
I	$O(m^{2n})$	$O(mn)$
II	$O(m^{2n})$	$O(mn)$
III or IV	$O(mn)$	$O((m+n)\log m)$

روش بیرون‌کشیدن نقطه

در روش بیرون‌کشیدن نقطه در هر مرحله یک نقطه از سرزمین را که خطای آن نسبت به همه نقاط کمتر است، انتخاب کرده و آن را از تین حذف می‌کنیم. با حذف این نقطه یک حفره در تین ایجاد می‌شود که باید تین را مجدداً مثلث‌بندی کرد. مثلث‌بندی مجدد می‌تواند بر اساس معیار Delaunay انجام شود. در شکل ۴ یک مرحله از این الگوریتم قابل مشاهده است.

نحوه مثلث‌بندی مجدد پس از حذف یک نقطه: پس از حذف یک نقطه v از تین، یک حفره ایجاد می‌شود که باید آن را مثلث‌بندی کرد. فرایند مثلث‌بندی مجدد در اکثر روش‌ها به این شکل است که همه همسایه‌های v را در نظر می‌گیرند، $W_1,W_2,\ldots,W_k$ حداکثر $O(k^2)$ خط مستقیم بین این رئوس برقرار است. نزدیک‌ترین خط به v را پیدا می‌کنیم (W_aW_b). ابتدا تمام لبه‌هایی که از v به W_iها وصل است را حذف می‌کنیم، سپس از بین W_a و W_b نزدیک‌ترین رأس به v را پیدا کرده و به تمام رئوس مقابلش بین مجموعه $W_1,W_2,\ldots,W_k$ وصل می‌کنیم. مراحل کار در شکل ۵ نشان داده شده است.

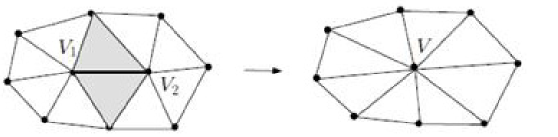
در اکثر روش‌های حذف گره، برای معیار خطا از فاصله استفاده می‌کنند ولی معیارهای دیگری نیز برای حذف می‌توان استفاده کرد. به طور مثال می‌توان بر اساس خصوصیات لبه‌های اطراف هر نقطه، حذف را انجام داد یعنی با استفاده از لبه‌هایی که به هر نقطه مرتبط است یک تابع ارزش برای آن تعریف می‌کنند.

الگوریتم‌های زیادی در این زمینه مطرح شده‌اند [۶] تا [۸]. در [۶] الگوریتمی برای ساده‌سازی بر اساس بیرون‌کشیدن نقطه بیان شده که از خصوصیات لبه‌های اطراف یک نقطه برای حذف آن استفاده می‌کند. برای هر گره زاویه داخلی[2] تعریف می‌شود. زاویه داخلی هر گره بیشینه زاویه‌ای است که دو لبه از آن گره می‌توانند بسازند. بنابراین از سطوحی که انحنای بالایی دارند صرف نظر شده و سطوح با انحنای کم را ساده می‌کنند و در نتیجه هر دو خصیصه انحنا و فاصله با هم نتایج بهتری می‌دهند.

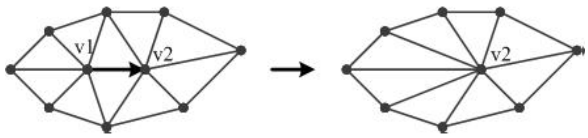
روش انقباض لبه

در روش انقباض لبه[3]، یک لبه (v_1,v_2) انتخاب می‌شود و به جای آن یک نقطه جایگزین می‌شود، به عبارتی لبه انتخابی در یک نقطه v منقبض می‌شود و بنابراین همه نقاطی که به v_1 و v_2 ارتباط داشته‌اند به v وصل می‌شوند [۹] تا [۱۲]. نقطه v می‌تواند به روش‌های مختلف انتخاب شود، به طور مثال می‌تواند v_1، v_2، $(v_1+v_2)/2$ و یا هر چیز دیگری انتخاب شود ولی اگر هدف این باشد که نقطه‌ای غیر از سرزمین انتخاب نشود باید از v_1 یا v_2 برای انقباض استفاده کرد. به طور معمول v را به نحوی انتخاب می‌کنند که خطای تین پس از انقباض لبه کمتر باشد. در شکل ۶ یک مدل از انقباض لبه نشان داده شده است.

2. Interior Angle
3. Edge Contraction



شکل ۶: مرحله‌ای از یک الگوریتم کاهشی با استفاده از انقباض لبه.



شکل ۷: مرحله‌ای از یک الگوریتم کاهشی با استفاده از انقباض نیم‌لبه.

در برخی از الگوریتم‌ها به جای لبه از نیم‌لبه استفاده می‌شود به این صورت که همه لبه‌های تین را به دو نیم‌لبه جهت‌دار تبدیل می‌کنند و در هر مرحله از انقباض نیم‌لبه استفاده می‌شود. به دلیل این که نیم‌لبه‌ها جهت‌دار هستند پس از انقباض، نیم‌لبه مربوط در نقطه مقصد نیم‌لبه منقبض می‌شود. در شکل ۷ یک مدل از انقباض نیم‌لبه آمده است.

در [۱۳] یک الگوریتم بر اساس انقباض جفت رأس بیان شده است به این صورت که اگر جفت انتخاب‌شده برای انقباض یک لبه را تشکیل دهند مانند شکل ۶ عمل می‌کند، ولی اگر جفت رأس انتخاب‌شده از هم مستقل باشند انقباض آنها باعث می‌شود که دو بخش مجزا از تین با هم الحاق شوند که این الحاق در شکل ۸ مشاهده می‌شود. در این الگوریتم از معیار خطای مربعی برای ساده‌سازی و انتخاب جفت رأس‌ها برای انقباض استفاده شده است که این معیار اولین بار توسط گارلند و هکبرت در [۱۳] بیان شد. برای مطالعه آن می‌توانید به کارهای گارلند و هکبرت در [۱۳] و [۱۴] مراجعه کنید. انقباض جفت رأس در مقاله گارلند و هکبرت بیان شد ولی در اغلب کارهای دیگر از انقباض لبه استفاده می‌کنند.

روش حذف مثلث

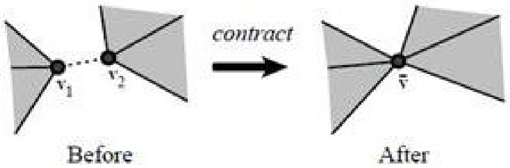
در کارهای بسیار محدودی [۱۵] برای ساده‌سازی در هر مرحله یک مثلث با کمترین خطا انتخاب می‌شود و از تین مربوطه حذف می‌شود که حفره ایجادشده پس از حذف مثلث باید مجدداً مثلث‌بندی شود، به عبارتی می‌توان گفت در این روش‌ها در هر مرحله سه نقطه از تین بیرون کشیده می‌شود.

روش خوشه‌بندی نقاط

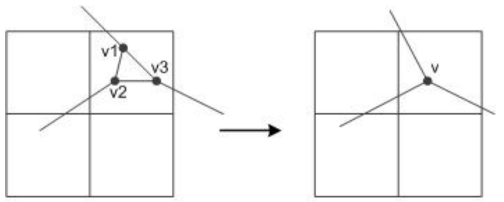
در این روش هر نقطه یک وزن دارد و در برخی از کارها معیاری که برای وزن‌دهی استفاده می‌کنند بر این اساس است: نقاطی که با مثلث‌های با وجوه بزرگ‌تری مواجه هستند و انحنای بیشتری دارند وزنشان نسبت به نقاطی که با مثلث‌هایی با وجوه کوچک‌تری مجاور هستند، بیشتر است. ابتدا یک چارچوب دور تین می‌کشیم و یک شبکه سه‌بعدی ایجاد می‌کنیم و نقاطی که در هر سلول خوشه‌بندی[1] شده‌اند با نقطه‌ای که ماکسیمم وزن را دارد جایگزین می‌شوند. در شکل ۹ یک مرحله از این روش نمایش داده شده است.

۳-۳ الگوریتم‌های تقریبی

همان طور که گفته شد مسأله ساده‌سازی سطح، یک مسأله ان پی- سخت است و بنابراین به دست آوردن جواب دقیق برای آن مقرون به صرفه نیست. از طرفی الگوریتم‌های مکاشفه‌ای نیز از نظر تئوری معیار دقیقی از دقت ساده‌سازی به ما نمی‌دهند و نمی‌توان به آنها کاملاً اعتماد



شکل ۸: مرحله‌ای از یک الگوریتم کاهشی با استفاده از انقباض جفت غیر لبه [۱۳].



شکل ۹: مرحله‌ای از یک الگوریتم کاهشی با استفاده از خوشه‌بندی نقاط.

کرد، بنابراین به دنبال الگوریتمی هستیم یک راه حل با زمان چندجمله‌ای برای مسأله ساده‌سازی ارائه دهد که دقتی کمتر از دقت راه حل بهینه داشته باشد، اما یک ضریب تقریب دقیق برای الگوریتم بیان شود به طوری که مشخص کند این الگوریتم به چه نسبتی به جواب بهینه نزدیک است. این تقریب می‌تواند روی هر دو معیار بحث‌شده در قبل اعمال شود، در ادامه تقریب روی هر یک از معیارها را بررسی می‌کنیم.

معیار تعداد نقاط پس از ساده‌سازی: در این مدل، ساده‌سازی را به نحوی انجام می‌دهد که تعداد m نقطه در خروجی داشته باشد که خطای این ساده‌سازی حداقل خطای ممکن نیست اما با یک نسبت، تفاوت خطا با خطای بهینه مشخص می‌شود، به این صورت که اگر جواب بهینه دارای خطای ε باشد این الگوریتم دارای خطایی بیشتر از ε است که دقیقاً با $\alpha\varepsilon$ مشخص می‌شود.

معیار حداکثر خطا: همان طور که قبلاً گفته شد در این موارد در ورودی، یک خطای حداکثر برای سطح ساده‌شده داده می‌شود. با الگوریتم تقریبی می‌توان سطح را در زمان چندجمله‌ای ساده کرد ولی دقت ساده‌سازی کمتر از جواب بهینه است. در اینجا ضریب تقریب را بر اساس تعداد نقاطی که به خروجی داده می‌شود، بیان می‌کنند به این صورت که اگر جواب بهینه با حداکثر خطای ε حداقل تعداد m نقطه در خروجی داشته باشد، الگوریتم تقریبی تعداد نقاطی بیشتر از m می‌دهد ولی به طور دقیق خطای این الگوریتم مشخص می‌شود، به طور مثال تعداد نقاط خروجی αm خواهد بود.

الگوریتم‌های تقریبی از نظر سرعت مناسب هستند و دقت آنها نیز قابل اندازه‌گیری است و بر این اساس می‌توان گفت در حالت عمومی الگوریتم‌های تقریبی از نظر کیفیت مابین الگوریتم‌های دقیق و مکاشفه‌ای قرار می‌گیرند. اولین الگوریتم تقریبی در این حوزه در سال ۱۹۹۴ توسط آگاروال و سوری [۱] بیان شد. در این الگوریتم سرزمین ورودی در زمان $O(n^8)$ به یک سرزمین با سایز $O(k\log k)$ ساده می‌شود که k سایز بهینه سرزمین ساده‌شده می‌باشد و معیار حداکثر خطا برای ساده‌سازی در نظر گرفته شده است. دباغی و قدسی [۱۶] این الگوریتم را با همان شرایط در زمان $O(n^7)$ ارائه دادند.

در سال ۱۹۹۷ آگاروال و دسیکان [۱۷] الگوریتم تقریبی دیگری مطرح کردند که بهبودی از الگوریتم قبل بود. ورودی الگوریتم یک مجموعه $\bar{S}$ از n نقطه در $\Re^3$ است که با یک تابع دومتغیره $z_p = f(x_p, y_p)$ مشخص می‌شود. $g(x,y)$ یک تقریب $\varepsilon > 0$ از $f(x,y)$ نامیده می‌شود اگر برای هر $p \in \bar{S}$ ، $|f(x,y) - g(x,y)| \le \varepsilon$ باشد. در اینجا با استفاده از یک الگوریتم تصادفی می‌توان یک تقریب ε از سطح را با

1. Clustering

سایز $O(k^2 \log^2 k)$ در زمان $O(n^{2+\delta} + k^3 \log k \log(n/k))$ محاسبه کرد که در اینجا k سایز تقریب بهینه ε است و δ یک مقدار کوچک مثبت می‌باشد.

٤- حل مسأله به صورت هیبرید

در این بخش یک الگوریتم هیبرید پیشنهاد شده است که علاوه بر ساده‌کردن سرزمین خاصیت Delaunay را حفظ می‌کند و مثلث‌بندی حاصل به این صورت است که رئوس هر مثلث از نقاط سرزمین تشکیل شده‌اند که در بسیاری از الگوریتم‌ها بر این قضیه تأکیدی نداشته‌اند. به طور خلاصه می‌توان این الگوریتم را در سه مرحله زیر خلاصه کرد:

۱) ابتدا نقاط سرزمین ورودی را به k خوشه تقسیم می‌کنیم.

۲) در این مرحله نقاط موجود در هر خوشه را به صورت مجزا ساده می‌کنیم.

۳) حال باید حفره‌های موجود در تین را با استفاده از مثلث‌های اضافی پر کرد تا تین ما کامل شود و به عبارتی در این مرحله به ادغام خوشه‌ها می‌پردازیم.

در ادامه به طور مفصل هر کدام از مراحل گفته‌شده را شرح می‌دهیم.

٤-۱ مرحله اول: خوشه‌بندی نقاط سرزمین

همان طور که مطرح شد در این مرحله باید نقاط موجود در سرزمین را به تعدادی خوشه تقسیم کرد. از آنجایی که نقاط ما در فضای اقلیدسی قرار دارند و معیار شباهت در خوشه‌بندی، فاصله نقاط می‌باشد، الگوریتم خوشه‌بندی k-means می‌تواند یک الگوریتم مناسب برای استفاده در این مرحله باشد.

هدف الگوریتم این است که n نقطه در ورودی را به k خوشه تقسیم کند به طوری که نقاط داخل هر خوشه از نظر فاصله اقلیدوسی خیلی به هم نزدیک باشند و نقاط موجود در دو خوشه متفاوت، از نظر فاصله اقلیدسی با هم تفاوت بیشتری داشته باشند. این الگوریتم گام‌هایی را به صورت تکراری انجام می‌دهد. در زیر می‌توان مراحل این الگوریتم را مشاهده کرد:

۱) ورودی الگوریتم: یک عدد k برای تعیین تعداد خوشه‌ها و یک عدد m که حداکثر تعداد تکرارها را مشخص می‌کند.

۲) k نقطه از نقاط ورودی را به صورت تصادفی به عنوان مرکز k خوشه انتخاب کن.

۳) $i \leftarrow 0$

۴) هر کدام از n نقطه ورودی را به خوشه‌ای اختصاص بده که فاصله اقلیدسی آن از مرکز خوشه کمتر از همه باشد.

۵) مرکز همه خوشه‌ها را به هنگام کن، به این صورت که مرکز هر خوشه برابر است با میانگین نقاط موجود در آن خوشه.

۶) $i \leftarrow i+1$

۷) اگر $i == m$ است متوقف کن، در غیر این صورت به مرحله ۴ برو.

مرتبه زمانی این مرحله برابر است با مرتبه الگوریتم k - means که از مرتبه $O(nmk)$ است.

٤-۲ مرحله دوم: ساده‌سازی هر خوشه به صورت مجزا

برای ساده‌سازی خوشه‌ها در مرحله اول مسأله را به فضای دوبعدی تبدیل می‌کنیم و تمام مثلث‌های معتبر را در فضای دوبعدی به دست می‌آوریم و حال برای انتخاب مثلث‌ها از الگوریتم حریصانه استفاده می‌کنیم. معیاری که برای انتخاب حریصانه در نظر می‌گیریم به صورتی است که از خاصیت مثلث‌بندی Delaunay استفاده می‌کند. به این صورت که مثلث‌ها را بر اساس مینیمم زاویه آنها به صورت نزولی مرتب می‌کنیم، بنابراین با انتخاب حریصانه دیگر مثلث‌هایی با زاویه‌های بسیار تند انتخاب نمی‌شوند و به اصطلاح عامیانه مثلث‌های چاق را انتخاب می‌کنیم. مراحل الگوریتم ساده‌سازی هر خوشه در زیر بیان شده است:

۱) ابتدا تصویر همه نقاط را در سطح xy به دست بیاور.

۲) تمام مثلث‌هایی را که می‌توان از نقاط تشکیل داد به دست بیاور و نقاطی را که داخل هر مثلث قرار می‌گیرند مشخص کن.

۳) مثلث‌هایی را که معتبر نیستند حذف کن.

۴) مینیمم زاویه داخلی را برای همه مثلث‌های معتبر محاسبه کن.

۵) مثلث‌ها را بر اساس مینیمم زاویه به صورت نزولی مرتب کن.

۶) $U \leftarrow \phi$

۷) مثلث Δ را از لیست مرتب‌شده مثلث‌ها انتخاب کن با این شرط که با مثلث‌های موجود در U اشتراک نداشته باشد.

۸) $U \leftarrow \Delta$

۹) اگر اجتماع مثلث‌های انتخاب‌شده در U کل نقاط سرزمین را پوشش می‌دهد توقف کن و در غیر این صورت به مرحله ۷ برو.

معتبربودن یک مثلث در سطح xy: یک مثلث ABC در سطح xy معتبر است اگر فاصله همه نقاط داخل آن مثلث در فضای سه‌بعدی از صفحه حاصل از سه نقطه A، B و C حداکثر ε باشد.

مرتبه زمانی این مرحله بر اساس نقاط موجود در خوشه‌ها محاسبه می‌شود. اگر نقاط موجود در خوشه iام را n_i در نظر بگیریم، مرتبه ساده‌سازی برای هر خوشه برابر است با $O(n_i^3 \log n_i)$ زیرا بیشترین هزینه برای مرتب‌سازی مثلث‌ها صرف می‌شود که چون در بدترین حالت $O(n_i^3)$ مثلث در خوشه داریم این مرتبه به دست می‌آید. ساده‌سازی برای k خوشه انجام می‌شود، بنابراین مرتبه کلی مرحله ساده‌سازی خوشه‌ها به صورت زیر می‌باشد

$$\sum_{i=0}^{k} O(n_i^3 \log n_i) \,\&\, \sum_{i=0}^{k} n_i = n \Rightarrow O(step2) = O(n^3 \log n) \qquad (2)$$

همان طور که مشخص است در این مرحله برای هر خوشه باید یک پیش‌پردازش برای به دست آوردن مثلث‌های معتبر انجام داد که همان طور که در بخش ۴-۱ اثبات شد، مرتبه زمانی این پردازش $O(n^4)$ است، بنابراین مرتبه زمانی این مرحله در کل $O(n^4)$ خواهد بود.

دلیل استفاده از روش گفته‌شده برای ساده‌سازی در این مرحله به این صورت می‌باشد که در این قسمت برای ساده‌کردن هر خوشه یک روش افزایشی اضافه‌کردن مثلث انتخاب شده است ولی چون هدف کلی الگوریتم هیبرید این است که معیار حداکثر خطا حفظ شود و روش اضافه‌کردن مثلث از معیار تعداد نقاط پس از ساده‌سازی استفاده می‌کند و تضمینی برای حداکثر خطا نمی‌دهد. به همین دلیل برای حفظ حداکثر خطا پس از ساده‌سازی از نگاشت نقاط به فضای دوبعدی و مفهوم مثلث معتبر استفاده شده است. پس از به دست آوردن کلیه مثلث‌های معتبر، اضافه‌کردن هر مثلث این اطمینان را ایجاد می‌کند که هیچ نقطه‌ای با خطای بیشتر از ε مطرح‌شده وجود نخواهد داشت.

٤-۳ مرحله سوم: ادغام خوشه‌های ساده‌شده

برای این مرحله باید با استفاده از رئوس مثلث‌های انتخاب‌شده در مرحله قبل مثلث‌هایی را ایجاد کرد که حفره‌های موجود در تین را پوشش دهد. در اینجا سعی کرده‌ایم برای اضافه‌کردن مثلث‌ها حتی‌الامکان

جدول ۲: جزئیات ساده‌سازی یک سرزمین ورودی با ۶۹۸ نقطه با استفاده از دقت‌های مختلف.

سرزمین	تعداد نقاط	تعداد مثلث‌ها
سرزمین اصلی	۶۹۸	۱۳۰۹
سرزمین ساده‌شده با $\varepsilon = ۰/۵$	۵۲۹	۹۷۷
سرزمین ساده‌شده با $\varepsilon = ۱$	۴۹۹	۹۱۶

خاصیت Delaunay حفظ شود که جزئیات آن در فصل پیاده‌سازی و ارزیابی بیان شده است. مرتبه زمانی این مرحله برابر با تعداد مثلث‌های اضافه‌شده یا همان $O(n)$ است.

۴-۴ تحلیل کلی الگوریتم هیبرید

در بدترین حالت مرتبه زمانی این الگوریتم از مرتبه $O(n^4)$ است زیرا همان طور که در تحلیل تک‌تک مراحل صحبت شد، مرحله پیش‌پردازش این الگوریتم برای تولید مثلث‌های معتبر زمانی معادل $O(n^4)$ لازم دارد ولی اگر بخواهیم حالت متوسط را در نظر بگیریم مرتبه الگوریتم از این مقدار کمتر می‌شود. در فصل پیاده‌سازی و ارزیابی به طور مفصل در مورد مقادیر ممکن برای انتخاب تعداد خوشه‌ها بحث شده است. به طور کلی اگر تعداد خوشه‌ها را بر اساس تعداد نقاط تعریف کنیم، می‌توان $O(\sqrt{n})$ خوشه در نظر گرفت که در حالت متوسط چون مراکز اولیه خوشه‌ها به صورت تصادفی انتخاب می‌شوند، می‌توان تعداد عناصر موجود در هر خوشه را $O(\sqrt{n})$ در نظر گرفت و بنابراین بیشترین مرتبه این الگوریتم، به دست آوردن مثلث‌های معتبر است. برای هر خوشه $O(\sqrt{n}^3)$ مثلث داریم که برای تست معتبربودن هر مثلث $O(\sqrt{n})$ زمان لازم داریم، در این صورت پیش‌پردازش هر خوشه دارای مرتبه زمانی $O(n^2)$ است، با توجه به این که $O(\sqrt{n})$ خوشه وجود دارد بنابراین مرتبه زمانی کلی تولید مثلث‌های معتبر، $O(n^{2.5})$ می‌باشد و در نتیجه مرتبه زمانی الگوریتم هیبرید در حالت متوسط $O(n^2\sqrt{n})$ است. البته اگر پردازش موازی را در نظر بگیریم، ساده‌سازی تمام خوشه‌ها را می‌توان به صورت موازی انجام داد که مرتبه زمانی الگوریتم $O(n^2)$ می‌شود.

۴-۵ مقایسه الگوریتم هیبرید با سایر الگوریتم‌ها

در الگوریتم ارائه‌شده ابتدا نقاط به تعدادی خوشه تقسیم می‌شوند و هر خوشه به صورت مجزا ساده می‌شود. این خوشه‌بندی در کاربردهایی که حجم داده‌ها بالا است مفید می‌باشد و می‌توان ساده‌سازی هر خوشه را در یک سیستم توزیع‌شده یا موازی انجام داد و سرعت اجرا را بالا برد.

همان طور که در بخش کارهای انجام‌شده بررسی شد، الگوریتم‌های تقریبی از معیار حداکثر خطا و الگوریتم‌های حریصانه از معیار حداکثر تعداد نقاط پس از ساده‌سازی استفاده می‌کنند. الگوریتم‌های حریصانه سرعت بسیار بالایی در ساده‌سازی دارند، به طور مثال الگوریتم ارائه‌شده توسط گارلند [۱] ساده‌سازی را در زمان $O(nm)$ انجام می‌دهد که n تعداد نقاط سرزمین اولیه و m تعداد نقاط پس از ساده‌سازی را مشخص می‌کند. مشکلی که الگوریتم‌های حریصانه دارند این است که معیار حداکثر تعداد نقاط پس از ساده‌سازی را در نظر می‌گیرند و هیچ تضمینی روی حداکثر خطای نقاط پس از ساده‌سازی ارائه نمی‌دهند. ما در این مقاله یک الگوریتم حریصانه ارائه کردیم که برای ساده‌سازی هر خوشه از معیار حداکثر خطا استفاده می‌کند. بنابراین در کاربردهایی که خطای سرزمین پس از ساده‌سازی مهم‌تر از تعداد نقاط باقیمانده باشد و بحث سرعت مطرح باشد این الگوریتم می‌تواند مورد استفاده قرار گیرد.

همان طور که گفته شد سرعت الگوریتم هیبرید از سایر الگوریتم‌های حریصانه کمتر است اما با توجه به این که الگوریتم هیبرید بر خلاف سایر الگوریتم‌های حریصانه از معیار حداکثر خطا استفاده می‌کند، برای مقایسه باید این روش را با روش‌های تقریبی مقایسه کرد که از معیار حداکثر خطا استفاده کرده‌اند. بهترین الگوریتم تقریبی از نظر مرتبه زمانی، الگوریتم تقریبی آگاروال و دسیکان [۱۷] می‌باشد که با استفاده از یک الگوریتم تصادفی می‌توان یک تقریب ε از سطح را با سایز $O(k^2 \log^2 k)$ در زمان $O(n^{2+\delta} + k^2 \log k \log(n/k))$ محاسبه کرد که در اینجا k سایز تقریب بهینه ε است و δ یک مقدار کوچک مثبت می‌باشد. همان طور که در بخش ۴-۴ مطرح شد، مرتبه زمانی الگوریتم هیبرید در حالت متوسط، $O(n^{2.5})$ می‌باشد که اگر عملیات ساده‌سازی خوشه‌ها را به صورت موازی انجام دهیم دارای مرتبه زمانی $O(n^2)$ می‌باشد. از آنجایی که ساده‌سازی هر خوشه کاملاً مستقل از سایر خوشه‌ها می‌باشد، می‌توان مرتبه زمانی الگوریتم هیبرید را $O(n^2)$ در نظر گرفت، بنابراین الگوریتم ارائه‌شده در این مقاله از نظر سرعت بهتر از سایر الگوریتم‌هایی است که از معیار حداکثر خطا استفاده می‌کنند.

یکی از معایب این روش این است که مانند سایر روش‌های حریصانه نمی‌توان هیچ براوردی از تفاوت دقت این روش با روش بهینه به دست آورد.

۵- پیاده‌سازی الگوریتم هیبرید برای ساده‌سازی سرزمین

در این بخش به پیاده‌سازی الگوریتم بیان‌شده در بخش ۴ می‌پردازیم. همان طور که قبلاً گفته شد، برای پیاده‌سازی این الگوریتم باید سه مرحله را در نظر گرفت. در مرحله اول نقاط سرزمین ورودی را در سطح xy تصویر کرده، سپس این نقاط را بر اساس فاصله اقلیدسی با استفاده از الگوریتم k - means خوشه‌بندی کرد. در حالت کلی تعداد خوشه‌ها برای ورودی مسأله $\sqrt{n}$ در نظر گرفته شده است. در مرحله دوم هر کدام از خوشه‌ها به صورت مجزا ساده می‌شوند که برای ساده‌سازی هر خوشه ابتدا تمام مثلث‌های معتبر آن خوشه محاسبه شده‌اند، سپس مثلث‌ها را بر اساس معیار چاق‌بودن به صورت نزولی مرتب کرده و به صورت حریصانه مثلث‌ها را انتخاب می‌کنیم به طوری که در هر مرحله چک می‌کنیم که مثلث انتخاب‌شده در فضای سه‌بعدی $\Re^3$ با مثلث‌های انتخاب‌شده قبلی تقاطع نداشته باشد و این مرحله برای هر خوشه زمانی متوقف می‌شود که تمام نقاط آن خوشه توسط مثلث‌های انتخاب‌شده پوشش داده شوند. خروجی به دست آمده از مرحله دوم یک سطح مثلث‌بندی شده است که حفره‌هایی در این سطح وجود دارد، بنابراین یک سری مثلث در مرحله سوم به آن اضافه می‌کنیم تا سطح به صورت کامل مثلث‌بندی شود.

همان طور که در بخش ۴ بیان شد، در اینجا معیار ساده‌سازی این است که در ورودی یک ε داریم که می‌خواهیم سرزمین به نحوی ساده شود که فاصله نقاط حذف‌شده از سطح ساده‌شده حداکثر ε باشد و خروجی ما حداقل تعداد نقاط ممکن را بر اساس این دقت بدهد.

بعد از نوشتن این کد یک سرزمین ۶۹۸ نقطه‌ای برای آزمایش با تقریب‌های متفاوتی ساده کرده‌ایم که در جدول ۲ می‌توان جزئیات آن را مشاهده کرد.

همان طور که در جدول ۲ مشخص است، ورودی الگوریتم یک سرزمین با سایز ۶۹۸ نقطه است که اگر آن را با مثلث‌بندی Delaunay مثلث‌بندی کنیم دارای ۱۳۰۹ مثلث خواهد بود. این سرزمین را با دقت‌های مختلف ساده کرده‌ایم و برای هر حالت تعداد نقاط و مثلث‌های موجود در

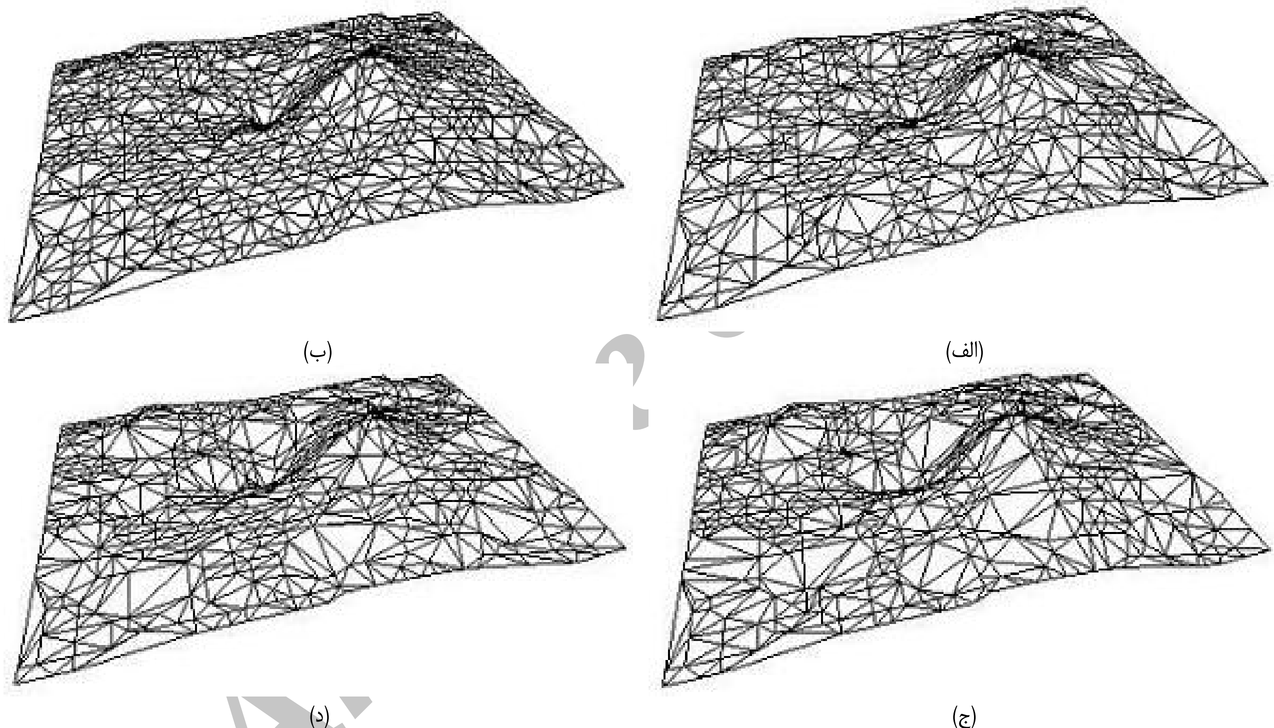
شکل ۱۰: تصویری از منطقه Denver.

جدول ۳: جزئیات ساده‌سازی بخشی از منطقه DENVER با ۶۲۵ نقطه با استفاده از دقت‌های مختلف.

سرزمین	تعداد نقاط	تعداد مثلث‌ها
سرزمین اصلی	۶۲۵	۱۱۵۲
سرزمین ساده‌شده با $\varepsilon = ۰٫۱$	۵۷۰	۱۰۴۴
سرزمین ساده‌شده با $\varepsilon = ۱$	۵۷۰	۱۰۴۴
سرزمین ساده‌شده با $\varepsilon = ۲٫۵$	۴۷۹	۸۶۴
سرزمین ساده‌شده با $\varepsilon = ۵$	۴۲۸	۷۶۵

جدول ۴: جزئیات ساده‌سازی بخشی از منطقه DENVER با ۵۲۰ نقطه با استفاده از دقت‌های مختلف.

سرزمین	تعداد نقاط	تعداد مثلثها
سرزمین اصلی	۵۲۰	۹۵۰
سرزمین ساده‌شده با $\varepsilon = ۰٫۵$	۴۴۵	۸۰۱
سرزمین ساده‌شده با $\varepsilon = ۱$	۳۸۲	۶۷۶



شکل ۱۱: مثالی از یک تین که با دقت‌های متفاوت ساده شده است، (الف) تین اصلی، (ب) تین ساده‌شده با دقت ۰٫۵، (ج) تین ساده‌شده با دقت ۱ و (د) تین ساده‌شده با دقت ۵.

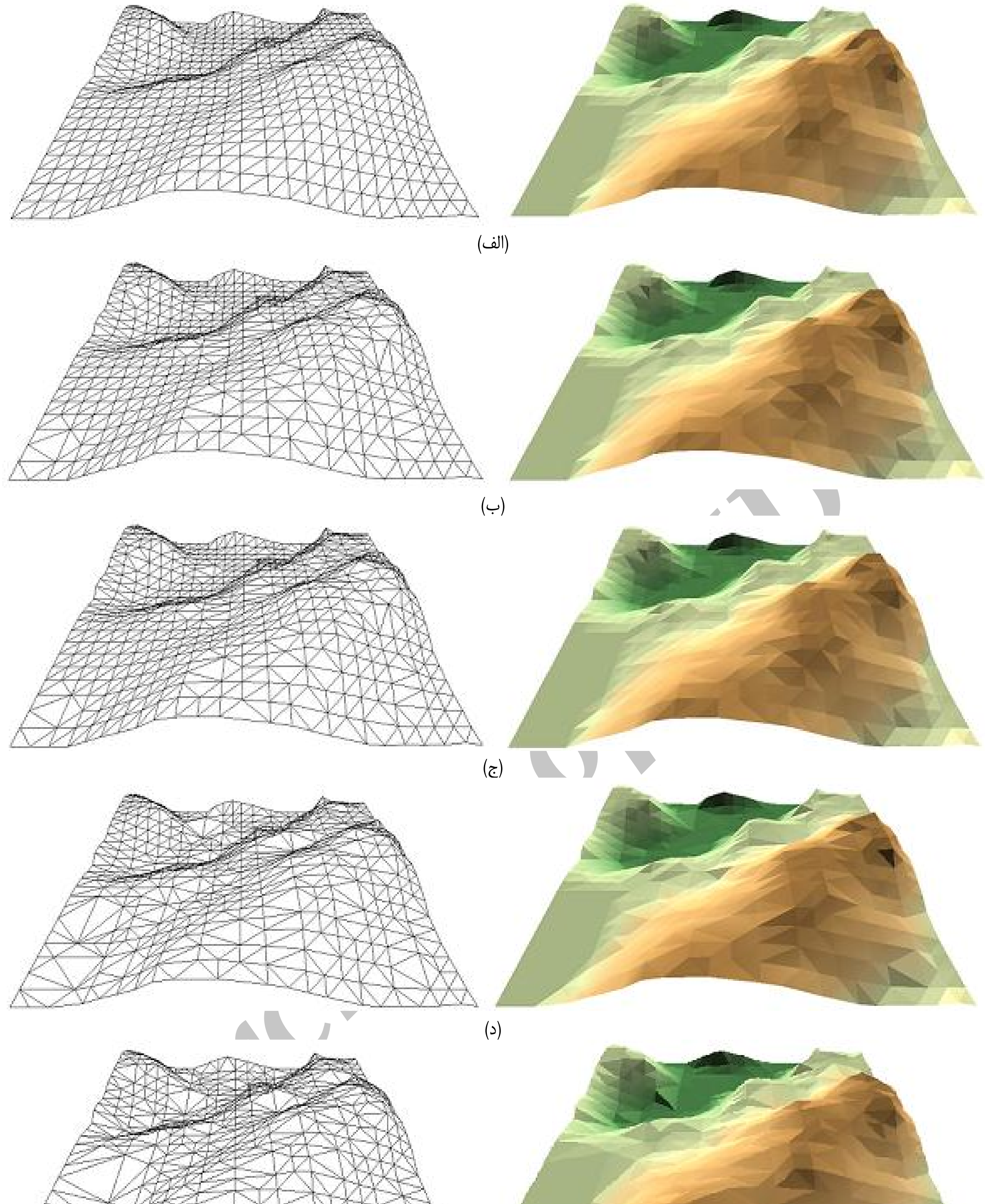
خروجی در جدول ذکر شده و تفاوت ارتفاع بین پایین‌ترین و بالاترین نقطه در این سرزمین ۲٫۵ متر می‌باشد.

همان طور که در بخش ۴ بیان شد این الگوریتم شکل تین اصلی و خاصیت Delaunay را تا حد ممکن حفظ می‌کند. در شکل ۲ می‌توان به صورت گرافیکی تین اصلی و تین‌های ساده‌شده را مشاهده کرد.

برای تست برنامه با داده‌های طبیعی از داده‌های موجود از منطقه Denver ایالت Colorado [۱۸] استفاده کرده‌ایم و تصویر واقعی این منطقه در شکل ۱۰ قابل مشاهده است. نقاط موجود در این سرزمین بسیار زیادند و برای راحتی تست برنامه قسمتی از این منطقه را انتخاب کرده‌ایم و به ساده‌سازی آن می‌پردازیم. در شکل ۱۱ و ۱۲ به‌ترتیب مراحل مختلف الگوریتم ساده‌سازی روی یک تین آزمایشی و بخشی از منطقه Denver با دقت‌های مختلف قابل نمایش است و در جدول ۳ جزئیات تعداد نقاط Denver اصلی و پس از ساده‌سازی‌ها بیان شده است.

در ستون اول جدول ۳ دقتی که برای ساده‌سازی سرزمین در نظر گرفته‌ایم آمده است و ستون دوم و سوم به ترتیب تعداد نقاط و تعداد مثلث‌های سرزمین را پس از ساده‌سازی نشان می‌دهند. توجه داشته باشید که این قطعه از سرزمین Denver دارای ۶۲۵ نقطه است که حداقل و حداکثر ارتفاع نقاط ۲۶۸۱ و ۲۸۰۳ متر می‌باشد. بنابراین حداکثر اختلاف ارتفاع بین دو نقطه می‌تواند ۱۲۲ متر باشد. ما برای ساده‌سازی حداکثر خطای ۰٫۵، ۱، ۲٫۵ و ۵ متر را در نظر گرفته‌ایم و نتایج حاصل را نشان داده‌ایم.

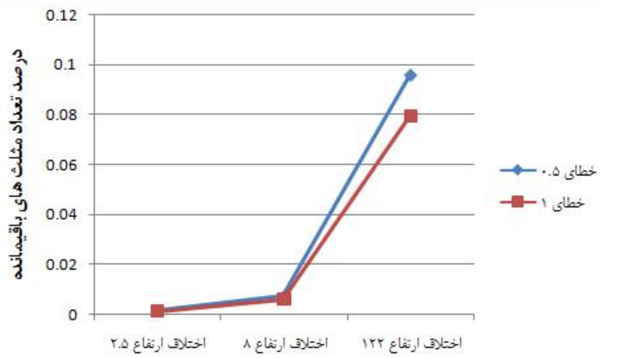
در جدول ۴ می‌توان نتایج حاصل از ساده‌سازی بخش دیگری از منطقه Denver را مشاهده کرد. در این منطقه سعی کردیم قطعه‌ای هموارتر نسبت به قطعه قبلی انتخاب کنیم و در اینجا تعداد نقاط ۵۲۰، حداقل و حداکثر ارتفاع ۱۸۸۱ و ۱۸۸۹ است که دو سطح ساده‌سازی با حداکثر خطای ۰٫۵ و ۱ متر روی آن اجرا شده است.



شکل ۱۲: ساده‌سازی بخشی از منطقه Denver با دقت‌های مختلف، (الف) تین اصلی، (ب) تین ساده‌شده با دقت ۰٫۵، (ج) تین ساده‌شده با دقت ۱، (د) تین ساده‌شده با دقت ۲٫۵ و (ه) تین ساده‌شده با دقت ۵.

همان طور که از نتایج به دست آمده در جدول ۴ مشاهده می‌شود، تعداد نقاط در خروجی ساده‌سازی به ازای εهای یکسان در تین‌های متفاوت به اختلاف ارتفاع (پستی و بلندی) نقاط آن بستگی دارد. به طور مثال درصد کاهش نقاط در تین جدول ۳ پس از ساده‌سازی به ازای $\varepsilon =$ ۰٫۵ از تین جدول ۴ کمتر است زیرا ناهمواری تین استفاده‌شده در جدول ۳ خیلی بیشتر از ناهمواری تین استفاده‌شده در جدول ۴ است.

با توجه به نتایج به دست آمده از سرزمین‌های مختلف می‌توان به این نتیجه رسید که هرچه اختلاف ارتفاع نقاط یک سرزمین بیشتر باشد ساده‌سازی کمتر انجام می‌گیرد، البته به تراکم نقاط در سرزمین نیز باید توجه کرد. برای واضح شدن این مطلب، معیار "درصد تعداد مثلث‌های باقیمانده پس از ساده‌سازی" را روی سرزمین‌های مورد آزمایش در نظر می‌گیریم. در شکل ۱۳ می‌توان نتایج حاصل از این معیار را روی سه سرزمین با پستی و بلندی‌های متفاوت به ازای ساده‌سازی $\varepsilon =$ ۰٫۵ یا $\varepsilon =$ ۱ مشاهده کرد.



شکل ۱۳: درصد مثلث‌های باقیمانده به ازای سرزمین‌ها با پستی و بلندی‌های متفاوت.

در این نمودار اعداد مشخص شده روی محور عمودی با استفاده از (۳) محاسبه شده‌اند به این صورت که ابتدا درصد مثلث‌های باقیمانده به دست آمده ولی چون تراکم سرزمین‌ها با هم تفاوت دارد، این اعداد را بر تعداد مثلث‌ها در واحد ارتفاع تقسیم کرده‌ایم. همان طور که مشخص است هرچه اختلاف ارتفاع به ازای ε یکسان بیشتر شود تعداد مثلث‌های حذف‌شده کمتر می‌شوند که این موضوع بدیهی است

$$\frac{\dfrac{\text{num of remind triangles}}{\text{num of total triangles}}}{\dfrac{\text{num of total triangles}}{\text{max height difference}}} \quad (۳)$$

٦- نتیجه‌گیری

در این مقاله به بررسی مسأله ساده‌سازی روی سرزمین‌های سه‌بعدی پرداختیم. مسأله ساده‌سازی این است که تعدادی از نقاط یک سرزمین حذف شود به نحوی که خطای سرزمین پس از ساده‌سازی، بیشتر از میزان تعیین‌شده نباشد. در ابتدا مروری بر روی کارهای انجام‌شده در این حوزه انجام شد و سعی کردیم یک دسته‌بندی کلی برای قالب اصلی الگوریتم‌های ساده‌سازی مطرح کنیم که در این دسته‌بندی الگوریتم‌های دقیق، تقریبی و حریصانه حاصل شد. هر کدام از این دسته‌ها به زیرشاخه‌های کوچک‌تری تقسیم می‌شود که برای هر زیرشاخه تا حد امکان، کاربردها و کارهایی که تاکنون انجام شده، مطرح شد.

کار اصلی انجام‌شده در این مقاله، ارائه یک الگوریتم هیبرید برای ساده‌سازی سرزمین است که در ورودی، یک سرزمین با n نقطه سه‌بعدی و یک درصد خطای ε دریافت می‌کند و در خروجی یک سرزمین ساده‌شده با تقریب ε حاصل شده و این ساده‌سازی در زمان $O(n^{۲}\sqrt{n})$ انجام می‌شود. در این الگوریتم ابتدا تمام نقاط موجود در سرزمین به خوشه‌هایی با خواص مشترک تقسیم‌بندی می‌شود و نقاطی که در یک خوشه قرار می‌گیرند از نظر فاصله اقلیدسی و پستی و بلندی خیلی نزدیک به هم هستند. در مرحله بعد نقاط موجود در هر خوشه به صورت مجزا ساده می‌شوند که در این مرحله یک الگوریتم ساده‌سازی در قالب الگوریتم‌های افزایشی مطرح شده است. پس از ساده‌سازی تمام خوشه‌ها، خوشه‌های ساده‌شده با هم ادغام می‌شوند.

با توجه به این که در الگوریتم پیشنهادی، ساده‌سازی در سطح هر خوشه انجام می‌گیرد و نقاط موجود در هر خوشه بسیار شبیه به هم هستند، دقت الگوریتم مربوط بالا می‌باشد. برای بررسی دقت الگوریتم، آن را روی چندین سرزمین واقعی با پستی و بلندی‌های متفاوتی مورد آزمایش قرار داده‌ایم که نتایج حاصل از ساده‌سازی روی سرزمین‌های متفاوت با معیارهای مشابهی مورد ارزیابی قرار گرفته‌اند.

با توجه به استفاده فراوان اطلاعات سرزمین‌ها در کاربردهای متداول مثل مسأله کوتاه‌ترین مسیرها، شبکه‌های رودخانه‌ای و سایر مسایل مرتبط به گراف، برای بالابردن سرعت پردازش‌ها می‌توان ابتدا به ساده‌سازی سرزمین‌ها پرداخت. از آنجایی که الگوریتم مطرح‌شده، ساده‌سازی را به صورت محلی و در سطح هر خوشه انجام می‌دهد، می‌توان از آن به عنوان یک روش مؤثر در ساده‌سازی استفاده کرد. از طرف دیگر اغلب الگوریتم‌های حریصانه بر اساس معیار تعداد نقاط پس از ساده‌سازی کار می‌کنند و هیچ تضمینی بر حداکثر خطا پس از ساده‌سازی ندارند اما این الگوریتم می‌تواند به صورت حریصانه مسأله را با معیار حداکثر خطای ε حل کند.

مراجع

[1] P. K. Agarwal and S. Surii, "Surface approximation and geometrics partitions," in *Proc. 5th ACM-SIAM Symp. Discrete Algorithms*, pp. 24-33, 1994.

[2] M. Garland and P. S. Heckbert, "Fast polygonal approximation of terrains and height fields," in *Technical Report, CMU-CS-95-181*, 1995.

[3] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry*, Springer-Verlag New York Inc, 3rd Edition, May 2008.

[4] J. Gudmundsson, M. Hammar, and M. van Kreveld, "Higher order delaunay triangulations," *Computational Geometry: Theory and Applications*, vol. 23, no. 1, pp. 85-98, Jul. 2002.

[5] M. Van Kreveld and R. I. Silveira, "Optimal higher order delaunay triangulations of polygons," *Computational Geometry: Theory and Applications*, vol. 42, no. 8, pp. 803-813, Oct. 2009.

[6] I. Dong Yun, K. Choo, and S. Uk Lee, "Mesh simplification using the edge attributes," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 1, pp. 1102-1115, Jan. 2002.

[7] C. Chuon and S. Guha, "Volume cost based mesh simplification," in *Proc. Int. Conf. on Computer Graphics, Imaging and Visualization*, pp. 164-169, 11-14 Aug. 2009.

[8] Y. Guoqing, C. Zhun, and W. Mao, "The new triangulation-simplify algorithm of tin," in *Proc. Int. Conf. on Computer Graphics and Virtual Reality*, *CGVR'06,* pp. 79-85, 2006.

[9] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. of the 24th Annual Conf. on Computer Graphics and Interactive Techniques*, *SIGGRAPH'97*, pp. 209-216, New York, NY, US, 1997.

[10] W. Zhao, T. Tang, H. Gong, F. Duan, and Y. Mo, "Dynamic data retrieval and distance decay of triangulated irregular network (TIN)," *Annals of GIS*, vol. 12, no. 1, pp. 21-26, 2006.

[11] W. Xie and J. Lu, "An improved simplification algorithm of 3d terrain visualization model," in *Proc. IAPRS Integrated System for Spatial Data Production, Custodian and Decision Suppor*t, pp. 545-548, 20-23 Aug. 2002.

[12] J. Wang, L. Wang, J. Li, and I. Hagiwara, "A feature preserved mesh simplification algorithm," *J. of Engineering and Computer Innovations*, vol. 2, no. 6, pp. 98-105, Jun. 2011.

[13] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proc. 24th Annual Conf. on Computer Graphics and Interactive Techniques*, *SIGGRAPH '97*, 1997.

[14] P. S. Heckbert and M. Garland, "Optimal triangulation and quadric-based surface simplification," *J. of Computational Geometry: Theory and Applications*, vol. 14, no. 1-3, pp. 49-65, Nov. 1999.

[15] J. Wei and Y. Lou, "Feature preserving mesh simplification using feature sensitive metric," *J. of Computer Science and Technology*, vol. 25, no. 3, pp. 595-605, May 2010.

[۱۶] ف. دباغی و م. قدسی، "یک الگوریتم تقریبی برای ساده‌سازی سرزمین،" *شانزدهمین کنفرانس ملی سالانه انجمن کامپیوتر ایران، دانشگاه صنعتی شریف، تهران*، صص. ۵۹۵-۵۹۱، ۱۳۸۹.

[17] P. K. Agarwal and P. K. Desikan, "An efficient algorithm for terrain simplification," in *Proc. 8th ACM-SIAM Symp. on Discrete Algorithms*, pp. 139-147, 1997.

[18] *Software, Lakes Environmental*, Web gis. http://www.webgis.com/

فهیمه دباغی زرندی تحصیلات خود را در مقطع کارشناسی و کارشناسی ارشد مهندسی کامپیوتر گرایش نرم‌افزار به‌ترتیب در سال‌های ۱۳۸۷ و ۱۳۸۹ در دانشگاه فردوسی مشهد و دانشگاه صنعتی شریف به پایان رسانده است و هم‌اکنون هیأت‌علمی دانشکده مهندسی دانشگاه ولی‌عصر(عج) رفسنجان می‌باشد. نام‌برده تحصیلات خود را در مقطع دکتری مهندسی کامپیوتر گرایش نرم‌افزار در سال ۱۳۹۲ در دانشگاه علم و صنعت ایران شروع کرده است. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: الگوریتم‌های کامپیوتری و کاربردهای آن، داده‌کاوی، شبکه‌های کامپیوتری و بهینه‌سازی مصرف انرژی در شبکه‌های کامپیوتری.

محمد قدسی تحصیلات خود را در مقطع کارشناسی مهندسی الکترونیک و کارشناسی ارشد مهندسی الکترونیک و علوم کامپیوتر به‌ترتیب در سال‌های ۱۳۵۴ و ۱۳۵۷ در دانشگاه صنعتی شریف و دانشگاه برکلی، کالیفرنیا به پایان رسانده است. نام‌برده تحصیلات خود را در مقطع دکتری علوم کامپیوتر در سال ۱۳۶۸ در دانشگاه پنسیلوانیا به پایان رسانده است و هم‌اکنون استاد دانشکده مهندسی کامپیوتر دانشگاه صنعتی شریف می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان عبارتند از: هندسه محاسباتی، الگوریتم‌ها و ساختار داده‌های کارا، الگوریتم‌های موازی.