



**زیربرنامه:**

ReDefineSeedNodes3D

|  |  |  |
| --- | --- | --- |
| **توسعه دهندگان** | مرتضی نامور |  |
| کامیار صفری |  |
| **تهیه کنندگان مستند** | مرتضی نامور، کامیار صفری | |
| **تاییدکنندگان** | مرتضی نامور | |
| **تاریخ تنظیم سند** | 24/10/1396 | |
| **شناسه سند** |  | |
| **زبان برنامه‌نویسی** | **Fortran 90** | |

1. وظایف

این زیربرنامه نقاط seed مربوط به ناحیه‌ی لایه‌بندی نشده(یا اشتباه لایه‌بندی شده) در مرحله قبل(wake) را تشخیص داده و به نقاط آن شماره لایه‌ی مناسب تخصیص میدهد.

1. توضیحات و تئوری

در مراحل قبل، شماره لایه‌ی نقاط ناحیه‌ی wake که به صورت اشتباه لایه بندی میشد، حذف شدند. قبل از لایه بندی دوباره ی این ناحیه، بایستی نقاط seed آن را بیابیم.

برای اینکار، ابتدا گره های لایه های لایه مرزی که قبلا لایه بندی کرده ایم را به ترتیب پیمایش میکنیم. در صورتی که در هرکدام از لایه ها بتوانیم گره ای به نام i پیدا کنیم که یک گره با شماره لایه‌ی صفر با اندیس j در همسایگی آن وجود داشته باشد، همچنین آن گره(j) در جهت بیشترین میزان کشیدگی گره ی i قرار داشته باشد، گره ی j را به عنوان گره ی شروع و شماره لایه‌ای که گره‌ی i دارد را به عنوان شماره لایه‌ی seed جدید در نظر میگیریم.

پس از تشخیص نقطه ی اولیه، از آن نقطه شروع کرده و به ترتیب نقاط همسایه‌ای که دارای شماره لایه‌ی صفر هستند و در جهت بیشترین میزان کشیدگی گره‌ی قبلی هستند را به مجموعه نقاط seed جدید اضافه میکنیم. شماره لایه‌ی نقاط مربوط به seed جدید را برابر با شماره لایه‌ی نقطه‌ی i قرار میدهیم.

توجه کنید که در شبکه‌های سه‌بعدی، سطح را نیز باید در پردازش‌ها لحاظ کنیم. به طوری که از سطح 1 که شامل تمام گره‌های مرزی می‌باشد عملیات را شروع کرده و به ترتیب سطوح دیگر را پردازش میکنیم. در هر مرحله، هر گره تنها با گره‌هایی هم سطح خودش مقایسه می‌شود. به عنوان مثال اگر گره‌ای شروط افزوده شدن به عنوان نقطه seed را داشته باشد اما هم سطح با نقطه‌ی فعلی نباشد، نمیتواند به عنوان نقطه‌ی seed جدید انتخاب شود.

در طول انجام عملیات، وضعیت گره‌ها در یک آرایه‌ی کمکی ذخیره خواهد شد. وضعیت‌های قابل اختصاص به گره‌ها شامل وضعیت "پردازش نشده" و "انتخاب شده" و "پردازش شده(قبلا انتخاب شده)" می باشد. در هربار پردازش گره‌ها، هر گره تنها با گره‌های مجاور خودش که وضعیت آنها "پردازش نشده" است و به حداقل یک گره‌ی "انتخاب شده" متصل هستند مقایسه میشود. چنین گره‌هایی بر روی یک سطح قرار دارند. پس از پایان هر فاز و پایان پردازش هر سطح، وضعیت گره های "پردازش نشده" ای که به گره های "انتخاب شده" متصل هستند به "انتخاب شده" تغییر یافته و گره های "انتخاب شده"ی قبلی به "پردازش شده(قبلا انتخاب شده)" تغییر وضعیت می‌یابند. اینکار تا زمانی که گره‌ای نتواند تغییر پیدا کند و یا سطح پردازش نشده ای باقی نمانده باشد ادامه می‌یابد.

1. بخش‌های زیربرنامه

در این قسمت، توضیح تمامی بخش‌های زیربرنامه، مطابق شماره‌گذاری انجام شده در متن برنامه کامپیوتری ارائه شده است.

1. مقداردهی اولیه متغیرهای مورد استفاده

در هربار تکرار حلقه‌ی اصلی، یکی از سطوح شبکه را پردازش میکنیم. گره های سطح فعلی با علامت "unknown" نشانه‌گذاری می‌شوند(همچنین به حداقل یک گره‌ی selected متصل هستند). گره های پردازش شده‌ی مرحله قبل با "selected" و به همین ترتیب گره‌هایی که قبلتر از آن پردازش شده‌اند با علامت "oldSelected" مشخص میشوند.

در ابتدای کار تمامی گره‌ها را با unknown علامت میزنیم. پردازش سطوح را از سطح مربوط به گره‌های مرزی(غیر از سطح داخلی ایرفویل) شروع میکنیم. بنابراین یک متغیر منطقی را برای تشخیص اینکه پردازش اول مربوط به گره‌های مرزی است با مقدار true مقداردهی میکنیم. همچنین یک متغیر که نشان دهنده‌ی ایجاد تغییر در حداقل یک گره است را با true مقداردهی میکنیم.

1. یافتن بزرگترین شماره لایه‌ی موجود در شبکه

برای یافتن نقاط seed جدید، باید تمامی لایه‌ها را به ترتیب پردازش کنیم تا بتوانیم تشخیص دهیم که کدام لایه میتواند به عنوان seed ادامه پیدا کند. برای اینکار در این بخش با استفاده از یک حلقه، بزگترین شماره لایه‌ی تخصیص داده شده به نقاط را شناسایی کرده و در یک متغیر نگداری میکنیم.

1. حلقه اصلی زیربرنامه برای پردازش تمامی گره ها و تمامی سطوح

در این بخش، در یک حلقه‌ی اصلی، با هر بار اجرای حلقه یکی از سطوح شبکه را پردازش میکنیم. در هربار تکرار حلقه تمامی لایه‌ها به ترتیب پیمایش میشوند. اجرای حلقه تا زمانی که حداقل یکی از گره‌های مربوط به یکی از لایه‌ها بتواند به لیست نقاط seed جدید اضافه شود ادامه پیدا میکند. اینکار را با یک متغیر منطقی کنترل میکنیم. در ابتدای کار مقدار آن را false کرده و در طول انجام عملیات در صورتی که گره‌ای به نقاط seed اضافه شود آن را true میکنیم تا حلقه‌ی اصلی بتواند حداقل یکبار دیگر نیز اجرا شود.

1. پیمایش تمامی لایه‌ها و نقاط هر لایه برای یافتن شماره لایه‌ای که میتوانیم به seed جدید تخصیص دهیم

در هر سطح، تمامی لایه‌ها را پیمایش میکنیم. در این بخش با استفاده از یک حلقه، از بین گره‌های هر لایه(و مربوط به سطح انتخاب شده)، شماره لایه‌ی مناسب برای تخصیص به seed جدید و همچنین نقطه‌ی seed(که بتوانیم از طریق آن سایر نقاط را به لایه‌ی seed اضافه کنیم) را پیدا میکنیم.

نقطه‌ای میتواند به عنوان نقطه‌ی seed انتخاب شود که در مجاورت آن، یک گره‌ی همسطح با شماره لایه‌ی صفر که در جهت بیشترین میزان کشیدگی آن قرار دارد وجود داشته باشد.

قبل از شروع حلقه، متغیرهای مربوط به شماره لایه‌ی seed جدید و همچنین آخرین نقطه(نقطه‌ی seed) را برابر با صفر قرار میدهیم.

درون حلقه با انتخاب هر لایه، تمامی نقاط متعلق به آن لایه که وضعیت آنها unknown(سطح پردازش نشده) است پیمایش میشوند.

1. بررسی اینکه نقطه انتخاب شده(نقطه اول) در سطح انتخاب شده قرار داشته باشد

در هربار پردازش، نقاطی را مورد بررسی قرار میدهیم که بر روی سطح فعلی قرار داشته باشند. میدانیم که منظور از سطح selected سطحیست که در مرحله قبل پردازش شده است. بنابراین در این بخش میتوانیم نتیجه بگیریم که اگر نقطه‌ی unknown انتخاب شده به حداقل یک نقطه‌ی selected متصل باشد، درنتیجه در سطح فعلی قرار دارد.

البته چون در مرحله‌ی اول وضعیت هیچ گره‌ای selected نیست، در ابتدا بررسی میکنیم اگر گره‌ی مورد نظر مرزی باشد و در حال پردازش اولین سطح باشیم، شرط درست بوده و گره بر روی سطح فعلی قرار دارد و میتوانیم پردازش آن را ادامه دهیم.

1. پیمایش نقاط متصل به نقطه انتخاب شده

هدف از این بخش، شناسایی نقطه‌ی seed جدید(بر روی سطح و لایه‎ی فعلی) است. فرض کنید نقطه‌ای که درحال پردازش آن هستیم p1 باشد. شناسایی نقطه‌ی seed جدید به این صورت است که نقاط مجاور نقطه‌ی انتخاب شده‌ی p1 را پیمایش میکنیم. در صورتی که در بین نقاط هم‌سطح مجاور آن نقطه نقطه‌ای وجود داشته باشد که شماره لایه‌ی آن صفر باشد و همچنین آن نقطه در جهت بیشترین میزان کشیدگی گره‌ی p1 باشد، درنتیجه p1 یک نقطه‌ی seed است.

1. بررسی اینکه نقطه دوم نیز در سطح انتخاب شده قرار داشته باشد

در این بخش با انتخاب هر نقطه‌ی مجاور نقطه‌ی مورد پردازش، بایستی بررسی کنیم که آیا دو نقطه بر روی یک سطح قرار دارند یا خیر. در صورتی که بر روی یک سطح نباشند از ادامه‌ی بررسی آن خودداری میکنیم.

1. بررسی اینکه نقطه‌ی دوم در جهت بیشترین میزان کشیدگی نقطه اول قرار داشته باشد

در این بخش و این مرحله دو نقطه را انتخاب کرده ایم. یکی نقطه‌ای که کاندید انتخاب شدن به عنوان نقطه‌ی seed است(آن را p1 می‌نامیم) و دیگری نقطه ایست که هم سطح p1 است و شماره لایه‌ی آن صفر است که میخواهیم بدانیم آیا در جهت بیشترین میزان کشیدگی p1 قرار دارد یا خیر(آن را p2 می نامیم). بنابراین در این بخش تمام نقاط متصل به نقطه p1 و هم‌سطح با آن را پیمایش میکنیم. از بین آنها بیشترین میزان کشیدگی را شناسایی میکنیم. در انتها و پس از پایان حلقه در صورتی که آن نقطه‌ی شناسایی شده همان p2 باشد، به این معنیست که p1 نقطه‌ایست که میتواند به عنوان نقطه seed انتخاب شود

1. شروع حلقه برای کامل کردن لایه‌ی جدید در سطح انتخاب شده

زمانی که نقطه‌ی seed جدید و همچنین شماره لایه‌ی seed شناسایی شد، میتوانیم عملیات تکمیل کردن زنجیره نقاط seed را شروع کنیم. برای اینکار یک حلقه با تعداد تکرار نامشخص را اجرا میکنیم که در هربار تکرار نقاط متصل به اخرین نقطه‌ی انتخاب شده برای لایه‌ی seed را پیمایش میکند. اینکار تا زمانی که هیچکدام از نقاط متصل به یکی از نقاط seed دیگر نتوانند به لیست نقاط seed اضافه شوند ادامه پیدا میکند. نقاط بررسی شده را در یک آرایه ذخیره میکنیم تا بتوانیم در انتها وضعیت آنها را به selected تغییر دهیم.

1. پیمایش تمامی نقاط متصل به نقطه‌ی seed شناسایی شده

در این بخش، در هربار به وسیله‌ی یک حلقه تمامی نقاط متصل به آخرین نقطه seed را پیمایش میکنیم. هدف از انجام اینکار شناسایی اندیس نقطه‌ای با بیشترین میزان کشیدگی در مجاورت نقطه‌ی seed(و در همان سطح) است که شماره لایه‌ی آن صفر باشد. چنین نقطه‌ای میتواند در ادامه به عنوان نقطه‌ی seed جدید به لیست نقاط seed اضافه شود.

1. بررسی اینکه نقطه انتخاب شده یک نقطه‌ی متعلق به ناحیه ناهمسانگرد باشد(در هر سطح)

پس از انتخاب گره‌ای با بیشترین میزان کشیدگی در مجاورت نقطه‌ی seed(و هم‌سطح با آن)، در این بخش کشیده بودن المانهای مجاور آن نقطه را بررسی میکنیم. به این صورت که با استفاده از متریک فیلد بیشترین و کمترین میزان کشیدگی آن را پیدا کرده و برهم تقسیم میکنیم. درصورتی که مقدار آن از یک حد مشخص بیشتر باشد، آن گره یک گره‌ی کشیده است. لازم به ذکر است که مقایسه تنها باید بر روی گره های روی یک سطح انجام گیرد. در صورتی که گره‌ی انتخاب شده یک گره‌ی کشیده نباشد نمیتواند به عنوان گره‌ی seed جدید انتخاب شود و عملیات جستجوی نقطه‌ی seed جدید بر روی آن سطح خاتمه می‌یابد.

1. تخصیص شماره لایه به نقطه‌ی seed جدید

در این بخش نقطه‌ی seed جدید به عنوان آخرین نقطه در متغیر مربوطه ذخیره شده و همچنین شماره لایه‌ی مربوط به آن برابر با شماره لایه‌ی seed قرار میگیرد. همچنین متغیرهای منطقی مربوط به بروز تغییر در لیست نقاط seed با مقدار true مقداردهی میشوند تا حلقه ها بتواند برای حداقل یکبار بیشتر اجرا شوند.

1. افزودن نقاط مجاور و هم‌سطح نقطه‌ی seed جهت تغییر وضعیت آنها به "انتخاب شده" برای پردازش سطوح بعدی

در این بخش نقاط متصل به آخرین نقطه‌ی seed پیمایش میشوند. آنهایی که وضعیت آنها unknown است و همچنین به حداقل یک نقطه‌ی با وضعیت selected متصل هستند، به یک آرایه ی کمکی افزوده میشوند. وضعیت گره های موجود در این آرایه کمکی در انتها به selected تبدیل میشود تا در ادامه بتوانیم سایر لایه ها را نیز پردازش کنیم.

1. تغییر وضعیت گره‌های انتخاب شده‌ی سطح قبلی به "پردازش شده"

در این بخش در صورتی که تغییری در وضعیت حداقل یکی از گره های شبکه رخ داده باشد، وضعیت تمامی گره های با وضعیت selected را به oldSelected تبدیل میکنیم تا دیگر در پردازشها تاثیر داده نشوند.

1. تغییر وضعیت گره‌های جدید به "انتخاب شده"

در این بخش گره‌های منتخب آخرین لایه‌ی پردازش شده را برابر با selected قرار میدهیم. توجه کنید که در تکرارهای بعدی سطحی پردازش میشود که وضعیت گره‌های آن unknown بوده و به حداقل یکی از این گره‌های selected متصل باشد.

در آخر نیز متغیری که مشخص کننده‌ی پردازش اولین سطح است را برابر با false قرار میدهیم، تا پس از اولین تکرار حلقه اصلی دیگر گره‌های مرزی به عنوان گره‌های موجود بر روی سطح فعلی در نظر گرفته نشوند.