یک الگوریتم تقریبی برای ساده‌سازی سرزمین

فهیمه دباغی[۱]، محمد قدسی[۲]

[۱] مربی، استاد مدعو گروه علوم‌کامپیوتر، دانشگاه شهیدباهنر، کرمان
dabaghi.fahim@gmail.com

[۲] استاد، گروه مهندسی کامپیوتر گرایش نرم‌افزار، دانشگاه صنعتی‌شریف، تهران
ghodsi@sharif.ir

چکیده

در این مقاله، یک الگوریتم تقریبی برای ساده‌سازی سرزمین[۱] مطرح شده است. هدف مسئله 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 اینکه پس از ساده‌سازی، m نقطه با حداقل خطا در سرزمین وجود داشته باشد یا اینکه، حداکثر خطا پس از ساده‌سازی به ازای کمترین تعداد نقاط، ϵ باشد. این مسئله در حوزه‌ی مسائل ان‌پی – سخت [۲] قرار دارد.

در این راستا، ما یک الگوریتم تقریبی برای ساده‌سازی سرزمین بیان کرده‌ایم که، یک سرزمین با n نقطه در فضای سه بعدی و حداکثر خطای $\epsilon > 0$ را دریافت می‌کند و در خروجی یک سرزمین ساده‌شده با سایز $O(k \log k)$ در زمان $O(n^7)$ حاصل می‌شود، که k سایز بهینه‌ی سرزمین ساده‌شده به ازای تقریب - ϵ می‌باشد.

کلمات کلیدی

سرزمین، ساده‌سازی[۳]، ذوزنقه‌ی کانونی، تقسیم‌بندی، الگوریتم تقریبی، برنامه‌سازی پویا.

۱- مقدمه

با توجه به استفاده فراوان داده‌های جغرافیایی در پردازش‌های آماری در حوزه‌هایی از قبیل سیستم‌های اطلاعات جغرافیایی، گرافیک‌های کامپیوتری، پردازش تصویر و غیره مسئله ساده‌سازی سطوح مطرح شده‌است. دلیل تمایل به ساده‌سازی این است که، وجود اطلاعات با جزئیات بسیار بالا باعث پایین آمدن سرعت محاسبات در پردازش‌ها شده‌است، 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 کم اهمیت 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 عمومی‌در مسائل ساده‌سازی، مسئله ساده‌سازی سرزمین است که به صورت زیر تعریف می‌شود.

یک مجموعه S از n نقطه در فضای سه بعدی (هر نقطه با تابع $z = f(x, y)$ مشخص می‌شود) و یک پارامتر $\epsilon > 0$ داده می‌شود. در خروجی الگوریتم ساده‌سازی یک سرزمین ساده‌شده با تقریب - ϵ ایجاد می‌شود که شامل یک مجموعه S' از m نقطه است که حداقل نقاط ممکن به ازای خطای داده شده را شامل می‌شود. نقاط خروجی با یک تابع $g(x, y)$ نمایش داده می‌شوند، که رابطه زیر برای آنها برقرار می‌باشد.

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

بنابراین خطایی که در اینجا برای ساده‌سازی در نظر گرفته‌ایم، برابر است با حداکثر فاصله عمودی نقاط حذف‌شده از سرزمین ساده‌شده.

آگاروال و سوری [1] اثبات کردند که این مسئله ان پی – سخت است. به همین دلیل اکثر کارها در این حوزه به صورت الگوریتم‌های حریصانه یا تقریبی هستند. در این مقاله، ما یک الگوریتم تقریبی بر

اساس الگوریتم تقریبی بیان‌شده توسط آگاروال و سوری [1] بیان کرده‌ایم. الگوریتم ارائه‌شده ایشان مسئله را در زمان $O(n^8)$ با سایز $O(k \log k)$ حل کرده است، که k سایز بهینه سرزمین ساده‌شده با حداکثر خطای ϵ می‌باشد. که ما با بهبودی در این الگوریتم مرتبه آن را یک توان کاهش داده‌ایم و آن را به $O(n^7)$ رسانده‌ایم.

تاکنون الگوریتم‌های زیادی در حوزه ساده‌سازی سرزمین بیان شده‌اند که اغلب آنها در حوزه مکاشفه‌ای و حریصانه هستند. چندین الگوریتم تقریبی نیز بیان شده‌است؛ همانند الگوریتم آگاروال و سوری [1] که بحث شد. آگاروال و دسیکان [2] در یک کار دیگر، یک الگوریتم تقریبی ارائه دادند که در زمان $O\left(n^{2+\delta} + c^3 \log^2 c \log \frac{n}{c}\right)$ یک سرزمین را به سرزمینی با سایز $O(c^2 \log^2 c)$ ساده می‌کند، که c سایز بهینه تقریب - ϵ با حداقل تعداد نقاط می‌باشد.

در بسیاری از کاربردها، نقاط یک سرزمین مثلث‌بندی می‌شوند، که سطح حاصل پس از مثلث‌بندی، شبکه نامنظم مثلث‌بندی شده[4](تین) نامیده می‌شود. از این رو به جای ساده‌سازی سرزمین می‌توان به ساده‌سازی تین پرداخت. عموماً دو روش حریصانه برای ساده‌سازی تین به کار برده می‌شود که روش‌های افزایشی[5] و کاهشی[6] نام دارند. در روش‌های افزایشی از یک تین کلی(عموماً دو مثلث) شروع می‌شود و در هر مرحله نقاط با بیشترین خطا به تین اضافه می‌شوند و تین مجدداً مثلث‌بندی می‌شود و در روش کاهشی از تین اصلی شروع می‌کنند و در هر مرحله نقاطی که کمترین خطا را ایجاد می‌کنند از تین حذف می‌شود و حفره ایجاد شده مجدداً مثلث‌بندی می‌شود.

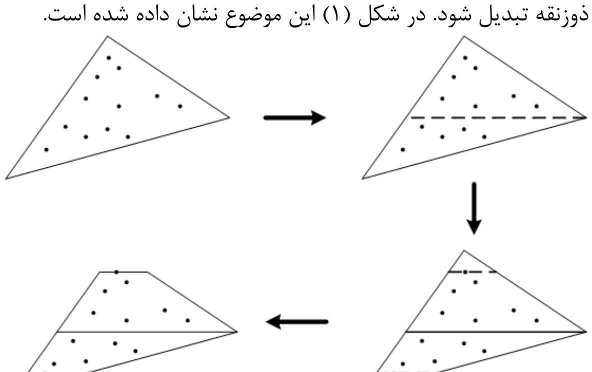
الگوریتم‌های بسیاری در حوزه کاهشی [3] بیان شده‌اند که برای کاهش در هر مرحله از انقباض لبه استفاده می‌کنند(در انقباض لبه، یک لبه با کمترین خطا انتخاب می‌شود و به یک نقطه تبدیل می‌شود) و در بسیاری از کارهای دیگر [4,5,6] برای کاهش از حذف یک نقطه استفاده می‌کنند.

الگوریتم‌های زیادی در حوزه افزایشی ارائه نشده‌اند زیرا قالب همه آنها یکسان می‌باشد و تنها تفاوت در خطای درنظرگرفته می‌باشد و در هر مرحله یه نقطه با بیشترین خطا به تین اضافه می‌شود. یکی از روش‌های خوب در این حوزه، کار گارلند و هکبرت [7] می‌باشد.

۲- مفاهیم اولیه

برای ساده‌سازی سرزمین، می‌توان از مفهوم تین در فضای دوبعدی استفاده کرد. به این صورت که ابتدا تمام نقاط روی سطح xy تصویر می‌شوند و هر نقطه $p \in S$ در فضای سه بعدی روی نقطه‌ی $\bar{p} \in \bar{S}$ در فضای دوبعدی تصویر می‌شود، حال می‌توان سطح را به مجموعه‌ای از مثلث‌های مجزا در دوبعد تقسیم کرد به نحوی که تمام نقاط را در برگیرند، بنابراین می‌توان تمام نقاط داخل یک مثلث را حذف کرد و مثلث‌های باقیمانده، تین حاصل پس از ساده‌سازی را نشان می‌دهند. البته برای این منظور باید مثلث‌های انتخابی معتبر باشند. به این صورت که، مثلث Δ_{abc} معتبر نامیده می‌شود اگر به ازای همه نقاط $p \epsilon S$ که تصویر آنها داخل مثلث Δ_{abc} قرار می‌گیرد این رابطه برقرار باشد که؛ حداکثر فاصله p از صفحه - abc در فضای سه بعدی ϵ باشد. در این صورت اگر تمام نقاط داخل مثلث‌ها بعد از تقسیم‌بندی حذف شوند، حداکثر خطای مجاز پس از ساده‌سازی، یعنی ϵ نیز رعایت می‌شود. با جزئیات گفته شده می‌توان مسئله ساده‌سازی سرزمین را به مسئله تقسیم‌بندی سرزمین به مثلث‌های معتبر دو به دو متمایز تبدیل کرد.

همان طور که گفته شد این مقاله بر اساس کار آگاروال و سوری [1] می‌باشد. در آنجا به جای مفهوم مثلث از ذوزنقه استفاده شده است، همانطور که مشخص است هر مثلث به راحتی می‌تواند به دو ذوزنقه تبدیل شود. در شکل (۱) این موضوع نشان داده شده است.



شکل (۱): مراحل تبدیل یک مثلث به دو ذوزنقه

بنابراین در ساده‌سازی می‌توان بجای مفهوم مثلث از ذوزنقه استفاده کرد که لم ۱ این موضوع را نشان می‌دهد.

لم ۱: اگر یک سرزمین ورودی با n نقطه داشته باشیم و بتوان یک تقریب - ϵ از آن با استفاده از مثلث‌ها با سایز k مثلث ایجاد کرد می‌توان یک تقریب - ϵ با استفاده از ذوزنقه‌ها با حداکثر سایز $2k$ ذوزنقه ایجاد کرد.

۲-۱- تقسیم‌بندی ذوزنقه ای کانونی

این بخش مطابق بخش ۳ در مقاله آگاروال و سوری [1] می‌باشد. ذوزنقه‌هایی که برای تقسیم‌بندی سرزمین استفاده می‌شوند باید دارای قالب مشخصی باشند. لبه‌های بالا و پایین ذوزنقه‌ها باید موازی محور xها باشند.

یک مجموعه $\bar{S}$ از n نقطه در فضای دو بعدی داده‌شده‌است که این نقاط از تصویر نقاط مجموعه S در فضای سه بعدی حاصل شده اند. یک خانواده از ذوزنقه‌های $\boldsymbol{\Delta} = \{\Delta_1, \Delta_2, \ldots, \Delta_{\mathrm{m}}\}$ یک تقسیم‌بندی ذوزنقه‌ای معتبر نامیده می‌شود اگر شرایط زیر را دارا باشد:

- همه‌ی ذوزنقه‌های $\Delta \in \boldsymbol{\Delta}$ باید معتبر باشند.
- $\boldsymbol{\Delta}$ باید همه نقاط موجود در $\bar{S}$ را پوشش دهد.
- ذوزنقه‌های موجود در $\boldsymbol{\Delta}$ باید دو به دو متمایز باشند.

حال بر طبق تعاریف بیان شده در مقاله آگاروال و سوری [1] ذوزنقه‌ی کانونی را تعریف می‌کنیم. یک ذوزنقه کانونی است، اگر تمام اضلاع آن جزو خطوط کانونی باشند. مجموعه خطوط کانونی شامل تمام خطوط افقی هستند که از نقاط موجود در $\bar{S}$ بگذرند و یا خطوطی که از دو نقطه در مجموعه $\bar{S}$ بگذرند، بنابراین این مجموعه حداکثر شامل $O(n^2)$ خط می‌باشد. حال یک تقسیم‌بندی ذوزنقه‌ای، تقسیم‌بندی ذوزنقه‌ای کانونی نامیده می‌شود، اگر همه ذوزنقه‌های آن کانونی باشند.

لم ۲: هر تقسیم‌بندی ذوزنقه‌ای روی $\bar{S}$ با k ذوزنقه می‌تواند به یک تقسیم‌بندی ذوزنقه‌ای کانونی روی $\bar{S}$ با حداکثر $4k$ ذوزنقه کانونی تبدیل شود.

اثبات: اثبات این لم در مقاله آگاروال و سوری [1] بیان شده است.□

ما در این مقاله یک تقسیم‌بندی ذوزنقه‌ای کانونی با محدودیت بیشتری به کار برده‌ایم. به این صورت که هر ذوزنقه کانونی باید قائم‌الزاویه باشد، یعنی یکی از اضلاع آن باید از یک خط عمودی که از نقاط موجود در $\bar{S}$ می‌گذرد، عبور کند. با اضافه‌کردن این محدودیت، خطوط عمودی که از نقاط $\bar{S}$ می‌گذرند نیز به عنوان خطوط کانونی درنظرگرفته می‌شوند. حال بر طبق لم ۲، لم ۳ را تعریف می‌کنیم.

لم ۳: : هر تقسیم‌بندی ذوزنقه‌ای روی $\bar{S}$ با k ذوزنقه می‌تواند به یک تقسیم‌بندی ذوزنقه‌ای کانونی قائم‌الزاویه روی $\bar{S}$ با حداکثر $8k$ ذوزنقه کانونی قائم‌الزاویه تبدیل شود.

اثبات: زیرا هر ذوزنقه کانونی با رسم یک خط عمودی از وسط می‌تواند به دو ذوزنقه کانونی قائم‌الزاویه تبدیل شود.□

۳- تقسیم‌بندی تفکیک‌پذیر بازگشتی

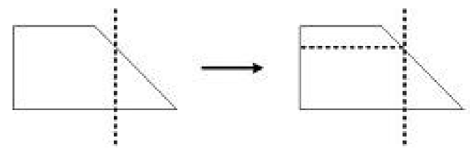
این بخش مطابق بخش ۵ از مقاله آگارول و سوری [1] می‌باشد. در این مقاله ما یک الگوریتم بر اساس روش برنامه‌سازی پویا پیشنهاد داده‌ایم که می‌تواند مسئله تقسیم‌بندی سرزمین به ذوزنقه‌های کانونی قائم‌الزاویه را به صورت بهینه حل کند. از آنجایی که در روش برنامه‌سازی پویا باید مسئله به صورت بازگشتی قابل تقسیم به زیرمسئله‌های مجزا باشد، ابتدا تقسیم‌بندی ذوزنقه‌ای تفکیک‌پذیر بازگشتی را تعریف می‌کنیم. یک تقسیم‌بندی ذوزنقه‌ای $\boldsymbol{\Delta}$، تفکیک‌پذیر بازگشتی نامیده می‌شود اگر شرایط زیر را دارا باشد:

- $\boldsymbol{\Delta}$ تنها یک ذوزنقه واحد را شامل شود، یا
- یک خط l وجود داشته باشد که با درون هیچ یک از ذوزنقه‌های $\Delta \in \boldsymbol{\Delta}$ تداخل نداشته باشد و مجموعه $\boldsymbol{\Delta}$ را به دو تقسیم‌بندی تفکیک‌پذیر بازگشتی $\boldsymbol{\Delta}^+ = \boldsymbol{\Delta} \cap l^+$ و $\boldsymbol{\Delta}^- = \boldsymbol{\Delta} \cap l^-$ تقسیم کند. که l^+ و l^- نیم صفحات تعریف شده توسط l هستند.

لم ۴: اگر $\bar{S}$ یک مجموعه محدود از n نقطه در صفحه باشد و $\boldsymbol{\Delta}$ یک تقسیم‌بندی ذوزنقه‌ای از $\bar{S}$ با k ذوزنقه باشد، آنگاه یک تقسیم‌بندی ذوزنقه‌ای قائم‌الزاویه تفکیک‌پذیر بازگشتی $\boldsymbol{\Delta}^*$ از $\bar{S}$ با $O(k \log k)$ ذوزنقه وجود دارد.

اثبات: فرض می‌کنیم که ذوزنقه T یک ذوزنقه کانونی قائم‌الزاویه‌ی باشد که برخی از ذوزنقه‌های داخل $\boldsymbol{\Delta}$ را پوشش دهد به نحوی که لبه‌های بالا و پایین آن از رئوس $\boldsymbol{\Delta}$ عبور کنند و لبه‌های چپ و راست T نیز جزئی از لبه‌های $\boldsymbol{\Delta}$ باشند و ذوزنقه T هیچکدام از ذوزنقه‌های $\boldsymbol{\Delta}$ را قطع نمی‌کند. بنابراین، $\boldsymbol{\Delta}_T$ به ذوزنقه‌های از $\boldsymbol{\Delta}$ اشاره می‌کند که داخل T قرار دارند و V_T نیز به نقاطی اشاره می‌کند که داخل T قرار می‌گیرند. لبه‌های $\boldsymbol{\Delta}_T$ نمی‌توانند لبه‌های چپ و راست T را قطع کنند، زیرا بخشی از این لبه‌ها هستند، بنابراین لبه‌های $\boldsymbol{\Delta}_T$ یا کاملاً داخل T قرار دارند یا لبه‌های بالا و پایین T را قطع می‌کنند. در ابتدا T، ذوزنقه کانونی قائم‌الزاویه بزرگی در نظر گرفته می‌شود که کل مجموعه $\boldsymbol{\Delta}$ را در خود جای می‌دهد.

اگر $|\boldsymbol{\Delta}_T| = 1$ باشد، $\boldsymbol{\Delta}_T$ را بر می‌گردانیم و توقف می‌کنیم. در غیر این صورت، به صورت گفته‌شده عمل می‌کنیم: اگر یک ذوزنقه $\Delta \in \boldsymbol{\Delta}$ وجود داشته باشد که کاملاً در T قرار داشته باشد(که لبه‌های بالا و پایین آن بر لبه‌های بالا و پایین T منطبق باشند) آنگاه: اگر Δ سمت چپ ترین(راست ترین) ذوزنقه از T باشد می‌توان T را به دو ذوزنقه T_1 و T_2 تقسیم کرد با استفاده از خط عبوری از لبه راست(چپ) ذوزنقه Δ، بنابراین T_1 شامل Δ و T_2 شامل بقیه ذوزنقه‌ها می‌شود. البته برای اینکه ذوزنقه‌های ما باید قائم‌الزاویه باشند یک محدودیت در انتخاب لبه غیر عمودی وجود دارد که در این صورت باید به این گونه عمل کرد: اگر لبه غیر عمود Δ برای تفکیک T استفاده شود برای اینکه باز خاصیت قائم‌الزاویه حفظ شود یک خط عمودی بر اساس نقطه با x میانه در بین نقاط داخل T_2 رسم می‌کنیم، که این خط عمودی ممکن است بعضی از ذوزنقه‌ها را خراب کند که در شکل (۲) (ذوزنقه سمت چپ) این حالت نشان داده‌شده‌است.



شکل (۲): اصلاح مثلث نامناسب بعد از تفکیک

در این حالت به ازای تمام ذوزنقه‌های خراب‌شده یک خط افقی از نقطه خرابی ذوزنقه رسم می‌کنیم که در صورت کانونی نبودن آن را به داخل انتقال می‌دهیم تا یکی از نقاط V_T را شامل شود، بنابراین ذوزنقه‌های موجود کانونی می شوند، این مراحل نیز در شکل (۲) (ذوزنقه سمت راست) قابل مشاهده می‌باشد.

حال اگر ذوزنقه‌ای وجود نداشته باشد که کاملاً در T قرار داشته باشد، یک خط افقی بر اساس نقطه با y میانه رسم می‌کنیم و T را به دو ذوزنقه تقسیم می‌کنیم.

با توجه به اینکه انتخاب خطوط عمودی بر اساس میانه است و در ابتدا فقط k خط عمودی داریم بنابراین این تقسیم‌بندی می‌تواند $\mathrm{O}(\log \mathrm{k})$ مرتبه ذوزنقه‌ها را تقسیم کند. همان طور که مشخص است هر خط عمودی(در صورت خراب کردن ذوزنقه ها) در هر مرحله می‌تواند $\mathrm{O}(\mathrm{k})$ خط افقی ایجاد کند. بنابراین تعداد خطوط افقی $\mathrm{O}(\mathrm{k} + \mathrm{k}\log \mathrm{k})$خواهد بود. هر خط افقی نیز بر اساس میانه انتخاب می‌شود و می‌تواند در هر مرحله ذوزنقه‌ها را به دو بخش تقسیم کند، پس در کل تعداد این مراحل می‌تواند $\mathrm{O}(\log\{\mathrm{k}\log \mathrm{k}\})$ باشد.

حال اگر تعداد تقسیمات خطوط افقی و عمودی را روی ذوزنقه‌ها محاسبه کنیم در نهایت $\{\mathrm{O}(\mathrm{k}\log \mathrm{k}) + \mathrm{O}(\mathrm{k}\log\{\mathrm{k}\log \mathrm{k}\})\}$ ذوزنقه ایجاد خواهد شد بنابراین اگر جواب بهینه k ذوزنقه داشته باشد در اینجا اثبات کردیم که جواب ما $\mathrm{O}(\mathrm{k}\log \mathrm{k})$ ذوزنقه کانونی قائم‌الزاویه خواهد داشت. □

با توجه به تبدیلات انجام‌شده می‌توان به جای حل ساده‌سازی سطح مثلث‌بندی‌شده از مسئله تقسیم‌بندی بر اساس ذوزنقه کانونی قائم‌الزاویه تقسیم‌پذیر بازگشتی استفاده کرد که برای حل این مسئله به راحتی می‌توان از روش برنامه سازی پویا یک راه حل دقیق ارائه داد.

قضیه۱: $\bar{\mathrm{S}}$ یک مجموعه از n نقطه در سطح xy می‌باشد. اگر یک تقسیم‌بندی مثلثی معتبر روی $\bar{S}$ با سایز k وجود داشته باشد، آنگاه یک تقسیم‌بندی ذوزنقه‌ای قائم‌الزاویه روی $\bar{S}$ با سایز $O(k\log k)$ وجود دارد.

در ادامه کار یک الگوریتم زمان چندجمله‌ای بر اساس برنامه سازی پویا بیان می‌شود که یک تقسیم‌بندی از سطح را می‌دهد. طبق قضیه ۱ این تقسیم‌بندی دارای سایز $O(k\log k)$ می‌باشد.

۴- الگوریتم تقریبی پیشنهادی

یک زیرمجموعه از نقاط $R \subseteq \bar{S}$ در نظر بگیرید که یک ذوزنقه کانونی قائم‌الزاویه Δ تمام نقاط آن را شامل می‌شود. $\sigma(\mathrm{R}, \Delta)$ به سایز تقسیم‌بندی بهینه‌ی R در Δ اشاره می‌کند. منظور از سایز یک تقسیم‌بندی، تعداد ذوزنقه‌های آن است. که با توجه به مطالب گفته‌شده فرمول بازگشتی آن به صورت زیر می‌باشد:

$$\sigma(\mathrm{R}, \Delta) = \begin{cases} 1 & \text{if } \Delta \text{ is valid} \\ \min\left\{\min\limits_{l} A, \min\limits_{l', l''} B\right\} & \text{otherwise} \end{cases} \qquad (۲)$$

در این معادله، A برابر است با $\sigma(\mathrm{R}^{+}, \Delta) + \sigma(\mathrm{R}^{-}, \Delta)$ و B برابر است با $\sigma(\mathrm{R}_1, \Delta) + \sigma(\mathrm{R}_2, \Delta) + \sigma(\mathrm{R}_3, \Delta)$ و l مجموعه همه خطوطی است که افقی یا عمودی هستند که Δ را خراب نمی‌کنند، و l'و l'' دو خط به این صورت هستند: اگر l' یک خط غیر عمودی باشد که با لبه‌های بالا و پایین Δ اشتراک دارد و Δ را به دو ذوزنقه Δ_1 و Δ_2 تقسیم کند و یکی از آنها قائم‌الزاویه نیست (فرض کنید Δ_1)، بنابراین l'' یک خط عمودی است که از نقطه‌ای با x میانه داخل Δ_1 رسم می‌شود. اگر l' یک خط عمودی باشد، آنگاه Δ را به دو ذوزنقه تقسیم می‌کند که یکی خراب است بنابراین طبق شکل (۲) ، خط l''یک خط افقی می‌شود که از نقطه خرابی عبور می‌کند. بنابراین R_1، R_2 و R_3 سه ناحیه ایجاد شده توسط خطوط l' و l'' می باشند. هدف برنامه سازی پویا این است که $\sigma(\bar{\mathrm{S}}, \Delta)$ را محاسبه کند. که Δ یک ذوزنقه است که تمام نقاط $\bar{\mathrm{S}}$ را شامل می‌شود.

هر ذوزنقه کانونی قائم‌الزاویه می‌تواند با یک پنج تایی $(\mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2)$ تعریف شود که همه اعدادی بین ۱ تا n هستند. دو عدد اول مربوط به دو نقطه $\mathrm{p_i}$ و p_j هستند که ضلع بالا و پایین Δ از آن عبور می‌کنند، p_k به نقطه‌ای اشاره‌ای می‌کند که خط عمودی آن شامل ضلع عمود ذوزنقه است و به همین صورت p_{l_1} و p_{l_2} به دو نقطه مربوط به خط مربوط به ضلع دیگر ذوزنقه اشاره می‌کنند. بنابراین $\Delta(\mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2)$ به پنج گانه $(\mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2)$ اشاره می‌کند و اگر ذوزنقه‌ای از این پنج گانه ایجاد نشود می‌گوییم $\Delta(i, j, k, l_1, l_2)$ تعریف نشده‌است. $\sigma(\mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2)$ به سایز یک تقسیم‌بندی بهینه از نقاط موجود در $\bar{\mathrm{S}}(i, j, k, l_1, l_2)$ اشاره می‌کند.

اکنون، ما یک الگوریتم مبتنی بر برنامه سازی پویا بیان می‌کنیم که می‌تواند مسئله تقسیم‌بندی تفکیک‌پذیر بازگشتی را حل کند.

اگر $\Delta\ (i, j, k, l_1, l_2)$ معتبر باشد، آنگاه $\sigma(\mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2) = 1$. در غیر اینصورت $\sigma(\mathrm{i}, \mathrm{j}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2)$ برابر است با مینیمم مقدار از بین مقادیر زیر:

- ✓ $\min_{\mathrm{i} \le \mathrm{u} \le \mathrm{j}}\{\sigma(\mathrm{i}, \mathrm{u}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2) + \sigma(\mathrm{u}, \mathrm{j}, \mathrm{k}, \mathrm{l}_1, \mathrm{l}_2)\}$

- ✓ $\min_{v_1,v_2 \in \bar{S}\cap\Delta(i,j,k,l_1,l_2),v_2 \le u \le l_1} \{\sigma(i,j,k,v_1,v_2) + \sigma i,j,u,v1,v2+\sigma i,j,u,l1,l2$

- ✓ $\min_{u \in \bar{S}\cap\Delta(i,j,k,l_1,l_2)} \sigma(i,j,k,u,u) + \sigma(i,j,u,l_1,l_2)$

همان طور که مشخص است، ما در اینجا تمام حالات برای تقسیم یک ذوزنقه کانونی قائم‌الزاویه، به ذوزنقه‌های کانونی قائم‌الزاویه کوچکتر را لحاظ کرده‌ایم و از بین آنها، آن تقسیم‌بندی که تعداد کمتری دارد را در نظر گرفته‌ایم. بنابراین تمام خطوط افقی و عمودی و مورب را برای تقسیم‌بندی در نظر گرفته‌ایم، و به دلیل اینکه این خطوط باید از نقاط موجود در مجموعه $\bar{S}$ عبور کنند (خطوط کانونی)، حداکثر $O(n^2)$ خط در هر مرحله برای بررسی تقسیم‌بندی وجود دارد.

اگر Ξ مجموعه ذوزنقه‌های کانونی قائم‌الزاویه باشد، آنگاه تعداد اعضای این مجموعه برابر است با $O(n^5)$ ، زیرا به ازای هر پنج گانه حداکثر یک ذوزنقه داریم. اگر $Q(n)$ زمان تصمیم گیری برای تعیین معتبر بودن Δ باشد(این زمان از مرتبه $O(n)$ می‌باشد)، برای بررسی این محدودیت روی همه ذوزنقه‌ها به $O(n^5Q(n))$ زمان لازم داریم. سپس آنهایی که $\sigma(\Delta) = 1$ نباشد را به صورت بازگشتی حساب می‌کنیم که زمان $O(n^2)$ برای بازگشت لازم است (طبق معادلاتی که برای برنامه سازی پویا در بالا ارائه شد، حداکثر $O(n^2)$ حالت برای محاسبه بازگشت لازم است). بنابراین در کل مرتبه الگوریتم $O(n^7)$ می‌شود.

قضیه ۲: یک مجموعه $\bar{S}$ از n نقطه در سطح داده شده است، یک تقسیم‌بندی هندسی معتبر از $\bar{S}$ با سایز $O(k \log k)$ ذوزنقه وجود دارد که k تعداد ذوزنقه‌ها در حالت بهینه می‌باشد. این الگوریتم در بدترین حالت در زمان $O(n^7 + n^5Q(n))$ اجرا می‌شود.

۵- نتیجه

در این مقاله، ما یک الگوریتم تقریبی برای ساده‌سازی سرزمین بیان کردیم که می‌تواند مسئله ساده‌سازی سرزمین را در زمان $O(n^7)$ با سایز $O(k \log k)$ حل کند. که k سایز بهینه تقریب - ϵ می‌باشد. همان طور که گفته این مقاله بهبودی برای مرتبه زمانی الگوریتم ارائه شده توسط آگاروال و سوری [1] می‌باشد.

مراجع

[1] Pankaj K. Agarwal, Subhash Suri. Surface approximation and geometric partitions. In 5th ACM-SIAM Sympos. Discrete Algorithms, pages 24-33, 1994.

[2] Pankaj K. Agarwal, Pavan K.Desikan. An efficient algorithm for terrain simplification. ACM-SIAM Symposium on Discrete Algorithms, 8, 1997.

[3] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 209{216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[4] Il Dong Yun, Sang Uk Lee, Kyowoong Choo. Mesh simplification using the edge attributes. EURASIP J. Appl. Signal Process, 1:1102-1115, 2002.

[5] Chansophea Chuon, Sumanta Guha. Volume cost based mesh simplification. Computer Graphics, Imaging and Visualization, International Conference on, 0:164-169, 2009.

[6] Yao Guoqing, Chen Zhun, Wang Mao. The new triangulation-simplify algorithm of tin. CGVR: International Conference on Computer Graphics and Virtual Reality, 2006.

[7] Michael Garland, Paul S. Heckbert. Fast polygonal approximation of terrains and height fields. Technical Report CMU-CS-95-181 Also appears in SIGGRAPH 96 Visual Proceedings: Technical Sketches, 1995.

زیرنویس‌ها

---

1 Terrain
2 NP-Hard
3 Simplification
4 Triangulated Irregular Network(TIN)
5 Refinement
6 Decimation