To Appear, Proceedings of 8th Iran Electrical Engineering Conference, May, 2000

شبکه‌بندی موازی سطوح سه‌بعدی

محمّد قدسی، سیّد علی اکرمی‌فر
آزمایشگاه پردازش موازی، دانشکده‌ی مهندسی کامپیوتر، دانشگاه صنعتی شریف

چکیده

شبکه‌بندی موضوعی مهم در هندسه محاسباتی است که کاربردهای فراوانی در برق، کامپیوتر، مکانیک و سایر رشته‌ها دارد. مهمترین نوع روش‌های شبکه‌بندی مثلث‌بندی است. در این مقاله شرح یک روش مثلث‌بندی، و نسخه‌ی موازی آن، برای پوسته‌ی اجسام سه‌بعدی آمده است. در این روش فضا به مکعب‌های ریز تقسیم می‌شود. با استفاده از مختصات نقاط پوسته، مکعب‌های متقاطع با آن مشخص می‌شوند. در نهایت با استفاده از مکعب‌های مجاور و کاملاً به صورت محلی، مثلث‌بندی انجام می‌گیرد. هر چند خطای شبکه‌های تولید شده، قابل توجه است؛ اما به عنوان یک الگوریتم مناسب برای شبکه‌بندی موازی، قابل تأمل است. نتایج شبکه‌بندی موازی، علاوه بر اختلاف کیفیت اندک با نوع ترتیبی، پیش‌بینی کم شدن زمان عملیات را نیز تأیید کرده‌اند.

کلید واژه‌ها

شبکه‌بندی *Mesh Generation*، پردازش موازی *Parallel Processing*، مدل‌سازی سطوح سه‌بعدی *3D Surface Modeling*

۱ مقدمه

فرض کنید اطلاعاتی از یک شیء خارجی موجود است. چگونه می‌توان این شیء را به کامپیوتر معرفی نمود؟ کامپیوتر چگونه می‌تواند آن را مورد پردازش قرار دهد؟ یکی از مهمترین روشها برای این منظور، مدل‌سازی هندسی است. هندسه‌ی محاسباتی که جایگاه تحقیقات پایه برای این مدل‌سازی محسوب می‌شود، در سال‌های اخیر پیشرفت چشمگیری داشته است. به مدل‌سازی هندسی شیء خارجی برای استفاده در کامپیوتر، شبکه‌بندی گویند. انواع مختلفی از شبکه‌بندی وجود دارد که نمونه‌هایی از آن در [۱، ۲، ۳، ۴، ۷] و [۸] آمده است. اطلاعات خارجی ممکن است مختصات مجموعه‌ای از نقاط جسم، سطح مقطع‌هایی از جسم که توسط صفحات موازی بریده شده است و یا اطلاعات جانبی مانند رنگ و چگالی باشند. مسأله‌ی شبکه‌بندی سطوح سه‌بعدی به این صورت است که مجموعه‌ای از نقاط پوسته‌ی یک جسم خارجی وجود دارد. می‌خواهیم به کمک این نقاط، یک پوسته‌ی تقریبی با عناصر مثلثی شکل، برای جسم به‌دست آوریم.

برای شبکه‌بندی موازی استفاده از ایده‌ی تقسیم فضا به مکعب‌های ریز که در روش هوپ[۱] [۵] نیز به کار رفته است، در نظر گرفته شد. براساس این ایده الگوریتم ترتیبی و موازی شبکه‌بندی، طراحی و پیاده سازی شدند. این روش فضا را به مکعب‌های ریز تقسیم نموده و ریزمکعب‌های متقاطع با پوسته را استخراج می‌کند و سپس به کمک آن‌ها مثلث‌بندی را انجام می‌دهد. ایده‌ی تقسیم مکعب‌ها برای موازی سازی مناسب است.

با توجه به کاربرد مکرر شبکه‌ها، کیفیت و درستی آن‌ها، اهمیت بیشتری نسبت به سرعت تولیدشان دارد. به همین دلیل است که بیشتر روش‌های شبکه‌بندی‌کند هستند. یافتن روش‌هایی برای ایجاد شبکه‌هایی باکیفیت مناسب در زمانی کوتاه، مشکل سرعت شبکه‌بندی را حل خواهد نمود. یکی از راه‌های سریع کردن الگوریتم‌ها،

[۱] hoppe عضو تیم تحقیقاتی مایکروسافت و دارای درجه‌ی دکتری از دانشگاه واشنگتن در زمینه‌ی بازسازی سطوح سه‌بعدی.

موازی سازی آن‌ها است. در شبکه‌بندی نیز در صورت استفاده از پردازش موازی، می‌توان به حل این مشکل امیدوار بود.

برای شبکه‌بندی موازی نقاط به صورت تقریباً یکسان، بین پردازنده‌ها تقسیم می‌شوند. برش نقاط از مرز مکعب‌ها و در راستای محوری که بیشترین پراکندگی را دارند انجام می‌گیرد. با این کار عمل تقسیم متوازن بهتر انجام می‌شود. هر پردازنده ناحیه‌ی مربوط به خود را شبکه‌بندی نموده و زیرشبکه‌ی حاصل را برمی‌گرداند. از اجتماع زیر شبکه‌ها، شبکه‌ی نهایی شکل می‌گیرد. شبکه‌ی نهایی نیاز به هیچ پس‌پردازشی برای اتصال زیرشبکه‌ها ندارد؛ زیرا در تقسیم نقاط، مرز به‌طور مشترک به پردازنده‌ها ارسال می‌گردد.

الگوریتم موازی در عمل نتایج خوبی داده است. اختلاف شبکه‌ی حاصل از این الگوریتم با گونه‌ی ترتیبی[۲] بسیار اندک است. نسبت افزایش سرعت نیز تقریباً برابر تعداد پردازنده‌ها است.

۲ روش ترتیبی شبکه‌بندی

یکی از بهترین روش‌های شبکه‌بندی پوسته‌ی سه‌بعدی، روش هوپ است. این روش ابتدا مجموعه نقاط نمونه‌ی موجود را به مکعب واحد نگاشت می‌دهد. سپس به کمک مختصات نقاط، پارامتر چگالی که نشان دهنده‌ی توزیع نقاط بر پوسته است و پارامتر خطای نمونه برداری، یک تابع فاصله برای نقاط به‌دست می‌آورد. به کمک این تابع می‌توان نقاطی از فضای مکعب واحد را که برای پوسته مناسبند، به‌دست آورد. با استفاده از پارامترهای شبکه‌بندی، مکعب واحد به تعداد زیادی ریزمکعب تقسیم می‌شود. سپس با استفاده از تابع فاصله، مکعب‌هایی که پوسته را قطع می‌کنند، مشخص شده و به‌کمک آن‌ها مثلث‌بندی انجام می‌گیرد[۵].

اما روش ما به این صورت است که در ابتدا با چند بار میانگین‌گیری از نقاط، خطای آن‌ها کاهش می‌یابد. در این مرحله هر نقطه، با میانگین خود و همسایگانش جایگزین می‌شود. پارامتر شبکه‌بندی، تنها پارامتر چگالی (ρ) است. هر دو نقطه با فاصله‌ی کمتر از ρ با یکدیگر همسایه‌اند. در مرحله‌ی بعد ریزمکعب‌هایی که نقاط نمونه در آن قرار دارند، به عنوان مکعب‌های پوسته مشخص می‌شود و در پایان مثلث‌بندی براساس آن‌ها انجام می‌شود.

۱.۲ فرضیات الگوریتم

فرض می‌کنیم نقاط نمونه بدون تکانه[۳]ی مشخصی هستند. برای رفع تکانه‌ی احتمالی با انجام چند بار عملیات میانگین‌گیری، نقاط نمونه به سمت پوسته‌ی واقعی حرکت می‌کنند. فاصله‌ی نقاط یا ضریب چگالی (ρ) توسط کاربر تعیین می‌شود. هر دو نقطه که فاصله‌ی آنها کوچکتر یا مساوی ρ باشد همسایه گوییم. نقاط نمونه توسط پویش‌گرهای سه‌بعدی[۴] نمونه‌برداری شده است و تنها اطلاع موجود مختصات (x, y, z) این نقاط است.

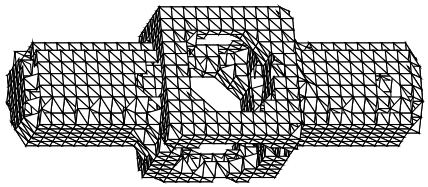
می‌خواهیم با استفاده از مجموعه نقاط ورودی $P = \{p_1, p_2, \ldots, p_n\}$، یک مثلث بندی مناسب، ایجاد کنیم. به منظور یکنواختی محاسبات و هماهنگ‌سازی در عملیات موازی، لازم است نقاط به مکعب واحد نگاشته شوند. نگاشت نقاط حتی در مراحل بعدی از قبیل نمایش شبکه یا چرخش آن کاربرد دارد. برای نگاشت و همچنین سایر عملیات محاسباتی هندسی مثل دوران، از عملگرهای پایه‌ی ماتریسی در فضای سه‌بعدی استفاده شده است.

ایده‌ی اصلی الگوریتم (الگوریتم ۱)، استخراج مکعب‌هایی است که پوسته را قطع می‌کنند. با در نظر گرفتن هر یک از این مکعب‌ها به عنوان یک رأس از شبکه و مثلث‌بندی مکعب‌های مجاور، می‌توان یک تقریب اولیه برای پوسته به‌دست آورد. باوجود ضعیف بودن تقریب، در سطح یک کار مقدماتی قابل قبول به‌نظر می‌رسد و می‌توان با روش‌های جدیدتر برای هرقسمت و یا بهینه‌سازی آن‌ها، کیفیت شبکه را بهبود بخشید.

الگوریتم ۱

---

[۲] serial
[۳] noise
[۴] 3d Scanner



شکل ۱: استخراج مکعب‌های متقاطع با پوسته.

۱) نقاط نمونه را به مکعب واحد، نگاشت بده.

۲) میانگین نقاط را چند[۵] مرتبه حساب کن.

۳) مکعب‌هایی را که در مجاورت پوسته قرار می‌گیرند، مشخص کن.

۴) با استفاده از مکعب‌های محیطی، مثلث‌های پوسته را مشخص کن.

با توجه به صورت الگوریتم۱، در مرحله‌ی ۱، که تنها با نقاط سروکار داریم، مرتبه‌ی الگوریتم، $O(n)$، است. در مرحله‌ی ۲، اگر نقاط m بار میانگین‌گیری شوند، مرتبه‌ی الگوریتم حداقل $O(m \cdot n \cdot log(n))$ خواهد بود. مرحله‌ی بعد با $O(n)$ و آخرین مرحله با $O(t)$ انجام می‌شود که در آن t، تعداد مکعب‌ها می‌باشد.

۲.۲ میانگین‌گیری از نقاط نمونه

برای رفع تکانه‌های احتمالی، به جای هر یک از نقاط، میانگین آن نقطه و همسایگانش، را جایگزین می‌کنیم. هر نقطه‌ای که فاصله‌ی آن تا نقطه‌ی مفروض p، کمتر از ρ باشد، یک همسایه‌ی p است. به این ترتیب از پراکندگی و تکانه‌ی نقاط نمونه کاسته شده و نقاط جدید به پوسته‌ی اصلی نزدیکتر خواهند بود. در حالت عادی، اگر m بار میانگین گرفته شود، مرتبه‌ی الگوریتم $O(n \cdot m \cdot k)$ می‌باشد که در آن k، متوسط تعداد همسایه‌ی هر رأس است. از آنجایی که برای هر دونقطه‌ی p, q، اگر p با q همسایه باشد، آن‌گاه q نیز با p همسایه خواهد بود، می‌توان پیچیدگی الگوریتم را به نصف کاهش داد.

در عمل به خاطر این‌که همسایگان هیچ نقطه در ابتدا مشخص نیست، یافتن همسایه‌ها نیازمند $O(n)$ هزینه است. بنابر این مرتبه‌ی الگوریتم $O(n^۲ \cdot m)$ می‌شود. اما با استفاده از روش زیر این مقدار در حالت متوسط به $O(n \cdot m \cdot (\log n + k))$، کاهش می‌یابد. ابتدا فاصله‌ی کلیه‌ی نقاط را از مبدأ مختصات حساب کرده، سپس آن‌ها را براساس فاصله تا مبدأ مختصات مرتب می‌نماییم($O(n \log(n))$). آن‌گاه برای هر نقطه‌ی P، تا اولین نقطه‌ای که فاصله‌ی آن از P، بیشتر از ρ باشد، پیش رفته و مقدار مختصات آن را به P و برعکس می‌افزاییم. برای هر نقطه شمارنده‌ای را یک واحد می‌افزاییم تا تعداد نقاط افزوده‌شده(همسایه) مشخص بشود. k بیشترین تعداد همسایه‌ای است که یک نقطه می‌تواند داشته باشد($O(n \cdot m \cdot k)$).

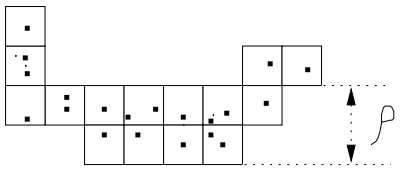
۳.۲ استخراج مکعب‌های محیطی

یکی از مراحل مهم، استخراج مکعب‌های محیطی است که طی آن، پوسته از حالت نقطه‌ای خارج و به صورت یک شکل هندسی تبدیل می‌گردد(شکل۱). برای انجام این کار مکعب واحد را به تعداد $\frac{۱}{\rho} \times \frac{۱}{\rho} \times \frac{۱}{\rho}$ مکعب کوچک تقسیم می‌کنیم. با استفاده از مختصات نقاط، شماره‌ی مکعبی را که هر نقطه در آن قرار می‌گیرد، محاسبه نموده و آن مکعب را درصورت تکراری نبودن، در یک فرهنگ‌داده[۶]، درج می‌کنیم($O(n)$). در پایان، مکعب‌های موجود در این فرهنگ داده‌ای، مکعب‌های پوسته هستند. برای آن‌که به هر یک از ریزمکعب‌ها، شماره‌ای انحصاری نسبت دهیم، از فرمول زیر استفاده می‌کنیم.

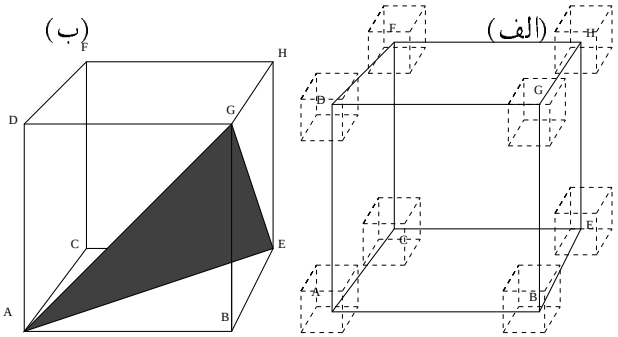
$$N = (((z \times MaxGrid) + y) \times MaxGrid) + x \qquad (۱)$$

[۵] تعداد مراتب میانگین‌گیری، باید به صورت تجربی تعیین شود.

[۶] dictionary



شکل ۲: نمونه‌ای دوبعدی از خطای استخراج مکعب‌های پوسته(مکعب‌های دو ردیفه).



شکل ۳: (الف)— نمای ۸ مکعب مجاور که در استخراج مثلث‌ها به‌کارمی‌روند.
(ب)— نمای یک مثلث که در راستای محورX دیده می‌شود.

که در آن (x,y,z)، مختصات کمینه‌ی ریزمکعب و $MaxGrid$، سقف مجاز برای تعداد مکعب‌ها در راستای هر یک از محورها است. برای تعیین مکعب‌های متناظر نقاط، کافی است مختصات نقاط در $\frac{1}{\rho}$ ضرب شود.

$$x = int(\frac{x}{\rho}), y = int(\frac{y}{\rho}), z = int(\frac{z}{\rho}) \qquad (۲)$$

۴.۲ استخراج مثلث‌ها

آخرین مرحله‌ی الگوریتم، استخراج مثلث‌ها از مکعب‌ها است. این کار به کمک مکعب‌های مجاور و به صورت کاملاً محلی، انجام می‌شود. لذا این مرحله می‌تواند توسط یک روش ریزدانه[۷] موازی انجام گیرد.

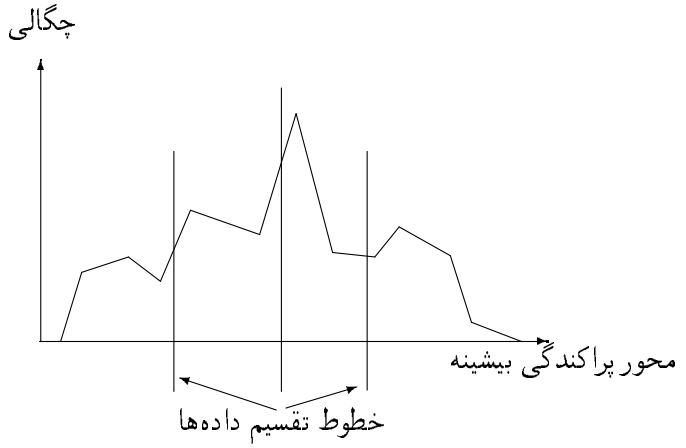
پس از تعیین مکعب‌های متقاطع با پوسته، مرکز هریک را به‌عنوان یک گره شبکه فرض می‌کنیم و برای هر مکعب A، ۷ مکعب مجاور آن را که مختصات آنها بزرگتر یا مساوی A است، در نظر می‌گیریم. به عنوان نمونه در شکل ۳—الف، مکعب‌های G,F,E,D,C,B و H، مجاور مکعب A، محسوب می‌شود. استخراج مثلث‌ها به این صورت است که در ابتدا تعیین می‌کنیم که کدامیک از مکعب‌ها به پوسته تعلق‌دارد(البته تعلق مکعب A به پوسته بدیهی است). سپس تمام مثلث‌های ممکن را که در راستای سه محور مختصات: X، Y و Z، و در جهت مثبت آن‌ها، دیده می‌شوند، مشخص می‌نماییم. به طور مثال اگر E و G تعریف شده باشند، ولی B تعریف نشده باشد، در راستای محور X، مثلث AEG به جای مثلث ABG دیده می‌شود(شکل۳—ب).

با توجه به تولید همه‌ی مثلث‌هایی که در راستای سه محور دیده می‌شوند، می‌توان نتیجه گرفت که‌پوسته‌ی سه‌بعدی از همه طرف پوشش داده می شود. اگر سطوح به اندازه‌ی کافی فاصله داشته و مکعب‌ها به خوبی استخراج شده باشند، این روش درست کار می‌کند. از ضعف‌های این الگوریتم ایجاد مثلث‌های تکراری در پوسته است. این مشکل در پیاده‌سازی، با استفاده از چند متغیر کمکی رفع شده است.

مرتبه‌ی کل الگوریتم برابر $O(n \cdot m \cdot (\log n + k))$، است که برابر بزرگترین مرتبه در مراحل مختلف می‌باشد.

---

[۷] Fine Grain



شکل ۴: نمودار چگالی : سطح زیر منحنی برابر تعداد نقاطی است که به هر پردازنده نسبت داده‌می‌شود.

۳ روش موازی شبکه‌بندی

روش‌های شبکه‌بندی به دلیل تولید شبکه برای یک بار و استفاده از آن برای بارها، بر نوع شبکه و کیفیت آن تأکید دارند. از این رو کندی الگوریتمی که شبکه‌ای مطلوب ایجاد می‌کند قابل قبول است. راه حل مشکل سرعت پردازش موازی است و آن روشی است که طی آن یک فرآیند کند و وقتگیر، توسط چند پردازنده و به صورت هم‌زمان انجام می‌شود تا از این طریق زمان انجام فرآیند کاهش یابد. در شبکه‌بندی نیز، به شرط این‌که بر کیفیت شبکه لطمه‌ی قابل ملاحظه‌ای وارد نشود، می‌توان از پردازش موازی استفاده نمود.

روش کار الگوریتم موازی که یک روش درشت دانه است، به این صورت است که در ابتدا مجموعه‌ی نقاط نمونه، به صورت تقریباً عادلانه توسط پردازنده‌ی اصلی تقسیم و به همراه پارامترهای لازم، برای سایر پردازنده‌ها ارسال می‌گردد. هر پردازنده، پس از دریافت نقاط، عملیات شبکه‌بندی را آغاز نموده و در پایان زیرشبکه‌ی[۸] حاصل را بلافاصله، به پردازنده‌ی اصلی برمی‌گرداند.

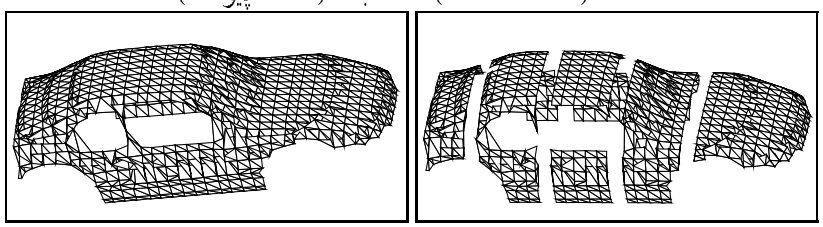
۱.۳ تقسیم داده‌ها

در شبکه‌بندی ترتیبی دیدیم که پس از نگاشت نقاط به مکعب واحد، این مکعب به چندین مکعب کوچکتر که اندازه‌ی ضلع هر یک ρ است، تقسیم می‌شود. در الگوریم موازی، از این مکعب‌های ریز برای تقسیم داده‌ها بین مجموعه‌ی پردازنده‌ها استفاده می‌کنیم. اگر p پردازنده‌ی موازی داشته باشیم، ساده‌ترین روش برش مکعب واحد توسط $p-1$ صفحه‌ی موازی از حد فاصل مکعب‌های ریز است. هر پردازنده‌ی جانبی، مسؤولیت شبکه‌بندی نقاط نمونه‌ی یکی از نواحی بریده شده را عهده‌دار می‌شود. البته تقسیم داده‌ها باید به گونه‌ای باشد تا موازنه‌ی بار، عادلانه انجام گیرد. در غیر این صورت ممکن است، همه‌ی نقاط به یک پردازنده فرستاده شوند و الگوریتم موازی، عملاً همان الگوریتم ترتیبی به‌علاوه‌ی هزینه‌ی انتقال داده‌ها[۹] باشد. در تقسیم داده‌ها می‌توان اولویت پردازنده‌ها را نیز منظور کرد. اگر پردازنده‌ها توان محاسباتی، سرعت و اندازه‌ی حافظه‌ی متفاوتی داشته باشند، اولویت‌های مختلفی خواهند داشت. فرض ما یکسان بودن شرایط همه‌ی پردازنده‌ها است. برای بهتر انجام شدن شبکه‌بندی موازی، لازم است تقسیم داده‌ها در راستای محوری که نقاط در آن بیشترین پراکندگی را دارند انجام گیرد. تعیین این محور با $O(n)$ و هنگامی‌که نقاط به مکعب واحد نگاشته می‌شوند انجام می‌گیرد.

---

[۸] SubMesh
[۹] IO

الف—(حالت گسسته) ب—(حالت پیوسته)



شکل ۵: شبکه‌بندی موازی خودرو، با ۵ پردازنده.

۲.۳ موازنه‌ی بار

اگر نقاط نمونه پراکندگی و انحنای یکنواختی بر پوسته داشته باشند، می‌توان توزیع عادلانه‌ی بار را با تقسیم مساوی نقاط در بازه‌های پیوسته انجام داد. این کار، بهترین روش موازنه‌ی بار است که در نتیجه‌ی آن همه‌ی پردازنده‌ها، تقریباً کاری برابر یکدیگر انجام می‌دهند. .برای این کار نمودار چگالی نقاط بر محور d، که نقاط در راستای آن بیشترین پراکندگی را دارند به‌دست می‌آوریم. سطح زیر نمودار برابر کل تعداد نقاط نمونه است. با تقسیم این نمودار به $p-1$ ناحیه با سطح یکسان، محل‌های برش به‌دست می‌آید(شکل۴). به این ترتیب به همه‌ی پردازنده‌ها تقریباً تعداد یکسانی نقطه تخصیص داده می‌شود.

۳.۳ زیر شبکه‌بندی

نقاط پس از تقسیم به همراه پارامترهای لازم مثل ρ، برای پردازنده‌های جانبی ارسال می‌گردد. هر پردازنده به صورت کاملاً مجزا و بدون هیچ ارتباط پیغامی با سایرین، محدوده‌ی خود را شبکه‌بندی می‌کند. روش شبکه‌بندی همان روش ترتیبی است که در بخش ۲ شرح داده شد. در پایان هر پردازنده زیر شبکه‌ی خود را به پردازنده‌ی اصلی برمی‌گرداند تا شبکه‌ی نهایی در آنجا شکل گیرد.

۴.۳ اتصال زیر شبکه‌ها

شبکه‌ی حاصل از زیر شبکه‌ها، یک شبکه‌ی گسسته در پردازنده‌ی اصلی است. شکل ۵ نتیجه‌ی شبکه‌بندی یک خودرو را توسط پنج پردازنده نشان می‌دهد. در نتیجه لازم است که یک الگوریتم پس‌پردازش، مرزها را به یکدیگر متصل نموده و شبکه‌ی نهایی را به‌دست آورد.

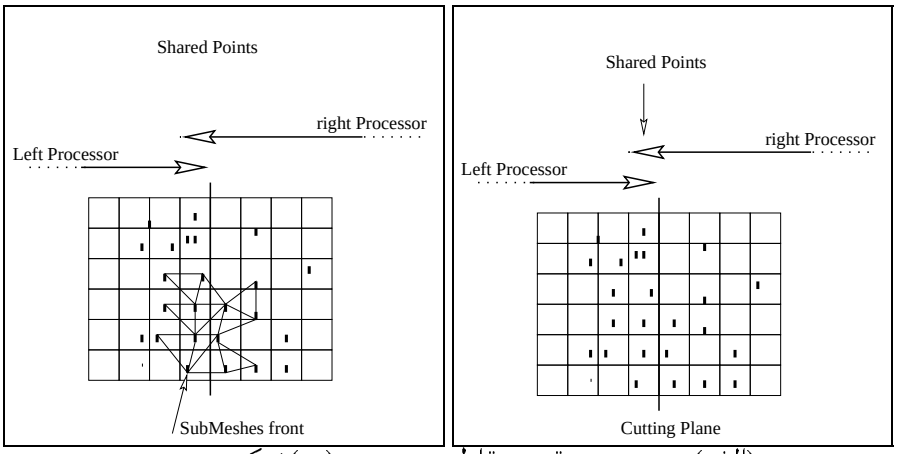
با توجه به طبیعت روش شبکه‌بندی جاری، با یک تغییر کوچک در مرحله‌ی تقسیم داده‌ها، می‌توان مرحله‌ی پس‌پردازش را حذف نمود. ایده‌ی کار برگرفته از شبکه‌بندی ترتیبی است که در آنجا استخراج مثلث‌ها به صورت کاملاً محلی و با مقایسه‌ی هر مکعب پوسته با ۷ همسایه‌ی تعریف شده‌اش[۱۰] انجام می‌گرفت. بنابراین در تقسیم داده‌ها نقاطی را که در آخرین ردیف طرف چپ صفحه‌ی برش قرار دارند، به صورت مشترک به هر دو پردازنده‌ی مجاور ارسال می‌کنیم(شکل۶—الف). پردازنده‌ی چپ، شبکه‌بندی ناحیه‌ی خود را مثل قبل به‌دست می‌آورد و شرایط برای او هیچ فرقی نکرده است. اما پردازنده‌ی راست، شبکه‌بندی خود را از یک ردیف عقبتر نسبت به قبل آغاز می‌کند. این تغییر هیچ تداخلی بین نتایج زیر شبکه‌ی راست و چپ ایجاد نمی‌کند(شکل۶—ب). در این روش، مرزها خود به خود پر می‌شوند و به نظر می‌رسد شبکه‌ی حاصل، تفاوتی با شبکه‌ی ترتیبی نداشته باشد.

در بخش ۲ دیدیم که مرتبه‌ی الگوریتم ترتیبی برابر $O(n \cdot m \cdot (\log n + k))$ است.اگر عملیات به صورت موازی انجام پذیرد، تقسیم نقاط بین پردازنده‌ها تنها بر n اثر می‌گذارد. لذا مرتبه‌ی الگوریتم به $O(\frac{n}{p} \cdot m \cdot (\log \frac{n}{p} + k))$ کاهش می‌یابد. هزینه‌ی ارتباطات به دلیل ناچیز بودن نسبت به هزینه‌ی شبکه‌بندی قابل چشم‌پوشی است. با فرض تقسیم عادلانه‌ی نقاط، تسریع تندی الگوریتم[۱۱] شبکه‌بندی، به اندازه‌ی p برابر قابل پیش‌بینی است.

---

[۱۰] به بخش ۴.۲ مراجعه شود

[۱۱] Speedup



(ب)شبکه‌ی مرزی (الف) محدوده‌ی تقسیم نقاط

شکل ٦: روش جدید تقسیم نقاط برای پردازش موازی.

۴ بررسی نتایج

نرم‌افزارهای شبکه‌بندی ترتیبی و موازی به زبان $C++$ تحت لینوکس پیاده سازی شده‌اند. محیط پردازش موازی اِم‌پی‌آی[۱۲] و بستر سخت‌افزاری، شامل ۴ عدد پردازنده‌ی پنتیوم ۲۳۳ با ۳۲ مگابایت حافظه و ۱ عدد پنتیوم ١٦٦ با ١٦ مگابایت حافظه است که با خط ارتباطی ۱۰۰ مگابیت بر ثانیه به یکدیگر متصل شده‌اند.

اشکالاتی در الگوریتم وجود دارد که باعث کاهش میزان درستی شبکه می‌شود. یک اشکال این الگوریتم ساده در محاسبه‌ی مکعب‌های محیطی می‌باشد که در نتیجه‌ی آن پوسته‌ای هرمی ایجاد می‌شود. حل این مشکل به پس پردازش نیاز دارد. با کاهش پارامتر چگالی این مشکل تاحدودی برطرف می‌شود. اشکال دیگر یکنواخت بودن اندازه‌ی عناصر شبکه است که برای اجتناب از آن باید نقاط نمونه توزیع یکنواختی بر پوسته داشته باشند.

الگوریتم‌های موازی و ترتیبی شبکه‌بندی روی ۱۲ پوسته‌ی مختلف اجرا شدند که چند نمونه از نتایج شبکه‌بندی این الگوریتم‌ها در شکل ۷ آمده است. هرچند خطای شبکه‌ی حاصل زیاد و کیفیت آن پایین است ولی اختلاف شبکه‌ی موازی نسبت به سریال بسیار کم است. تفاوت دو شبکه‌ی موازی و ترتیبی با استفاده از فرمول $\epsilon_p = \frac{1}{2}(\frac{\Delta_v}{f} + \frac{\Delta_v}{v})$ به طور متوسط ۱/۴۱ درصد می‌باشد. مهمترین حسن روش موازی آن است که تنها در دو مرحله انجام می‌گیرد. شبکه‌بندی موازی پیش‌بینی‌های تند شدن[۱۳] الگوریتم را برآورده است. این روش می‌تواند به عنوان شروعی برای شبکه‌بندی موازی با کیفیت‌های بهتر باشد.

در ادامه‌ی این کار، موازی سازی روش هوپ و یا اصلاح الگوریتم جاری را، می‌توان مورد بررسی قرار داد.

مراجع

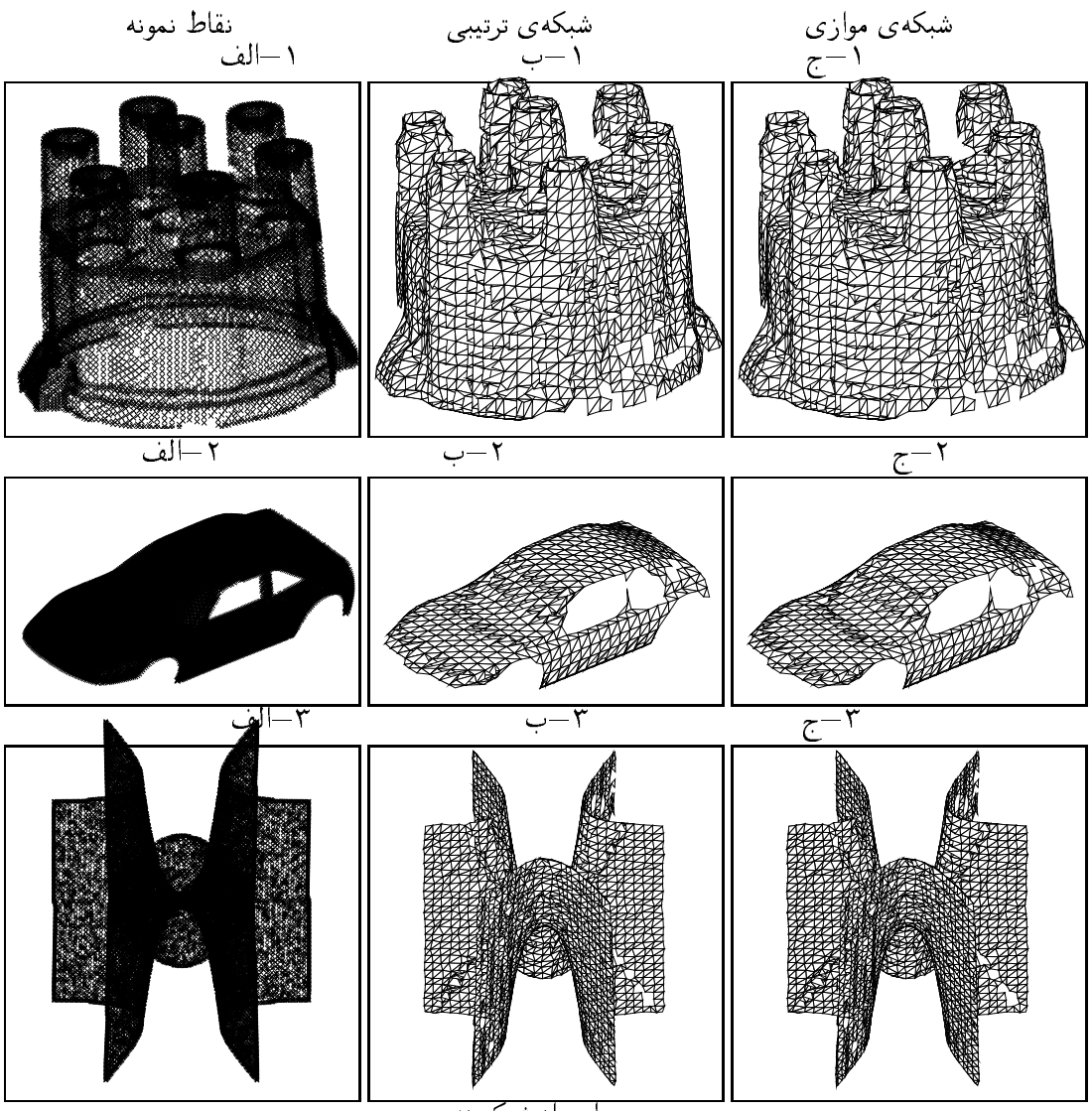
[1] M.Bern, D.Eppstein, and J.Gilbert, "Provably Good Mesh Generation," *Proc. 31'st Symp. on Foundation of Computer Science,* pp. 231-241, 1990.

[2] H.Borouchaki, F.Hecht, E.Saltel, and P.L.George, "Reasonably efficient Delaunay based mesh generation in 3 dimensions," *Proceedings 4th International Meshing Roundtable*, pp. 3-14, Oct. 1995.

[3] F.J.Bossen, and P.S.Heckbert, "A Plaint Method for Anisotropic Mesh Generation," *International Meshing Roundtable*, pp. 63-80, 1996.

[4] L.D.Florini, and P.Magillo, "Triangle-Based Surface Models," *Information and Computer Science Department(DISI)* University of Genova.

۱۲ Message Passing Interface(MPI)
۱۳ Speedup



نمونه‌هایی از شبکه‌بندی

[5] H.Hoppe, "Surface Reconstruction from Unrecognized Points," *Ph.D thesis in washington university*, 1994.

[6] Y.Kono, M.Seto, K.Nishimatsu, H.Fukumori, and Y.Muraoka, "Parallel Mesh Generation of Finite Element Method,...," *School of Science and Engineering, WASEDA Uinversity japan.*

[7] D.Rypl, and Z.Bittnar, "Direct Triangulation of 3D Surfaces Using The Advancing Front Technique," *Department of Structural Mechanics, Czech University-prague.*

[8] D.Rypl, and P.Krysl, "Triangulation of 3D structurs," *Accepted by Engineering with Computers*, 1995.

[9] D.Rypl, and Z.Bittnar, "Parallel 3D Mesh Generator," Department of Structural Mechanics, Technical University in Prague.