To Appear, Proceedings of 7th Electrical Engineering Conference, May 17-19, 1999

مدلی چندسطحی از شبکه‌های نامنظم مثلث‌بندی شده

محمّد قدسی فرشاد رستم‌آبادی

دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف

چکیده

دریافت و پردازش اطلاعات ماهواره‌ای یکی از کاربردهای مهم کامپیوتر در سال‌های اخیر است که عموماً تحت نام سیستم‌های اطلاعات جغرافیایی شناخته می‌شود. اطلاعات مناطق جغرافیایی در این سیستم‌ها به صورت‌های متفاوتی نگه‌داری می‌شود که شبکه‌های نامنظم مثلث‌بندی شده (تین) یکی از مدل‌های مهم برای این کار است. یکی از مهم‌ترین کاربردهای تین در نمایش گرافیکی تصاویر است. هنگامی‌که بخواهیم بسته به نقطه دید بخش‌های مختلف تصویر از درجه وضوح مختلفی برخوردار باشند، این کار با مدل چندسطحی در تین انجام می‌شود و حجم اطلاعات و زمان انجام پردازش‌های مختلف را به نحو چشم‌گیری کاهش می‌دهد.

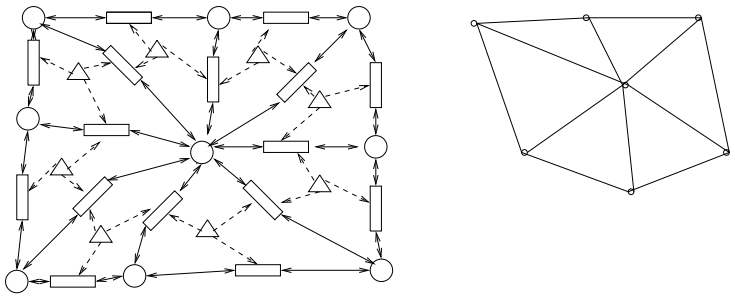
در این مقاله، یک روش جدید برای ذخیره و بازیابی مدل چندسطحی تین‌ها ارائه شده است که ضمن ساده بودن پیاده‌سازی آن و صرفه‌جویی در حافظه، اعمال مختلف را می‌تواند به‌سرعت انجام دهد. مثلاً با این روش می‌توان مثلث‌های مجاور هر مثلث را در زمان ثابت پیدا کرد. این مدل با یک پیش‌پردازش تهیه می‌شود و با استفاده از یک تابع تعیین میزان خطای قابل قبول برای هر نقطه، و با ادغام نتایج حاصل از پیش‌پردازش، تین ساده‌شده را برای نمایش و یا پردازش‌های بعدی آماده می‌کند. این مدل تغییر تابعِ تعیین خطا و نیز سؤالات مربوط به هم‌جواری را نیز در زمان ثابت انجام می‌دهد.

۱ مقدمه

دریافت و پردازش اطلاعات دریافتی از تصویرهای ماهواره‌ای از سطح زمین از جمله کاربردهای مهم کامپیوتر در دهه‌ی اخیر می‌باشد. این تصویرها که حاوی میلیون‌ها نقطه می‌باشند، حجم زیادی را به خود اختصاص می‌دهند، و لذا انجام پردازش‌های مختلف بر روی آن‌ها زمان زیادی می‌گیرد. لذا مدل‌های دیگری پیشنهاد شده است که ضمن حفظ ویژگی‌های اصلی تصویر، حجم اطلاعات لازم به مقدار زیادی کاهش پیدا کرده و نیز انجام پردازش‌های لازم در زمان کوتاهی ممکن باشد. یکی از این مدل‌های شبکه‌ی نامنظم مثلث‌بندی شده[1] است که علاوه بر داشتن مزایای ذکرشده، به لحاظ نظری بسیار غنی بوده و مورد بررسی زیادی قرار گرفته است [۶].

یکی از کاربردهای مهم تین‌ها در مدل چندسطحی، نمایش گرافیکی سریع تصویر سطح‌ها است. وقتی که اهمیت نواحی دورتر به نقطه‌ی دید، کم‌تر از نواحی نزدیک است، می توان با استفاده از مدل‌های چند سطحی مناطق دور را با دقت کمی نمایش داد. از چنین مدلی در نرم‌افزارهای شبیه‌سازی پرواز، یا بازی‌های کامپیوتری استفاده می‌شود تا عملیات برخط را بتوان با سرعت‌های بالاتری انجام داد، چراکه نمایش گرافیکی کامل این

[1] Triangulated Irregular Network(TIN)



شکل ۱: یک تین نمونه و مدل گراف نگه‌داری آن

تصاویر توان پردازش زیادی طلب می کند. در این مقاله ابتدا به مرور چند مدل پیشنهاد شده می‌پردازیم و سپس یک مدل جدید برای این عمل پیشنهاد و بررسی می‌کنیم.

در مدل ارائه شده، از یک تابعِ تعیینِ مقدار خطای قابل قبول $f(x,y)$ برای هر نقطه $P=(x,y)$، استفاده می‌کند. به عنوان مثال تابع $f_1(x,y)=\sqrt{(x_p-x_v)^2-(y_p-y_v)^2}$ می‌تواند برای نقطه دید $P_v=(x_v,y_v)$ سطح دقت هر نقطه $P=(x_p,y_p)$ را تعیین کند. پس از انجام پیش پردازش مورد نیاز برای ساخت مدل، تغییر تابعِ تعیینِ خطا در زمان ثابت صورت می‌گیرد. به عبارت دیگر در هر لحظه دلخواه می‌توان تابع f را تغییر داد و از تابع دیگری به جای آن استفاده نمود. ویژگی دیگر این مدل، ساخت افزایشی تین ساده شده می‌باشد. در زمان انتخاب یک تابعِ تعیین خطای جدید، تین ساده شده ساخته نمی‌شود؛ بلکه اعمال مجاز این مدل مثلث‌های شرکت کننده در تین را انتخاب می‌نمایند و به تین اضافه می‌نمایند. اعمال مجاز مدل شامل عمل مکان‌یابی یک نقطه داده شده، سؤالات هم‌جواری مثلث‌ها، اضلاع و نقاط تین می‌باشد. از دیگر ویژگی مهم این مدل آن است که ارجاع‌های بعدی به یک مثلث از تین ساده شده، به شرط عدم تغییر تابع $f(x,y)$، به محاسبات دیگری نیاز ندارد و در زمان ثابت صورت می‌گیرد.

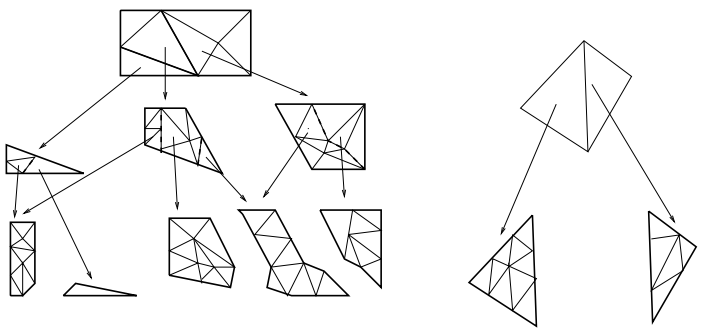
۲ شبکه‌های نامنظم مثلث‌بندی شده

ماهواره برای تهیه یک تصویر از یک منطقه آنرا به تعدادی نقطه تقسیم می‌نماید و سپس اطلاعات مورد نیاز (مثلاً ارتفاع) را برای هر نقطه تعیین و ارسال می‌دارد. بدین ترتیب تصویرِ بدست آمده، یک نما از بالایِ منطقه می‌باشد که اصطلاحاً به آن تصویر ۲/۵ بعدی گفته می‌شود.

یک تصویر ۲/۵ بعدی را می‌توان به روش‌های مختلفی نگه‌داری کرد: مثلاً در روش جدول نقاط تمام اطلاعات یک تصویر بدون هیچ خلاصه‌سازی، به صورت سه‌تایی‌های (x,y,z) و یا در یک جدول دو بعدی، ذخیره می‌شوند، که به فضای زیادی برای ذخیره‌سازی نیاز دارد. روش دیگر، استفاده از خطوط تراز است. دراین روش، که یکی از روش‌های قدیمی نقشه‌کشی است، به‌جای نگهداری تمام نقاط، خطوط مرزی دارای یک ارتفاع کشیده و نگهداری می‌شوند. ارتفاع نقاطی که دقیقاً روی مرز قرار ندارند، با استفاده از درون‌یابی تعیین می‌شود.

در روش شبکه‌های نامنظم مثلث‌بندی شده سطح مورد نظر توسط تعدادی مثلث تقریب‌زده می‌شود و بجز رئوس مثلث‌ها که ارتفاع خود را حفظ کرده‌اند، برای بدست آوردن ارتفاع سایر نقاط باید از درون‌یابی استفاده شود. خروجی مثلث‌بندی شده این روش را شبکه نامنظم مثلث‌بندی شده و یا به اختصار تین می‌نامند. یک تین نمونه در شکل ۱ آورده شده است. معمولاً سعی می‌شود مثلث‌های تولید شده دارای خواصی، مثل خاصیت دلانی، داشته باشند تا کیفیت مثلث‌ها در سطح قابل قبولی قرار گیرد.

یکی از روش‌های معمول ذخیره‌سازی تین، یک گراف سه بخشی جهت‌دار است[۶]. هر نقطه با یک رأس گرد، هر یال با یک رأس مستطیل و هر مثلث نیز با یک رأس مثلث مشخص می‌شود. به ازای هر نقطه و یک یال متصل به آن، یک ضلع از رأس گرد متناظر با نقطه به رأس مستطیل متناظر با یال و نیز یک یال در جهت عکس همین یال، از رأس مستطیل به رأس گرد در گراف سه



(ب) (الف)

شکل ۲: الف) سازماندهی درختی مدل چندسطحی، ب) سازماندهی گرافی مدل چندسطحی

بخشی می‌گذاریم. به‌علاوه، برای هر مثلث و هر یال آن، یک ضلع از رأس مستطیل به رأس مثلث وصل می‌شود. در هر رأس گرد لیستی از رئوس مثلث‌های متصل به آن نیز نگه‌داری می‌شود. یال‌های گراف این امکان را می‌دهند که، با صرف زمان ثابت، از هر مثلث می‌توان به نقطه‌های تشکیل دهنده آن رسید، و از هر یال به مثلث‌هایی که در آن شرکت دارند و نیز رئوس آن یال دست یافت. مثالی از این مدل در ۱ آورده شده است که در آن لیست داخلی رئوس مثلث نمایش داده نشده‌اند. این مدل به دلیل پاسخ‌گویی سؤال‌های مجاورت (مثل یال‌های مجاور به یال، مثلث‌های مجاور یک رأس و غیره) در زمان ثابت، به عنوان پایه مدل پیشنهادی برای نگه‌داری درخت چندسطحی تین در مدل ارائه شده می‌باشد.

۳ مدل‌های چندسطحی

مدل‌های چند سطحی از تین به عنوان مدل نگه‌داری تصویر استفاده کرده‌اند. مدل‌های چند سطحیِ نگه‌داریِ تین امکان تغییر میزان دقت هر ناحیه از تین را می‌دهند. این مدل‌ها ابتدا با استفاده از حداکثر دقت ممکن در تین، پیش‌پردازشی انجام داده و مراحل آن را برای استفاده‌های بعدی ذخیره می‌نمایند. این مراحل که عموماً تین‌های بسیار کوچکی می‌باشند، در یک ساختار گراف سازماندهی می‌گردند. در زیر دو نمونه از این سازماندهی‌ها را به اختصار ذکر می‌کنیم.

روش اول استفاده از یک درخت می‌باشد[۵]. ریشه درخت حاوی شکلِ مثلث‌بندی شده‌ی پوش محدب کل نقاط می‌باشد. فرزندان این گره مثلث‌های تشکیل دهنده مثلث‌بندیِ پوش می‌باشند. هر کدام از این مثلث‌ها، ریشه یک زیر درخت دیگر می‌باشند که شامل مثلث‌بندی رئوس مثلث و چند نقطه درون آن مثلث می‌باشد. بدیهی است که به علت عدم ارتباط بین زیر درخت‌ها، با کنار هم قرار دادن مثلث‌های تولید شده در گره‌های مختلف، لزوماً یک مثلث‌بندی دلانی تولید نمی‌شود[۱]. مثالی از این روش در شکل ۲.الف نمایش داده شده است.

روش دیگر تا حدودی شبیه به مدل قبل می‌باشد با این تفاوت که بجای ساختن یک درخت، یک گراف جهت‌دار بدون دور(DAG) می‌سازد و تضمین می‌نماید که تمام تین‌های ممکن که از این ساختار به دست می‌آیند، دلانی می‌باشند. این روش نیز از یک حالت اولیه شروع به ساختن گراف می‌نماید. اولین گره گراف، مانند مدل درختی، شکل مثلث‌بندی شده پوش محدب می‌باشد. حال با استفاده از یک الگوریتم جزئی‌سازی (Refinement) شروع به اضافه کردن نقطه در این مدل می‌نماید تا وقتی‌که تمام نقاط به تین وارد شوند. نحوه‌ی تولید این گراف به صورت زیر است: در هر مرحله با افزایش یک نقطه به تین جاری، تعدادی از مثلث‌ها از تین حذف می‌شوند و مثلث‌های جدیدی جایگزین آن‌ها می‌شوند. در این حالت، گره‌هایی از گراف که حاوی مثلث‌های قدیمی هستند، به دو گره تقسیم می‌شوند، که یکی از آن‌ها شامل هیچ مثلث قدیمی نیست و دیگری فقط از مثلث‌های قدیمی درون گره اولیه تشکیل شده است. در این‌جا، یک گره جدید دیگر به گراف اضافه می‌شود، که شامل مثلث‌بندی نقطه‌ی جدید و مجموعه‌ی نقاط مثلث‌های قدیمی می‌باشد. این گره‌ی جدید به

عنوان فرزند به تمام گره‌هایی در قسمت قبل تجزیه شدند، پیوند زده می‌شود. این روند تا ورود تمام نقاط به ساختار تکرار می‌شود. در شکل ۲.ب یک نمونه از چنین ساختاری را در زیر می آوریم[۳].

از جمله نکات جالب روش دوم، می‌توان به موارد زیر اشاره کرد: این روش به الگوریتم جزئی‌سازی وابستگی ندارد. به علاوه، بر خلاف روش اول، همواره تین به‌دست آمده از این روش، دلانی است. با توجه به این نکته که، حالت اولیه (مثلث بندی پوش محدب) و حالت نهایی (مثلث بندی با استفاده از حداکثر نقاط) یکتا می‌باشند، می‌توان به جای ساخت گراف از حالت اولیه توسط روش‌های افزودن نقطه، آن را از حالت نهایی و با استفاده از روش‌های ساده‌سازی (برای حذف نقاط) بصورت برعکس به دست آورد.

۴ مدل چند سطحی پیشنهادی

این مدل براساس مدل گراف نگهداری تین و سازماندهی گرافی پیش‌پردازش بنا شده است. ایده‌ی اصلی در این مدل این است که گراف تین بصورت پویا خود را با استفاده از قطعه تین‌های کوچک به‌دست آمده از پیش‌پردازش، تغییر دهد. به‌عبارت دیگر، در هنگام پیمایش مثلث‌های تین، با تغییر میزان خطای مجاز، ممکن است مثلث‌های تین خود را با سِری دیگری از مثلث‌ها که از قبل تهیه شده است، تعویض نمایند تا میزان دلخواه دقت به دست آید. در این جا ابتدا ساختارهای مورد نیاز برای نگه‌داری یک تین ارائه می‌شود، سپس سایر ساختارها برای افزودن چندسطحی بودن به آن اضافه می‌گردد.

برای نگه‌داری یک تین، از مدل گراف آن ایده گرفته شده‌است و تغییر داده شده در آن این است که رئوسی مربوط به یال‌ها، حذف شده‌اند. البته این کار، پیدا کردن مثلث مجاور یک یال را کندتر می‌کند، ولی بعداً می‌بینیم که این زمان به طور متوسط ثابت باقی می‌ماند. حذف یال‌ها به منظور سرعت بخشیدن به عمل به‌هنگام سازی تین در مواقعی که احتیاج به تغییر میزان خطا در یک منطقه مورد نیاز است، صورت گرفته است. علاوه بر این، در هر مثلث به جای سه یال، سه راس آن ذخیره می‌شوند.

در ساختارهای مورد استفاده فرض شده است که هر عنصر دارای یک شماره یکتا می‌باشد و از طریق این شماره با زمان ثابت قابل دسترسی است. این شماره در زمان پیاده‌سازی می‌تواند آدرس خانه‌ی حافظه‌ی حاوی اطلاع مورد نظر باشد. ساختارهای ۴ برای نگه‌داری نقاط و مثلث‌ها ارائه می‌شوند. توجه کنید که هر مثلث حاوی شماره نقاط سه راس آن می‌باشد و هر نقطه شماره‌ی لیستی از شماره مثلث‌هایی را دارد که در یک رأس آن‌هاحضور دارد.

یک یال در ساختار جدول ۴ بصورت زوج مرتبی از شماره‌ی مثلث و شماره‌ی ترتیبی یال نمایش داده می‌شود. شماره‌های ترتیبی یال‌های یک مثلث به صورت زیر تعیین می‌گردند

- یال شماره ۱ یال بین راس های V_1, V_2 مثلث.
- یال شماره ۲ یال بین راس های V_2, V_3 مثلث.
- یال شماره ۳ یال بین راس های V_1, V_3 مثلث.

برای افزودن خاصیت چندسطحی بودن به این ساختار تغییرات زیر لازم می‌باشند: باید به جدول مثلث‌ها برای هر یال یک مثلث، مؤلفه‌ای دوقسمتی اضافه شود که در آن مقدارِ تغییرِ سطحِ خطایِ ممکنِ بعدی (هم‌برای خطای کمتر و هم خطای بیشترِ بعدی) برای آن یال نگه‌داری گردد. بدین منظور، در هنگام ساختن ساختار، با هر بار کاسته شدن یک نقطه از تین، یک سطر برای نقطه‌ی کاسته شده تخصیص می‌دهیم. شماره‌ی شناسایی سطر مزبور را در اطلاعات مربوط به یال درگیر در هر مثلث تغییر یافته، با ذکر مقدارِ تغییر خطای مثلث جدید مجاور به آن یال، ذخیره می‌کنیم.

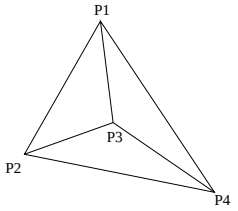
۱.۴ تولید ساختار

برای تولید ساختار صحیح، باید از یک حالت اولیه شروع نمود و نقطه، نقطه رئوس را از ساختار کم نمود (یا به آن اضافه کرد). در این بخش با استفاده از یک الگوریتم ساده‌سازی ساختار را تولید می‌کنیم.

حالت شروع می‌تواند شکل مثلث‌بندی شده کامل با استفاده از تمام نقاط باشد. از این به بعد، تا زمانی‌که الگوریتم نقطه‌ای مثل P_1 را برای حذف انتخاب می‌کند، اعمال زیر را انجام می‌دهیم.

شماره مثلث مثلث	مختصات رئوس مثلث (V_1, V_2, V_3)
T_1	(P_1, P_2, P_3)
T_2	(P_2, P_3, P_4)
T_3	(P_1, P_3, P_4)
...	...

شماره نقطه	مختصات نقطه (x, y, z)	شماره لیست مثلث‌های مجاور
P_1	(10, 20, 40)	L_1
P_2	(21, 21, 13)	L_2
P_3	(25, 10, 3)	L_3
P_4	(10, 10, 10)	L_4
...	...	...



شماره لیست	لیست شماره مثلث‌های درون این لیست
L_1	(T_1,T_3)
L_2	(T_1,T_2)
L_3	(T_1,T_2,T_3)
...	...

جدول ۱: مدل اولیه: بدون اطلاعات چندسطحی

۱) نقطه P_1 را در جدول نقاط پیدا کرده و لیست مثلث‌های مجاور آنرا L_1 می‌نامیم.

۲) با استفاده از لیست L_1، لیستی از نقاطی تهیه می‌کنیم که توسط یک یال به نقطه P_1 متصلند.

۳) برای هر نقطه در لیست L_2، یک کپی از لیست مثلث‌های مجاور آن تهیه می‌کنیم با این شرط که هیچ یک از مثلث‌های لیست L_1 در آن نباشد.

۴) نقاط درون لیست L_2 رئوس چند ضلعی ستاره‌ای[۲] متصل به نقطه P_1 هستند، این نقاط را مثلث بندی دلانی کرده و مثلث‌ها را در لیست‌های نقاط رئوس آنها اضافه می‌کنیم.

۵) آخرین عمل ضبط تغییرات است. برای این منظور مراحل زیر باید انجام گیرند.

- یک سطر در جدول اصلاحات برای نقطه P_1 تخصیص داده شده و لیستی از سه‌تایی مرتب زیر در آن نگه‌داری شود: شماره لیست مثلث‌ها قبل از حذف، شماره لیست مثلث‌ها بعد از حذف، شماره نقطه

  بدیهی است تمام نقاط درون لیست L_2 می‌باید در این لیست حضور داشته باشند.
- برای هر مثلث که به جدول مثلث‌ها افزوده شده‌است، اطلاعات تغییر خطا برای یال مرزی مثلث (یالی مرز چند ضلعی ستاره‌ای) بدین ترتیب به‌هنگام می‌شود: شماره عمل اصلاح انجام شده(مربوط به حذف P_1) در مؤلفه‌ی CoarseID یال مرزی مثلث‌های جدید قرار می‌گیرد و همین شماره در مؤلفه‌ی FinerID یال مرزی مثلث‌های قدیمی نیز گزارده می‌شود.

با انجام تغییرات بالا، تاریخچه اعمال ساده سازی ذخیره می‌شوند و می‌توان از آن‌ها برای تولید (یا تصحیح) تین استفاده نمود. توجه کنید که تاریخچه‌ها بروی مثلث‌های جدید و قدیم درگیر تغییر نگه‌داری می‌شود به عبارت دیگر برای مثلث‌های مجاور به یک چند ضلعی ستاره‌ای تغییر یافته، اطلاعی را بهنگام نمی‌کنیم. با این کار اطلاعات مؤلفه‌های CoarserID و FinerID یک یال برای ورود به مثلث (در حقیقت چندضلعی ستاره‌ای) معتبر می‌باشند نه برای زمانی که در مثلث قرار داریم و می‌خواهیم از طریق یک یال از آن خارج شویم. نحوه استفاده از این داده‌ها در بخش بعد آورده شده است.

در جدول مثلث‌ها یک مؤلفه وجود دارد که تا به حال از آن استفاده نشده است و آن مؤلفه‌ی TimeStamp می‌باشد. در زمان ساخت پویای تین از این مؤلفه برای

[۲] Star shape polygon

شماره مثلث مثلث	مختصات رئوس مثلث (V_1, V_2, V_3)	شماره‌های اصلاح برای یال ۱ (Coarser,Finer)	شماره‌های اصلاح برای یال ۲ (Coarser,Finer)	شماره‌های اصلاح برای یال ۳ (Coarser,Finer)	زمان مهر
T_1	(P_1, P_2, P_3)	$(U_1,-)$	(-,-)	(-,-)	۰
T_2	(P_2, P_3, P_4)	(-,-)	$(U_1,-)$	(-,-)	۰
T_3	(P_1, P_3, P_4)	$(-,U_1)$	(-,-)	(-,-)	۰
...	...	...	...	...	...

شماره تغییر	شماره نقطه حذف شده	خطای نقطه	لیست تغییرات در هر نقطه قبل و بعد از عمل حذف
U_1	P_3	E_{P_3}	$((P_1, L_1, L_4),(P_2, L_2, L_5))$
U_2	P_2	E_{P_2}	$((P_1, L_4, L_6),(P_4, L_4, L_7))$
...	...	...	...

جدول ۲: مدل تکامل یافته برای نگه‌داری اطلاعات چندسطحی

تشخیص این که آیا یک مثلث بعد از آخرین تغییر تابع میزان خطای قابل قبول، پیمایش شده است یا خیر، استفاده می‌شود. در حین ساخته شدن مثلث‌ها باید به این مؤلفه مقدار صفر نسبت داده شود.

۵ نحوه‌ی استفاده از مدل

این ساختار برای نمایش سه‌بعدی حرکت در تین در زمان حقیقی بهینه شده است. اعمال تعریف شده در این ساختار به این منظور می‌باشند.

مکان‌یابی یک نقطه داده شده: برای شروع، یک نقطه مکان‌یابی می‌شود. با هر بار مکان‌یابی یک شماره سریال تولید می‌شود و در هر لحظه فرض بر این است تمام مثلث‌هایی که دارای این شماره در مؤلفه‌ی TimeStamp خود باشند، قطعاً در تین حضور دارند. نتیجه عمل مکان‌یابی شماره یک مثلث خواهد بود که نقطه داده شده در آن قرار دارد.

برای عمل مکان‌یابی، ابتدا با استفاده از الگوریتم‌های مکان‌یابی تین، در تینِ با حداقل مثلث (کمترین دقت)، مثلث حاوی نقطه داده شده مکان‌یابی می‌گردد. سپس با استفاده از تابع تعیین مقدار خطای مجاز $f(x, y)$ مقدار خطای ممکنِ نقطه تعیین می‌شود. سپس تا زمانی‌که مثلث به سطح مورد نظر از خطا نرسیده است، با استفاده از عمل تصحیح که در ادامه آمده است، دقت تا سطح مطلوب بالا برده می‌شود.

مثلث مجاور به یک یال از یک مثلث با خطای داده شده: برای اینکار با استفاده از لیست مثلث‌های راس‌های یال در مثلث اول، مثلث دوم را که در یال خواسته شده با مثلث اول مجاور است، پیدا می‌کنیم. بعد از پیدا کردن این مثلث، اگر مثلث دارای TimeStamp صحیح باشد، جواب است. در غیر اینصورت، عمل تصحیح را آنقدر انجام می‌دهیم تا به سطح خطای مورد نظر دست یابیم.

عمل تصحیح:
اگر خطای مطلوب بین خطای نقطه FinerID و CoarserID نباشد، یک عمل به‌هنگام سازی (یا عکس آن را) انجام می‌دهیم تا به سطح خطای مورد نظر دست یابیم. شرط آنکه عمل تصحیح انجام پذیر باشد، آن است که تمام مثلث‌هایی که در لیست مجاورت راس سوم مثلث مورد تصحیح قرار دارند،دارای TimeStamp صحیح نباشند.

٦ نتیجه

در این مقاله یک روش ذخیره و بازیابی برای مدل‌های چندسطحی ارائه شد. این روش از مدل گراف برای نگه‌داری تین در حافظه استفاده می‌کند. مدل چند سطحی گرافی به منظور افزودن قابلیت‌های چند سطحی به مدل گراف تین اضافه شده است. پیاده‌سازی این

ساختار با استفاده از امکانات کتابخانه‌ای LEDA [۷] بر روی سیستم عامل Linux پیاده‌سازی شده است.

از جمله موارد قدرت این ساختار باید به سادگی نسبی آن برای تولید و نیز میزان حافظه مصرفی اشاره کرد. این ساختار برای بیشتر الگوریتم‌های قابلیت دید[۳] اعمال پایه‌ای مورد نیاز را ارائه می‌دهد [۲]. علاوه بر این، موارد زیر نیز از مزایای این مدل محسوب می‌شوند:

- پیدا کردن مثلث مجاور در زمان مورد انتظار ثابت
- هر مثلث تنها یک بار جهت تعیین میزان دقت پردازش می‌شود
- تغییر تابع خطای قابل قبول در زمان ثابت صورت می‌گیرد

نقاط ضعف این ساختار را می‌توان در عدم امکانِ داشتنِ چندین تابعِ تعیینِ خطای مستقل و کندی عمل مکان یابی برای حالت کلی برشمرد. کار روی این مدل، برای بهبود این جنبه‌ها ادامه دارد.

مراجع

[1] J. H. Clark, "Hierarchical geometric models for visible surface algorithms", *CACM, 19(10):547-554*, Oct. 1976.

[2] Daniel Cohen-Or and Yishay Levanoni, "Temporal continuity of Levels of Delaits", In *Proceedings IEEE Visualization '96*, 1996.

[3] L. De Floriani, P. Magillo, and E. Puppo, "Building and traversing a Surface at Variable Resolution", In *Proceedings IEEE Visualization '97*, pp. 103-110, Oct. 1997.

[4] L. De Floriani, P. Magillo, and E. Puppo, "Multiresolution Representation and Reconstruction of Triangulated Surfaces", In *Advances in Visual Form Analysis*, pp.140-149, Singapore, 1997.

[5] L. De Floriani, C. Montani, and R. Scopingo, "Parallelizing visibilty computations on triangulated trains", Int Geographical Information Vol 8, No.6, 515-531, 1994.

[6] P. S. Heckbert, and M. Garland, "Survey of polygonal surface simplification algorithms", *Multiresolution Surface Model Course, SIGGRAPH'97*, Carnegy Mellon University, May 1, 1997.

[7] K. Mehlhorn and S. Naher, "LEDA: A Platform for Combinatorial and Geometric Computing", 1994.

[۳] Visibility