



دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر

**پایان نامه کارشناسی**

**جستجوی محتوایی  
بر اساس ویژگی های شکل در تصاویر**

نگارنده:

**مرتضی عالمگیر**

استاد راهنما:

**دکتر حمیدرضا ربیعی**

به نام خدا

دانشگاه صنعتی شریف  
دانشکده مهندسی کامپیوتر

پایان نامه کارشناسی

عنوان: جستجوی محتوایی بر اساس ویژگی های شکل در تصاویر

نگارش: مرتضی عالمگیر

امضاء:

استاد راهنما: دکتر حمید رضا ربیعی

تقدیم به پدر و مادرم

## تشکر و قدردانی

در اینجا لازم می‌دانم از جناب آقای دکتر ربیعی به خاطر سرپرستی پروژه و رهنمودهایی که در جهت پیشبرد پروژه به نگارنده نموده‌اند تشکر کنم.

همچنین مراتب سپاس و قدردانی خود را از جناب آقای دکتر قنبری به خاطر تمام زحماتی که در طول انجام پروژه متقبل شدند اعلام می‌کنم.

در نهایت از تمامی اعضای آزمایشگاه DML به خصوص خانم شیرین قنبری و آقایان شهریار شریعت، علی جلالی و جعفر محمدی به خاطر راهنمایی‌هایشان تشکر می‌کنم.

## چکیده

به علت گسترش تولید محتوای چند رسانه ای در شبکه توزیع شده اینترنت در سالهای اخیر، جستجوی متنی در بسیاری از کاربردها کارآیی لازم را ندارد. از طرف دیگر جستجوی پایگاه داده های تصویری بر اساس کلید واژه نیز نیاز موجود را رفع نمی نماید. با توجه به اینکه برای انتخاب کلمات کلیدی برای هر شکل استاندارد واحدی وجود ندارد، لازم است که جستجوی پایگاه های داده تصویری بر اساس ویژگیهای آماری تصاویر مانند بافت، رنگ و شکل صورت پذیرد.

البته راهکارهایی بر اساس روابط معنایی (*semantic*) نیز در سالهای اخیر مطرح شده اند. در این پروژه پس از مرور روشهای موجود، راهکار جدیدی برای جستجوی پایگاه داده های تصویر بر اساس شکل اشیا ارائه میشود. در این روش برای استخراج ویژگیهای اشیا در تصاویر از الگوریتم جدیدی جهت توصیف اشیا بر اساس نقاط مرزی آنها استفاده خواهد شد. در پایان با شبیه سازی صحت و توانایی این روش مورد بررسی قرار خواهد گرفت.

**کلمات کلیدی:** جستجوی پایگاه های داده تصویری، توصیف شکل، روش CSS، توصیفگرهای ناوردا

1. جستجوی چندرسانه‌ای - تصویر ..... 7
- 1-1- مشخصه‌های پرس‌وجوهای تصویر ..... 9
- 2-2- روش‌های جستجو و بازیابی تصاویر ساختارمند ..... 14
- 3-1- روش‌های جستجو و بازیابی تصاویر عمومی و بدون ساختار ..... 15
- بازیابی بر اساس ویژگی رنگ ..... 16
- بازیابی بر اساس ویژگی طرح‌بندی رنگ ..... 16
- بازیابی بر اساس ویژگی بافت ..... 17
- بازیابی بر اساس ویژگی شکل ..... 18
- تکنیک‌های موجود برای بازیابی بر اساس محتوی در سطح ۲ ..... 19
- تکنیک‌های موجود برای بازیابی بر اساس محتوی در سطح ۳ ..... 21
2. بررسی روشهای بازیابی بر اساس ویژگی شکل ..... 22
- نمایشگرهای ناوردا: ..... 22
- 1-1- ناورداهای گشتاور: ..... 22
- 2-2- استخوان بندی شکل ..... 24
- 3-2- نگاشت لگاریتمی-قطبی ..... 25
- 4-2- توصیف گرهای هندسی شکل ..... 26
- خصوصیات ..... 27
- نمودار  $(r, \theta)$  ..... 27
- نمودار  $(s, \theta)$ ، روش توصیف بر اساس انحنا ..... 28
- مرحله پیش پردازش ..... 30
- 5-2- بررسی پوش محدب و مفاهیم مرتبط ..... 33
- 6-2- پیشنهاد یک روش تقریب زنی خطی بر اساس تابع انحنا ..... 35
3. تعمیم و بهبود تابع CURVATURE ..... 37
- 1-3- پیشنهاد معیاری جدید برای توصیف خمها ..... 40
- 3-2- یک *DISTANCE FUNCTION* با استفاده از  $l_n(t)$  ..... 41
- 3-3- بررسی خواص مشتق تعمیم یافته: ..... 44
- 4-3- بررسی حساسیت روش پیشنهادی در مقابل خطا: ..... 48
4. پیش نهاد راه کارهای آتی: ..... 50
5. جمع بندی: ..... 51
6. منابع و مراجع: ..... 51
- ضمیمه 1: کد منبع پیاده‌سازی الگوریتم ..... 53

## مقدمه

در سال‌های اخیر حجم مجموعه صوت‌ها، تصاویر و ویدیوهای دیجیتال رشد بی‌سابقه‌ای داشته است. این حجم عظیم اطلاعات بدون روش‌های مرور، جستجو و بازیابی موثر، نمی‌توانند مفید واقع شوند. بر گسترش کاربردهای این داده‌های چندرسانه‌ای نیز روز به روز افزوده می‌شود. به عنوان مثال تولید و بازیابی و استفاده از تصاویر، امروزه در فیلم‌های بسیاری نظیر گالری‌های هنری، مدیریت موزه‌ها، طراحی‌های مهندسی و معماری، طراحی داخلی، پردازش‌های از راه دور و مدیریت منابع زمین، سیستم‌های اطلاعات جغرافیایی، روزنامه‌نگاری و تبلیغات، تشخیص پزشکی، مدیریت پایگاه داده‌های علمی، پیش‌بینی آب و هوا، طراحی مد و پارچه، مدیریت پایگاه داده‌های حقوق انحصاری و علائم تجاری، کشف مجرمان و قانون‌شکنان، سیستم‌های آرشیو تصاویر، سیستم‌های ارتباطی، آموزش و سرگرمی و ... بصورت وسیعی مطرح است.

کاربران معمولی ممکن است در مواجهه با یک پایگاه داده چندرسانه‌ای عظیم نیازهای چندانی نداشته باشند و یا به مرور کردن داده‌های پایگاه داده با فراغت بال کفایت کنند ولی کاربران حرفه‌ای معمولاً به تکه‌ای خاص از داده‌های چندرسانه‌ای نیازمندند که باید در زمانی کوتاه در میان خیل عظیمی از اطلاعات به آن دسترسی پیدا کنند. امروزه این نیازهای حرفه‌ای مرتب و دایماً در حال افزایش بوده و نیاز به روش‌های بازیابی اطلاعات که بتواند به این کاربران حرفه‌ای کمک کند، شدیداً احساس می‌شود. پس ما نیازمند روش‌هایی هستیم که بتواند به جستجو و بازیابی اطلاعات در میان حجم عظیمی از داده‌های چندرسانه‌ای بپردازد.

برای جستجو و بازیابی این اطلاعات چندرسانه‌ای، دو روش عمده وجود دارد: روش‌های بر پایه متن و روش‌های بر پایه محتوی (روش‌های بر پایه مشاهده و شنود). در روش‌های بر پایه متن، ابتدا اطلاعاتی در

مورد داده چندرسانه‌ای توسط افراد مطلع در قالب داده‌های متنی ذخیره می‌شود که معرف کامل داده چندرسانه‌ای می‌باشد. سپس جستجو بر روی این داده‌های متنی صورت گرفته و اطلاعات چندرسانه‌ای متناظر داده‌های متنی، به کاربران نشان داده می‌شود. از فعالیت‌های انجام گرفته در این دسته می‌توان به [CF 79، CYDA 88] اشاره نمود. دو مقاله [SCAH 92، TY 84] به بررسی کامل فعالیت‌های انجام گرفته در این دسته می‌پردازند. مشکل اصلی این دسته، این است که با رشد حجم داده‌ها، حجم کار دستی انجام گیرنده بر روی داده‌ها نیز، خارج از حد قابل پذیرش می‌شود. از طرفی برداشت افراد مختلف از حتی یک تصویر مشخص نیز حتی متفاوت است و این باعث می‌شود که اطلاعات متنی تولیدی برای پایگاه داده‌های چندرسانه‌ای یکدست نباشد. همچنین چون حجم داده‌های چندرسانه‌ای شدیداً و سریعاً در حال افزایش است و دسته‌بندی‌های انسانی و اطلاعات متنی‌ای که برای این داده‌ها تولید می‌شود، محدود می‌باشد، به ازای هر جستجویی، حجم خیلی زیادی داده برگشت داده می‌شود که عملاً کارایی این روش‌ها را زیر سوال می‌برد. این مشکل در حالی در این روش‌ها خودنمایی می‌کند که تصاویر از لحاظ اطلاعاتی بسیار غنی بوده و قابلیت آن را دارند که به دسته‌های بسیار دقیق و جزئی طبقه‌بندی شوند. مجموعه مشکلات بیان شده، راه را برای مطرح شدن و رشد دسته دوم روش‌های جستجو هموار کرد. در این روش‌ها که روش‌های بازیابی مبتنی بر محتوی<sup>1</sup> نامیده می‌شوند، داده‌های چندرسانه‌ای بر اساس ویژگی‌هایی که بصورت خودکار از آنها استخراج می‌گردد، اندیس‌گذاری شده و مورد جستجو و بازیابی قرار می‌گیرند.

شرکت‌های چندرسانه‌ای، سرویس‌های خبری، شبکه‌های رادیویی و تلویزیونی هر ساله میلیون‌ها دلار جهت اندیس‌گذاری دستی داده‌های چندرسانه‌ای‌شان هزینه می‌کنند. در نتیجه تلاش‌های بسیاری در گروه‌های تحقیقاتی متعددی در سراسر جهان در حال انجام است که پروسه اندیس‌گذاری فایل‌های چندرسانه‌ای به یک پروسه خودکار یا نیمه خودکار تبدیل گردد. بازیابی مبتنی بر محتوی در داده‌های چندرسانه‌ای هنوز مبحث بازی بوده و محصولات چندانی در این زمینه موجود نمی‌باشند.

موتورهای جستجوی چندرسانه‌ای سعی می‌کنند در میان خیل عظیمی از اطلاعات بدنبال پاسخی مناسب برای پرس‌وجوهای کاربرانشان بگردند. بر اساس این هدف براحتی می‌توان چالش‌ها و مشکلات موجود بر سر راه این موتورهای جستجوگر را در چند دسته اصلی دسته‌بندی کرد: نحوه ذخیره داده‌ها، مشکل تعدد فرمت‌ها، نحوه ورود پرس‌وجو، استراتژی‌های جستجو، نحوه ارائه نتایج و نحوه برخورد با چالش انفجار اطلاعات و حجم بالای داده‌های موجود مهمترین دغدغه‌های موتورهای جستجو می‌باشند.

## 1. جستجوی چندرسانه‌ای - تصویر

یک تصویر در حالت کلی می‌تواند یک عکس، مثال تصویری، *Clipart*، نقاشی، متن اسکن شده و یا هر چیزی دیگری باشد که با بینایی انسان در ارتباط است.

مشخصه‌های پرس‌وجوها و نحوه ارائه آنها در موتورهای جستجوی تصویری دارای تفاوت اساسی با پرس‌وجوهای در موتورهای جستجوی متنی می‌باشد. در یک موتور جستجوی متنی، جستجو بر اساس کلمات کلیدی می‌باشد، ولی در جستجوی تصویری، جستجو بر اساس ویژگی‌های تصویر می‌باشد که بیان آنها با کلمات، مشکل می‌باشد. در نتیجه یکی از مسایل مهم در جستجوی تصویر، بررسی انواع پرس‌وجوها و نحوه ارائه آنها می‌باشد.

ویژگی‌هایی از تصاویر که می‌بایستی در هنگام جستجو و بازیابی مورد توجه قرار گیرند:

- ویژگی‌های توصیف‌گر محتوی: صحنه تصویر (نظیر کوه، ماشین، صورت و ...) را توصیف می‌کنند.
- ویژگی‌های وابسته به محتوی: ویژگی‌های تصویر (نظیر اطلاعات رنگی، بافتی و شکلی و ...) را توصیف می‌کنند.
- ویژگی‌های مستقل از محتوی: ویژگی‌های داده‌ای (نظیر فرمت فایل، سایز و ...) را توصیف می‌کنند.

بهترین ویژگی‌هایی که برای بازیابی در موتورهای جستجو می‌توانند مورد استفاده قرار گیرد، ویژگی‌های توصیف‌گر محتوی هستند. ولی متأسفانه تولید توصیف‌گرهای محتوی به صورت خودکار، بسیار مشکل بوده و در حال حاضر پیشرفت‌های بسیار اندکی را در این زمینه شاهد هستیم. به همین دلیل موتور

---

1 illustration

2 Content-descriptive features

3 Content-dependent features

4 Content-independent features

جستجویی که از این ویژگی‌ها استفاده کند، موجود نمی‌باشد. مهمترین ویژگی‌هایی که در موتورهای جستجوی امروزی برای بازیابی تصاویر مورد استفاده قرار می‌گیرند ویژگی‌های وابسته به محتوی هستند. گر چه بکارگیری این ویژگی‌ها برای بازیابی اطلاعات در تصاویر و برای اندیس‌گذاری آنها مناسب هستند و می‌توان همبستگی معنایی بین این ویژگی‌ها و محتویات تصویر برقرار کرد، ولی بهتر است ابتدا تصاویر را به دسته‌های گوناگونی بر اساس محتوای آنها تقسیم نموده و روش‌های جستجو و بازیابی منطقی‌تری را بر پایه این دسته‌بندی‌ها تعریف نمود. یک دسته بندی مناسب برای تصاویر می‌تواند بصورت زیر تعریف شود [TALMUD]:

- تصاویر ساختارمند

- متن‌های اسکن شده

- متن‌های تایپ ماشینی اسکن شده

- متن‌های دست‌نویس اسکن شده

- تصاویر صورت انسان

- بازیابی بر اساس صورت فرد خاص (تشخیص صورت)

- بازیابی بر اساس احساسات صورت افراد

- دیاگرام‌ها

- و ...

- تصاویر غیر ساختاری

همانطور که انتظار داریم، روش‌های جستجو و بازیابی تصاویر ساختارمند با تصاویر بدون ساختار با هم کاملاً متفاوت می‌باشند. در بازیابی تصاویر ساختارمند، از ویژگی‌های آن دسته خاص از تصاویر استفاده شده و روش‌های مورد استفاده در آن کاملاً منحصر به همان دسته می‌باشد. ولی در بازیابی تصاویر بدون

ساختار، سعی می‌شود تا ویژگی‌های کلی و عمومی‌ای از تصاویر استخراج گردیده و بازیابی بر اساس آنها انجام گیرد.

در ادامه این بخش، ابتدا مشخصه انواع پرس‌وجوهای کاربران و نحوه ارائه آنها را مورد بررسی قرار می‌دهیم. سپس به توضیحاتی فنی کاملاً مختصری در مورد روش‌های بازیابی تصاویر ساختارمند اشاره کرده و در نهایت به بررسی انواع ویژگی‌های مورد استفاده در بازیابی تصاویر بدون ساختار می‌پردازیم و نحوه استفاده از آنها را توضیح می‌دهیم.

## 1-1- مشخصه‌های پرس‌وجوهای تصویر

برای آگاهی از اینکه کاربران چه نوع پرس‌وجوهایی را دوست دارند در اختیار داشته باشند، باید از نیازهای کاربران آگاه بود و اینکه برای رفع چه نیازهایی عمل جستجو را انجام می‌دهند؟ چه استفاده‌ای از نتایج بدست آمده بعمل می‌آورند؟ قضاوت آنها در مورد نتایجی که به آنها برگشت داده می‌شود چیست؟ و ... . هنوز تحقیق جامعی در این زمینه وجود ندارد اما می‌توان بطور شهودی به نتایجی دست یافت. یک دستور میتواند به یکی از دلایل زیر مورد نیاز باشد:

- برای توصیف حالات و جملاتی که براحتی قابل بیان نیستند و یا اینکه درک واضحی به شنونده یا خواننده نمی‌دهند.

- نمایش جزئیات یک شی برای تجزیه و تحلیل آن شی (نظیر تصاویر رادیولوژی)

- برای کاربردهایی نظیر طراحی و ...

بسته به هر کدام از کاربردهای فوق، تصاویری با ویژگی‌های خاصی مد نظر است. تصاویر می‌توانند از لحاظ ویژگی‌هایی که دارا هستند به گروه‌های زیر تقسیم شوند:

- تصاویر دارای دادگان ' مشخص (مثلاً چه کسی تصویر را ساخته است؟ کجا و کی؟)

- تصاویر دارای ترکیب معینی از رنگ، بافت و شکل (مثلا: ستاره‌های سبز).
- تصاویر حاوی ترتیبی از اشیای خاص مورد نظر (مثلا صندلی‌هایی دور یک میز).
- تصاویر معرف یک رویداد خاص (مثلا مسابقه فوتبال).
- تصاویر حاوی رویداد، مکان یا فرد خاص (مثلا تصویر تخت جمشید).
- تصاویر حاوی هیجانات درونی فردی (نظیر شادی)

هر کدام از موارد فوق (به جز مورد اول که به محتوای تصویر نامربوط می‌باشد) تجرید سطح بالاتری از موارد قبلی ترش را در خود دارد و بازیابی بر اساس آن بدون داشتن دانش خارجی سخت‌تر است. بر این اساس می‌توان پرس‌وجوها را از لحاظ پیچیدگی به چند دسته تقسیم نمود [JPE 96, JPE 98]:

**پرس‌وجوهای سطح ۱:** شامل بازیابی توسط ویژگی‌های ابتدایی نظیر رنگ، بافت، شکل و موقعیت مکانی عناصر تصویر. نمونه‌های از این پرس‌وجوها می‌تواند به این شکل بیان شود: «تصاویری را با اشیای نازک تیره در گوشه سمت چپ و بالای تصویر پیدا کن»، «تصاویر حاوی ستاره‌های زرد به شکل حلقه چیده شده را پیدا کن» و یا در حالتی عمومی‌تر «تصاویری را که بیشترین شباهت با این تصویر دارند را پیدا کن». این نوع جستجوها از ویژگی‌هایی از تصویر استفاده می‌کنند که بدون نیاز به اتصال به هیچ منبع دانش خارجی‌ای می‌توانند پاسخ پرس‌وجو را بیابند.

**پرس‌وجوهای سطح ۲:** بازیابی را بر اساس ویژگی‌های اشتقاقی‌ای انجام می‌دهند که حاوی اطلاعات استنتاجی منطقی‌ای از اشیای موجود در تصویر می‌باشند. این پرس‌وجوها خود می‌توانند به دو دسته زیر تقسیم شوند:

- بازیابی اشیایی از نوع داده شده (مثال: تصاویری را پیدا کن که حاوی یک اتوبوس دو طبقه باشند).
- بازیابی بر اساس اشیای مشخص (مثال: تصاویر برج ایفل را پیدا کن).

برای پاسخگویی به این پرس و جوها (و بخصوص پرس و جوهای در مورد اشیای مشخص) نیاز به ذخیره سازی مقداری دانش خارجی وجود داریم. پاسخگویی به پرس و جوهای این سطح به مراتب مشکل تر از پاسخگویی به پرس و جوهای سطح ۱ می باشد، در حالیکه نیاز اکثر کسانی که به جستجوی تصاویر می پردازند از این دسته می باشد.

**پرس و جوهای سطح ۳:** شامل بازیابی بر اساس ویژگی های انتزاعی ای می باشد که نیازمند استدلال های سطح بالا از تصویر و درک معنی و مفهوم آن تصویر می باشد. پرس و جوهای این دسته نیز می توانند به دو زیر دسته تقسیم گردند:

- بازیابی رویدادهای مشخص یا نوع خاصی از فعالیت (مثال: تصاویری را پیدا کن که در آن شخصی در حال دویدن باشد).

- بازیابی تصاویر حاوی هیجانانگیز یا حالات روحی خاص (مثال: تصاویری را بیاب که در آن کسی درد را تحمل می کند).

پاسخگویی به پرس و جوهای این سطح، بسیار پیچیده تر از یک جستجو می باشند. استدلال پیچیده و قضاوت ذهنی لازمه ارتباط دادن مفاهیم با محتویات تصویر می باشد. این پرس و جوها موارد استفاده کمتری از پرس و جوهای سطح ۲ دارند ولی به مراتب از آنها مشکل تر می باشند.

حال که انواع پرس و جوهای تصویر را بررسی کردیم، به بررسی نحوه ارائه این پرس و جوها می پردازیم. مطالبی که در این بخش به آن می پردازیم در حوزه جستجوهای سطح ۱ می باشد که در حال حاضر، نمونه های از موتورهای جستجو در این سطح موجود می باشد.

اولین سیستم های ارائه شده، استفاده از پرس و جوهای شبیه *SQL* را پشتیبانی می کردند (مثلا سیستم *GRIM DBMS*). از آنجایی که بیان داده های شهودی توسط کلمات مشکل به نظر می رسید روش

پرس و جو با مثال<sup>۱</sup> مطرح شد که در آن کاربر با ارائه تصویری شبیه آنچه که مد نظرش می باشد، از سیستم تقاضای نتایج می کند. امروزه تقریباً تمامی سیستم موجود از این شیوه پشتیبانی می کنند. تصویر نمونه دریافت شده، جستجو انجام می پذیرد و سپس در یک صفحه تعدادی از نزدیک ترین پاسخ های یافته شده به ترتیب شباهت برای کاربر نمایش داده می شوند.

می توان دسته بندی زیر را بر اساس برای انواع پرس و جوهای کاربر ارائه داد:

- جستجو با مثال<sup>۲</sup>: جستجو با مثال یا قالب کلاسیک ترین روش در جستجو و بازیابی اطلاعات تصویری می باشد. در این روش در یک واسط گرافیکی تعاملی، کاربر تصویر مورد نظر خود را انتخاب کرده ، نواحی ای از تصویر را که به آنها علاقمند است مشخص می نماید و سپس معیارهای خود برای جستجو (مثلاً همبستگی روشنایی<sup>۳</sup> یا شباهت ویژگی<sup>۴</sup>) را انتخاب می نماید و پرس و جو خود را ارسال می کند.
- جستجو با ویژگی ها<sup>۵</sup>: از آنجایی که کاربر همیشه تصویری شبیه آنچه را که لازم دارد در اختیار ندارد، روشهای دیگری برای این منظور پیشنهاد شد که همه آنها بر پایه ایده مطرح شده در سیستم *IBM QBIC* می باشند. در این سیستمها، کاربر تصاویری را مطابق یک سری از ویژگی های آنها (نظیر رنگ، بافت، شکل، جابجایی و یا ساختار زمانی-مکانی<sup>۶</sup> نواحی آن) که مد نظرش می باشد، تقاضا می کند برای بازیابی بر اساس رنگ، به کاربر اجازه انتخاب رنگ از یک جعبه رنگ<sup>۷</sup> داده می شود. جستجوهای بافت می تواند بر اساس انتخاب نمونه بافت مورد نظر از یک مجموعه بافت مختلف انجام پذیرد.

---

Query by Example 1

2 Search by example

3 Intensity correlation

4 Feature similarity

5 Search by features

6 Spatio-Temporal structures

Palette 7

- جستجو با طرح‌ها: حالت خاصی از جستجو با ویژگی‌ها می‌باشد که ویژگی مورد استفاده در آن، شکل می‌باشد. در این سیستم‌ها ابزارهای گرافیکی پیشرفته‌ای در اختیار کاربران قرار می‌گیرد تا طرح بصری کلی چیزی را که در ذهن دارند و دنبال آن می‌گردند نشان دهند. این طرح‌های کلی به عنوان پرس‌وجو به سیستم فرستاده می‌شود تا نتایج متناظر با آن برگشت داده شوند در این سیستم‌ها کاربر می‌تواند طرح کلی تصویر مورد نظر خود را بر روی صفحه‌ای ترسیم کند یا در صورتی که هنر ترسیم خوبی نداشته باشد، می‌تواند طرحی از شکل مورد نظر خود را با مستطیل‌ها و دایره‌هایی که سیستم در اختیار او قرار می‌دهد، برای سیستم نشان دهد.

- جستجو با نواحی<sup>۲</sup>: این حالت، تلفیقی از جستجو با ویژگی‌ها و جستجو با مثال می‌باشد که در آن بر اساس ویژگی‌های اولیه تصویر ورودی، تصویر، قطعه‌بندی شده و به نواحی تشکیل دهنده آن تقسیم می‌شود. سپس این ناحیه‌ها به کاربر نشان داده شده و کاربر ناحیه مورد نظر خود را انتخاب می‌نماید تا جستجو بر اساس آن انجام پذیرد. تصاویری از پایگاه داده که حاوی نواحی‌ای شبیه به ناحیه مورد جستجو داشته باشند، به عنوان نتایج برگشت داده می‌شوند. به عنوان نمونه‌ای از این سیستم‌ها می‌توان به *BlobWorld* اشاره نمود که در آن ناحیه‌بندی بر اساس ویژگی‌های اولیه رنگ و بافت انجام می‌پذیرد.

- جستجو همرا با پالایش تدریجی: تکنیک مناسب دیگری که می‌تواند مطرح شود، بهره‌گیری از بازخورد ارتباط<sup>۳</sup> می‌باشد که بر پایه ایده پالایش تدریجی نتایج استوار است و اولین بار برای جستجوی متن ارائه گردید. این تکنیک در بازیابی تصاویر نخست در سیستم‌های بازیابی بر اساس کلمات کلیدی بکار گرفته شد و امروزه بصورت موفق‌تری در بسیاری از سیستم‌های بازیابی بر اساس محتوا نیز بکار گرفته شده است. در این سیستم‌ها مجموعه‌ای از تصاویر تولید شده

---

1 Search by sketches

2 Search by Region

Relevance Feedback 3

تصادفی به کاربر نشان داده می‌شود و کاربر از بین آنها تصویری را که بیشترین شباهت را با تصویر مورد نظرش دارد انتخاب می‌کند. در این صورت مجموعه تصاویری منطبق بر نظر کاربر برای وی برگشت داده خواهد شد که کاربر از میان این تصاویر برگشتی تصویری را که به تصویر مد نظرش نزدیک‌تر است انتخاب می‌کند و دوباره عمل بازیابی انجام می‌گیرد. این حلقه به همین صورت ادامه پیدا می‌کند تا در نهایت کاربر تصویر مورد نظر خود را پیدا نماید.

دو نکته در انتهای بحث باید ذکر شوند، یکی اینکه، در هر کدام از سیستم‌های فوق، می‌توان به کاربران اجازه داد تا برای هر کدام از اجزای پرس‌وجویشان وزن‌هایی را انتخاب نمایند. و دیگر اینکه هنوز جا برای تحقیق در مورد واسط‌هایی که بتوانند معیارهای سنجش شباهت مورد نظر کاربران را دریافت کرده و بر اساس آنها عمل نمایند خالی است.

## 1-2- روش‌های جستجو و بازیابی تصاویر ساختارمند

بازیابی در هر دسته از تصاویر ساختارمند متفاوت با سایر دسته‌ها و همچنین متفاوت با روش‌های بازیابی تصاویر بدون ساختار می‌باشد. عملیاتی که در بازیابی هر دسته مورد استفاده قرار می‌گیرند بصورت خلاصه در زیر ذکر بیان شده‌اند [TALMUD]:

**متن‌های تایپ ماشینی اسکن شده:** در این سیستم‌ها اسناد تصویری موجود باید به اسناد متنی تبدیل شده و جستجو در متون استخراجی انجام پذیرد. برای تبدیل یک سند تصویری به سند متنی، ابتدا می‌بایستی برای حذف پس‌زمینه و سایه‌ها، نویززدایی و ارتقای کیفیت متون باقیمانده، اقدامات لازم صورت پذیرفته و سپس عملیات تشخیص حروف و کلمات ( $OCR^1$ ) بر روی آن انجام پذیرد.

**متن‌های دست‌نویس اسکن شده:** متون نوشته شده با فونت‌های غیرمعمول، یا با کیفیت پایین علاوه بر متون دست‌نویس در این دسته قرار می‌گیرند. برای تشخیص متن از تکنیک «تشخیص کلمه»<sup>2</sup> استفاده

---

<sup>1</sup> Optical Character Recognition

<sup>2</sup> Word Spotting

می‌شود که متن به کلمات متصل به هم سگمنت‌بندی شده و از تکنیک‌های «تطبیق تصویر» به جای *OCR* استفاده می‌شود. جهت آموزش تکنیک‌های تطبیق تصویر، در ابتدا می‌بایستی نمونه‌هایی از تصاویر کلماتی که برای ما اهمیت دارند، به سیستم معرفی شوند و عملیات آموزش سیستم بر روی آنها انجام پذیرد.

**دیگرام‌ها:** گسترشی از برنامه‌های *OCR* هستند که باید قادر به درک بعضی از نمادهای از قبل تعریف شده نیز باشند.

**تشخیص صورت در تصاویر:** برای این منظور از تکنیک‌هایی نظیر تحلیل مولفه اصلی<sup>۱</sup> (*PCA*)، تحلیل تفکیک کننده اصلی (*LDA*)<sup>۲</sup>، تبدیل موجک گبور<sup>۳</sup> و جت‌های گبور، شبکه‌های عصبی و ... استفاده می‌شود.

**تشخیص احساسات در تصاویر صورت:** اکثر تکنیک‌های مورد استفاده در تشخیص صورت در اینجا هم مورد استفاده قرار می‌گیرد.

## 1-3- روش‌های جستجو و بازیابی تصاویر عمومی و بدون ساختار

جستجو بر روی داده‌های بدون ساختار عموماً از جستجوهای از سطح ۱ می‌باشند. در این جستجوها بر خلاف جستجو در حوزه تصاویر ساختارمند از ویژگی‌های عمومی و سطح پایین تصاویر استفاده می‌شود. از مهم‌ترین ویژگی‌های مورد استفاده در روش‌های بازیابی این دسته، می‌توان از رنگ، بافت، شکل، طرح‌بندی مکانی، تبدیل‌های چند وضوحی<sup>۴</sup> شدت پیکسل‌ها نظیر تبدیل موجک و فیلترهای گاوسی چند مقیاسی<sup>۵</sup> نام برد.

---

1 Principle Component Analysis

2 Linear Discriminate Analysis

3 Gabor Wavelet Transform

Multi Resolution 4

Multi-Scale Gaussian filtering 5

در ادامه این بخش ابتدا تکنیک‌های بکار رونده در بازیابی تصاویر بر اساس مهمترین ویژگی‌های مورد استفاده یعنی رنگ، طرح‌بندی، بافت و شکل مورد بررسی قرار گرفته و در نهایت توضیحات مختصری در مورد جستجوهای سطوح بالاتر ارائه می‌شود.

### بازیابی بر اساس ویژگی رنگ

مهمترین نمایش رنگی مورد استفاده در بازیابی تصاویر هیستوگرام‌های رنگی می‌باشند. مهمترین تکنیک‌هایی که از این نمایش رنگی استفاده می‌کنند عبارتند از: اشتراک هیستوگرام‌ها<sup>۱</sup>، مقایسه هیستوگرام‌ها و هیستوگرام رنگ تجمیعی<sup>۲</sup> (جهت کاهش اثرات نویز). نمایش‌های دیگری نیز برای ویژگی‌های رنگی جهت بازیابی تصاویر مورد استفاده قرار گرفته است. مثلاً می‌توان به گشتاورهای رنگی<sup>۳</sup> و مجموعه‌های رنگی<sup>۴</sup> اشاره نمود.

### بازیابی بر اساس ویژگی طرح‌بندی رنگ

استفاده تنها از ویژگی رنگی بر روی پایگاه داده‌های با تعداد تصاویر بالا باعث می‌شود تا میزان پاسخ‌های برگشتی اشتباه بسیار بالا برود و استفاده از این ویژگی را ناکارآمد کند. به همین دلیل استفاده از ویژگی طرح‌بندی رنگی<sup>۵</sup> (ترکیب ویژگی رنگ و رابطه مکانی پیکسل‌ها). ایده کلی این روش استخراج ویژگی‌های رنگی در زیر نواحی مشخصی از تصویر می‌باشد که از آن جمله می‌توان به استفاده از ویژگی رنگ در درختان چهارتایی اشاره نمود. قطعه‌بندی تصویر و استفاده از ویژگی رنگ در هر ناحیه می‌تواند نتایج دقیق‌تری را تولید کند ولی به همان نسبت پیچیدگی کار را نیز بالاتر می‌برد. از جمله روش‌های دیگری که با این دید به مساله پرداخته‌اند می‌توان به موارد زیر اشاره نمود: استفاده از سه گشتاور رنگی اول در

---

Histogram Intersection 1

Cumulated Color Histogram 2

Color Moments 3

Color Sets 4

Color Layout 5

نواحی در چند ناحیه همپوشان از قبل تعیین شده ، روش بر پایه تمایز قایل شدن بین پیکسل‌های شبیه و غیر شبیه هر ناحیه به میانگین رنگی آن ناحیه و ... . طرح‌بندی مکانی را می‌توان با سایر ویژگی‌های دیگر نیز (نظیر بافت و ...) بکار برد.

### بازیابی بر اساس ویژگی بافت

بافت به الگوهای بصری‌ای گفته می‌شود که توسط نمایش رنگی قابل ارائه نیستند. مدل‌های مختلفی برای نمایش و استفاده بافت‌ها ارائه شده است. بازیابی بر اساس بافت به تنهایی کاربرد چندانی ندارد. اما استفاده از شباهت بافت بین نقاطی که دارای رنگ همسانی هستند (نظیر آسمان و دریا و یا برگ‌ها و علف) بسیار مفید می‌باشد. تکنیک‌های مختلفی جهت شناخت بافت پیشنهاد شده است. استفاده از ماتریس هم‌وقوعی<sup>۱</sup> جزو اولین تلاش‌ها در این زمینه بوده است که سابقه آن به خیلی قبل از مبحث بازیابی تصویر برمی‌گردد. از مهم‌ترین روش‌های این زمینه می‌توان به روش *Tamura* که بر پایه مطالعات روانشناسانه و استخراج ویژگی‌های بافتی دارای مفهوم می‌باشد، اشاره نمود.

مهمترین تکنیک‌های موجود برای بازیابی بر اساس بافت بر پایه مقایسه مقادیر آماره‌های درجه دوم<sup>۲</sup> بین مورد پرس‌وجو و داده‌های بانک اطلاعاتی کار می‌کنند. با استفاده از این آماره‌ها اندازه‌گیری ویژگی‌هایی از بافت تصویر نظیر کنتراست، زبری<sup>۳</sup>، جهت بافت<sup>۴</sup> و نظم و باقاعدگی<sup>۵</sup> بافت یا حالت تناوبی، جهت بافت و تصادفی بودن بافت ممکن می‌شود. رهیافت‌های دیگری برای اندازه‌گیری بافت عبارتند از: نمایش‌های بر پایه تبدیل موجک ، فیلترهای گبور و فراکتال‌ها. مقاله‌های مختلفی در جهت مقایسه کارایی این روش‌ها در کاربردهای پردازش تصویر ارائه شده است.

---

Co-occurrence matrix 1

Second-order statistics 2

Coarseness 3

Directionality 4

Regularity 5

## بازیابی بر اساس ویژگی شکل

بازیابی بر اساس شکل شهودی‌ترین نوع بازیابی در سطح ۱ می‌باشد. یکی از مهمترین ویژگی‌هایی که آن را ویژگی شکل را برای کاربردهای پردازش تصویر مناسب ساخته است این است که یک ویژگی شکل می‌تواند در برابر انتقال، دوران و تغییر مقیاس هویت خود را حفظ کند.

در این روش‌ها تعدادی از ویژگی‌های مشخصه اشیای داخل تصاویر (که مستقل از اندازه و جهت می‌باشند)، برای هر تصویر ذخیره شده و بازیابی بر اساس آنها انجام می‌شود. ویژگی‌های شکل بر اساس اینکه به چه صورت استخراج می‌شوند، به دو دسته تقسیم می‌شوند، ویژگی‌های بر پایه لبه‌ها و ویژگی‌های بر پایه نواحی. مهمترین ویژگی‌های مشخصه اشیا که به کار گرفته می‌شوند، عبارتند از: ویژگی‌های سراسری نظیر نسبت ابعاد شی<sup>۱</sup>، مدور بودن<sup>۲</sup> و ثوابت گشتاور<sup>۳</sup>، توصیف‌گرهای فوریه<sup>۴</sup> (استفاده از لبه‌های تبدیل فوریه یافته) و ویژگی‌های محلی نظیر سگمنت‌های لبه‌های متوالی<sup>۵</sup>؛ از روش‌های دیگری که برای انطباق اشکال پیشنهاد شده‌اند، می‌توان از روش دگردیسی کشسان قالب‌ها<sup>۶</sup>، مقایسه هیستوگرام جهتی<sup>۷</sup> لبه‌های تصاویر و نمایش اسکلتی<sup>۸</sup> اشیای تصویر که با تکنیک‌های انطباق گراف با هم مقایسه می‌شوند، روش عناصر محدود<sup>۹</sup>، توابع گردشی<sup>۱۰</sup> و توصیف‌گرهای موجک نام برد.

---

Aspect ratio 1

Circularity 2

Moment Invariants 3

Fourier Descriptors 4

Consecutive Boundary Segments 5

Elastic Deformation of Templates 6

Directional Histogram 7

Skeletal Representation 8

Finite Element Method 9

Turning Functions 10

## تکنیک‌های موجود برای بازیابی بر اساس محتوی در سطح ۲

اکثر سیستم‌های بازیابی چندرسانه‌ای موجود بر پایه ویژگی‌های ابتدایی هستند. برخی از پژوهش‌ها برای کوتاه کردن فاصله بین بازیابی سطح ۱ و سطح ۲ در حال انجام هستند. پژوهش‌هایی که در این راستا انجام می‌گیرند، بیشتر در دو راستا می‌باشد: شناسایی و تحلیل صحنه و شناسایی اشیای موجود در صحنه. شناسایی صحنه در تصاویر بسیار مهم می‌باشد، چون هم می‌توان اطلاعاتی راجع به صحنه و هم راجع به اشیای موجود در صحنه استخراج کرد. از سیستم‌هایی که با این دید به مساله پرداخته‌اند می‌توان به سیستم *IRIS* اشاره نمود که با استفاده از ویژگی‌های رنگ، بافت، ناحیه و اطلاعات مکانی به تفسیر صحنه پرداخته و توصیف‌هایی متنی در مورد تصویر را تولید می‌نماید. نمونه‌ای دیگری از این گروه، استخراج اجزای با فرکانس پایین تصویر می‌باشد که بعداً توسط شبکه‌های عصبی مصنوعی مورد تجزیه و تحلیل قرار می‌گیرند یا در نمونه دیگر اطلاعات رنگی همسایگی از تصاویر با رزولوشن پایین استخراج می‌شوند تا قالبهای تعریف شده توسط کاربر را بسازند. تلاش‌های زیادی نیز برای تشخیص اشیا در تصاویر انجام گرفته است. از آن جمله می‌توان به سیستم‌هایی اشاره نمود که سعی در تشخیص اشیا با الهام‌گیری از فرایندهایی که در مغز انسان رخ می‌دهد پرداخته‌اند. سیستم *SIMPLICity* و سیستم‌های بر پایه پردازش‌های سلسله‌مراتبی تحلیلی<sup>۱</sup>، جدیدترین تلاش‌ها، از این گروه می‌باشد. در سیستم *SIMPLICity* سعی در کلاسه‌بندی معنایی تصاویر بصورت خودکار بر اساس ویژگی‌های سطح پایین آن می‌شود. بر اساس ویژگی‌هایی نظیر رنگ و بافت، این سیستم سعی می‌کند تا تصاویر را به کلاس‌های نظیر (بافتی، غیر بافتی)، (عکس طبیعی، تصویر ساختگی)، (تصویر داخلی<sup>۲</sup>، تصویر بیرونی<sup>۳</sup>) و ... کلاسه‌بندی نموده و بازیابی محتوایی را براساس آنها انجام دهد. و در سیستم‌های بر پایه پردازش‌های سلسله‌مراتبی تحلیلی (*AHP*)، ابتدا یک ساختار سلسله‌مراتبی معنایی از مفاهیمی که در پایگاه داده ممکن است ظاهر شوند،

---

process (AHP) 1 analytic hierarchical

2 Indoor

3 Outdoor

تشکیل می‌شود. در مرحله آموزش ویژگی‌های سطح پایین متناظر با هر گره از ساختار سلسله مراتبی به آن گره نسبت داده می‌شود. سپس در مرحله استفاده، هر تصویر با ورود به پایگاه داده در ساختار سلسله مراتبی از ریشه شروع به حرکت نموده و با مقایسه ویژگی‌هایش با ویژگی‌های هر نود فرزند در ساختار، به سمت برگ‌ها حرکت می‌کند تا در نهایت در کلاسی از برگ‌ها، که از لحاظ مفهومی به آن شباهت بیشتری دارد، قرار بگیرد. در این روش، جستجو بر اساس مفاهیم و برچسب‌های داده شده به تصاویر انجام می‌پذیرد.

علاوه بر سیستم‌های فوق که تمام خودکار می‌باشند، سیستم‌هایی نیز ارائه شده‌اند که بر اساس بازخوردی که از کاربران می‌گیرند سعی در یادگیری و ایجاد ارتباط بین مفاهیم معنایی و ویژگی‌های سطح پایین دارند که از آن نمونه به *FourEyes* از دانشگاه *MIT* اشاره نمود. در این سیستم از کاربران خواسته می‌شود تا در تصاویری که به آنها نشان داده می‌شود، برچسب‌هایی را برای نواحی مشخص شده عنوان کنند. سیستم از این توضیحات کاربران استفاده نموده و از آنها برای برچسب‌گذاری نواحی‌ای از سایر تصاویر که دارای ویژگی‌های اولیه شبیه این ناحیه می‌باشند، استفاده می‌کند. نمونه دیگری از این دسته مفهوم «قالب بصری مفهومی»<sup>۱</sup> می‌باشد که در مطرح شد. در این روش کاربر تقاضای یک مفهوم می‌کند. سیستم از کاربر می‌خواهد محدوده رنگ، بافت، شکل و پارامترهای جابجایی مفهوم مورد نظرش را معین نماید و براساس این ویژگی‌ها تصاویر بدست آمده را به وی نشان می‌دهد. این فرآیند در یک حلقه بازخورد کاربر-اصلاح سیستم پیش می‌رود تا کاربر تصویر مورد نظر خود را بیابد. سیستم واژه مورد جستجو را به تصویر پیدا شده برچسپ می‌زند تا برای کاربران آینده مورد استفاده و مفید واقع شود. بعد از گذشت مدتی از زمان این پایگاه داده حاوی گنجینه‌ای از اطلاعات مفهومی بر پایه ویژگی‌های اولیه خواهد بود.

### تکنیک‌های موجود برای بازیابی بر اساس محتوی در سطح ۳

در این سطح، تقریباً کار خاصی انجام نشده است. تنها نمونه‌هایی که شاید بتوان آنها را در این دسته جا داد، تحقیقاتی است که در آنها بازیابی بر اساس رنگ‌ها و مطرح کردن مفاهیم گرم و سرد بودن رنگ‌ها و یا اینکه آیا دو رنگ بهم می‌آیند یا خیر در آنها مطرح شده است.

## 2. بررسی روشهای بازیابی بر اساس ویژگی شکل

در این قسمت به بررسی روشهای مختلف بازیابی بر اساس ویژگی شکل می‌پردازیم. برای اینکار به مرور روشهای توصیف شکل (*shape description*) می‌پردازیم.

معمولا توصیف شکل بعد از *segmentation* انجام می‌گیرد. ما در اینجا فرض می‌کنیم که مرز (*boundary*) شکل را به طریقی بدست آورده‌ایم و سعی در توصیف بر اساس شکل خواهیم داشت.

### نمایشگرهای ناوردا<sup>1</sup>:

ایده اصلی در استفاده از نمایشگرهای ناوردا بدست آوردن مجموعه‌ای فشرده از توصیف‌هاست که برای توصیف همه نماهای مربوط به یک شکل کافی باشد. خواص نوردایی که انتظار داریم برای ما مهم باشد جابجایی، دوران و تغییر مقیاس است. گستره وسیعی از الگوریتم‌ها برای تشخیص اجسام 2 بعدی ارائه شده که هر از یک خصیصه‌های متفاوت و روشهای تطبیق مختلفی استفاده می‌کنند.

### 2-1- ناورداهای گشتاور<sup>2</sup>:

استفاده از گشتاور به عنوان ناوردا در اشکال دودویی برای اولین بار در [8] 1961 ارائه شد. *Hu* توانست با موفقیت از این روش برای تشخیص حروف دست‌نویس استفاده کند. گشتاور یک تصویر  $M$  در  $N$  به صورت زیر تعریف می‌شود:

$$u_{pq} = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} i^p j^q f(i, j)$$

که  $f(x,y)$  شدت رنگ هر پیکسل (در اینجا 0 یا 1) در مختصات  $(x,y)$  می‌باشد و  $p+q$  مرتبه گشتاور نامیده می‌شود.

---

<sup>1</sup> Invariant Representations

<sup>2</sup> Moment Invariants

چون محاسبه تابعی از فاصله پیکسلها و مرکز ثقل تصویر می باشد (که حاصل باید ناوردا باشد)، مختصات مرکز ثقل طبق فرمول زیر محاسبه می گردد:

$$i = \frac{u_{10}}{u_{00}} \quad j = \frac{u_{01}}{u_{00}}$$

پس گشتاورهای مرتبط با فرمول گشتاورهای مرکزی زیر محاسبه می گردد:

$$u_{pq} = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} (i - i)^p (j - j)^q f(i, j)$$

هر یک از گشتاورها به تنهایی نمی تواند به صورت یکتا اشکال دلخواه را توصیف کند، ولی مجموعه ای از توابع بر حسب این گشتاورها می تواند اینکار را انجام دهد.  $Hu$  مجموعه ای از 7 تابع ناوردا نسبت به دوران ارائه کرد که روش مناسبی برای توصیف اشکال را تشکیل می دهد.

$$M_1 = (u_{20} + u_{02})$$

$$M_2 = (u_{20} - u_{02})^2 + 4u_{11}^2$$

$$M_3 = (u_{30} - 3u_{12})^2 + (3u_{21} - u_{30})^2$$

$$M_4 = (u_{30} + u_{12})^2 + (u_{21} + u_{03})^2$$

$$M_5 = (u_{30} - 3u_{12})(u_{30} + u_{12}) ((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2) \\ + (3u_{21} - u_{03})(u_{21} + u_{03}) (3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2)$$

$$M_6 = (u_{20} - u_{02}) ((u_{30} + u_{12})^2 - (u_{21} + u_{03})^2) + 4u_{11}(u_{30} + 3u_{12})(u_{21} + u_{03})$$

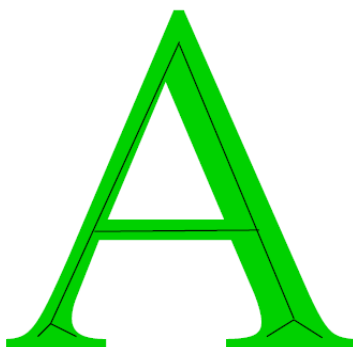
$$M_7 = (3u_{21} - u_{03})(u_{30} + u_{12}) ((u_{30} + u_{12})^2 - 3(u_{21} + u_{03})^2) \\ - (u_{30} - 3u_{12})(u_{21} + u_{03}) (3(u_{30} + u_{12})^2 - (u_{21} + u_{03})^2)$$

طبقه بندی اشکال با تطبیق بردارهای خاصیت بدست آمده توسط این توابع با بردارهای بدست آمده از مجموعه تست صورت می گیرد. برای بهبود نمایش اشکال می توان از نرمال کردن مقیاس برای ناوردایی

نسبت به مقیاس استفاده کرد. روش گشتاور را می‌توان برای تصاویر دودویی قطعه قطعه شده یا برای مرز اشکال به کار برد.

## 2-2- استخوان بندی شکل<sup>1</sup>

یکی از روشهای توصیف اشکال یافتن استخوان بندی برای شکل است. اسکلت بندی شکل با نازک کردن متوالی شکل تا بدست آمدن خطی به پهنای 1 پیکسل بدست می‌آید. برای مثال در شکل زیر استخوان بندی رسم شده است.



فلسفه استفاده از استخوان بندی این است که در تشخیص اشکال بیشترین اطلاعات در مورد یک شکل در توپولوژی آن شکل قرار دارد. برای مثال در تشخیص دست نوشته حروف با پهنای نوشته شدن آنها تشخیص داده نمی‌شوند بلکه با استفاده از توپولوژی و نحوه اتصال قطعه‌ها شناخته می‌گردند. تشخیص اشکال با استفاده از استخراج اسکلت بندی از تصاویر و تطابق آن با مجموعه *training* صورت می‌گیرد. نازک کردن<sup>2</sup> شکل اغلب با زدودن تدریجی مرز شکل صورت می‌گیرد به طوری که در این حین توپولوژی شکل تغییر نکند. یکی از الگوریتمهای کارا در این زمینه در [9] آورده شده است. شبکه توپولوژی حاوی اکثر خاصیت‌های نوردای مورد نیاز برای استفاده در سیستمهای بینایی است. البته در بیشتر کاربردها ساختار توپولوژی محض کافی نمی‌باشد و برخی از اطلاعات ساختاری نیز باید اضافه

---

<sup>1</sup> Shape Skeleton

<sup>2</sup> thinning

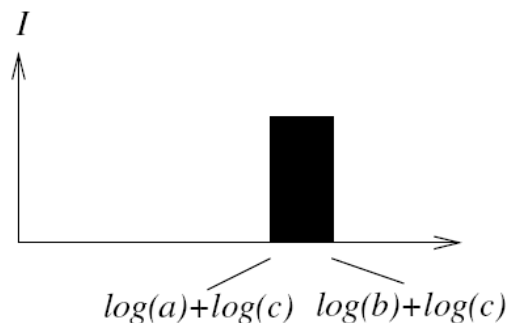
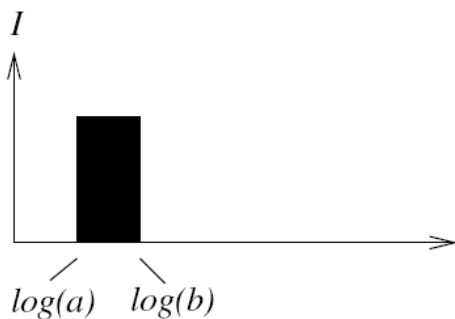
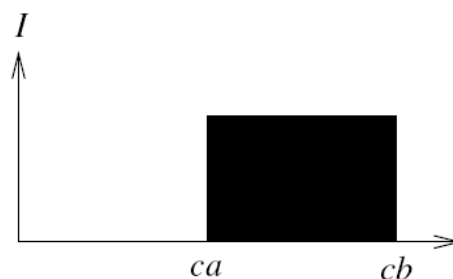
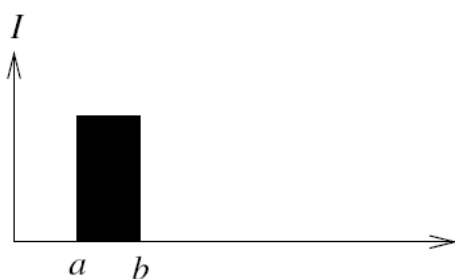
گردد. الگوریتمهای نازک کردن حساسیت چندانی به نویز ندارند و با روشهایی می توان حساسیت نسبت به نویز را حذف کرد، ولی این الگوریتمها حساسیت زیادی نسبت به هم پوشانی اشکال دارند.

## 2-3-نگاشت لگاریتمی-قطبی<sup>1</sup>

یکی از نمایشهای ناوردای شکل، با استفاده از نگاشت لگاریتمی-قطبی انجام میگردد. در این روش نقاط در صفحه تصویر به نقاطی در صفحه لگاریتمی-قطبی نگاشته میشوند. یک نقطه  $z$  در فضای تصویر در نظر بگیرید به طوریکه  $z=x+iy$

نگاشت لگاریتمی-قطبی این نقطه را به نقطه  $w$  در فضای جدید تبدیل می کند به طوریکه  $w=\ln(z)$  یا به بیان دیگر  $w=\ln(|z|)+j\theta$

خاصیت ناوردایی تبدیل از آنجا حاصل میشود که تغییر در مقیاس و جهت تصویر در فضای تبدیل به جابجایی نگاشته میشود. تبدیل مقیاس به جابجایی به آسانی در حالت یک بعدی قابل مشاهده است :



همانطور که از شکل معلوم است، نقاط  $a$  و  $b$  که با ضریب  $c$  تغییر مقیاس یافته اند، در فضای تبدیل با جابجایی به اندازه  $\log(c)$  مشخص میشود. با اینکه در نگاشت لگاریتمی-قطبی ناوردایی مقیاس و دوران

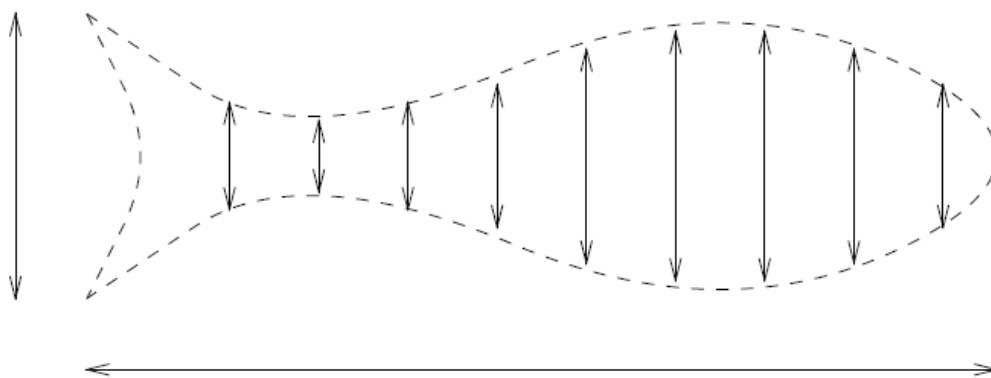
<sup>1</sup>Log-polar Mapping

به طور کامل وجود ندارد، ولی میتوان با تبدیل آنها به جابجایی به راحتی به این ناوردایی دست یافت. در  $[10]$  و  $[11]$  از تبدیل فوریه از صفحه لگاریتمی-قطبی و مقادیر آن برای دست یافتن به ناوردا استفاده شده است. به خاطر اینکه ناوردایی ذاتی نسبت به جابجایی ناوردا نیست، شکل مورد نظر با استفاده از مرکز ثقل آن به مرکز مختصات انتقال می‌یابد.

با بررسی بیشتر این روش معلوم شد که تغییرات کوچک در یافتن مرکز ثقل موجب تغییرات بزرگی در نمایش لگاریتمی قطبی می‌شود. یکی دیگر از مشکلات نقطه تکین در  $\log(0)$  است که مشکلاتی در محاسبه تبدیل آن به وجود می‌آورد.

## 2-4- توصیف گرهای هندسی شکل

با داشتن مرز شکلی دلخواه میتوان با معیارهای هندسی ساده یک توصیف برای شکل ساخت. مثالهایی از این اندازه گیری ها عرض، ارتفاع، محیط، مساحت و موارد مشابه دیگر می باشد. توصیفگر شکل با ترکیب تعدادی از این معیارها بدست می آید. تشخیص اشیا با مقایسه بردارهای توصیف مجموعه آموزشی و بردارهای استخراج شده از تصویر جاری صورت میگیرد. برای اینکار به معیار فاصله مناسبی نیز نیازمندیم. در شکل زیر یک توصیفگر هندسی شکل با استفاده از 11 طول بدست آمده از شکل آورده شده است.



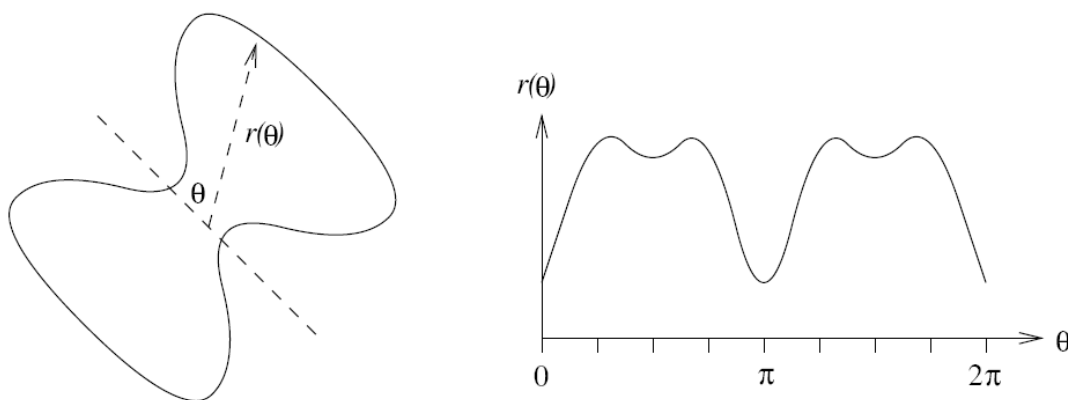
این اندازه گیریها شامل طول و عرض ماهی، و 9 عرض دیگر در فواصل مساوی از طول ماهی میباشد. ناوردایی با تغییر مقیاس ماهی به مساحت واحد و تنظیم آن در راستایی محور اصلی شکل بدست می آید.

## خصوصیات مرز :

برای اینکه بتوان تطبیق مرز شکل‌ها را آسانتر انجام داد، می‌توان مرز اشکال را از دامنه تصویر دوبعدی به خصوصیات مرز یک‌بعدی تبدیل کرده و سپس به دنبال تطبیق اشکال بود.

## نمودار $(r, \theta)$

یکی از فرمهای نمایش در یک بعد توصیف یک مرز در مختصات قطبی نسبت به مرکز ثقل آن است. هر نقطه در مرز با استفاده از فاصله آن از مرکز ثقل  $r$  و زاویه آن نسبت به یک محور مرجع  $\theta$  توصیف می‌شود.



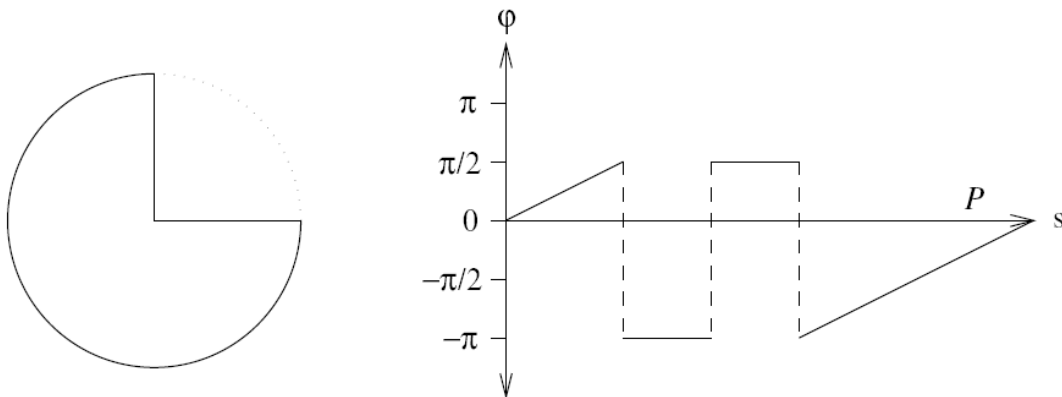
تشخیص اشکال با استفاده از تطبیق نمودارهای استخراج شده از تصویر جاری با نمودارهای استخراج شده از تصاویر آموزشی صورت می‌گیرد. اگر شکلها بتوانند در جهتهای مختلفی قرار گیرند نمودارهای آنها نسبت به محور جابجا می‌شود.

یکی از مشکلاتی که در نمودار  $(r, \theta)$  با آن مواجه هستیم آن است که حتی در شکل‌های ساده نیز نمودار در برخی نقاط چند مقداری می‌شود. این بدان معنی است که نمودار ما دوبعدی می‌شود و برای تطبیق مجدداً به مشکل کار با نمودارهای دوبعدی مواجه می‌شویم.

مانند همه نمایش‌هایی که از مرکز ثقل استفاده می‌کنند نمودار  $(r, \theta)$  نیز در صورت برخورد با همپوشانی غیر کارا خواهد بود.

## نمودار $(s, \theta)$ ، روش توصیف بر اساس انحنا

نمودار  $(s, \theta)$  جایگزینی برای نمودار قبلی است که چندمقداره نمی‌شود و از مرکز ثقل نیز استفاده نمی‌کند.



همه ما تصویری ذهنی از شدت یا ضعف خمیدن یا انحنای یک خم پارامتری داریم. معمولاً تصور ما از انحنا شدت انحراف یک خم از خط راست بودن است. مثلاً انحنای یک دایره با شعاع بزرگ را کوچکتر از انحنای یک دایره با شعاع کوچک می‌دانیم. برای تعریف دقیق ریاضی از انحنا به طریق زیر عمل می‌کنیم. نخست بحث ما محدود به خم‌های  $f: I \rightarrow R^2$  خواهد بود. هر خم در  $R^2$  را می‌توان به صورت پارامتری زیر نشان داد:

$$r(u) = (x(u), y(u))$$

یک خم می‌تواند پیمایش‌های گوناگونی داشته باشد. یکی از این پیمایش‌ها بر اساس طول خم، پیمایش طبیعی (*natural*) نام دارد. یک پیمایش طبیعی را می‌توان از روی هر پیمایش با معادله زیر بدست آورد:

$$s(u) = \int_0^u |\dot{r}(v)| dv$$

که در اینجا

$$\dot{r}(v) = (\dot{x}(v), \dot{y}(v))$$

$$|\dot{r}(v)| = (\dot{x}(v)^2 + \dot{y}(v)^2)^{\frac{1}{2}}$$

نیم خطی را به عنوان یک جهت مرجع انتخاب می‌کنیم، مثلاً نیم خط افقی به طرف راست. بنابراین جهت مثبت خط مماس نیز مشخص است. حال زاویه  $\theta$  را به ازای هر  $t$  برابر زاویه از نیم خط مرجع به نیم خط مثبت مماس تعریف می‌کنیم. انحنا باید آهنگ تغییر  $\theta$  نسبت به مسافت طی شده روی خم

$$\kappa = \frac{d\theta}{ds} \text{ یعنی باشد،}$$

$\kappa$  یا تابع انحنا یک توصیف برای خم است. این توصیف برای بیان شکلها یک توصیف مناسب است. به

قضیه زیر توجه کنید :

قضیه اساسی خم ها : فرض کنید  $\alpha, \beta : I \rightarrow R^2$  دو خم پرمایش شده بر حسب طول باشند. انحنای دو

خم را با  $\kappa_1$  و  $\kappa_2$  نشان می‌دهیم. اگر به ازای هر  $s \in I$  داشته باشیم  $\kappa_1(s) = \kappa_2(s)$  آنگاه با یک

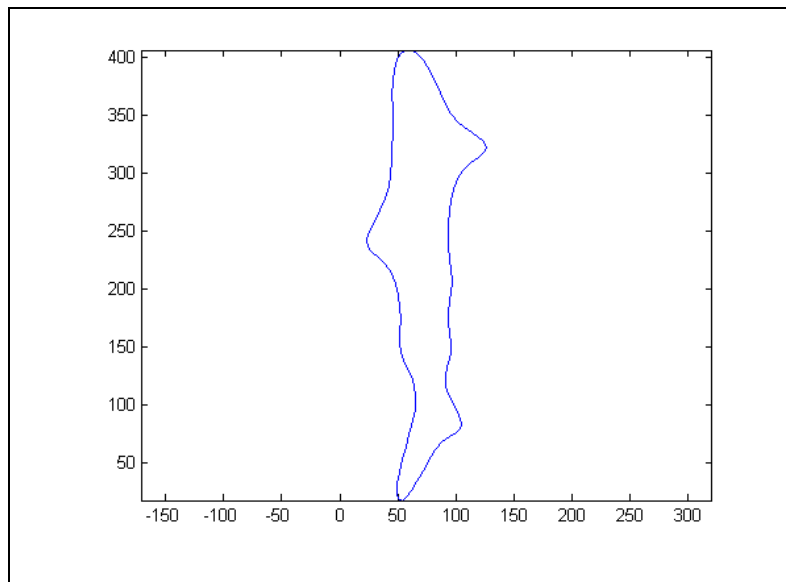
حرکت اقلیدسی (انتقال + دوران) می‌توان تصویر دو خم را بر هم منطبق کرد.

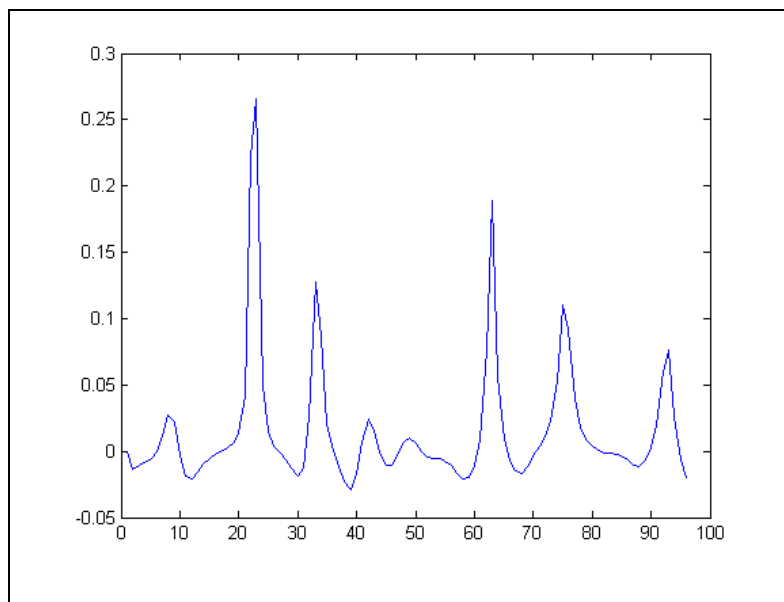
با استفاده از این قضیه می‌توان گفت که توصیف ما مستقل از دوران و انتقال شکل است و به نظر می‌رسد

که توصیفی با خواص مناسب است.

برای اختلاف در مقیاس در عمل پرمایشها را یکسان انتخاب می‌کنیم. در زیر نتیجه پیاده‌سازی این روش

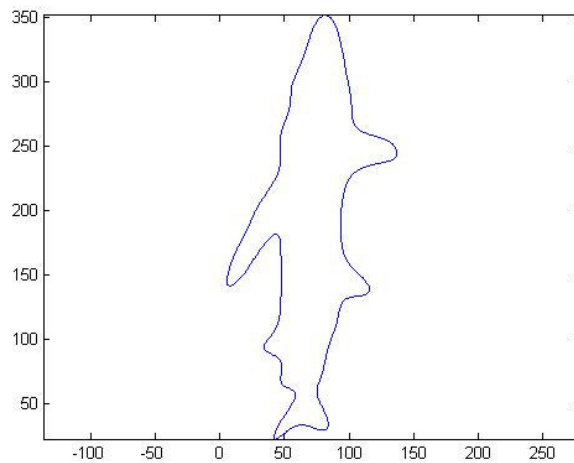
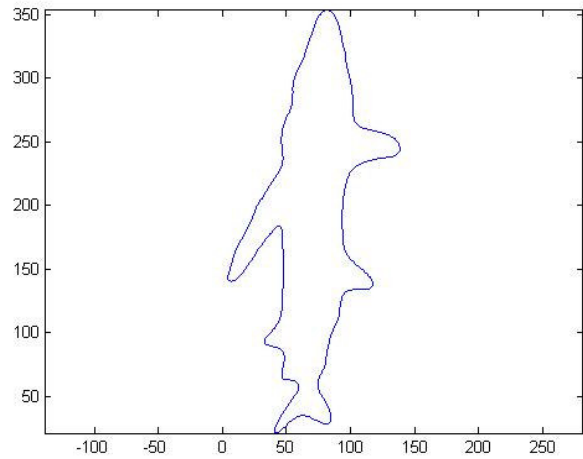
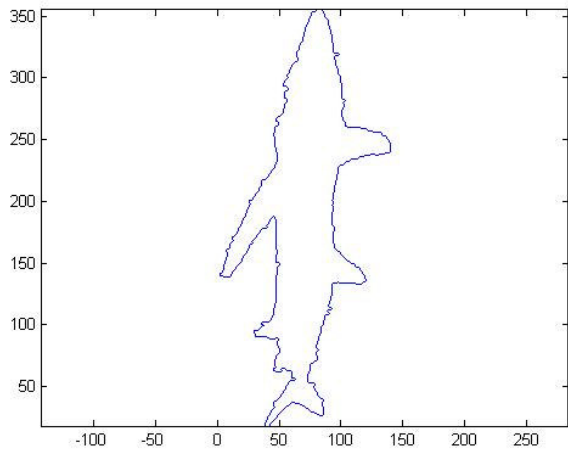
برای یک مثال دیده می‌شود :





### مرحله پیش پردازش

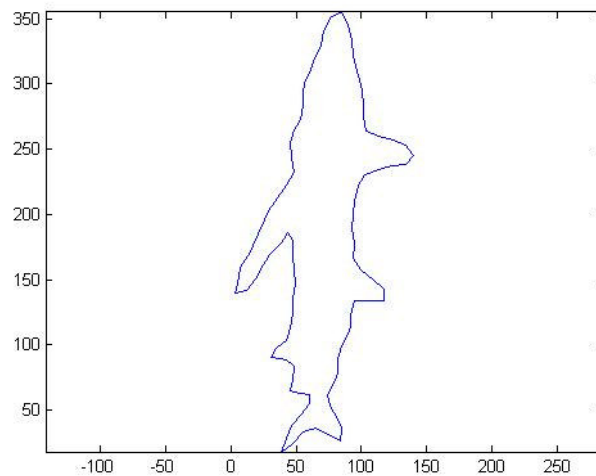
در این مرحله به هموار سازی تصویر برای از بین بردن اثر نویزها می پردازیم. خروجی این مرحله در قسمتهای دیگر استفاده میشود. در روش اول با فیلتر *gaussian* بر روی مرزها به هموار سازی تصویر میپردازیم. در این روش هموار سازی به کندی صورت میگیرد و حدود 50 مرحله تکرار برای بدست آوردن تصویر مناسب لازم است.



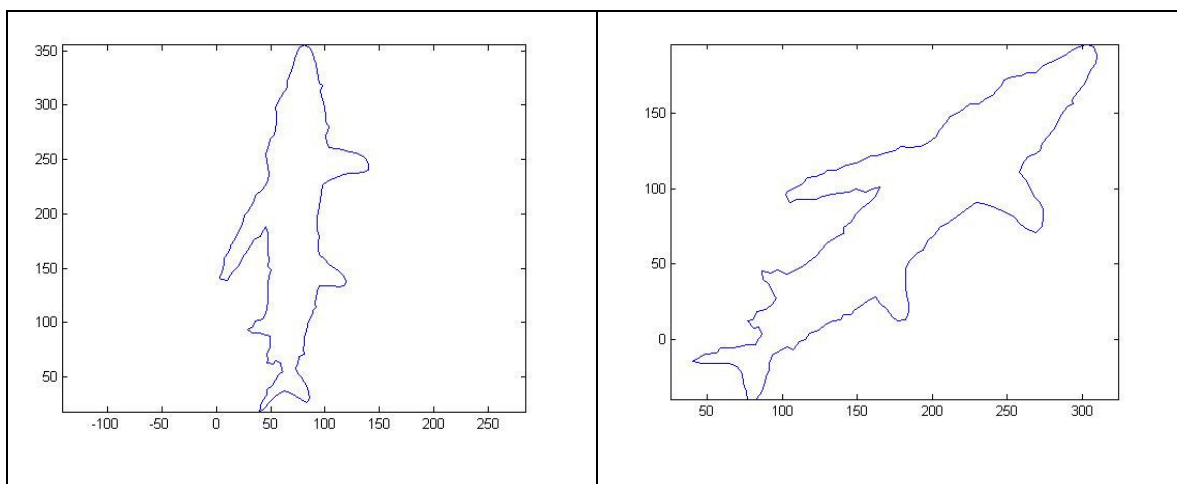
هموار سازی نقطه ای منحنی ها - تصویر اصلی - 50-مرحله - 100 مرحله

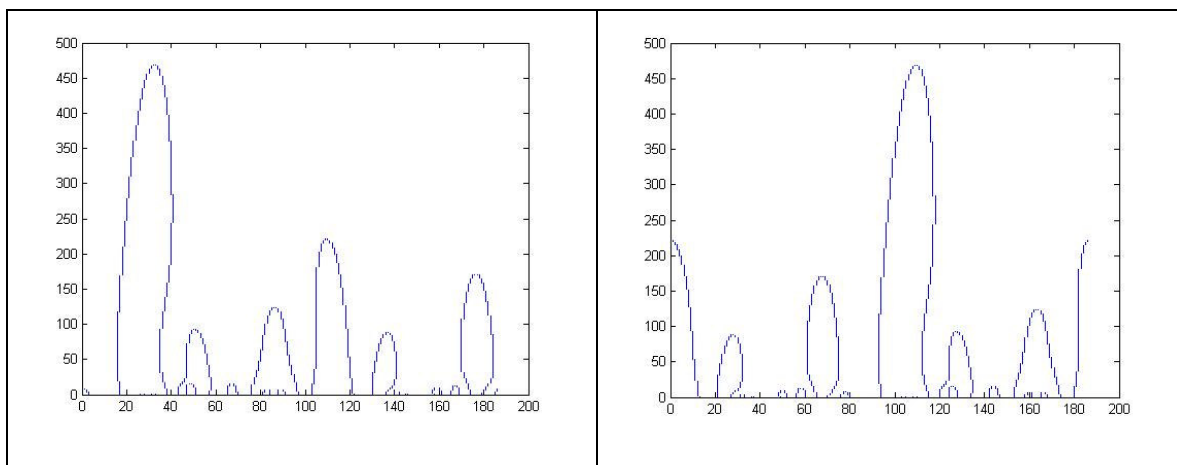
در عمل این روش دارای کارایی پایین میباشد

در روش دیگر ابتدا طول منحنی را بدست آورده و سپس 50 تا 200 نقطه همفاصله روی آن انتخاب میکنیم. در این روش با یک مرحله انتخاب نقاط تصویر مناسب هموار شده ای بدست می آید و میتوان روی آن یک مرحله فیلتر گاوسی اعمال کرده تا نتیجه بهتری داشت. این روش حتی بدون اعمال فیلتر نتیجه مناسبی میدهد و محاسبه آن بسیار سریعتر است.



یکی از مهمترین روشهای پیشنهاد شده در این زمینه روش پیشنهاد شده در [7] است. در این روش  $zero$   $crossing$  های تابع انحنای در مراحل مختلف بدست آورده شده و رسم می‌گردد. در هر مرحله با استفاده از فیلتر گاوسی تصویر هموارتر شده و مجدداً  $zero$   $crossing$  ها محاسبه می‌گردد. تصویر نهایی که از این روش بدست می‌آید مستقل از دوران و انتقال می‌باشد. این روش پیاده‌سازی شده و در زیر نتایج آن دیده می‌شود .





نتیجه پیاده سازی روش پیشنهادی در [7]

با چرخاندن و تغییر نقطه شروع نمودار حاصله فقط یک جابجایی پیدا میکند و به همین دلیل با بررسی تعداد *zero crossing* ها در تابع انحنا در *iteration* های مختلف میتوان معیاری برای تشابه تصاویر بدست آورد. در این روش با استفاده نقاط *maximum* در نمودار *scale space* بدست آمده و الگوریتم تطبیق به بررسی و دسته بندی اشکال می پردازد.

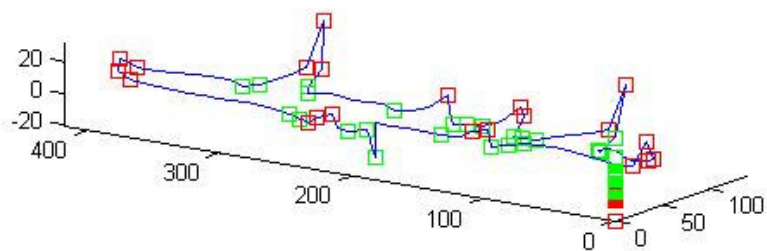
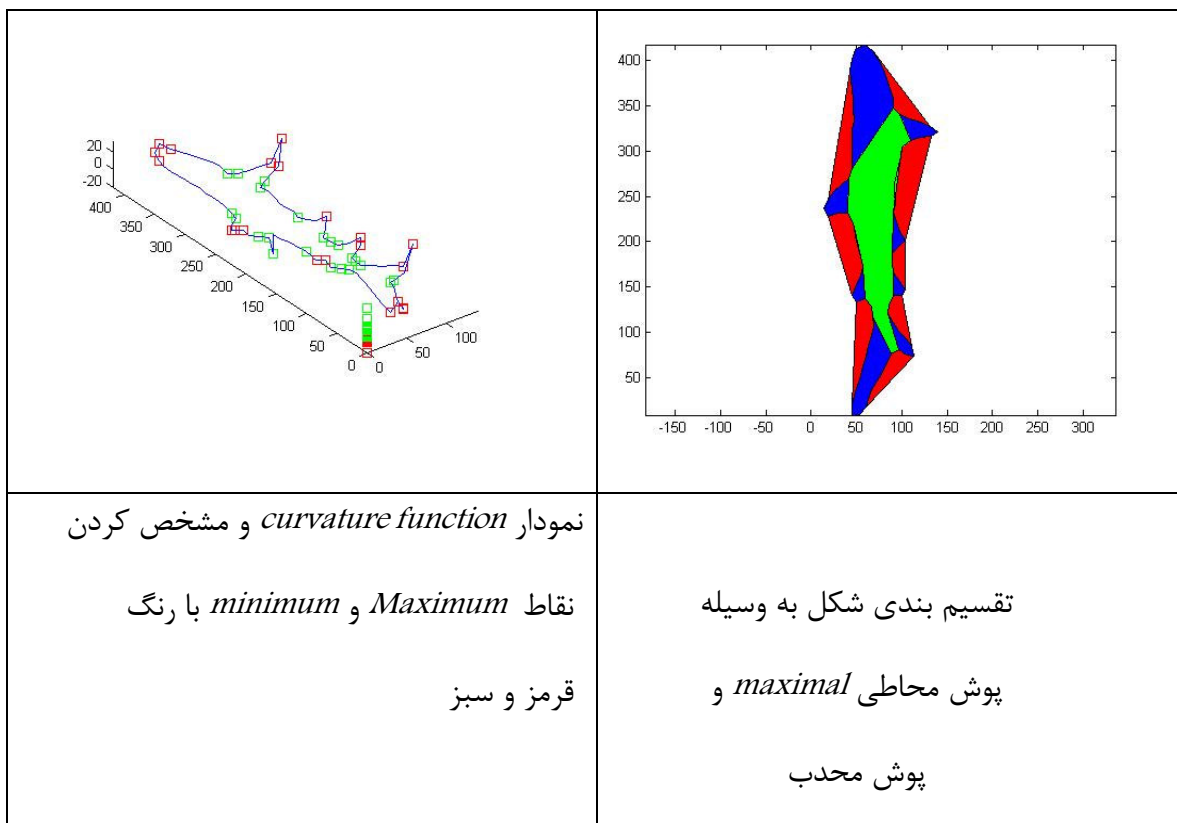
این روش بسیار رایج شده و در استاندارد *MPEG7* هم از آن برای توصیف اشکال استفاده می شود. در کل الگوریتمهای زیادی بر اساس تابع انحنا ارائه شده و استفاده می گردد و اگر بتوان مفهوم تابع انحنا را بهبود داد، نتایج بسیاری از الگوریتمها بهبود می یابد.

## 2-5- بررسی پوش محدب و مفاهیم مرتبط

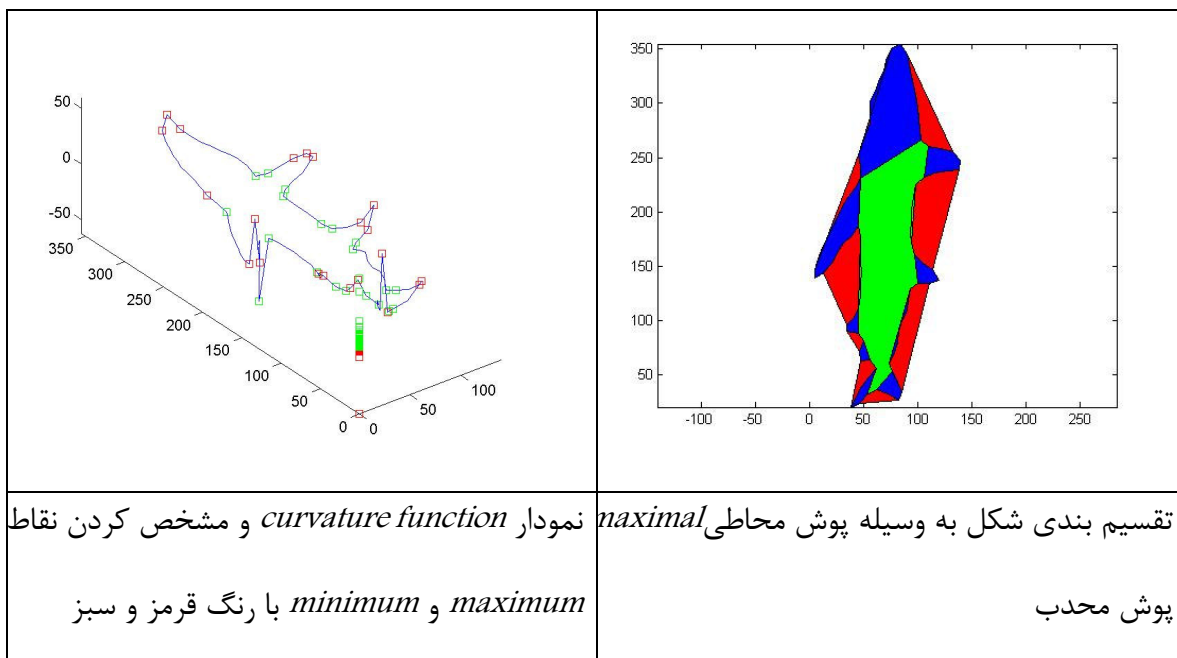
برای اشکال؛ تعریف چیزی مانند بزرگترین زیر مجموعه محدب مفید خواهد بود. در اینجا سعی کرده ایم این مفهوم را شبیه سازی کرده و از آن در تقسیم بندی شکلها استفاده کنیم. مفهوم ذکر شده خوش تعریف نیست و به دنبال پیدا کردن تعریفی مناسب و الگوریتم کارا برای محاسبه آن هستیم. در زیر نتیجه تقسیم بندی شکل با تقریب پیدا شده برای یافتن زیر مجموعه محدب *maximal* دیده میشود.

همچنین نمودار سه بعدی برای تابع انحنا ترسیم شده که دید خوبی نسبت به روش پیشنهادی در [7] میدهد. در حقیقت روش ذکر شده در آن مقاله پیدا کردن نقاط تقاطع این تصویر با صفحه  $Z=0$  است. ولی از نموداری که

به ترتیب پیش نهاد شده و بطور سه بعدی رسم میشود میتوان اطلاعات بیشتری کسب کرد که کسب اطلاعات بیشتر در دست بررسی است.

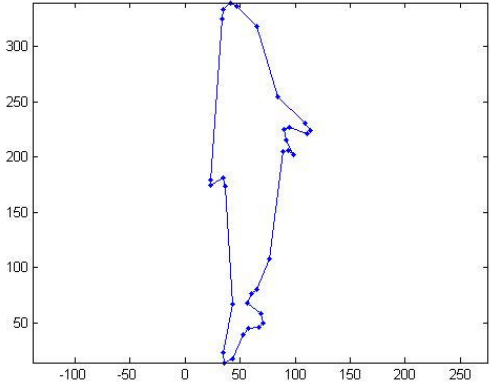
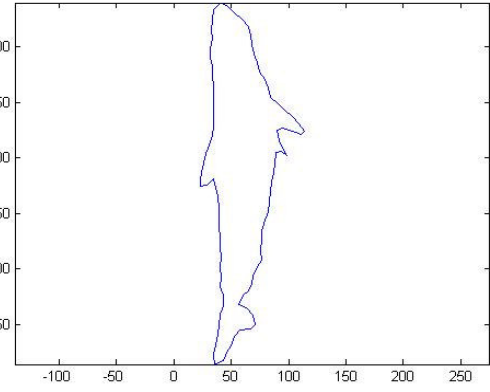
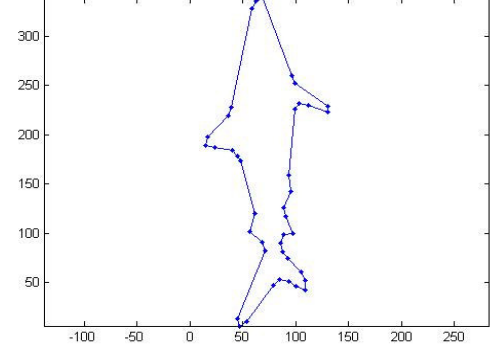
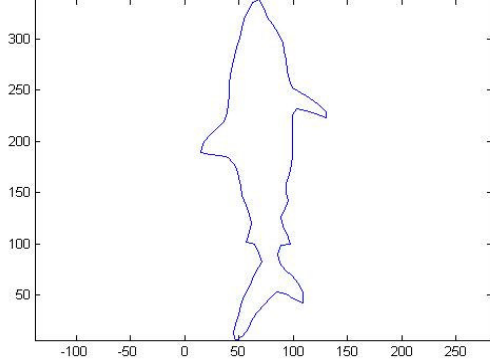
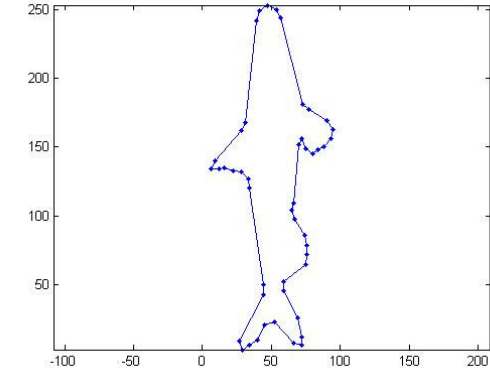
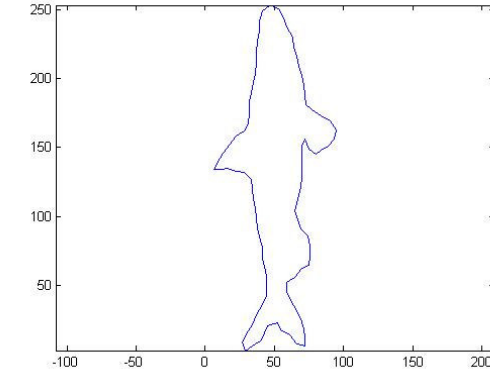


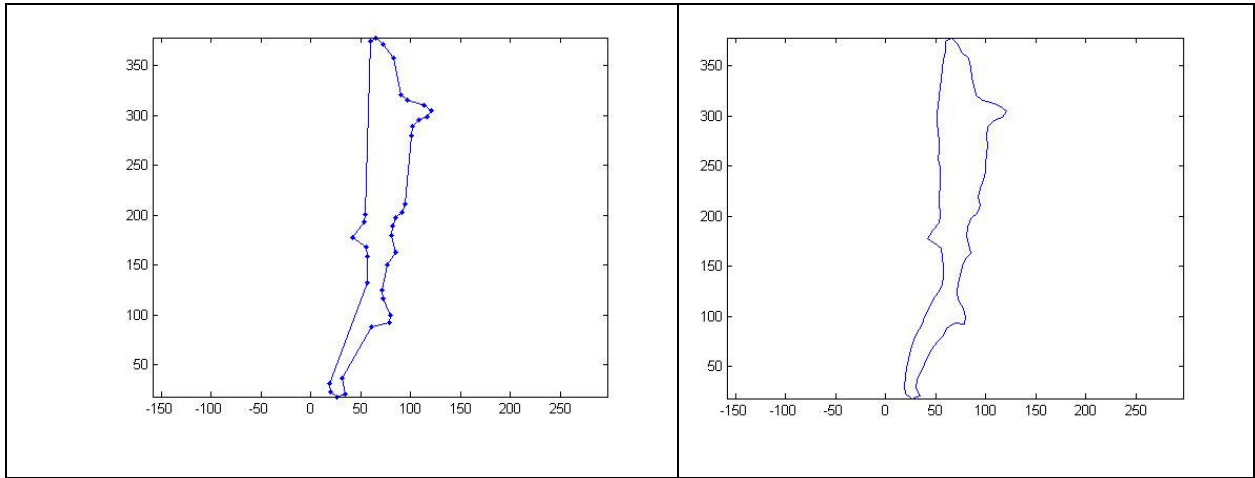
نمودار سه بعدی شده *curvature function* از نمای دیگر (*curvatureto3d.m*)



## 2-6- پیش نهاد یک روش تقریب زنی خطی بر اساس تابع انحنا

در اینجا نتایج یک تقریب زنی خطی نیز اضافه شده است. با انتخاب نقاط *max* و *min* از تابع انحنا می-توان یک تقریب خطی از تصویر را بدست آورد. در حقیقت نقاط *max* و *min* به عنوان نماینده‌ای برای نقاط تحدب و تقعر محسوب می‌گردند. این روش اشکالاتی نیز دارد، برای مثال در اشکالی که تغییر ناگهانی نداریم و تغییر به طور پیوسته صورت می‌گیرد، این تقریب ممکن است جواب مناسبی ندهد. نتایج این تقریب زنی برای چند تصویر در زیر آورده شده است:

تصویر تقریب زده شده	تصویر اصلی
	
	
	



### 3. تعمیم و بهبود تابع curvature

قسمت اصلی کار انجام شده بررسی اشکالهای تابع انحنا و سعی در تعمیم و بهبود این تابع می‌باشد. ابتدا روشی برای تعمیم تابع انحنا ارائه گشته و در نهایت جایگزینی برای آن ارائه می‌گردد. در اینجا سعی در تعمیم و بهبود تابع *curvature* برای استفاده در توصیف *curve* ها داریم. برای اینکار ابتدا تعمیمی برای مماس در نقطه  $s(u)$  را ارائه می‌دهیم :

$$v(t) = \dot{s}(t) = \lim_{h \rightarrow 0} \frac{s(t+h) - s(t-h)}{2h} = (\dot{x}(t), \dot{y}(t))$$

$$\dot{s}(t) = \lim_{h \rightarrow 0} \dot{s}_h(t) \rightarrow v_h(t) = \dot{s}_h(t) = \frac{s(t+h) - s(t-h)}{2h} = (\dot{x}_h(t), \dot{y}_h(t))$$

که در رابطه فوق :

$$\dot{x}_h(t) = \frac{x(t+h) - x(t-h)}{2h}$$

$$\dot{y}_h(t) = \frac{y(t+h) - y(t-h)}{2h}$$

در اینجا رابطه تعمیم با تعریف عادی مماس نیز معلوم می‌گردد. در حقیقت مماس حد  $\dot{s}_h(t)$  است وقتی

که  $h \rightarrow 0$ . با استفاده از تعمیم فوق می‌توان  $\theta_h(t)$  و  $\kappa_h(t)$  را نیز تعریف کرد :

$$\kappa_h(t) = \frac{\dot{x}_h(t)\ddot{y}_h(t) - \ddot{x}_h(t)\dot{y}_h(t)}{(\dot{x}_h(t) + \dot{y}_h(t))^{3/2}}$$

$$\theta_h(t) = \arg(v_h(t)) = \arctan\left(\frac{\dot{y}_h(t)}{\dot{x}_h(t)}\right)$$

و نیز تعریف می‌کنیم :

$$l_h(t) = \|v_h(t)\|$$

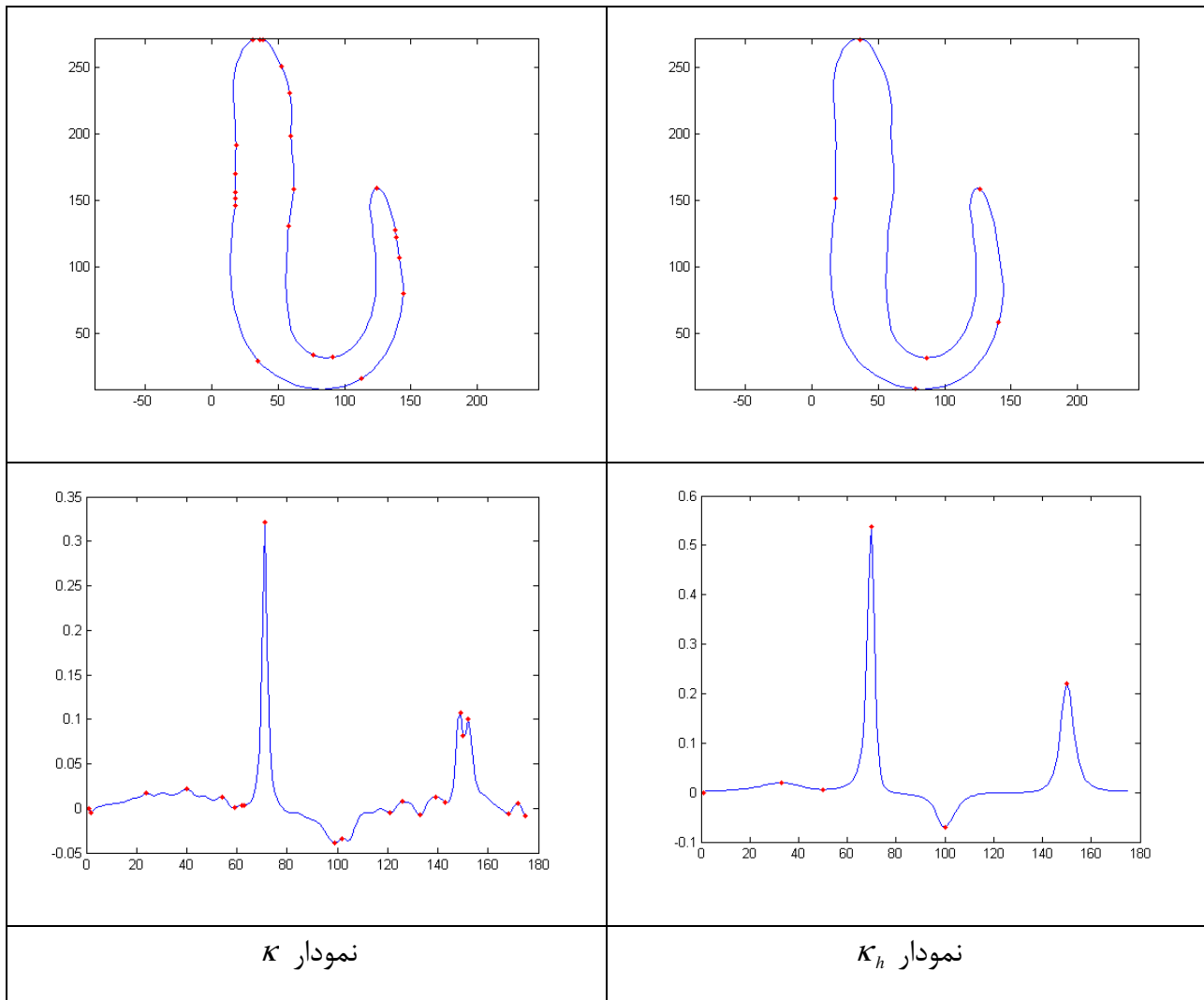
اگر  $\kappa_h$  را بدست آوریم خواهیم دید که رفتار عمومی تری از خم را نسبت به  $\kappa$  نشان می‌دهد.

هر چه  $h$  از 0 فاصله بگیرد،  $\kappa_h$  رفتار عمومی تری از خم در آن نقطه نشان می‌دهد و هر چه  $h$  به 0

نزدیک گردد رفتارهای محلی بیشتر نشان داده می‌شود به طوری که اگر  $h=0$  در این صورت  $\kappa_h = \kappa$ .

در مثال زیر  $h = \frac{s}{10}$  است که  $s$  نشان دهنده طول خم است. برای اینکه بتوانیم با تصاویر در مقیاس‌های

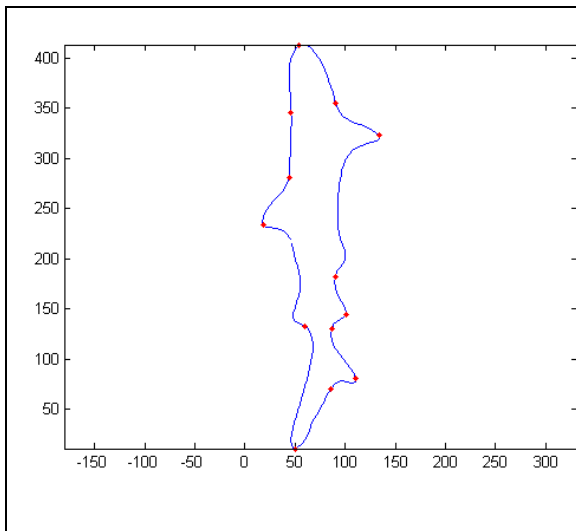
مختلف کار کنیم  $h$  را نسبت به طول خم بیان کرده‌ایم.



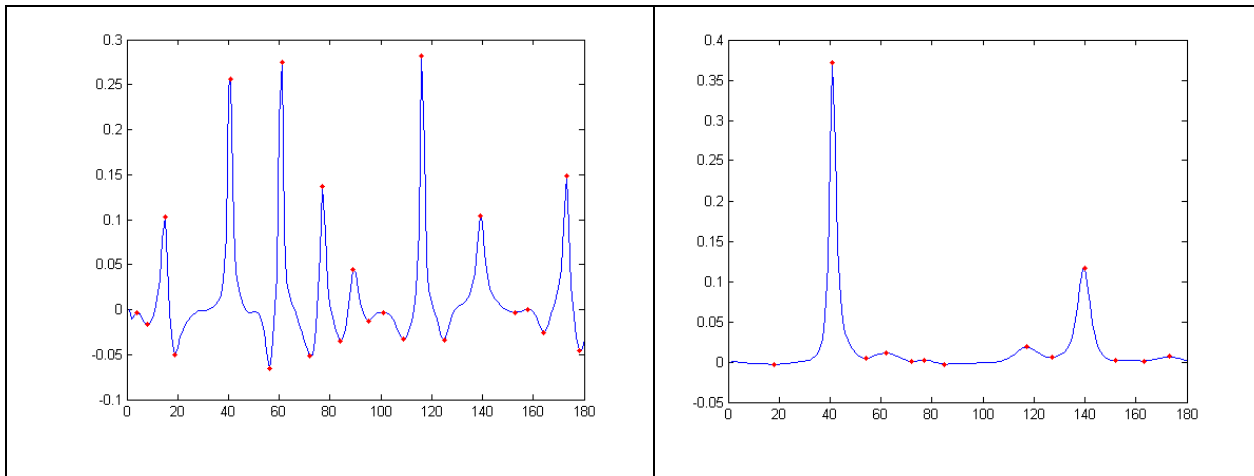
البته هر چه  $h$  بیشتر از 0 دور شود به احتمال حضور  $bias$  در نمایاندن مکان نقاط بحرانی خم افزوده خواهد شد.

اگر در نمودار  $\kappa_h$  دقت کنیم می‌بینیم که تغییرات محلی از نمودار حذف شده و فقط تغییرات  $global$  مشاهده می‌شود. ولی هنوز تغییرات  $global$  به خوبی نمایش داده نشده و در مواردی که تغییر  $global$  به آرامی صورت گیرد، نمی‌توان آن را با دقت بالا فهمید که این موضوع در نمودار  $\kappa$  در  $Figure 1$  مشهود است.

به نظر می‌رسد کارایی زاویه برای نشان دادن تغییرات عمومی مناسب نیست و در صورت نشان دادن تغییرات عمومی، نمی‌توان آن را از تغییر محلی تشخیص داد.



در نمودار سمت چپ، نقاط تغییر  $global$  نیز به عنوان نقاط بحرانی تشخیص داده شده است، ولی نمی‌توان آن را تغییرات محلی تمایز داد ولی در نمودار سمت راست به راحتی می‌توان این نقاط را تمایز داد.



### 3-1- پیشنهاد معیاری جدید برای توصیف خمها

در اینجا با توجه به موارد گفته شده معیاری جدید برای تشخیص نقاط بحرانی و در کنار آن نقاط تغییرات *global* ارائه می شود.

دیدیم که مماس را می توان به وسیله  $(l_h(t), \theta_h(t))$  نشان داد. به جای استفاده از  $\frac{d\theta_h(t)}{ds}$  برای نمایاندن

ویژگی های تصویر، از  $l_h(t)$  برای اینکار استفاده می کنیم.

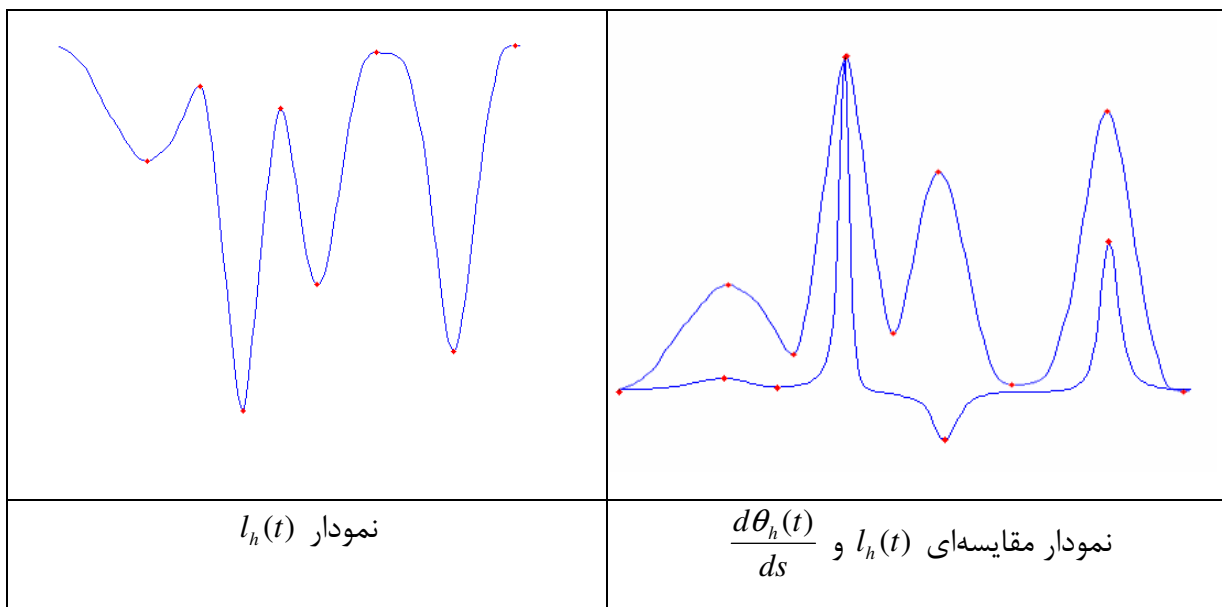
برخی خواص  $l_h(t)$ ، آن را تبدیل به وسیله ای مناسب برای *feature extraction* از خمها می کند. برای

مثال از  $l_h(t)$  تغییرات *global* را می توان بهتر تشخیص داد و به دست آوردن آن نیز محاسبه کمتری

لازم دارد.

در زیر نمودار  $l_h(t)$  و مقایسه آن با نمودار  $\frac{d\theta_h(t)}{ds}$  برای تصاویر قبل آورده شده است. البته در نمودار

بدست آمده تغییرات لازم داده شده تا با نمودار  $\frac{d\theta_h(t)}{ds}$  قابل مقایسه باشد.



همان‌طور که دیده می‌شود، تغییرات *global* بسیار مناسب‌تر از  $\frac{d\theta_h(t)}{ds}$  نشان داده شده است. اگر تعمیم ذکر شده برای مشتق را بررسی کنیم، به رابطه نزدیک محاسبه  $v_h(t)$  و عمل *evolution* پی می‌بریم:

$$v_h(t) = \dot{s}_h(t) = \frac{s(t+h) - s(t-h)}{2h} = \frac{1}{2h} \cdot \int_{t-h}^{t+h} s'(x) dx = \frac{\partial \left( \frac{1}{2h} \cdot \int_{t-h}^{t+h} s(x) dx \right)}{\partial t} = \frac{\partial g(t)}{\partial t}$$

$$g(t) = \frac{1}{2h} \cdot \int_{t-h}^{t+h} s(x) dx$$

یعنی در  $v_h(t)$  مشتق تابع  $g(t)$  را محاسبه کرده‌ایم که تابع  $g(t)$  نوعی *evolution* برای  $s(t)$  محسوب می‌شود. مزیت این‌گونه برخورد با مساله، سرعت بالای محاسباتی آن است.

### 3-2- یک distance function با استفاده از $l_h(t)$

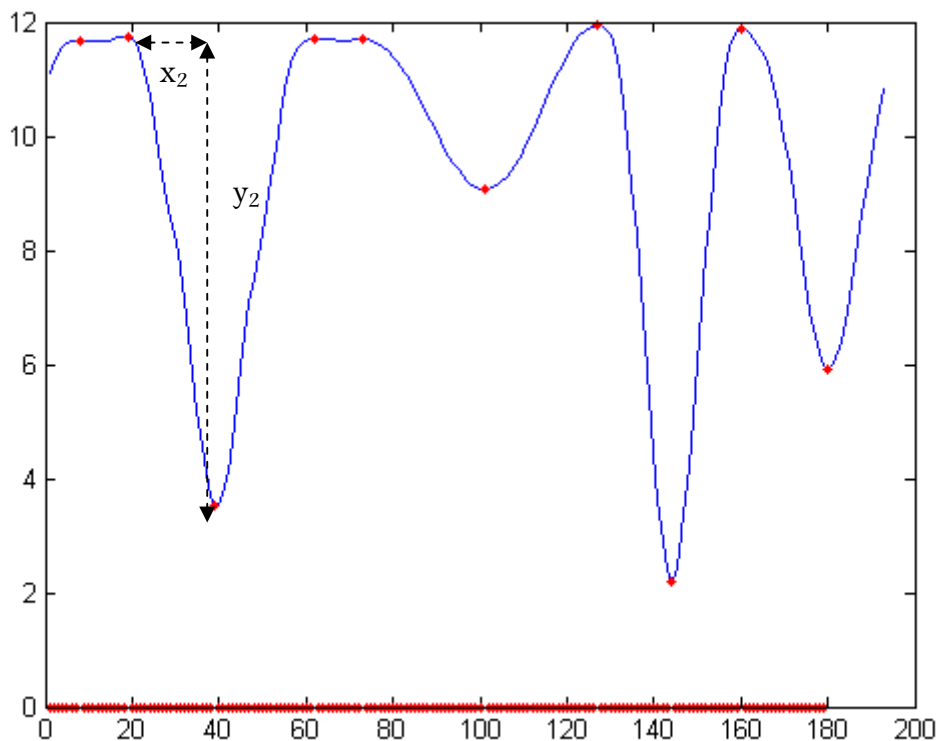
ما با استفاده از  $l_h(t)$  نقاط تغییرات خم را ثبت کرده و از آن برای مقایسه و بازیابی تصاویر استفاده می‌کنیم.

در اینجا الگوریتمی برای مقایسه خم‌ها با استفاده از  $l_h(t)$  ارائه می‌کنیم.

برای اینکار نقاط *maximum* و *minimum* نسبی در  $l_h(t)$  را بدست آورده و فاصله افقی و عمودی هر

نقطه از نقطه قبل را حساب می‌کنیم. فرض کنید دنباله به دست آمده  $(x_1, y_1)(x_2, y_2) \dots (x_n, y_n)$

باشد.



می‌خواهیم الگوریتمی برای *match* کردن این نقاط و پیدا کردن *distance* بین دو دنباله ارائه دهیم.

الگوریتم پیشنهادی اینکار را در  $O(n^2)$  انجام میدهد. با توجه به اینکه حدود  $n$  برای شکل‌های مختلف به

طور متوسط 15 است، انجام *matching* در  $O(n^2)$  بسیار کارا می‌باشد. البته چون نوردای ذکر شده

نسبت به نقطه شروع حساس بود، دوران‌های مختلف یکی از رشته‌های بدست آمده را در نظر می‌گیریم که

با توجه به  $n$  دوران ممکن، مرتبه زمانی اجرای الگوریتم  $O(n^3)$  خواهد بود که با توجه به تعداد متوسط

$n$  کارا خواهد بود.

صورت فرمال مساله بدین ترتیب میشود :

دو رشته  $(x_1, y_1)(x_2, y_2) \dots (x_m, y_m)$  و  $(a_1, b_1)(a_2, b_2) \dots (a_n, b_n)$  را در نظر بگیرید. میخواهیم بزرگترین زیر رشته مشترک را پیدا کنیم به طوری که فاصله بین دو زیر رشته (بر اساس معیار تعریف شده-در اینجا مساحت) کمینه باشد. (می خواهیم نقاط مربوط به دو شکل را چنان متناظر کنیم که مساحت بین دو شکل کمینه شود) برای نقاطی که  $match$  نمی شوند، میتوان نمره منفی در نظر گرفت که آنرا  $penalty(i)$  می نامیم.

اگر فاصله  $(x_1, y_1)(x_2, y_2) \dots (x_j, y_j)$  و  $(a_1, b_1)(a_2, b_2) \dots (a_i, b_i)$  را  $match(i, j)$  بنامیم، داریم :

$$match(i, j) = \min[ (match(i-1, j) + penalty(i)) , (match(i, j-1) + penalty(j)) , (match(i-1, j-1) + penalty(i) + penalty(j)) ]$$

برای اینکه الگوریتم را پیاده سازی کنیم از *dynamic programming* استفاده می کنیم. برای این منظور

جدول دو بعدی درست می کنیم و به ترتیب از  $match(1,1)$  تا  $match(1,m)$ ، از  $match(2,1)$  تا  $match(2,n)$  و ... شروع به پر کردن جدول می کنیم تا  $match(m,n)$  را بدست آوریم. در این صورت اگر  $n > m$  باشد مرتبه اجرای الگوریتم  $O(n^2)$  می شود.

این مقدار را برای دوران های مختلف یکی از رشته ها حساب می کنیم، یعنی :

$$(x_2, y_2) \dots (x_m, y_m)(x_1, y_1)(x_2, y_2) \text{ و } (x_3, y_3) \dots (x_m, y_m)(x_1, y_1)(x_2, y_2) \dots$$

و  $min$  آن ها را حساب کرده و به عنوان فاصله دو بردار در نظر می گیریم.

**نکته :**

می توان با  $l_h(t)$  را به عنوان تابعی از دو متغیر  $h$  و  $t$  برخورد کرده و آن را با  $l(h,t)$  نشان داد. از این

تابع می توان به آسانی  $\theta_h(t)$  را محاسبه کرد :

$$\theta_h(t) = \frac{l(h/2, t-h/2)^2 + l(h/2, t+h/2)^2 - l(h, t)^2}{2l(h/2, t-h/2)l(h/2, t+h/2)}$$

### 3-3- بررسی خواص مشتق تعمیم یافته :

$$f(x) = x \Rightarrow f'_h(x) = 1$$

$$f(x) = x^2 \Rightarrow f'_h(x) = 2x$$

$$f(x) = x^3 \Rightarrow f'_h(x) = 3x^2$$

$$f(x) = \sin(x) \Rightarrow f'_h(x) = \left(\frac{\sin(h)}{h}\right) \cos(x)$$

$$f(x) = \cos(x) \Rightarrow f'_h(x) = -\left(\frac{\sin(h)}{h}\right) \sin(x)$$

$$(f^2(x))'_h = \frac{f^2(x+h) - f^2(x-h)}{2h} = 2f(x+h)f'_h(x) - 2h[f'_h(x)]^2$$

$$[a f(x) + b g(x)]'_h = a f'_h(x) + b g'_h(x)$$

(خاصیت خطی بودن)

$$|f'_h(t)| = \left| \frac{f(t+h) - f(t-h)}{2h} \right|$$

برای یافتن نقاطی که در آن خم ما تغییر ناگهانی داده است باید سعی کنیم  $\min |f'_h(t)|$  را بیابیم.

در ابتدا به بررسی نمودار توابع یک متغیره میپردازیم.

برای مثال نمودار تابع  $y = x^2, y = x^3$  را بررسی می‌کنیم.

$$f(x) = (x, x^2) \Rightarrow |f'_h(t)| = \sqrt{1 + (2h)^2} = \sqrt{1 + 4x^2}$$

برای یافتن  $\min |f'_h(t)|$  کافی است مینیموم  $4x^2$  را بیابیم، که در  $x = 0$  اتفاق می‌افتد. یعنی به ازای هر

$h$  در  $x = 0$  تابع  $f$  تغییر ناگهانی میدهد.

$$f(x) = (x, x^3) \Rightarrow |f'_h(t)| = \sqrt{1 + (3x^2 + h^2)^2}$$

برای یافتن  $\min |f'_h(t)|$  کافی است مینیموم  $3x^2 + h^2$  را پیدا کنیم که به ازای هر  $h$  در  $x = 0$  می‌افتد.

برای توابع  $f(x) = (x, u(x))$  داریم:

$$f(x) = (x, u(x)) \Rightarrow |f'_h(t)| = \sqrt{1 + u'_h(x)^2}$$

در اینجا باید مینیموم  $u'_h(x)^2$  را پیدا کنیم:

چون  $u'_h(x)^2 > 0$  پس باید مینیموم محلی داشته باشد، یعنی باید داشته باشیم:

$$\frac{d}{dx}(u'_h(x)^2) = 2u''_h(x)u'_h(x) = 0$$

یعنی یکی از این دو حالت را داریم:

$$u'_h(x) = 0 \quad \bullet$$

$$u''_h(x) = 0 \quad \bullet$$

در حالت  $h = 0$ ، یعنی وقتی از عملگر مشتق عادی استفاده کنیم، این دو حالت به معنی وجود ماکزیمم یا مینیموم محلی یا نقطه عطف در نمودار ما است که مورد آخر یعنی نقاط عطف همان نقاطی است که در روش مختاریان مورد توجه و استفاده قرار گرفته است. البته در آنجا به خاطر مقابله با مشکل محلی بودن اطلاعات مشتق، از *evolution* استفاده شده است که مستلزم هزینه محاسباتی بالاست، ولی در اینجا با وارد کردن پارامتر  $h$  انتظار نتایجی مشابه داریم.

برای محاسبه  $u''_h(x)$  باید توجه کرد که مشتق اول با حضور پارامتر  $h$  و مشتق دوم به صورت مشتق عادی انجام میشود. برای اینکه بتوانیم بین اینها تمایز قایل شویم از نمادگذاری زیر استفاده می کنیم:

$$u'_h(x) = \frac{d}{d_h x} u(x)$$

$$u''_x(x) = \frac{d}{dx} u'_h(x) = \frac{d^2}{dx d_h x} u(x)$$

در مورد  $\frac{d^2}{dx d_h x} u(x)$  میتوان گفت:

$$\frac{d^2}{dx d_h x} u(x) = \frac{d^2}{d_h x dx} u(x)$$

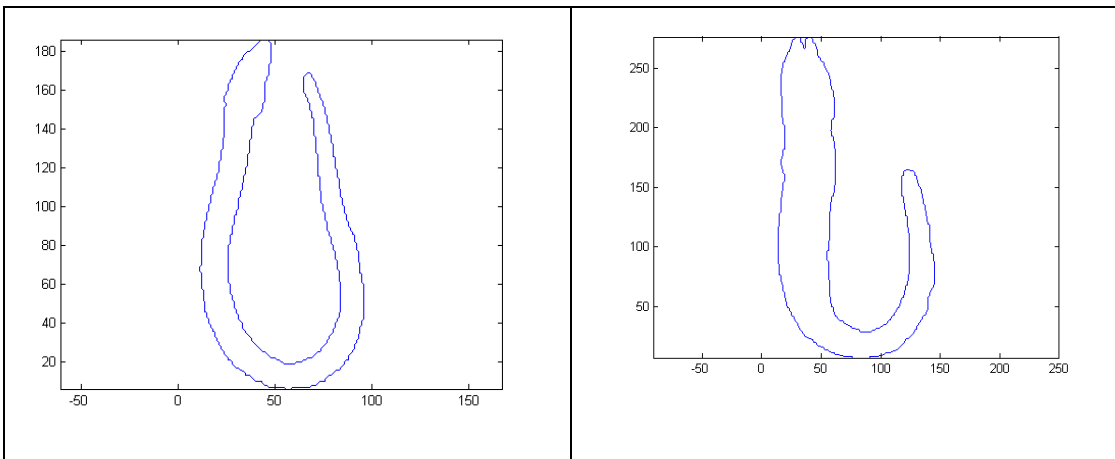
چون

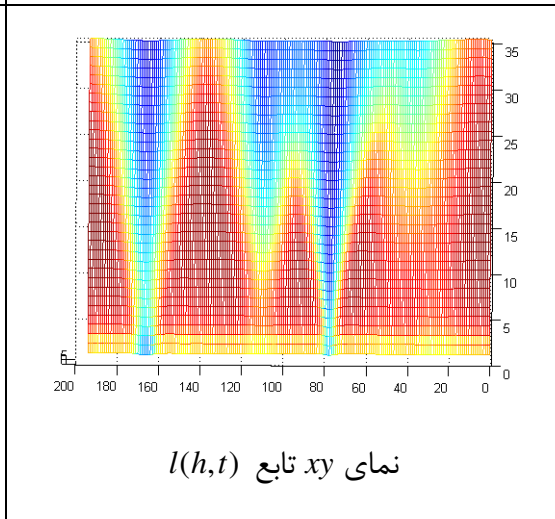
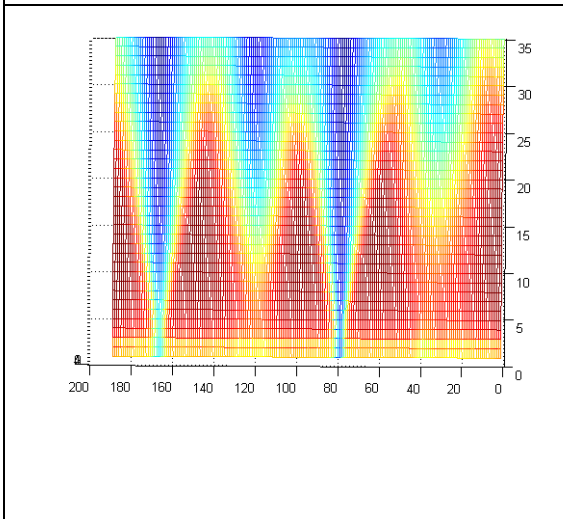
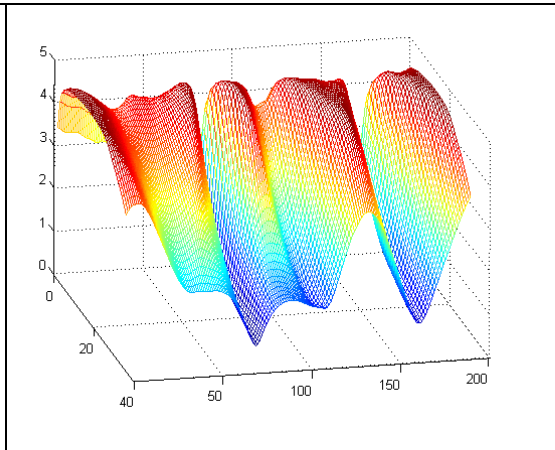
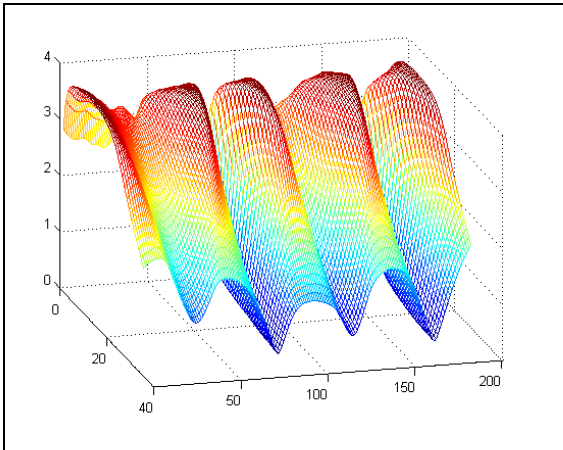
$$\begin{aligned}
\frac{d^2}{dx d_h x} u(x) &= \lim_{m \rightarrow 0} \frac{u'_h(x+m) - u'_h(x-m)}{2m} \\
&= \lim_{m \rightarrow 0} \left[ \frac{\frac{u(x+m+h) - u(x+m-h)}{2h} - \frac{u(x-m+h) - u(x-m-h)}{2h}}{2m} \right] \\
&= \lim_{m \rightarrow 0} \left[ \frac{\frac{u(x+h+m) - u(x+h-m)}{2m} - \frac{u(x-h+m) - u(x-h-m)}{2m}}{2h} \right] \\
&= \left[ \frac{\lim_{m \rightarrow 0} \frac{u(x+h+m) - u(x+h-m)}{2m} - \lim_{m \rightarrow 0} \frac{u(x-h+m) - u(x-h-m)}{2m}}{2h} \right] \\
&= \frac{u'(x+h) - u'(x-h)}{2h} = \frac{d}{d_h x} u'(x) = \frac{d^2}{d_h x dx} u(x)
\end{aligned}$$

پس نماد  $u''_h(x)$  دارای ابهام نمی‌باشد.

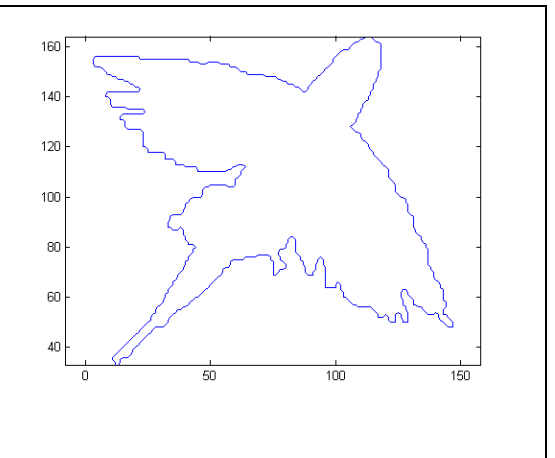
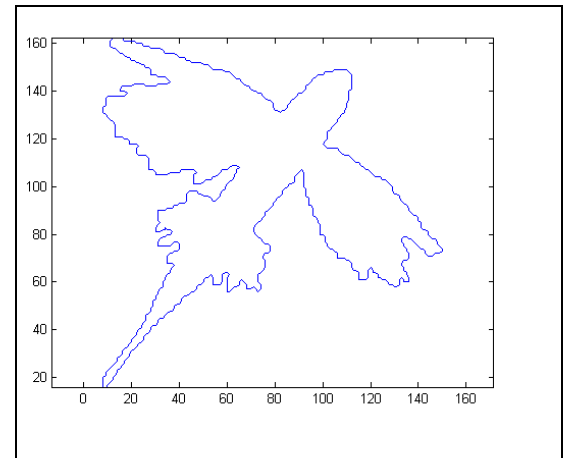
تا به حال تابع  $l_h(t)$  را برای  $h$  ثابت بررسی می‌کردیم، در اینجا برای هر  $t$ ، یک تابع بر حسب تغییرات  $h$  نسبت می‌دهیم. یعنی یک تابع دو متغیره  $l(h, t)$  داریم. این تابع نیز خواص جالبی از خود نشان می‌دهد که در اینجا چند نمونه از آن را بررسی می‌کنیم.

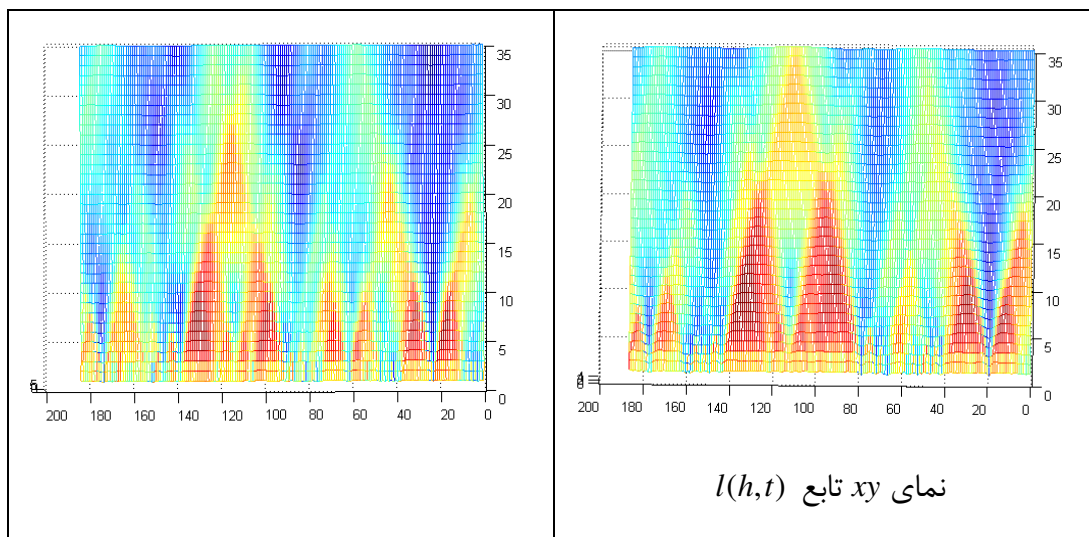
$$g_{t_0}(h) = l(h, t_0) = \frac{f(t_0 + h) - f(t_0 - h)}{2h}$$





نمای  $xy$  تابع  $l(h,t)$

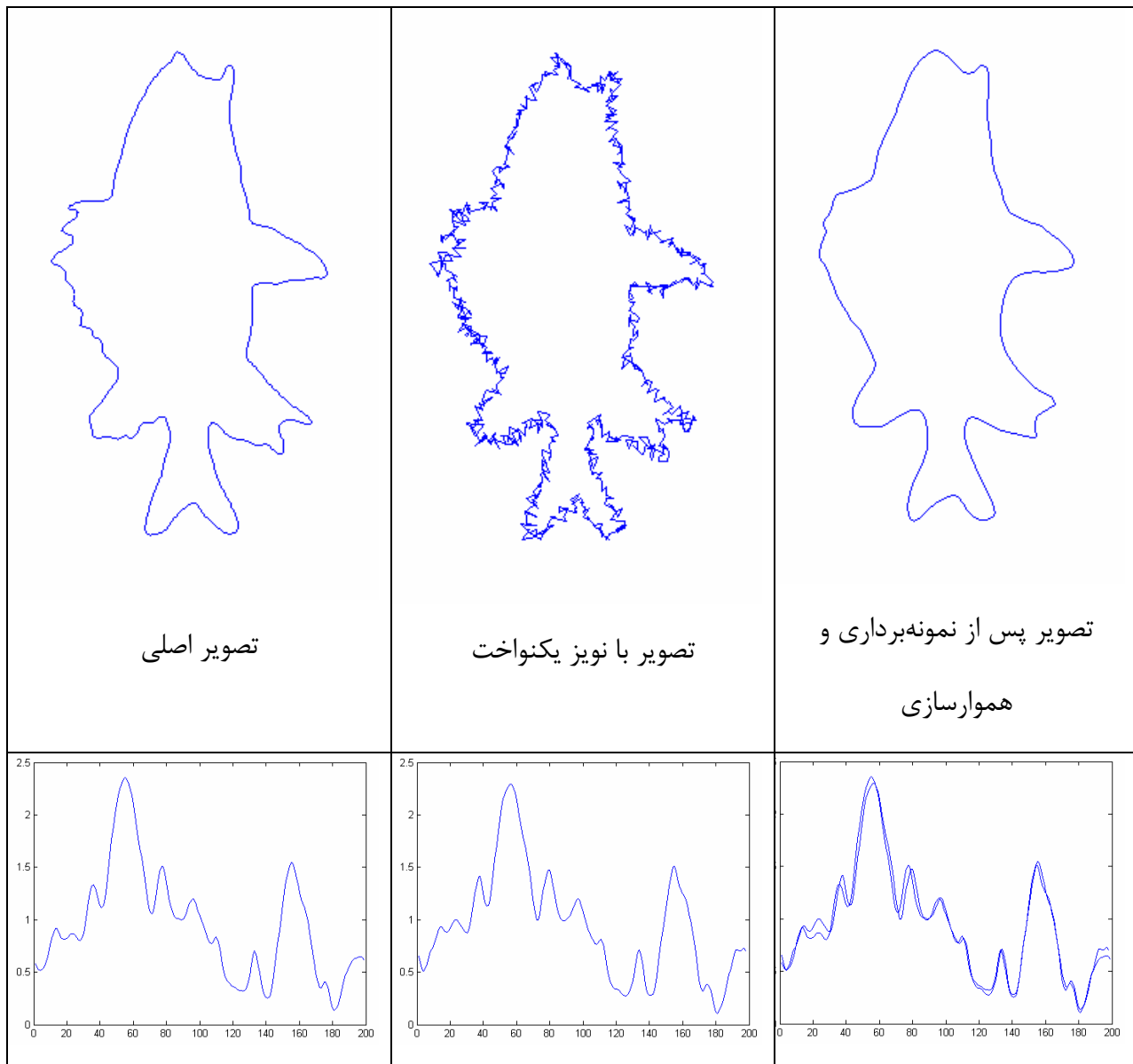




### 3-4- بررسی حساسیت روش پیشنهادی در مقابل خطا :

همانطور که در فلسفه روش ارائه شده گفته شد، هدف نشان دادن تغییرات عمومی و نه تغییرات محلی است، پس نباید روش ارائه شده به نویز حساسیت داشته باشد.

به تصویر نویز یکنواخت اعمال شده و نتیجه برای  $l_h(t)$  و  $\kappa_h$  و  $\kappa$  نشان داده شده است. ( $h = \frac{s}{10}$ )



همان طور که دیده می شود و از ماهیت  $l_h(t)$  نیز آشکار است، چون  $l_h(t)$  بیشتر نشانگر تغییرات عمومی است، نسبت به حضور نویز و خطا در  $shape$  بسیار مقاوم است و بیشتر نویزها در تصاویر لبه در  $l_h(t)$  تاثیر زیادی ندارد.

لازم به ذکر است که انتخاب نقاط هم فاصله نیز یکی از دلایل کاهش تاثیر نویز است.

#### 4. پیش نهاد راه کارهای آتی :

یکی از مواردی که می توان از نظر تئوری بررسی کرد، یکتایی تابع  $l_h(t)$  است. با استفاده از  $l_h(t)$  می توان به طور سلسله مراتبی به طبقه بندی اشکال پرداخت که خواص  $l_h(t)$  اینکار را ممکن می سازد. برای مثال ابتدا با  $h$  های بزرگ از روی خواص عمومی به دسته بندی پرداخت و سپس با کاهش  $h$  مقایسه ها را دقیقتر کرد.

همچنین بررسی دقیقتر تابع  $l(h, t_0)$  و مشتق های آن می تواند خواصی جالب از شکل را به ما نشان دهد.

می توان با جایگزینی تابع  $l_h(t)$  به جای تابع انحنا در الگوریتم های مختلفی که از انحنا استفاده می کنند، اثر و بهبود حاصل را بررسی کرد.

## 5. جمع بندی :

در این پروژه پس از مرور روشهای موجود، راهکار جدیدی برای جستجوی پایگاه داده های تصویر بر اساس شکل اشیا ارائه گشت. در این روش برای استخراج ویژگیهای اشیا در تصاویر از الگوریتم جدیدی جهت توصیف اشیا بر اساس نقاط مرزی آنها استفاده شد. این روش پایدار و دارای حساسیت پایین نسبت به حضور نویز می باشد. همچنین تعمیمی برای عملگر مشتق ارائه شد که بتواند خواص منطقه ای را علاوه بر خواص محلی نشان دهد.

## 6. منابع و مراجع :

- [1] [CYDA 88] Shi\_Kuo Chang, C. W. Yan, Donald C. Dimitroff and Timothy Arndt. *An intelligent image database system. IEEE Trans. on Software Engineering*, 14(5), May 1988.
- [2] [TALMUD] J. V. Sanchez, J. G. Rodriguez, E. Boertjes, M. Kholer, D. Walshe, M. L. Demarie, G. L. Bello, "Retrieval and Filtering methods and techniques", *Talmud Project, PIR 2.3*.
- [3] [CF 79] N. S. Chang and K. S. Fu, *A relational database system for images. Technical report TR-EE 79-28, Purdue University, May 1979*
- [4] [JPE 96] Eakins, J P, "Automatic image content retrieval – are we getting anywhere?" *Proceedings of Third International Conference on Electronic Library and Visual Information Research (ELVIRA3)*, De Montfort University, Milton Keynes, pp 123-135, 1996.
- [5] [JPE 98] Eakins J P, "Techniques for image retrieval" *Library and Information Briefings, in press, 1998*.

- [6] [MKL 97] Babu M. Mehtre, M. Kankanhalli and Wing Foon Lee, *Shape measures for content based image retrieval: A comparison*, *Information Processing & Management*, 33(3), 1997.
- [7] F. Mokhtarian and A. K. Mackworth. *A theory for multiscale, curvature based shape representation for planar curves*. *Transactions on Pattern Analysis and Machine Intelligence*, 14:789–805, 1992.
- [8] Hu, M. K., *Visual Pattern Recognition by Moment Invariants*", *IRE trans. Information Theory*, Vol. 8, pp.179-187.
- [9] Xia, Y., "Skeletonization via the Realization of the Fire Front's Propagation and Extinction in Digital Binary Shapes", *IEEE trans. Pattern Analysis and Machine Intelligence*, Vol. 11 No. 10, pp. 1077-1086, 1989.
- [10] Rak, S. J. and Kolodzy, P. J., "Performance of a Neural Network Based 3-D Object Recognition System", *SPIE Automated Object Recognition*, Vol. 1471, pp. 177-184.
- [11] Wechsler, H. and Zimmerman, G. L., "2D Invariant Object Recognition Using Distributed Associative Memory", *IEEE trans. Pattern Analysis and Machine Intelligence*, Vol. 10, No. 6, pp. 811-821, 1988

- [12] Davies, E. R., "Machine Vision - Theory, Algorithms, Practicalities", Academic Press, 1990.
- [13] M. Bober, "MPEG-7 Visual Shape Descriptors", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 11, no. 6, pp. 716-719, June 2001.
- [14] F. Mokhtarian, and R. Suomela, "Robust Image Corner Detection Through Curvature Scale Space", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1376-1381, December 1998.
- [15] A. Del Bimbo, *Visual Information Retrieval*, 270 pages, Morgan Kaufmann Publishers, San Francisco, California, 1999.
- [16] M. Tuceryan, and A.K. Jain, *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, by C.H. Chen, L.F. Pau, and P.S.P Wang (eds.), pp. 207-248, World Scientific Publishing Co., 1998.
- [17] P. Reijonen, "Shape Analysis for Content-Based Image Retrieval", *Master of Science Thesis, Tampere University of Technology*, p. 71, October 2001.
- [18] جعفر محمدی، طرح مطالعاتی نرم افزارهای متن باز جهت موتور جستجوی چندرسانه‌ای، پژوهشگاه ارتباطات و فناوری اطلاعات ایران

ضمیمه 1 : کد منبع پیاده‌سازی الگوریتم

Curvature function:

```

clear all;
clc;
%%%%%%%%%%
pointNumber = 100;
input = load('kk6.c');
n = length(input);
s = 0;
input(n+1,:) = input(1,:);
for i=1:n
    d = (input(i,1)-input(i+1,1))^2 + (input(i,2)-
input(i+1,2))^2;
    s = s+sqrt(d);
end
d = s/pointNumber;
%%%%%%%%%%
% points => selected point's array
t = 0;
points(1,1) = input(1,1);
points(1,2) = input(1,2);
pCounter = 1;
for i=1:n
    disToNextPoint = sqrt((input(i,1)-input(mod(i,n)+1,1))^2 +
(input(i,2)-input(mod(i,n)+1,2))^2);
    t = t+disToNextPoint;
    if (t>d)
        pCounter = pCounter+1;
        points(pCounter,1) = input(i,1);
        points(pCounter,2) = input(i,2);
        t = 0;
    end
end
n = length(points)
for j=1:5
    for i=1:n
        newPoints(i,1)= (points(mod(i-
2,n)+1,1)+points(mod(i,n)+1,1)+points(i,1))/3;
        newPoints(i,2)= (points(mod(i-
2,n)+1,2)+points(mod(i,n)+1,2)+points(i,2))/3;
    end
    points(:,:)=newPoints(:,:);
end
figure;
points(n+1,1) = points(1,1);
points(n+1,2) = points(1,2);
plot(points(:,1),points(:,2));
axis equal;
%%%%%%%%%%
%%%%%dfx
for i=1:n
    dfx(i) = points(mod(i,n)+1,1)-points(mod(i-2,n)+1,1);
    dfy(i) = points(mod(i,n)+1,2)-points(mod(i-2,n)+1,2);
end
for i=2:n
    d2fx(i) = dfx(mod(i,n)+1)-dfx(mod(i-2,n)+1);
    d2fy(i) = dfy(mod(i,n)+1)-dfy(mod(i-2,n)+1);
end
%%%%%%%%%%

```

```

%%%%%%%%calc kappa
for i=1:n
    k(i)=(dfx(i)*d2fy(i)-
dfy(i)*d2fx(i))/((dfx(i)^2+dfy(i)^2)^1.5);
end
%%%%%%%%
figure;
plot((1:n),k(:)); hold on;

```

### *Mokhtarian CSS*

```

clear all;
%close all;
clc;
%%%%%%%%
pointNumber = 200;
input = load('kk1.c');
n = length(input);
s = 0;
for i=1:n-1
    d = (input(i,1)-input(i+1,1))^2 + (input(i,2)-
input(i+1,2))^2;
    s = s+sqrt(d);
end
d = s/pointNumber;
%%%%%%%%
% points => selected point's array
t = 0;
points(1,1) = input(1,1);
points(1,2) = input(1,2);
pCounter = 1;
for i=1:n-1
    disToNextPoint = sqrt((input(i,1)-input(i+1,1))^2 +
(input(i,2)-input(i+1,2))^2);
    t = t+disToNextPoint;
    if (t>d)
        pCounter = pCounter+1;
        points(pCounter,1) = input(i,1);
        points(pCounter,2) = input(i,2);
        t = 0;
    end
end
end
n = length(points)
%%%%%%%%
%%%%%%%%rotate teta degree
% teta = pi/4;
% for i=1:n
%     x=points(i,1);
%     y=points(i,2);
%     points(i,1) = int16(x*cos(teta)+y*sin(teta));
%     points(i,2) = int16(-x*sin(teta)+y*cos(teta));
% end
% for i=1:n
%     newpoints(i,1) = points(mod(i+108,n)+1,1);
%     newpoints(i,2) = points(mod(i+108,n)+1,2);
% end

```

```

% points(1:n,1:2) = newpoints(1:n,1:2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
points(n+1,:)=points(1,:);
figure;
plot(points(1:n+1,1),points(1:n+1,2));hold on;
axis equal;
% plot(points(1:n,1),points(1:n,2),'--rs');
figure;
itrNumber = 0;
axis equal;
for j=1:400
    for i=1:n
        newPoints(i,1)= (points(mod(i-
2,n)+1,1)+points(mod(i,n)+1,1)+points(i,1))/3;
        newPoints(i,2)= (points(mod(i-
2,n)+1,2)+points(mod(i,n)+1,2)+points(i,2))/3;
    end
    points(1:n,1)=newPoints(1:n,1);
    points(1:n,2)=newPoints(1:n,2);
    points(n+1,:)=points(1,:);
%     if (mod(j,5)==0)
%         plot(points(1:n+1,1),points(1:n+1,2),'r');hold on;
%     end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%dfx
for i=2:n
    dfx(i) = points(i+1,1)-points(i-1,1);
    dfy(i) = points(i+1,2)-points(i-1,2);
end
dfx(1) = points(2,1)-points(n,1);
dfy(1) = points(2,2)-points(n,2);
dfx(n+1) = dfx(1);
dfy(n+1) = dfy(1);
for i=2:n
    d2fx(i) = dfx(i+1)-dfx(i-1);
    d2fy(i) = dfy(i+1)-dfy(i-1);
end
d2fx(1) = dfx(2)-dfx(n);
d2fy(1) = dfy(2)-dfy(n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%calc kappa
for i=1:n
    k(i)=(dfx(i)*d2fy(i)-
dfy(i)*d2fx(i))/((dfx(i)^2+dfy(i)^2)^1.5)*100;
end

    for i=1:n
        if ((k(i)>0 && k(mod(i,n)+1)<0) || (k(i)<0 &&
k(mod(i,n)+1)>0))
            plot(i,itrNumber);
            hold on;
        end
    end
    itrNumber = itrNumber + 1;
end

```

```

%
plot(points(1:n+1,1),points(1:n+1,2),'k','LineWidth',1.5);hold
on;
% figure;
% plot(points(1:n+1,1),points(1:n+1,2));
% axis equal;
return;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%dfx
for i=2:n
    dfx(i) = points(i+1,1)-points(i-1,1);
    dfy(i) = points(i+1,2)-points(i-1,2);
end
dfx(1) = points(2,1)-points(n,1);
dfy(1) = points(2,2)-points(n,2);
dfx(n+1) = dfx(1);
dfy(n+1) = dfy(1);
for i=2:n
    d2fx(i) = dfx(i+1)-dfx(i-1);
    d2fy(i) = dfy(i+1)-dfy(i-1);
end
d2fx(1) = dfx(2)-dfx(n);
d2fy(1) = dfy(2)-dfy(n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%calc kappa
for i=1:n
    k(i)=(dfx(i)*d2fy(i)-
dfy(i)*d2fx(i))/((dfx(i)^2+dfy(i)^2)^1.5)*100;
end

for i=1:n
    if (k(i)>0 && k(mod(i,n)+1)<0)
        plot(i,itrNumber);
        hold on;
    end
end
itrNumber = itrNumber + 1;

threshold = 0.6*(max(k)-min(k));
thresholdedK = zeros(n,2);
for i=1:n
    if(abs(k(i))>(max(k)-threshold))
        thresholdedK(i,1) = points(i,1);
        thresholdedK(i,2) = points(i,2);
    end
end
plot(thresholdedK(1:n,1),thresholdedK(1:n,2),'*');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% function [] = linepredict(fname)
% clear all;
% close all;
% clc;
%%%%%%%%%%
pointNumber = 200;
% input = load(fname);
input = load('kk6.c');
n = length(input);
s = 0;
for i=1:n-1
    d = (input(i,1)-input(i+1,1))^2 + (input(i,2)-
input(i+1,2))^2;
    s = s+sqrt(d);
end
d = s/pointNumber;
%%%%%%%%%%
% points => selected point's array
t = 0;
points(1,1) = input(1,1);
points(1,2) = input(1,2);
pCounter = 1;
for i=1:n-1
    disToNextPoint = sqrt((input(i,1)-input(i+1,1))^2 +
(input(i,2)-input(i+1,2))^2);
    t = t+disToNextPoint;
    if (t>d)
        pCounter = pCounter+1;
        points(pCounter,1) = input(i,1);
        points(pCounter,2) = input(i,2);
        t = 0;
    end
end
end
%%%%%%%%%%smooth
n = length(points);
for j=1:8
    for i=1:n
        newPoints(i,1)= (points(mod(i-
2,n)+1,1)+points(mod(i,n)+1,1)+points(i,1))/3;
        newPoints(i,2)= (points(mod(i-
2,n)+1,2)+points(mod(i,n)+1,2)+points(i,2))/3;
    end
    points(1:n,:) = newPoints(1:n,:);
end
points(n+1,:)=points(1,:);
%%%%%%%%%%
figure;
subplot(1,2,1);
plot(points(1:n+1,1),points(1:n+1,2));
axis equal;
%%%%%%%%%%
%%%%%dfx
dfx = zeros(n+1);
dfy = zeros(n+1);
for i=1:n
    dfx(i) = points(mod(i,n)+2,1)-points(mod(i-3,n)+1,1);
    dfy(i) = points(mod(i,n)+2,2)-points(mod(i-3,n)+1,2);

```

```

end
dfx(n+1) = dfx(1);
dfy(n+1) = dfy(1);
d2fx = zeros(n);
d2fy = zeros(n);
for i=2:n
    d2fx(i) = dfx(i+1)-dfx(i-1);
    d2fy(i) = dfy(i+1)-dfy(i-1);
end
d2fx(1) = dfx(2)-dfx(n);
d2fy(1) = dfy(2)-dfy(n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%calc kappa
k = zeros(n+1);
for i=1:n
    k(i)=(dfx(i)*d2fy(i)-
dfy(i)*d2fx(i))/((dfx(i)^2+dfy(i)^2)^1.5)*100;
end
k(n+1) = k(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%predict
len = 1;
ptList = zeros(n,3);
for i=1:n
    if(abs(k(i))>1)
        ptList(len,1) = points(i,1);
        ptList(len,2) = points(i,2);
        ptList(len,3) = k(i);
        len = len+1;
    end
end
ptList(len,:) = ptList(1,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(1,2,2);
plot(ptList(1:len,1),ptList(1:len,2)); hold on;
plot(ptList(1:len,1),ptList(1:len,2),'.');
axis equal;
figure;
plot(k);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

### 3d Curvature Function

```
clear all;
close all;
clc;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pointNumber = 200;
input = load('kk6.c');
n = length(input);
s = 0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
input(n+1,:) = input(1,:);
for i=1:n
    d = (input(i,1)-input(i+1,1))^2 + (input(i,2)-
input(i+1,2))^2;
    s = s+sqrt(d);
end
d = s/pointNumber;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% points => selected point's array
t = 0;
points(1,1) = input(1,1);
points(1,2) = input(1,2);
pCounter = 1;
for i=1:n-1
    disToNextPoint = sqrt((input(i,1)-input(i+1,1))^2 +
(input(i,2)-input(i+1,2))^2);
    t = t+disToNextPoint;
    if (t>d)
        pCounter = pCounter+1;
        points(pCounter,1) = input(i,1);
        points(pCounter,2) = input(i,2);
        t = 0;
    end
end
end
% points(pCounter+1,1:2) = points(1,1:2)
n = length(points)
figure;
points(n+1,1) = points(1,1);
points(n+1,2) = points(1,2);
plot(points(1:n,1),points(1:n,2));
axis equal;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%dfx
for i=2:n
    dfx(i) = points(i+1,1)-points(i-1,1);
    dfy(i) = points(i+1,2)-points(i-1,2);
end
dfx(1) = points(2,1)-points(n,1);
dfy(1) = points(2,2)-points(n,2);
dfx(n+1) = dfx(1);
dfy(n+1) = dfy(1);
for i=2:n
    d2fx(i) = dfx(i+1)-dfx(i-1);
    d2fy(i) = dfy(i+1)-dfy(i-1);
end
end
d2fx(1) = dfx(2)-dfx(n);
```

```

d2fy(1) = dfy(2)-dfy(n);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%calc kappa
for i=1:n
    k(i)=(dfx(i)*d2fy(i)-
dfy(i)*d2fx(i))/((dfx(i)^2+dfy(i)^2)^1.5)*100;
end
k(n+1) = k(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%threosholding
pthreshold = 0.1*max(k);
nthreshold = 0.1*min(k);
pthresholdedK = zeros(n,2);
ptList(1,1) = 0;
ptList(1,2) = 0;
ptList(1,3) = 0;
len = 0;
for i=1:n
    if(k(i)>2)
        len = len+1;
        pthresholdedK(i,1) = points(i,1);
        pthresholdedK(i,2) = points(i,2);
        ptList(len,1) = points(i,1);
        ptList(len,2) = points(i,2);
        ptList(len,3) = k(i);
    end
end
nthresholdedK = zeros(n,2);
len = 0;
for i=1:n
    if(k(i)<-2)
        len = len+1;
        nthresholdedK(i,1) = points(i,1);
        nthresholdedK(i,2) = points(i,2);
        ntList(len,1) = points(i,1);
        ntList(len,2) = points(i,2);
        ntList(len,3) = k(i);
    end
end
end
if(exist('ptList'))
    len = length(ptList)+1;
    ptList(len,:) = ptList(1,:);
end
if(exist('ntList'))
    len = length(ntList)+1;
    ntList(len,:) = ntList(1,:);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
plot(pthresholdedK(1:n,1),pthresholdedK(1:n,2),'rs');
plot(nthresholdedK(1:n,1),nthresholdedK(1:n,2),'gs');
axis equal;
axis vis3d;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
ptlen = length(ptList);
ntlen = length(ntList);
plot3(ptList(2:ptlen,1),ptList(2:ptlen,2),ptList(2:ptlen,3),'r
');

```

```

fill3(ptList(2:ptlen,1),ptList(2:ptlen,2),ptList(2:ptlen,3),'r
');
plot3(ntList(2:ntlen,1),ntList(2:ntlen,2),ntList(2:ntlen,3),'g
');
fill3(ntList(2:ntlen,1),ntList(2:ntlen,2),ntList(2:ntlen,3),'g
');
plot(ptList(2:ptlen,1),ptList(2:ptlen,2),'r');
fill(ptList(2:ptlen,1),ptList(2:ptlen,2),'r');
fill(points(:,1),points(:,2),'b');
plot(ntList(2:ntlen,1),ntList(2:ntlen,2),'g');
fill(ntList(2:ntlen,1),ntList(2:ntlen,2),'g');

```

*Classification with l function and new derivative*

```

clear all;
clc;
chickens = 446;
for k=1:chickens
    fname = sprintf('chicken\\c_%d.txt',k);
    t = diagonalpoint(fname,9);
    s = size(t);
    l1(k,1:s(1),1:2) = t;
end
v = double(zeros(chickens));
for j=1:1
    l1=[];
    l1(:,1) = nonzeros(l1(j,:),1));
    l1(:,2) = nonzeros(l1(j,1:length(l1),2));
    for i=1:chickens
        if(j~=i && v(i,j)==0)
            l2 = [];
            l2(:,1) = nonzeros(l1(i,:),1));
            l2(:,2) = l1(i,1:length(l2),2);
            d1 = match2(l1,l2);
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            l2 = l2(length(l2):-1:1,:);
            d2 = match2(l1,l2);
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            v(i,j) = min(d1,d2);
        end
    end
end
End

```

### *Calculate described feature vector*

```
function [l] = diagonalpoint(fname, stepDiv);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pointNumber = 100;
input = load(fname);
n = length(input);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s1 = 0;
for i=1:n
    d = (input(i,1)-input(mod(i,n)+1,1))^2 + (input(i,2)-
input(mod(i,n)+1,2))^2;
    s1 = s1+sqrt(d);
end
d = s1/pointNumber;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = 0;
points(1,1) = input(1,1);
points(1,2) = input(1,2);
pCounter = 1;
for i=1:n
    disToNextPoint = sqrt((input(i,1)-input(mod(i,n)+1,1))^2 +
(input(i,2)-input(mod(i,n)+1,2))^2);
    t = t+disToNextPoint;
    if (t>=(d-0.4))
        pCounter = pCounter+1;
        points(pCounter,1) = input(i,1);
        points(pCounter,2) = input(i,2);
        t = 0;
    end
end
n = length(points);
for j=1:5
    for i=1:n
        newPoints(i,1)= (points(mod(i-
2,n)+1,1)+points(mod(i,n)+1,1)+points(i,1))/3;
        newPoints(i,2)= (points(mod(i-
2,n)+1,2)+points(mod(i,n)+1,2)+points(i,2))/3;
    end
    points(:, :) = newPoints(:, :);
end
n = n - mod(n, stepDiv);
m = n/stepDiv;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:n
    x = points(mod(i+m-1,n)+1,1)-points(mod(i-m-1,n)+1,1);
    y = points(mod(i+m-1,n)+1,2)-points(mod(i-m-1,n)+1,2);
    dist(i) = sqrt(x^2+y^2);
    x = points(mod(i+m-1,n)+1,1)-points(mod(i-1,n)+1,1);
    y = points(mod(i+m-1,n)+1,2)-points(mod(i-1,n)+1,2);
    dd(i) = sqrt(x^2+y^2);
end
dist = dist/mean(dist);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c = 0;
```

```

for i = 1:n
    if ((dist(mod(i,n)+1)>=dist(i) && dist(mod(i-
2,n)+1)>=dist(i)) || (dist(mod(i,n)+1)<=dist(i) && dist(mod(i-
2,n)+1)<=dist(i)))
        c = c + 1;
        lis(c,1) = i;
        lis(c,2) = dist(i);
    end
end
cc = 1;
for i = 1:c
    if ((cc>=1 && abs(lis(mod(i-2,c)+1,1)-lis(i,1))>=4) ||
(cc>1 &&abs(lis(i,1)-lis2(cc-1,1))>=3))
        lis2(cc,:) = lis(i,:);
        cc = cc+1;
    end
end
lis2(cc,:) = lis2(1,:);
l = zeros(cc-1,2);
for i=1:cc-1
    l(i,1) = lis2(i+1,2)-lis2(i,2); %h
    l(i,2) = lis2(i+1,1)-lis2(i,1); %a
end
l(cc-1,2) = n + l(cc-1,2);
l(:,2) = l(:,2)/n;

```

### *Match function match2.m*

```

function [mvalue] = match2(l1,l2);
n = length(l2);
l2(n+1:2*n,:) = l2(1:n,:);
for i=1:n
    % m(i) = match3(l1,l2(i:n+i-1));
    m(i) = match3(l1(:,1),l1(:,2),l2(i:n+i-1,1),l2(i:n+i-
1,2));
end
mvalue = min(m);

```

```

function [mvalue] = match3(h1,l1,h2,l2);
c1 = length(l1);
c2 = length(l2);
l = zeros(c1+1,c2+1);

l(1,1) = 0;
for i=2:c1+1
    l(i,1) = l(i-1,1) + abs(l1(i-1)*h1(i-1));
end
for i=2:c2+1
    l(1,i) = l(1,i-1) + abs(l2(i-1)*h2(i-1));
end
for i=1:c1
    for j=1:c2
        s1 = l(i,j+1) + abs(l1(i)*h1(i));
        s2 = l(i+1,j) + abs(l2(j)*h2(j));
        s3 = l(i,j) + abs((l1(i)-l2(j))*(h1(i)-h2(j)))/3;
        l(i+1,j+1) = min([s1 s2 s3]);
    end
end
end
mvalue = l(c1+1,c2+1);

```

### *Linear prediction with l function*

```

clear all;
close all;
clc;
fname = 'brd\kk30.c';
pl = 1;
stepDiv = 8;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pointNumber = 80;
input = load(fname);
n = length(input);
input2 = input;
input2(n+1:n+n,:) = input(1:n,:);
input(1:n,:) = input2(n+269:-1:270,:);
n1 = n;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
s1 = 0;
pointDist(1) = 0.0;
for i=1:n
    d = (input(i,1)-input(mod(i,n)+1,1))^2 + (input(i,2)-
input(mod(i,n)+1,2))^2;
    s1 = s1+sqrt(d);
    pointDist(i+1) = s1;
end
d = s1/pointNumber;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t = 0;
points(1,1) = input(1,1);
points(1,2) = input(1,2);
pCounter = 1;

```

```

for i=1:n
    disToNextPoint = sqrt((input(i,1)-input(mod(i,n)+1,1))^2 +
(input(i,2)-input(mod(i,n)+1,2))^2);
    t = t+disToNextPoint;
    if (t>=(d-0.4))
        points(pCounter,1) = input(i,1);
        points(pCounter,2) = input(i,2);
        ind(pCounter) = i;
        t = 0;
        pCounter = pCounter+1;
    end
end
n = length(points);
for j=1:3
    for i=1:n
        newPoints(i,1)= (points(mod(i-
2,n)+1,1)+points(mod(i,n)+1,1)+points(i,1))/3;
        newPoints(i,2)= (points(mod(i-
2,n)+1,2)+points(mod(i,n)+1,2)+points(i,2))/3;
    end
    points(:,:)=newPoints(:,:);
end
points(n+1,:) = points(1,:);
figure;
subplot(1,2,1);
plot(points(:,1),points(:,2)); hold on;
plot(points(1:n,1),points(1:n,2),'.r');
plot(points(1:20:n,1),points(1:20:n,2),'.k');
axis equal;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
hold on;
curveLength = round(s1);
for m = 71:71
    stepDiv = j;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for i=1:n
        a = (points(mod(i-m-1,n)+1,1)-points(mod(i+m-
1,n)+1,1))^2+(points(mod(i-m-1,n)+1,2)-points(mod(i+m-
1,n)+1,2))^2;
        dist(i) = sqrt(a);
    end
    dist = dist/round(s1);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j=1:1
        for i=1:n
            newdist(i)= (dist(mod(i-
2,n)+1)+dist(mod(i,n)+1)+dist(i))/3;
        end
        dist=newdist;
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    c = 1;
    for i=1:n
        if ((dist(mod(i,n)+1)>=dist(i) && dist(mod(i-
2,n)+1)>=dist(i)) || (dist(mod(i,n)+1)<=dist(i) && dist(mod(i-
2,n)+1)<=dist(i)))
            crit(c,:) = points(i,:);

```

```

        c = c+1;
    end
end
crit(c,:) = crit(1,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(1,2,2);
plot(crit(:,1),crit(:,2));
axis equal;
hold on;
plot(crit(:,1),crit(:,2),'.r');
length(crit)
figure;
plot(dist);
end

```