

توسعه روش گراف در شناسایی فیچرهای ماشین کاری

کیوان رحمانی^۱؛ بهروز آرزو^۲

چکیده

شناسایی فیچرهای ساخت از مدل CAD مهم‌ترین مرحله در راه ایجاد یک سیستم اتوماتیک برنامه‌ریزی فرآیند ساخت است. فیچرها نسبت به اجزای استفاده شده در مدل CAD عناصر سطح بالاتری هستند که اطلاعات طراحی را به فعالیت‌های ساخت مرتبط می‌کنند. یکی از متداول‌ترین روش‌ها برای شناسایی فیچرها استفاده از گراف توصیف کننده نحوه ارتباط وجوه و همین‌طور ویژگی‌های لبه‌های یک قطعه است که به آنها روش‌های متکی بر گراف گفته می‌شود. در این مقاله، یک روش توسعه یافته متکی بر گراف ارائه شده است که نسبت به روش‌های معمول متکی بر گراف قابلیت‌های بالاتری دارد؛ به این ترتیب که این روش، راه حل کلی‌تری را برای حل مساله فیچرهای متقاطع ارائه می‌کند و از نظر محاسباتی نیز مناسب‌تر از روش‌های گذشته است. همچنین الگوریتم‌های متکی بر گراف پیشنهاد شده تاکنون بیشتر به قطعات چند وجهی و فیچرهای ۲/۵ بعدی پرداخته‌اند؛ اما در این مقاله استفاده از روش‌های متکی بر گراف به محدوده‌ای از فیچرهای غیر چند وجهی و سه بعدی نیز توسعه داده شده است.

کلمات کلیدی

CAD, CAM, CAPP, شناسایی فیچرها, Graph Theory.

Machining Feature Recognition Using an Extended Graph Theory

K. Rahmani; B. Arezoo

ABSTRACT

Manufacturing feature recognition is a basic component for systems that automatically generate manufacturing processes. When compared to the set of low level entities existing in a component's CAD model, features are a set of higher level entities that model the correspondence between design information and manufacturing activities. Graph-based methods, which use the part adjacency graph to recognize features, are among the most common techniques in the existing feature recognition literature. In this paper, some techniques are proposed which significantly improve the performance of traditional graph-based algorithms. The approach is more general and efficient than common graph-based approaches dealing with the problem of feature interactions. The scope of the graph-based approaches is also extended in this paper to include curved and 3D features in addition to polyhedral features, where most of the existing graph-based algorithms have focused only on the polyhedral objects.

KEYWORDS

CAD, CAM, CAPP, Feature recognition, Graph Theory.

^۱ کارشناس ارشد مهندسی مکانیک (ساخت و تولید)، دانشگاه صنعتی امیرکبیر.

^۲ دانشیار دانشکده مهندسی مکانیک، دانشگاه صنعتی امیرکبیر: barezoo@aut.ac.ir

محدب یا مقعر بودن لبه‌ها نیز اضافه می‌شود. گراف مورد نظر سپس به کمک یک heuristic به زیرگراف‌های^{۱۱} خود تجزیه می‌شود. زیرگراف‌های حاصل نیز برای تعیین نوع فیچر آنها تحلیل می‌شوند. در روش، از الگوریتم‌های شباهت‌یابی بین زیرگراف‌ها^{۱۲} برای مقایسه زیرگراف‌های تجزیه شده با یک سری گراف‌های الگو استفاده می‌شود. این روش در شناسایی فیچرهای جدا از هم بخوبی عمل می‌کند، ولی در شناسایی فیچرهای متقاطع عملکرد خوبی ندارد.

"معرفت"^۶ [۶]، [۷]، [۸] با استفاده از تکنیک‌های تست فرضیه^۲، سعی که تا قابلیت شناسایی فیچرهای متقاطع را در یک روش متکی بر گراف بهبود بخشد. او ملاحظه کرد که بر اثر تقاطع فیچرها، قوس‌های موجود در گراف قطعه ممکن است حذف شده باشند؛ بنابراین تلاش کرد به طریقی این قوس‌های حذف شده را به گراف قطعه بازگرداند. روش او به قطعات چند وجهی^۴ محدود بود و در بازگرداندن قوس‌های ناپدید شده و شناسایی فیچرها همیشه موفق عمل نمی‌کند.

"زو"^۹ [۹] نیز الگوریتمی را برای شناسایی فیچرهای فرم در قطعات ۲/۵ بعدی ارائه کرده است. در روش او، با توجه به اطلاعات تحدب و تقعر وجوه و لبه‌ها، به جستجوی نشانه‌های وجود فیچر پرداخته می‌شود و فیچرها بر اساس این اطلاعات شناخته می‌شوند. البته فیچرهایی که در این روش شناسایی می‌شوند، مستقیماً به عملیات ماشین‌کاری مربوط نمی‌شوند. در اینجا یک طبقه‌بندی کلی از فیچرها ارائه می‌شود که برای استفاده در ماشین‌کاری باید بیشتر تجزیه و تحلیل شود.

"کیم"^{۱۰} [۱۰]، [۱۱]، [۱۲] روش تجزیه محدب^{۱۵} را برای شناسایی فیچرها پیشنهاد کرد که یک روش چند مرحله‌ای است. در اینجا ابتدا از تفاضل منظم^{۱۱} مدل قطعه از پوسته محدب^{۱۷} آن یک حجم واسطه تشکیل می‌شود. حجم واسطه حاصل نیز به همان ترتیب از پوسته محدب خود تفریق می‌شود و این عمل به همین شکل برای حجم‌های واسطه بعدی نیز ادامه می‌یابد تا زمانی که یک مجموعه تهی، حاصل تفاضل باشد. در مرحله بعد کلیه حجم‌های واسطه ایجاد شده به صورت متناوب جمع و تفریق می‌شوند و اجزای حاصل از این جمع و تفریق متناوب به عنوان فیچر فرم در نظر گرفته می‌شوند. در ادامه روش‌هایی نیز برای ترکیب کردن این فیچرهای فرم و ایجاد فیچرهای ماشین‌کاری از آنها پیشنهاد شده است. این روش ممکن است به ایجاد یک مجموعه ناخواسته از فیچرها منجر شود. همچنین این روش ذاتاً برای قطعات چند وجهی مناسب است.

"ساکورای"^{۱۳} [۱۳]، [۱۴] از تکنیک تجزیه سلولی^{۱۸} برای شناسایی فیچرها استفاده کرد. در این روش ابتدا تمام سطوح و

برنامه‌ریزی فرآیند تولید به کمک کامپیوتر^۱ (CAPP) واسطه ارتباط دهنده CAD و CAM^۲ در یک سیستم طراحی و ساخت اتوماتیک است. در چنین سیستمی یکی از مهم‌ترین مسائل این است که چگونه می‌توان اطلاعات موجود در CAD را به صورتی که در CAPP قابل استفاده باشند، تفسیر کرد؛ چرا که CAD مدل قطعه را به صورت هندسی بیان می‌کند؛ ولی به اطلاعاتی که در CAPP احتیاج است، فیچرهای ساخت هستند که اطلاعات مربوط به آنها باید به نحوی از مدل CAD استخراج و استنتاج شود. شناسایی فیچرها^۴ یکی از مهم‌ترین روش‌هایی است که تاکنون برای ایجاد ارتباط اتوماتیک بین CAD و CAPP ارائه شده است. در شناسایی فیچرها، هدف اصلی شناسایی اجزای سطح بالاتر (فیچرها) از مجموعه اجزای سطح پایین‌تر موجود در مدل هندسی قطعه، نظیر نقاط، لبه‌ها و سطوح است.

تحقیقات در زمینه شناسایی فیچرها بیش از دو دهه پیش آغاز شده است؛ اما نتایج این تحقیقات بر خلاف حجم بسیار زیاد کارهای انجام شده، به میزان بسیار کمی وارد صنعت شده است؛ زیرا هنوز تا دستیابی به سیستمی که به طرز معقولی قادر به شناسایی فیچرها در کاربردهای صنعتی باشد و فیچرهای شناسایی شده به وسیله آن کاملاً اطلاعات مورد نیاز یک سیستم CAPP را تامین کنند، راه درازی باقی مانده است [۱].

یکی از مهم‌ترین موضوعاتی که در کارهای تحقیقاتی انجام شده در زمینه شناسایی فیچرها به چشم می‌خورد، مسأله تقاطع فیچرها^۲ است [۲]. تقاطع فیچرها به حالتی گفته می‌شود که دو حجم فیچر یکدیگر را قطع کنند. در چنین حالتی، اطلاعات استفاده شده برای شناسایی فیچرها یا تغییر می‌کنند و یا کاملاً از بین می‌روند، که باعث دشوار شدن شناسایی فیچرها می‌گردد. از آنجا که روش‌های متکی بر گراف تاکنون توانایی محدودی را در شناسایی فیچرهای متقاطع ارائه کرده‌اند، هدف بعدی مورد نظر در این مقاله، توسعه بیشتر توانایی شناسایی فیچرهای متقاطع در روش‌های سنتی متکی بر گراف است. تشخیص الگوی گرامری^۱ و استفاده از سیستم‌های خبره^۳؛ از جمله روش‌های به کار گرفته شده در شناسایی فیچرها هستند که امروزه دیگر چندان مورد توجه نیستند [۲]، [۴].

تبدیل نمایش مرزی (B-Rep)^۵ قطعه به یک گراف را؛ که گره‌های^۹ آن معرف وجوه قطعه و قوس‌های^{۱۱} آن معرف لبه‌های قطعه می‌باشند، نخستین بار "جوشی و چانگ" فرمول‌بندی کردند [۵]. به این گراف اطلاعاتی اضافی نظیر

نیم فضاهاى موجود در حجم دلتا^{۱۹} (حجمى که از تفاضل محصول از قطعه خام ایجاد مى‌شود) با یکدیگر تقاطع داده مى‌شوند و به این ترتیب حجم دلتا به کوچک‌ترین سلول‌هاى ممکن تجزیه مى‌شود. سپس با ترکیب زیرمجموعه‌هاى این سلول‌ها، احجام ماشین‌کاری ایجاد مى‌شوند؛ که در مرحله بعدى به صورت یک فیچر ماشین‌کاری طبقه‌بندى مى‌شوند. پیچیدگى زمانى محاسبات انجام شده در این روش از مرتبه‌نمایی^{۲۰} بوده و با افزایش تعداد فیچرها به شدت افزایش مى‌یابد^{۲۱} و از سوى دیگر، فیچرهاى ایجاد شده ممکن است با هیچ مجموعه از پیش تعریف شده و شناخته شده‌اى از فیچرها مطابقت نداشته باشند.

"ساندارارجان"^[۱۵] روشى را برای شناسایی فیچرهاى با فرم دلخواه^{۲۲} ارائه کرد. در کار او فیچرهاى فرم دلخواه، سطوح آزادى هستند که به یک کانتور مسطح محدود شده‌اند. در این روش، سطوح منحنى با سطوح تختى، که در برگیرنده آنهاست، جایگزین و برای ایجاد حجم فیچر سطح پوشش دهنده به سمت سطح آزاد متناظر آن جاروب مى‌شوند و به این ترتیب، حجم ماشین‌کاری فیچر ایجاد مى‌شود که به سطح آزاد محدود شده است.

"وندن برند"^[۱۳]، استدلالى متكى بر ردیابی^{۲۳} فیچرها را برای حل مساله فیچرهاى متقاطع ارایه کرد. او اصل وجود فیچر را به این صورت تعریف کرد که یک فیچر و عملیات ماشین‌کاری مرتبط با آن حتى زمانى که فیچرها تقاطع مى‌کنند باید از خود اثرى در مرز قطعه به جا گذاشته باشد. این ردپاها ممکن است هندسى، فیچرهاى طراحی، تئرانس‌ها و دیگر صفات طراحی باشند. به طور مثال، وجود دو وجه موازى و داراى جهت مختلف نشانه وجود شیاری خواهد بود که این سطوح مطابق با سطوح جانبى آن مى‌باشند. در مرحله بعدى یک حجم برای این رد پا ایجاد شده و سپس صحت این حجم ارزیابى

مى‌شود.

در بین روش‌هاىی که در این قسمت شرح داده شدند، استفاده از گراف یکى از متداول‌ترین روش‌هاى موجود در زمینه شناسایی فیچرهاست. این روش‌ها مزایایى نسبت به روش‌هاى دیگر که از آن جمله مى‌توان به موارد زیر اشاره کرد [۱۶]:

۱- آنها را مى‌توان در محدوده‌هاى دیگری علاوه بر ماشین‌کاری نیز به کار برد.

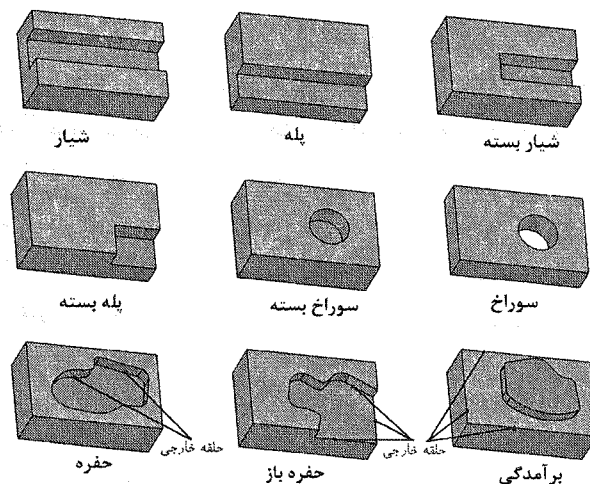
۲- این روش‌ها به کاربر اجازه مى‌دهد که انواع جدیدى از فیچرها را بدون تغییر متن و کد برنامه اضافه کند.

۳- این روش‌ها در شناسایی فیچرهاى غیر متقاطع بسیار موفق عمل مى‌کنند.

بنابراین اگر به نحوى اشکالات روش‌هاى متكى بر گراف تصحیح و قابلیت‌هاى آن بهبود داده شوند، توانایی شناسایی فیچرها در آنها بسیار بهبود خواهد یافت. این موضوع هدف مقاله حاضر است که در قسمت‌هاى بعدى به این موضوع پرداخته خواهد شد.

۲- دسته بندى فیچرها

فیچرهاى بررسی شده در این مقاله به ۹ گروه تقسیم‌بندى شده‌اند: شیار^{۲۴}، پله^{۲۵}، شیار بسته^{۲۶}، پله بسته^{۲۷}، سوراخ بسته^{۲۸}، سوراخ راه به در^{۲۹}، حفره^{۳۰}، حفره باز^{۳۱} و برآمدگى^{۳۲} که مثال‌هاىی از هر یک در شکل (۱) مشاهده مى‌شود. در این طبقه‌بندى سعی شده است تا بیشتر فیچرهاىی که متناظر با ایجاد یک شکل هندسى و براده بردارى در قطعه هستند، در طرح گنجانده شوند. طرح ارائه شده حاوى فیچرهاى منحنى نیز است و برای برخى از فیچرهاى سه بعدى نیز قابل استفاده مى‌باشد.



شکل (۱): طبقه‌بندى فیچرهاى بررسی شده

۳- تعاریف مقدماتی

پیش از توضیح نحوه عملکرد الگوریتم‌های این مقاله، به دلیل استفاده مکرر از مفاهیمی نظیر وجه، سطح، سطوح تحلیلی، لبه و نیم لبه لازم است این موارد به طور دقیق و مختصر شرح داده شوند.

۳-۱- سطوح

به طور کلی سطوح مورد استفاده در سیستم‌های CAD را می‌توان به دو دسته تقسیم کرد: سطوح تحلیلی^{۲۳} و سطوح تصنعی^{۲۴}. سطوح تحلیلی شامل صفحه، سطح، مخروط، کره و سطوح مدور^{۲۶} و استوانه غیر مدور^{۲۷} می‌شوند. توجه شود که به این ترتیب سطوحی نظیر استوانه مدور، مخروط، کره و چنبره زیر مجموعه‌ای از سطوح مدور خواهند بود [۱۷].

در این مقاله از بین این سطوح تحلیلی از این پس پنج سطح صفحه‌ای، استوانه مدور، مخروطی، کره‌ای و چنبره‌ای^{۲۸} سطوح تحلیلی ساده نامیده می‌شوند. سطوح تصنعی نیز شامل سطوحی نظیر Bezier و B-spline می‌گردند که از انطباق (fitting) یک سطح آزاد بر مجموعه‌ای از نقاط حاصل می‌شوند [۱۷].

۳-۲- مدل توپر^{۲۹}

در کار حاضر از نمایش (B-rep)^{۳۰} قطعه برای شناسایی فیچرها استفاده شده است. امروزه استفاده از مدل‌سازهای B-rep به دلیل مزایایی که دارند به سرعت رو به گسترش است. نمایش مرزی بر یک ایده توپولوژیکی استوار شده است و آن این است که یک جسم فیزیکی با مجموعه‌ای از وجوه^{۳۱} پوشانده شده است. این وجوه زیر مجموعه‌های محدودی از یک سری سطوح بسته جهت‌پذیر^{۳۲} هستند که خود را قطع نمی‌کنند. سطح بسته به سطحی گفته می‌شود که پیوسته و مشتق‌پذیر باشد. سطح جهت‌دار نیز سطحی است که می‌توان از جهت بردار نرمال آن برای اشاره به داخل یا خارج مدل توپر استفاده کرد. مرز یک وجه به یک یا چند حلقه^{۳۳} محدود می‌شود. حلقه‌ها مجموعه‌های مرتبی از رئوس^{۳۴} و لبه‌های^{۳۵} جسم هستند که به صورت یک در میان قرار گرفته‌اند. یک رأس در مدل توپر به یک نقطه منحصر به فرد در فضا گفته می‌شود و یک لبه به منحنی محدود و جهت‌داری گفته می‌شود که به دو رأس محدود شده و خود را قطع نمی‌کند.

۳-۳- فیچرهای سه بعدی

موضوع دیگری که توضیح آن در اینجا مناسب است، تعریف فیچرهای سه بعدی است. پیش از تعریف فیچرهای سه

بعدی ابتدا باید مفهوم امتداد یک مجموعه^{۳۶} تشریح شود:

برای امتداد یک مجموعه، هر نقطه از مجموعه در امتداد یک مسیر مشخص؛ که به همان نقطه وابسته است، حرکت داده می‌شود. فرض شود S مجموعه‌ای شامل نقاط P باشد ($S = \{P\}$) و مجموعه‌ای از منحنی‌های متصل به این نقاط با نام $C(P)$ وجود داشته باشد؛ به طوری که نقطه انتهایی این منحنی‌ها در P واقع شده باشد؛ در این صورت امتداد مجموعه S که $Ext(S)$ نامیده می‌شود فضایی است که نقاط P با حرکت در امتداد منحنی‌های $C(P)$ متناظرشان جاروب می‌کنند. به عبارت دیگر:

$$S = \{P\} \rightarrow Ext(S) = \bigcup_{P \in S} C(P) \quad (1)$$

اکنون بر اساس تعریف فوق، فیچرهای ۲/۵ بعدی و سه بعدی را می‌توان به صورت زیر تعریف کرد:

اگر در (۱)، S مجموعه نقاط واقع بر یک صفحه و $\{C(P)\}$ مجموعه‌ای از خطوط راست موازی عمود بر این صفحه باشد، آنگاه حجم جاروب شده نقاط P را ۲/۵ بعدی می‌نامیم؛ در غیر این صورت حجم مذکور سه بعدی نامیده می‌شود. در قسمت‌های بعدی ملاحظه خواهد شد که روش اجرا شده در کار حاضر علاوه بر فیچرهای ۲/۵ بعدی، فیچرهای سه بعدی را نیز در بر می‌گیرد.

۴- گراف نحوه همسایگی ۴۷

مبنای روش پیشنهادی در مقاله حاضر را گراف نحوه همسایگی تشکیل می‌دهد.

گراف نحوه همسایگی را، نخستین بار "جوشی" [۵] معرفی کرد و پس از آن در بسیاری از کارهای مرتبط بعدی استفاده شده است. این گراف را می‌توان به این صورت تعریف کرد:

سه‌تایی مرتب^{۳۸} $G = (N, L, T)$ که دارای ویژگی‌های زیر است:

- N مجموع ناتهی گره‌های این گراف است به نحوی که به ازای هر وجه f_i از مدل قطعه تنها یک گره n_i در G وجود دارد و برعکس.

- L مجموعه قوس‌های گراف است به نحوی که به ازای هر لبه e_{ij} از قطعه؛ که وجوه f_i و f_j را به هم متصل کرده است، یک قوس واحد l_{ij} در L وجود دارد که گره‌های n_i و n_j را به هم متصل می‌کند. توجه شود که در این تعریف e_{ji}, e_{ij}, j, i است.

- T مجموعه حالت‌های^{۳۹} لبه‌های گراف است؛ به این ترتیب که به هر قوس l_{ij} یک برچسب $t_{ij} \in T$ اختصاص داده

می‌شود که مشخص کننده حالت آن قوس است. حالت‌ها با تابع $\Psi(l): L \rightarrow \{0,1\}$ تعریف می‌شوند. اگر لبه e_{ij} متناظر با قوس l_{ij} مقعر باشد $\Psi(l_{ij})=0$ و اگر محدب باشد $\Psi(l_{ij})=1$.

یک لبه را مقعر گویند، اگر زاویه بین وجوه متناظر با آن در داخل قطعه بزرگتر از 180° باشد و آن را محدب گویند اگر این زاویه کوچکتر از 180° باشد. در شکل (۲) نمونه‌ای از این گراف برای یک قطعه ساده ملاحظه می‌شود. در این شکل، لبه‌های مقعر با خطوط توپر و لبه‌های محدب با خط چین نمایش داده شده‌اند.

۲- لبه‌های محدب/مقعر^۱: فرض شود که θ زاویه بین دو وجه f_i و f_j در قسمت توپر جسم باشد. در این صورت لبه e_{ij} مشترک بین این دو وجه می‌تواند به یکی از صورت‌های زیر باشد:

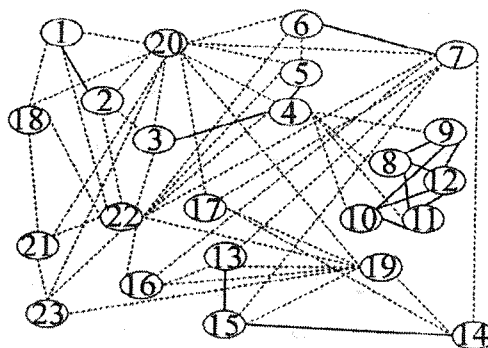
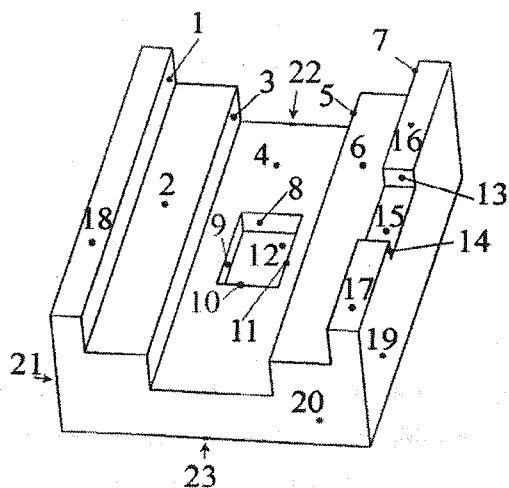
- محدب است اگر $0 < \theta < 180$.
 - محدب صاف^۲ است اگر f_i و f_j بر یکدیگر در طول این لبه مماس، و f_i و f_j هر دو محدب یا یکی محدب و دیگری خنثی باشد.

- مقعر است اگر $180 < \theta < 360$.
 - مقعر صاف^۳ است اگر f_i و f_j بر یکدیگر در طول این لبه مماس، و f_i و f_j هر دو مقعر یا یکی مقعر و دیگری خنثی باشد.

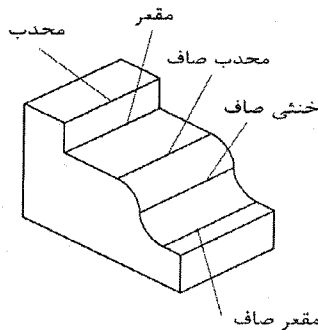
- خنثی صاف^۴ است اگر a و b بر هم مماس، و یکی از آنها محدب و دیگری مقعر باشد.

محدب یا مقعر بودن لبه‌ها در کار جوشی و چانگ، تنها برای قطعات چند وجهی تعریف شده‌اند؛ بنابراین سیستم آنها تنها قادر به شناسایی فیچرهای چند وجهی بوده است. برای توسعه بیشتر این روش ابتدا باید تعاریف کلی‌تری برای تقعر و تحدب لبه‌ها و وجوه قطعه ارائه شود. تعاریف "زو" [۹] می‌تواند در این زمینه بسیار مفید واقع شود، که در زیر به برخی از آنها اشاره می‌شود:

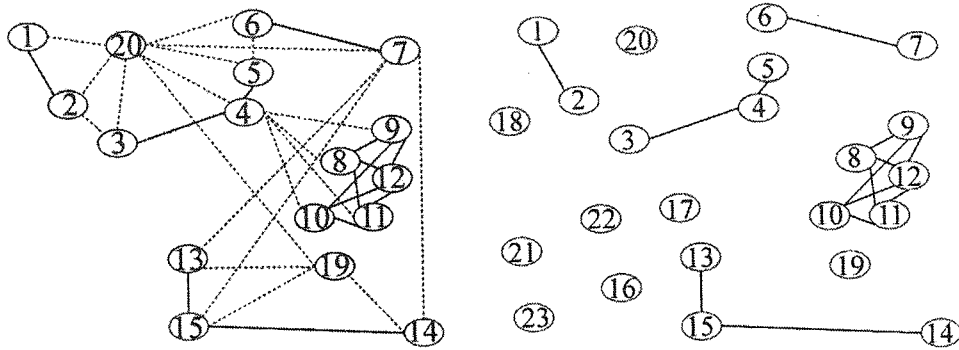
۱- وجوه محدب/مقعر/خنثی^۵: در تعاریف آنها تنها سطوح تحلیلی ساده (مسطح، استوانه‌ای، مخروطی، کروی و چنبره‌ای) در نظر گرفته می‌شوند. به یک وجه، محدب گفته



شکل (۲): گراف نحوه همسایگی برای یک قطعه ساده



شکل (۳): لبه‌های محدب، مقعر و صاف



شکل (۴): زیرگراف‌های حاصل از تجزیه گراف شکل (۲). (a) تجزیه با حذف وجوهی که با لبه‌های محدب محاصره شده‌اند. (b) تجزیه به روش ارائه شده در این مقاله

دستورالعمل در شکل (۴-ا) ملاحظه می‌شود. زیرگراف‌های حاصل از این تجزیه برای شناسایی فیچرها مورد تحلیل قرار می‌گیرند. این روش شکستن گراف نحوه همسایگی، علی‌رغم توانایی قابل ملاحظه‌ای که در شناسایی فیچرهای جدا از هم ارائه می‌کند، در شناسایی فیچرهای متقاطع با مشکلات زیادی روبرو می‌شود. برای مثال همان گونه که در شکل (۴-ا) ملاحظه می‌شود، زیرگراف حاصل هر دو فیچر متقاطع (۳-۴-۵) و (۸-۹-۱۰-۱۱-۱۲) را در بر دارد و برای شناسایی هر یک از آنها باید از الگوریتم‌های شباهت‌یابی بین زیرگراف‌ها استفاده شود. به این صورت که زیرگراف حاصل با مجموعه‌ای از گراف‌های الگو^{۵۵} مقایسه می‌شود و به این ترتیب هر یک از فیچرها از زیرگراف اولیه استخراج و شناسایی می‌شوند.

در استفاده از گراف‌ها به آنچه که باید توجه شود، این نکته است که اطلاعات موجود در گراف بر اثر تقاطع فیچرها ممکن است تغییر یابد یا از بین بروند، که در این صورت این روش با مشکل مواجه می‌شود. از سوی دیگر الگوریتم‌های شباهت‌یابی بین زیرگراف‌ها از نظر زمان محاسباتی اصلاً مقرون به صرفه نیستند. نکته مهمی که در این مقاله به آن پرداخته می‌شود این است که نحوه تجزیه گراف نحوه همسایگی و heuristicهای استفاده شده می‌تواند اثر قابل توجهی بر توانایی روش در شناسایی فیچرهای متقاطع داشته باشد. در کار حاضر، از ایده متفاوتی استفاده می‌شود: به جای جستجوی اطلاعات گم‌شده زیرگراف‌ها که بر اثر تداخل فیچرها از بین رفته‌اند، و بازیابی این اطلاعات و سپس شباهت‌یابی بین زیرگراف‌ها، در اینجا زیرگراف‌هایی ایجاد می‌شوند که کم‌ترین اثر را از تغییرات پذیرفته‌اند. این ایده در واقع مشابه ایده‌ای است که در تمامی روش‌های منکی بر ردیابی استفاده می‌شود: «جستجوی حداقل اطلاعات لازم برای شناسایی یک فیچر». با این تفاوت که در روش پیشنهادی این مقاله این اطلاعات با تجزیه گراف قطعه ایجاد می‌شوند. از سوی دیگر با توجه به

نمونه‌ای از هریک از موارد فوق در شکل (۳) نشان داده شده است. در این مقاله، تعاریف قبلی لبه‌های محدب و مقعر کمی بیشتر توسعه داده شده‌اند، تا برای استفاده در الگوریتم‌های اجرا شده مفیدتر شوند. به این صورت که این تعاریف را نه تنها برای لبه‌های خطی بلکه برای لبه‌های منحنی با فرم دلخواه نیز با توجه به ملاحظه‌ای که در این تحقیق انجام شده است، می‌توان بکار برد:

در یک قطعه $2/5$ بعدی، فرض شود f و g دو وجه دلخواه است که سطوح متناظر با آنها به دلخواه تحلیلی یا تصنعی هستند و این دو وجه در لبه e مشترک باشند. حال اگر زاویه بین f و g در یک نقطه دلخواه در طول e محدب باشد، آنگاه زاویه بین f و g در تمام طول e برای چنین سطوحی محدب خواهد بود و به مقعر تبدیل نخواهد شد. به همین ترتیب اگر زاویه بین f و g در یک نقطه دلخواه در طول e مقعر باشد، آنگاه زاویه بین آنها در تمام طول e مقعر خواهد بود.

بنابراین برای محاسبه حالت یک لبه منحنی، تنها کافیست که حالت آن در یک نقطه از طول آن محاسبه شود. در گراف استفاده شده در کار حاضر برای شناسایی فیچرها، حالت لبه‌های منحنی به علاوه حالت هر یک از وجوه قطعه نیز با گراف همراه می‌شود و این گراف شکل توسعه یافته‌تری به خود می‌گیرد که برای شناسایی محدوده قابل توجهی از فیچرهای موجود در قطعات سه بعدی کفایت می‌کند.

۵- تجزیه گراف نحوه همسایگی

پس از تشکیل گراف نحوه همسایگی، برای شناسایی فیچرها می‌بایست این گراف تجزیه (شکسته) شود. «جوشی» [۵]، این گراف را با استفاده از ملاحظه زیر تجزیه کرد:

«وجهی که همه لبه‌های آن محدب هستند جزئی از یک فیچر نیست و از گراف حذف می‌شود.»

نتیجه حاصل از تجزیه گراف شکل (۲) بر اساس این

مناسب بودن الگوریتم پیشنهادی تجزیه گراف نحوه همسایگی از نظر محاسباتی (نسبت به روش‌های متداول)، سودمندی زیادی خواهد داشت.

برای توضیح روش پیشنهادی تجزیه گراف در این مقاله، ابتدا می‌توان ملاحظه کرد که تقاطع فیچرهای ساده نظیر شیار و پله با یکدیگر معمولاً به ایجاد لبه محذب در ناحیه تقاطع منجر می‌شود. به طور مثال، در شکل (۲) لبه مشترک بین شیار ۴-۵ و ۳ و پله ۶-۷ یک لبه محذب است. بنابراین این دسته از فیچرها را می‌توان با حذف لبه‌های محذب از گراف نحوه همسایگی از یکدیگر جدا کرد. این موضوع در شکل (۴) نمایش داده شده است. همان گونه که در شکل (۴-ب) ملاحظه می‌شود، حذف لبه‌های محذب باعث شده است که فیچرهای ۴-۵ و ۳-۴ و ۱۱-۱۲ و ۸-۹-۱۰ براحتی از یکدیگر جدا و قابل شناسایی شوند. مزیت دیگر روش حذف لبه‌های محذب نسبت به روش جوشی [۵] این است که در اینجا اطلاعات مربوط به وجوه قطعه از بین نمی‌روند. به طور مثال، هر یک از گره‌های ۱۸، ۱۹، ۲۰، ۲۱، ۲۲ و ۲۳ در شکل (۴-ب) می‌توانند نماینده یک سطح کف‌تراشی شده باشند؛ اما این اطلاعات در شکل (۴-ا) از بین رفته‌اند.

با تجزیه گراف نحوه همسایگی مطابق روش پیشنهادی در این مقاله، زیرگراف‌هایی حاصل خواهند شد که تمامی قوس‌های آنها مقعر هستند. این زیرگراف‌ها از این پس در این مقاله زیرگراف‌های مقعر نامیده می‌شوند. توجه شود که اگر چه حفره‌ها، حفره‌های باز و برآمدگی‌ها می‌توانند همان گونه که در شکل (۱) نیز ملاحظه می‌شود حاوی لبه‌های محذب نیز باشند؛ اما حذف قوس‌های متناظر با این لبه‌های محذب عملاً تاثیری بر شکل کلی زیر گراف آنها نمی‌گذارد؛ چرا که در این نوع فیچرها همواره یک وجه پایه وجود دارد که تمامی وجوه دیگر با لبه‌های مقعر به آن متصل هستند و حذف لبه‌های محذب باعث تجزیه بیشتر زیر گراف آنها نمی‌شود.

برای توضیح بیشتر روش تجزیه گراف در این مقاله، فرض شود G معرف گراف نحوه همسایگی قطعه باشد و تجزیه G به ایجاد مجموعه‌های $D = \{g_1, g_2, \dots, g_n\}$ و $Y = \{y_1, y_2, \dots, y_r\}$ منجر شده باشد. در اینجا D مجموعه زیرگراف‌هایی است که تمامی قوس‌های آن مقعر است و در آنها بیش از یک گره وجود دارد. برای مثال، در شکل (۴-ب)، $D = \{g_1, g_2, \dots, g_5\}$ که در آن $g_1 = 1-2$ ، $g_2 = 3-4-5$ ، $g_3 = 6-7$ ، $g_4 = 13-15-14$ و $g_5 = 8-9-10-11-12$ همچنین Y مجموعه زیر-گراف‌هایی است که تنها از یک گره مجزا^۶ تشکیل شده‌اند که از آنها هیچ قوسی نمی‌گذرد. برای مثال در شکل (۴)، $Y = \{F1, F2, F3, F4, F5, F6, F10\}$ است. همچنین فرض شود $A_i = \{a_1, a_2, \dots, a_m\}$ معرف مجموعه

قوس‌های موجود در زیرگراف g_i بوده و $T_i = \{t_1, t_2, \dots, t_m\}$ مجموعه حالت‌های مربوط به قوس‌های A_i است. اگر D با حذف تمامی قوس‌های محذب از G ایجاد شده باشد، آنگاه:

۱- تمامی قوس‌های موجود در g_i مقعر هستند. به عبارت

$$\sum_{j=1}^m t_j = 0 \text{ دیگر}$$

۲- g_i ها معادل یکی از زیرگراف‌های الگوی ابتدایی هستند که برای شناسایی فیچرها بکار برده خواهند شد. این الگوهای ابتدایی در قسمت بعدی تشریح خواهد شد.

$$3- \left(\bigcup_{i=1}^n g_i \right) \cup X \cup Y = G \text{ که در آن } X \text{ مجموعه تمام}$$

قوس‌های محذب G است.

در تعاریف بالا مجموعه D برای شناسایی تمامی انواع فیچرها بجز سوراخ‌های راه بدر بکار می‌رود. برای شناسایی سوراخ‌های راه بدر از مجموعه Y استفاده می‌شود. از میان گره‌های موجود در Y تنها آنهایی اهمیت دارند که وجه متناظر با آنها یک استوانه مقعر باشد؛ در این صورت این وجه به عنوان سوراخ راه بدر شناسایی می‌شود.

در این مقاله برای استخراج زیرگراف‌های مقعر حاصل از شکستن گراف نحوه همسایگی با حذف لبه‌های محذب، از عملیات همجوشی گره‌ها^۷ استفاده می‌شود که دارای پیچیدگی زمانی $O(n^2)$ است [۱۸]. در روش همجوشی گره‌ها ابتدا یک گره از گراف همسایگی در نظر گرفته می‌شود و بعد تمامی گره‌هایی که با این گره همسایه هستند، به آن جوش داده می‌شوند تا یک گره جدید ایجاد شود. سپس همین مراحل برای گره جدید تکرار می‌شود تا زمانی که گرهی در همسایگی برای جوش دادن وجود نداشته باشد. در چنین حالتی یک زیرگراف متصل از گراف مادر بازیابی شده است. در اینجا منظور از جوش دادن این است که یک گره بر گره دیگر منطبق شود؛ در این صورت گره ترکیبی جدید تمامی قوس‌های هر دو گره قبلی را در بر خواهد داشت. این قوس‌ها خود مشخص کننده این است که چه گره‌هایی عضو این زیرگراف هستند. در فرم ماتریسی گراف نحوه همسایگی، همجوشی گره z با گره i از طریق عمل منطقی OR بین این دو انجام می‌شود؛ به این صورت که برچسب‌های (حالت‌ها) سطر z با سطر i و همین طور برچسب‌های ستون z نیز با ستون i جمع (OR) می‌شوند. همجوشی گره i زمانی پایان می‌پذیرد که پس از همجوشی تغییری در تعداد گره‌های غیر همسایه با i حاصل نشود.

۶- الگوهای استفاده شده برای شناسایی فیچرها

- اگر $\sum_{i=1}^q \Psi(e_i) = 0$ ، آنگاه زیرگراف مربوط به یک حفره است.

- اگر $0 < \sum_{i=1}^q \Psi(e_i) < q$ ، آنگاه زیرگراف مربوط به یک حفره باز است.

- اگر $\sum_{i=1}^q \Psi(e_i) = q$ ، آنگاه زیرگراف مربوط به یک برآمدگی است.

همان گونه که ملاحظه می‌شود، الگوهای به کار رفته در اینجا تا حد ممکن کلی است. مثلاً با استفاده از تعریفی که در بالا برای حفره ارائه شده است، انواع وسیعی از حفره‌ها شناسایی می‌شوند. در شناسایی آنها به هیچ وجه محدودیتی از نظر شکل مرز بیرونی آنها و بودن یا نبودن جزیره در آنها وجود ندارد و داشتن حلقه خارجی مقعر به تنهایی نشانه وجود حفره خواهد بود. همچنین توجه شود که این الگوها این ویژگی قابل توجه را نیز دارند که جهت ماشین‌کاری فیچر را نیز به خاطر تشخیص وجه پایه آن مشخص می‌کنند، که می‌تواند در تعیین مراحل ماشین‌کاری^۹ آنها نیز بکار برده شود. نکته قابل توجه در اینجا این است که پیچیدگی زمانی رویه‌های^{۱۰} استفاده شده در الگوریتم شناسایی فیچرها از درجه چند جمله‌ای است. برای مثال رویه تحلیل اینکه آیا زیرگراف g_i یک حفره/حفره باز/برآمدگی است به صورت زیر است:

Procedure Check_pocket_open.pocket_prot(g_i)

1. For every node $N_j \in g_i$
Calculate Degree(N_j)
Next j
2. Find node N_{max} which has maximum degree
3. $Extloop(N_{max}) \leftarrow$ get the external loop of N_{max}
4. count number of concave edges in $Extloop(N_{max})$
5. If all Edges in $Extloop(N_{max})$ are Concave
Then $Featue_Type(g_i) = pocket$
Else
If all Edges in $Extloop(N_{max})$ are convex
Then $Featue_Type(g_i) = protrusion$
Else $Featue_Type(g_i) = open_pocket$
End if

در رویه بالا، اگر n تعداد کل وجوه قطعه باشد، آنگاه کران بالایی تعداد گره‌های موجود در یک زیرگراف برابر با n خواهد بود. پس حداکثر تکرار حلقه For در مرحله ۱ از این رویه برابر با n^2 است. الگوریتم یافتن ماکزیمم در دنباله‌ای از اعداد در مرحله ۲ نیز از مرتبه خطی است. تعداد تکرار لازم در مرحله ۴ برای محاسبه تعداد لبه‌های مقعر در زیرگراف نیز از مرتبه خطی و معمولاً بسیار کوچک‌تر از n است. در بلوک‌های If نیز

پس از تجزیه گراف، شناسایی فیچرها از روی زیرگراف‌های حاصل با استفاده از مجموعه الگوهایی انجام می‌شود که برای تحلیل این زیرگراف‌ها به کار برده می‌شوند. در این مقاله این الگوها از این قرار هستند:

۱- شیار: الگوی آن یک زیرگراف مقعر است که در آن سه گره وجود دارد. یکی از این گره‌ها از درجه ۲ (دارای دو قوس) و دوتای دیگر از درجه ۱ می‌باشد (درجه یک گره تعداد قوس‌هایی است که از آن خارج شده است).

۲- پله: یک زیرگراف مقعر با دو گره است.

۳- شیار بسته: یک زیرگراف مقعر و غیر سیلیکی (cyclic) با بیش از ۳ گره است که وجوه متناظر همه آنها مسطح است و یکی از آنها از درجه ۳ و بقیه از درجه ۲ می‌باشند.

۴- پله بسته: یک زیرگراف سیلیکی مقعر با سه گره است که همگی به هم متصل هستند و درجه یکسانی دارند.

۵- سوراخ بسته: یک زیرگراف مقعر با دو گره است که وجه متناظر با یکی از آنها استوانه مقعر است.

۶- سوراخ سرتاسری: یک گره مجزای مربوط به یک وجه استوانه‌ای مقعر در مجموعه Y .

۷- حفره/حفره باز/برآمدگی: زیرگراف مقعر مربوط به حفره حداقل سه عدد گره دارد که فقط یکی از آنها از درجه ماکزیمم است. اگر حلقه خارجی^۸ وجه متناظر با این گره تماماً از لبه‌های مقعر تشکیل شده باشد، زیرگراف مورد نظر مربوط به یک حفره است. اگر این حلقه خارجی تماماً از لبه‌های محدب تشکیل شده باشد، زیرگراف مورد نظر مربوط به یک برآمدگی است و اگر مخلوطی از لبه‌های محدب و مقعر باشد، این زیرگراف مربوط به یک حفره باز می‌باشد. این موضوع در شکل (۱) نمایش داده شده است. همان گونه که در شکل (۱) ملاحظه می‌شود، حلقه خارجی تماماً از لبه‌های مقعر تشکیل شده است. در مورد برآمدگی این حلقه از لبه‌های محدب، در مورد حفره باز ترکیبی از لبه‌های محدب و مقعر می‌باشد.

به بیان ریاضی فرض شود f_{max} معرف وجهی از فیچر

باشد که درجه ماکزیمم دارد

و $Exloop[f_{max}] = \{e_1, e_2, \dots, e_q\}$ مجموعه لبه‌های موجود

در حلقه خارجی f_{max} و

$T[f_{max}] = \{\Psi(e_1), \Psi(e_2), \dots, \Psi(e_q)\}$ مجموعه حالت‌های

مربوط به این لبه‌ها باشد (در اینجا از علامت کروشه برای پرهیز از اندیس‌گذاری مضاعف استفاده شده است)؛ در این صورت:

تنها عمل مقایسه صورت می‌گیرد. بنابراین پیچیدگی زمانی کل رویه $O(n^2)$ خواهد بود.

پیش از توضیح چگونگی شناسایی فیچرهای متقاطع در این مقاله، توصیف خاصیت قابل توجه دیگری از زیرگراف‌های تجزیه شده با روش پیشنهادی این مقاله نیز مناسب به نظر می‌رسد. در کار حاضر، زیرگراف‌های مربوط به حفره‌ها خاصیتی را از خود به نمایش می‌گذارند که می‌تواند به تنهایی برای شناسایی آنها بکار برده شود. توجه شود که پس از تجزیه گراف به روشی که شرح آن گذشت، زیرگراف حفره، تنها زیرگرافی خواهد بود که در آن تعداد قوس‌های مقعر احاطه کننده وجه پایه با تعداد گره‌های جانبی برابر است. به عبارت دیگر، در حفره‌ها وجهی به نام وجه پایه وجود دارد که به تمام وجه مرتبط با خود، به وسیله لبه‌های مقعر متصل شده است. به طور مثال، در شکل (۴-ب)، وجه شماره ۱۲ پایه یک حفره است که به وسیله لبه‌های مقعر به وجوه جانبی ۸، ۹، ۱۰ و ۱۱ وصل شده است. ملاحظه می‌شود که وجه ۱۲، ۴ لبه مقعر دارد که با تعداد وجوه جانبی (تعداد دیگر گره‌های باقیمانده در گراف) برابر است. این مطلب را به بیان ریاضی می‌توان به صورت زیر شرح داد:

فرض شود در یک زیرگراف m تعداد لبه‌هایی باشد که مرز وجه دلخواه A از این زیرگراف را تشکیل می‌دهند. در این صورت لبه i ام از مرز این وجه را می‌توان به کمک رابطه زیر تعریف کرد:

$$E[i]_A = \{P_{Ai}(u), P_{Ai}(0), P_{Ai}(1)\}, i = 1, 2, \dots, m \quad (2)$$

که در آن $P_{Ai}(u) = \{X(u), Y(u), Z(u)\}$ در واقع نمایش پارامتری منحنی یک لبه سه بعدی در مختصات دکارتی است؛ به عبارت دیگر یک لبه در اینجا به کمک معادله منحنی آن و نقاط ابتدا و انتهای آن مشخص شده است.

اکنون فرض شود که دو وجه A و B به ترتیب m و n لبه در مرز خود داشته باشند؛ اگر این دو وجه جدا از هم نباشند (حداقل یک لبه مشترک داشته باشند)، آنگاه رابطه زیر برقرار است:

$$\prod_{i=1}^m \left(\prod_{j=1}^n (E[i]_A^2 - E[j]_B^2) \right) = 0 \quad (3)$$

که در آن $E[i]_A^2 = \{P_{Ai}^2(u), P_{Ai}^2(0), P_{Ai}^2(1)\}$ است. رابطه (۳) اصل اشتراک یک لبه بین دو وجه را بیان می‌کند. توجه شود که رابطه (۳) به صورت حاصل ضرب بیان شده است، بنابراین اگر دو وجه A و B تنها یک وجه مشترک داشته باشند، مثلاً $E[i]_A = E[j]_B$ ، آنگاه حاصل ضرب رابطه (۳) برابر با صفر خواهد شد. علت استفاده از مجذور $E[i]_A$ در

رابطه (۳) این است که منحنی‌ها به صورت پارامتری در نظر گرفته شده‌اند و این امر مشکل تفاوت علامت را مرتفع می‌سازد. از آنجا که در حفره‌ها، وجه پایه به همه وجوه دیگر متصل است؛ بنابراین رابطه (۳) بین تمامی وجوه جانبی و وجه پایه برقرار خواهد بود، پس شرط اینکه وجه A پایه یک حفره باشد، برقراری رابطه زیر است:

$$\sum_{k=1}^r \left[\prod_{i=1}^m \left(\prod_{j=1}^{n_k} (E[A]_i^2 - E[A]_j^2) \right) \right]^2 = 0 \quad (4)$$

که در آن r تعداد وجوه مجاور با A ، m تعداد لبه‌های A و n_k تعداد لبه‌های وجه k ام است. توجه شود که خواص توصیف شده در رابطه‌های (۳) و (۴) تنها برای شناسایی حفره‌ها قابل استفاده هستند و این روابط را نمی‌توان برای شناسایی دیگر انواع فیچرها بکار برد. در ادامه، چگونگی شناسایی فیچرهای متقاطع در کار حاضر توصیف خواهد شد.

۷- تقاطع فیچرها

در تقاطع فیچرها، حالت‌های بسیار متنوعی ممکن است اتفاق بیفتد؛ برای مثال، دو نوع کلی از تقاطعاتی که در نظر گرفته شده‌اند، از این قرار هستند:

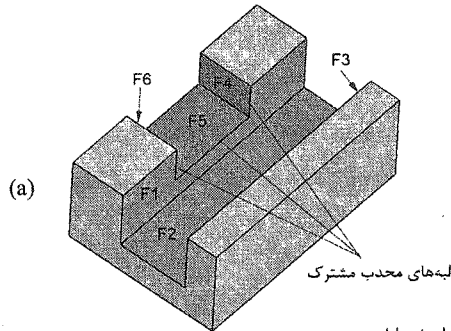
۱- تقاطعاتی که در آنها دو فیچر تنها در لبه‌هایی مشترک هستند (شکل ۵-ا).

۲- تقاطعاتی که در آنها فیچرها در یک وجه مشترک هستند (شکل ۵-ب).

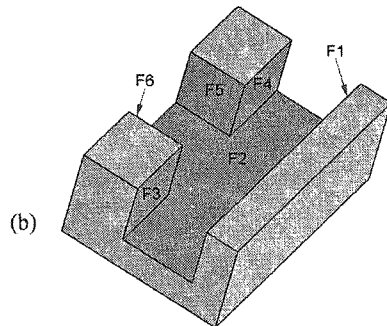
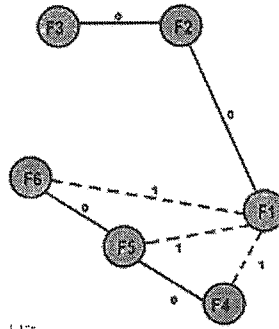
در حال حاضر، تقاطعات نوع اول، بدون نیاز به استفاده از heuristic‌های بیشتر به طور کامل بوسیله الگوریتم شناسایی می‌شوند. دلیل این امر این است که تقاطع فیچرها در این حالت همواره به ایجاد لبه‌های محدب در مرز تقاطع منجر می‌شود. از آنجا که لبه‌های محدب از گراف قطعه حذف می‌شوند، بنابراین زیرگراف‌های مربوط به دو فیچر متقاطع به طور اتوماتیک از یکدیگر جدا می‌شود و نوع آنها بر اساس روشی که در بخش ۷ توضیح داده شد، به سهولت قابل تشخیص است و به استفاده از الگوریتم‌های شباهت‌یابی بین گراف‌ها و یا تجزیه بیشتر زیرگراف‌ها برای جداسازی فیچرها نیازی نیست.

تمامی تقاطعات نوع دوم در کار حاضر در گروه «حفره‌های باز» طبقه‌بندی می‌شوند؛ زیرا آنها را می‌توان نه تنها به صورت ترکیبی از فیچرهای متقاطع، بلکه با حرکت پیوسته یک ابزار انگشتی نیز ایجاد کرد. در واقع، تعریف و دسته‌بندی فیچرها امروزه با توجه به استفاده روز افزون از ماشین‌های CNC در صنعت باید بازبینی شود. برای مثال، طبقه‌بندی فیچر نشان

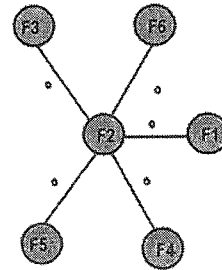
ساده‌تر مجدداً به عملیات پردازش بعدی نیاز خواهد داشت و کارهای بیشتری باید در این زمینه انجام شود. به این ترتیب، محدوده بسیار وسیعی از تقاطع‌ها بین فیچرها بدون نیاز به استفاده از دستورالعمل‌های متعدد، قابل شناسایی خواهند بود که در قسمت بعدی این موضوع با ارائه مثال‌هایی روشن‌تر خواهد شد.



تقاطع فیچرهای نوع اول



تقاطع فیچرهای نوع دوم



شکل (۵): (a) تقاطع فیچرها از نوع اول که در آن فیچرهای متقاطع در لبه‌های محذب مشترک هستند. (b) تقاطع از نوع دوم که در آن یک وجه بین دو فیچر مشترک است (این دسته از تقاطعات بهتر است حفره باز در نظر گرفته شوند).

بعدی توانایی‌های الگوریتم پیشنهادی را آشکار می‌سازند:

مثال ۱:

در الگوریتم‌های شناسایی استفاده شده در این مقاله، جهت‌گیری فیچرها در فضا و نحوه ایجاد آنها اهمیتی ندارد و در واقع، این الگوریتم‌ها مستقل از محورهای مختصات هستند. در شکل (۶) نتایج عملکرد نرم افزار بر روی قطعه‌ای نمایش داده شده است که در آن دو شیار با جهات دلخواهی روی قطعه قرار گرفته‌اند. از مدل این قطعه، دو شیار، یک پله و یک شیار بسته شناسایی شده است.

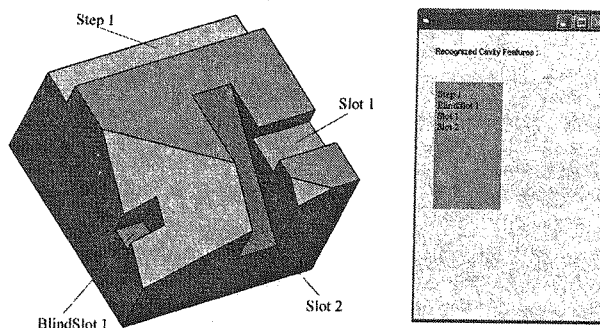
مثال ۲:

شکل (۷) نمایانگر عملکرد الگوریتم بر روی فیچرهای سه بعدی است. از این قطعه یک حفره و یک حفره باز متقاطع با هم شناسایی شده است. از آنجایی که الگوریتم‌های پیشنهادی عمدتاً بر اطلاعات توپولوژیکی متکی هستند؛ آنها را می‌توان

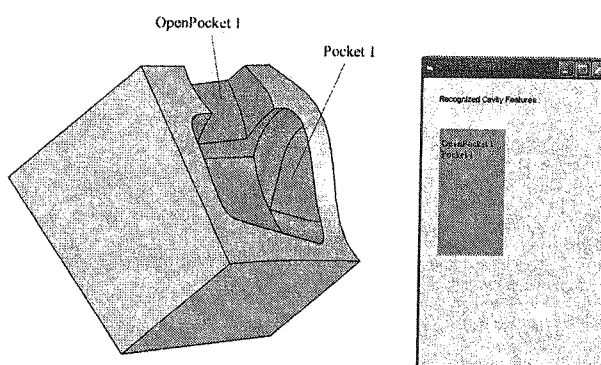
۸- مثال‌ها و نتایج

در این فصل، نتایج حاصل از عملکرد نرم افزار بر روی چند قطعه نمونه ارائه خواهند شد. الگوریتم طراحی شده در کار حاضر؛ که در قسمت‌های قبلی توضیح داده شد، به وسیله زبان برنامه نویسی Visual Basic نوشته شده است و از توابع OLE نرم‌افزار Solid Works برای دستیابی به اطلاعات هندسی وجوه و لبه‌های قطعه از مدل B-rep استفاده می‌کند. پس از استخراج اطلاعات هندسی، در مرحله بعدی حالت‌های وجوه و لبه‌ها بر اساس این اطلاعات محاسبه می‌شوند و این حالت‌ها با گراف قطعه؛ که به شکل ماتریسی ایجاد شده است، همراه می‌شوند. گراف مورد نظر سپس با روشی که شرح داده شد، شکسته شده و زیرگراف‌های آن استخراج می‌شوند. در مرحله بعدی نوع هر یک از این زیرگراف‌ها به کمک الگوهایی ارائه شده تعیین و به این ترتیب فیچرها شناسایی می‌شوند. مثال‌های

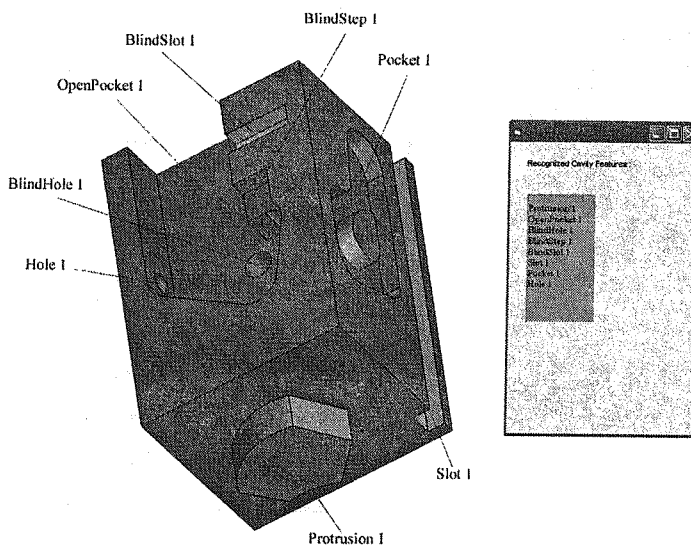
براحتی برای فیچرهای سه بعدی نیز بکار برد. در این مثال نیز حفره و حفره باز؛ که با هم متقاطع نیز هستند، تنها با استفاده از زیرگراف مقعر خود شناسایی شده‌اند، که کاملاً شبیه زیرگراف



شکل (۶): مثال ۱



شکل (۷): مثال ۲



شکل (۸): مثال ۳

برآمدگی و یک حفره با شکل منحنی شناسایی شده‌اند؛ که بیانگر توانایی الگوریتم‌های ارائه شده در شناسایی فیچرهای منحنی شکل است.

مثال ۳:

شکل (۸) نشان می‌دهد که این نرم افزار توانایی زیادی در شناسایی فیچرهای متقاطع دارد. در یک سمت قطعه نشان داده شده، یک حفره قرار دارد که دو سوراخ، یک شیار و یک شیار بسته با آن متقاطع هستند. در دیگر قسمت‌های آن یک شیار، یک

مقاله، آن دسته از فیچرهای سه بعدی که قابل طبقه‌بندی بر اساس الگوی فیچرهای ۲/۵ بعدی هستند، در نظر گرفته شده‌اند؛ نظیر حفره، حفره باز و برآمدگی. باید توجه شود همه انواع فیچرهای ۲/۵ بعدی دارای معادلی سه بعدی نیستند؛ مثلاً یک شیار دارای معادل سه بعدی نیست و شاید مناسب‌تر باشد که یک فیچر سه بعدی با سه سطح؛ که حداقل یکی از آنها فرم دلخواهی دارد، به صورت حفره یا حفره باز طبقه‌بندی شود تا یک شیار.

از دیگر نکاتی که در اینجا ذکر آن شایسته به نظر می‌رسد این است که از خروجی این نرم‌افزار می‌توان در در دیگر مراحل برنامه‌ریزی فرایند استفاده کرد؛ برای مثال، در ادامه فعالیت‌های کنونی، به تعیین ترتیب (setup) و مراحل (sequence) ماشین‌کاری قطعه نیز می‌توان توجه کرد. استخراج کدهای CNC از فیچرهای شناسایی شده برای ماشین‌کاری آنها و انتخاب ابزار و پارامترهای ماشین‌کاری برای این فیچرها نیز از دیگر فعالیت‌هایی هستند که در راستای این مقاله قابل انجام هستند.

۱- مراجع

- [۱] Han, J.H.; Pratt, M.; Regli, W.C.; "Manufacturing feature recognition from solid models: A status report", IEEE Trans. On Robotics and Automation, vol. 16 p.p. 782-796, 2000.
- [۲] Regli, W.C.; "Geometric algorithms for recognition of features from solid models", PhD dissertation, Univ. Maryland, College Park MD, 1995.
- [۳] Henderson, M.R.; "Extraction of feature information from three dimensional CAD data", PhD Thesis, Purdue University, west Lafayette, IN, USA, 1984.
- [۴] Jakubowski, R.; "Syntactic Characterization of machine parts shapes", Cybernetics and Systems: An International Journal, vol. 13 p.p. 1-24 1982.
- [۵] Joshi, S.; Chang, T.C.; "Graph-based heuristics for recognition of machined features from 3D solid model", Computer-Aided Design, Vol. 20 p.p. 58-66, 1988.
- [۶] Ji, Q.; Marefat, M.; "Bayesian approach for extracting and identifying features", CAD, vol. 27, p.p. 435-454, 1995.
- [۷] Ji, Q.; Marefat, M.; "A Dempster-Shafer approach for recognizing machine features from CAD models", PATTERN RECOGNITION, PERGAMON, vol. 36, p.p. 1355-1368 2003.
- [۸] Marefat, M.; Kashyap, R.L.; "Geometric reasoning for recognition of the three dimensional object features", IEEE Trans. On Pattern Analysis and Mach. Intelligence, vol. 12 p.p. 949-965, 1990.
- [۹] Xu, X.; Hinduja, S.; "Recognition of rough machining features for 2.5D components", CAD, vol. 30, p.p. 503-516, 1998.
- [۱۰] Kim, Y.; "Convex decomposition and solid geometric modeling", PhD dissertation, Stanford Univ, 1990.
- [۱۱] Kim, Y.; Wang, E.; "Recognition of machining features for cast then machined parts", CAD, vol. 34, p.p. 71-87, 2002.
- [۱۲] Wang, E.; Kim, Y.; "1997 status of the form feature recognition method using convex decomposition", in

در این مقاله نرم‌افزاری که برای شناسایی فیچرها از نمایش مرزی مدل توپر تهیه شده است، توضیح داده شد. در نرم افزار ارائه شده تلفیقی از ایده‌های موجود در الگوریتم‌های متکی بر گراف و متکی بر ردیابی برای شناسایی فیچرها استفاده شده است.

موارد زیر را می‌توان به عنوان دستاوردهای این مقاله در نظر گرفت:

۱- توسعه روش تحلیل الگوی گراف در شناسایی فیچرها. مبنای شناسایی بیشتر فیچرها در این مقاله، شکل توسعه یافته‌تری از گراف نحوه همسایگی است. با این توضیح که در کار حاضر، محدوده فیچرهای قابل شناسایی بوسیله روش‌های متکی بر گراف گسترده‌تر شده است و برآمدگی‌ها و حفره‌های باز نیز به گروه‌های قابل شناسایی اضافه شده و شکل فیچرها نیز از چند وجهی به منحنی و حتی سه بعدی توسعه یافته است. در مقایسه با روش‌های متکی بر گراف، الگوریتم‌های ارائه شده در کار حاضر، قادر به شناسایی محدوده وسیع‌تری از فیچرهای متقاطع بوده و برای این کار به heuristic ها و قوانین اضافی دیگری نیاز ندارند.

در کار حاضر، گراف نحوه همسایگی به صورتی تجزیه می‌شود که مؤلفه‌های حاصل تنها بیانگر یک فیچر باشند (نه ترکیبی از فیچرها)؛ از این رو، این روش به روش‌های متکی بر ردیابی شباهت دارد.

۲- پیچیدگی زمانی.

الگوریتم استفاده شده در شناسایی فیچرها پیچیدگی زمانی نسبتاً مناسبی دارد. بدترین پیچیدگی زمانی کل الگوریتم شناسایی همان طور که اشاره شد، $O(n^2)$ است. برای مقایسه توجه شود که پیچیدگی زمانی روش‌های خام تحلیل الگوی گراف عموماً از درجه نامایی است.

باید توجه شود که اگرچه روش پیشنهادی در شناسایی محدوده وسیعی از فیچرهای متقاطع موفق عمل می‌کند، اما هنوز تا ارائه یک روش کامل؛ که قابل استفاده در صنعت باشد، راه درازی باقی مانده است. تقاطع فیچرها در مواردی ممکن است به حذف اطلاعات توپولوژیکی گراف قطعه منجر شود که برای شناسایی فیچرها در چنین حالتی باید کارهای بیشتری صورت گیرد. همچنین برای توسعه بیشتر الگوریتم‌های پیشنهادی در محدوده فیچرهای سه بعدی نیز باید فعالیت‌های بیشتری صورت گیرد. برای این منظور ابتدا باید یک طبقه‌بندی دقیق از فیچرهای سه بعدی ارائه و انواع مختلف آن مشخص شود، که این خود به تنهایی کار بسیار مشکلی است. در این

- 39 Solid
- 40 Boundary Representation
- 41 Face
- 42 Closed and Orientable
- 43 Loop
- 44 Vertex
- 45 Edge
- 46 Set Extension
- 47 Attributed Adjacency Graph
- 48 Ordered Triple
- 49 Attribute
- 50 Convex/Concave/Neutral Faces
- 51 Convex/Concave Edges
- 52 Smooth Convex
- 53 Smooth Concave
- 54 Smooth Neutral
- 55 Template
- 56 Isolated Node
- 57 Node Fusion Operation
- 58 External Loop
- 59 Setup
- 60 Procedure

- Proc. ASME Comp. in Eng. Conf., 1997.
- Vandenbrande, J.H.; Requicha, A.A.G.; "Spatial reasoning for the automatic recognition of machinable features in solid models", IEEE Trans. On Pattern Analysis and Mach. Intelligence, vol. 15 p.p. 1269-1285 1993. [۱۳]
- Woo, Y.; Sakurai, S.; "Recognition of maximal features by volume decomposition", CAD, vol. 34, p.p. 195-207, 2002. [۱۴]
- Sandararajan, V.; Wright, P.; "Volumetric feature recognition for machining components with free form surfaces", CAD, vol. 36 p. 1125 2004. [۱۵]
- Gao, S.; Shah, J.J.; "Automatic recognition of interacting machining features based on minimal condition sub-graph", CAD, vol. 30 p.p. 727-739, 1998. [۱۶]
- Zeid, I.; CAD/CAM Theory and Practice, McGraw-Hill, Inc., 1991. [۱۷]
- Deo, N.; Graph Theory with Applications to Engineering and Computer Science, Prentice-Hall, Inc., Englewood Cliffs, NJ., 1974. [۱۸]

۱۱- زیر نویس ها

- 1 Computer-Aided Process Planning
- 2 Computer-Aided Design
- 3 Computer-Aided Manufacturing
- 4 Feature Recognition
- 5 Feature Intersections
- 6 Syntactic Pattern Recognition
- 7 Expert Systems
- 8 Boundary Representation
- 9 Node
- 10 Arc
- 11 Sub-graph
- 12 Sub-graph Isomorphism
- 13 Hypothesis Testing
- 14 Polyhedral
- 15 Convex Decomposition
- 16 Regularized Set Difference
- 17 Convex Hull
- 18 Cell Decomposition
- 19 Delta Volume
- 20 Exponential Time Complexity
- 21 Scalability
- 22 Free-form
- 23 Hint-based Reasoning
- 24 Slot
- 25 Step
- 26 Blind Slot
- 27 Blind Step
- 28 Blind Hole
- 29 Through hole
- 30 Pocket
- 31 Open Pocket
- 32 Protrusion
- 33 Analytical
- 34 Synthetic
- 35 Ruled Surface
- 36 Surface of Revolution
- 37 Tabulated Cylinder
- 38 Toroidal