دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

گروه مستقل مهندسی رباتیک

تمرین پنجم درس بینایی ماشین

استاد درس:

دکتر صفابخش

تدریس‌یار:

مهندس مجد

نام دانشجو:

نوید خزاعی

۹۲۱۳۵۰۰۸

تیر ۱۳۹۳

فهرست مطالب

۱ بخش اول

۱.۱ روش‌های ارایه‌ی اشکال

پرسش: در درس روش های متعددی برای ارایه اشکال توضیح داده شد. از بین این روش ها کدام یک از آن ها توسط OpenCV پیاده سازی شده اند؟ چه روش های دیگری در OpenCV برای ارایه اشکال پیشنهاد شده است؟

پاسخ: از روش‌هایی که در درس مربوطه آمده‌اند، روش‌های زیر در OpenCV پیاده‌سازی شده‌اند:

- کد زنجیره‌ای فری‌من[۱]: برای محاسبه‌ی آن، لازم است پیرامون شی را داشته باشیم که در تمرینات قبلی، چگونگی به‌دست آوردن آن توسط کانتورها را دیدیم.

```
CvSeq* cvApproxChains(CvSeq* src_seq, CvMemStorage* storage,
int method=CV_CHAIN_APPROX_SIMPLE, double parameter=0,
                int minimal_perimeter=0, int recursive=0 )
```

  - src_sec : اشاره‌گری به زنجیره‌ی فریمن ورودی است که می‌تواند به زنجیره‌های دیگری نیز اشاره‌کند. در اصل باید کانتور یافت‌شده از شکل را به عنوان ورودی بدهیم.

    راه دیگری نیز پیاده‌سازی شده‌است که نیازی به این تابع ندارد و فقط با تابع پیدا کردن کانتورها، کد را ارائه می‌دهد. دقت کنید که باید در فراخوانی تابع پیداکردن کانتورها به این شکل عمل کنیم:

```
cvFindContours(image, storage, (CvSeq**)&chain, sizeof(*chain),
                CV_RETR_LIST, CV_CHAIN_CODE);
```

    که کانتور خروجی chain را که از نوع CvChain* تعریف شده‌است، به نوع مناسب برای استفاده در تابع cvFindContours تبدیل کنیم. همچنین روش محاسبه‌ی کانتور باید با مقدار CV_CHAIN_CODE مشخص شود تا خروجی یک کد زنجیره‌ی فریمن باشد. روش نمایش این کد در حل قسمت بعدی از همین تمرین پیاده‌سازی و ارایه شده‌است.

  - storage : فضای لازم برای ذخیره‌سازی موقت محاسبات مربوط به خطوط.

[۱] Freeman Chain Code

- method : روش تخمین که مانند پارامتری با همین نام در تابع cvFindContours است و در تمرینات قبلی توضیح‌داده شده‌است.
- parameter : این آرگومان مورد استفاده قرار نگرفته‌است.
- minimal_perimeter : مشخص می‌کند کانتورهایی که محیطشان از این اندازه کم‌تر است، دخالت داده‌نشوند.
- recursive : در صورت یک بودن، به صورت بازگشتی تمام کانتورهای سلسله‌مراتبی را نیز محاسبه می‌کند. در غیر این صورت کانتور اولیه‌ی بیرونی انتخاب می‌شود.

- توصیف‌گرهای فوریه (تبدیل فوریه‌ی گسسته):

void dft(InputArray src, OutputArray dst, int flags=0, int nonzeroRows=0)

- src : آرایه‌ی ورودی.
- dst : آرایه‌ی خروجی.
- flags :
  - DFT_INVERSE : به‌جای تبدیل معمولی که رو به جلو است، تبدیل معکوس فوریه را حساب می‌کند.
  - DFT_SCALE : مقیاس‌بندی جواب نهایی را مشخص می‌کند، به این ترتیب که حاصل را بر طول آرایه تقسیم می‌کند.
  - DFT_ROWS : برای هر سطر از ورودی، تبدیل را جداگانه حساب می‌کند (برای کم‌کردن سربار محاسباتی در تبدیل‌های سه‌بعدی و بیشتر کاربرد دارد).
  - DFT_COMPLEX_OUTPUT : درصورت استفاده از این پرچم، پردازش بر روی ورودی حقیقی انجام خواهد شد و خروجی با اندازه‌ی اصلی و جواب‌های مختلط خواهد بود. در حالت عادی این آرایه، به اندازه‌ی ورودی درخواهد آمد و شامل جواب‌های حقیقی است.
  - DFT_REAL_OUTPUT : تبدیل معکوس بر روی ورودی مختلط انجام می‌دهد. در حالت عادی جواب آرایه‌ای با اندازه‌ی ورودی و جواب‌های مختلط خواهد بود. در صورت استفاده از این پرچم حاصل به صورت حقیقی محاسبه خواهدشد.

- nonzeroRows : وقتی صفر نیست، تابع فرض می‌کند تنها این تعداد سطر غیرصفر از ورودی (در صورت اعمال نکردن پرچم DFT_INVERSE) یا این تعداد از سطرهای غیرصفر خروجی (در صورت اعمال کردن پرچم DFT_INVERSE) شامل عناصر غیر صفر هستند و به‌این‌ترتیب، در محاسبه‌ی بقیه‌ی سطرها صرفه‌جویی می‌شود. این پرچم برای محاسبه‌ی همبستگی متقابل[۲] و یا محاسبه‌ی کانولوشن با تبدیل فوریه کاربرد دارد.

- گشتاورهای آماری: برای محاسبه‌ی این ویژگی‌ها، از تابع زیر استفاده می‌کنیم:

```
Moments moments(InputArray array, bool binaryImage=false)
```

- array : آرایه‌ی ورودی، تک کاناله و هشت بیتی یا دوبعدیِ ممیز شناور.
- binaryImage : اگر درست باشد، تمام مقادیر غیر صفر، یک در نظر گرفته‌می‌شوند. این گزینه فقط در صورتی اعمال خواهد شد که ورودی تصویر باشد.

خروجی از جنس Moments می‌باشد که کلاسی شامل گشتاورهای شکل تا درجه‌ی سوم است. این کلاس سه نوع گشتاور را در خود ذخیره می‌کند: گشتاورهای فضایی[۳]، گشتاورهای مرکزی[۴]، و گشتاورهای مرکزی نرمال‌شده.

- گشتاورهای هو[۵][۱]:

```
void HuMoments(const Moments& m, OutputArray hu)
```

یا

```
void HuMoments(const Moments& moments, double hu[7])
```

که گشتاورهای آماری شکل را دریافت می‌کند و هفت گشتاور معرفی‌شده در [۱] را محاسبه می‌کند و در آرایه‌ی خروجی hu قرار می‌دهد.

[۲] cross-correlation
[۳] Spatial moments
[۴] Central moments
[۵] Hu Moments

- اسکلت: در بخش دوم به طور مفصل بر روی روش‌های پیاده‌سازی‌شده برای استخراج اسکلت شکل بحث خواهد شد.

- بهترین برازش بیضی:

RotatedRect fitEllipse(InputArray points)

این روش، آرایه‌ی ورودی شامل تعدادی نقاط را دریافت می‌کند و بهترین بیضی را به آن برازش می‌کند. نتیجه در قالب یک مستطیل دوران‌داده شده که بیضی در آن محیط می‌شود، محاسبه می‌شود.

- بهترین برازش دایره:

void minEnclosingCircle(InputArray points, Point2f& center, float& radius)

نقاط را دریافت می‌کند و مرکز و شعاع را در دو آرگومان مربوطه قرار می‌دهد.

- بهترین برازش مستطیل:

RotatedRect minAreaRect(InputArray points)

به نقاط ورودی، بهترین مستطیل را برازش می‌کند.

همچنین تابع

Rect boundingRect(InputArray points)

مستطیل محاط بر شکل را می‌دهد.

همچنین می‌توان به کمک توابع پیاده‌سازی شده برای پیداکردن کانتورها، خواص هندسی مانند محیط، مساحت (به کمک تابع contourArea)، عدد اویلر (تعداد سطح کانتورهای تودرتو) و گریز از مرکز[۶] (درازی) را نیز مستقیما به دست‌آورد. این مشخصات در کلاس دربردارنده‌ی کانتور به عنوان ویژگی‌های کانتور محاسبه می‌شوند.

علاوه بر روش‌های فوق که در درس آمده‌اند، روش‌های دیگری نیز پیاده‌سازی شده‌اند که در زیر آورده شده‌اند.

[۶] Eccentricity

- برازش خم چندجمله‌ای :

void approxPolyDP(InputArray curve, OutputArray approxCurve, double epsilon, bool closed)

  - curve : نقاط ورودی.
  - approxCurve : خروجی تقریب زده‌شده.
  - epsilon : دقت در تقریب است که در تعیین فاصله از جواب موثر است. بیشترین فاصله‌ی تقریب تا خود شکل از این مقدار بیشتر نمی‌شود.
  - closed : در صورت درست بودن این پرچم، خمِ تقریب‌زده شده بسته خواهد بود.

- چندضلعی محدب[۷]:

void convexHull(InputArray points, OutputArray hull, bool clockwise=false, bool returnPoints=true)

این تابع بع مجموعه‌ای از نقاط، یک چندضلعی محدب اختصاص می‌دهد.

  - clockwise : جهت نقاط را در نتیجه‌ی نهایی hull مشخص می‌کند.
  - returnPoints : در صورت درست بودن، هنگامی که ورودی ماتریس است، نقاط چندضلعی برگردانده می‌شوند و در غیر این صورت، اندیس نقاط در ماتریس را برمی‌گرداند.

پرسش: روش‌های Chain code، Fourier و بهترین مستطیل و بیضی برازش را بر روی تصاویری متناسب با روش‌ها اعمال کرده و نتایج را گزارش کنید.

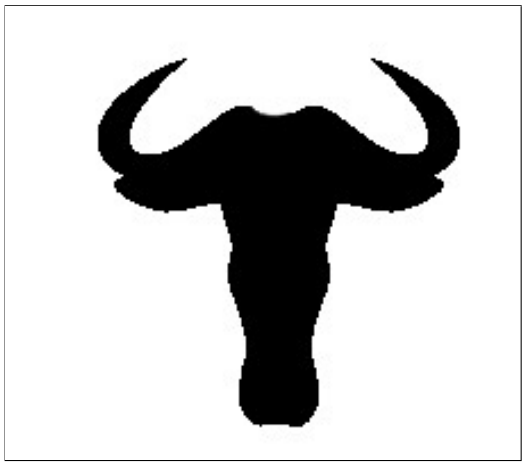
پاسخ: از روش‌هایی که در بالا آورده‌شد استفاده نمودیم.

در روش ارایه با کد فریمن برای شکل ۱۲، کدی به طول ۳۳۶۸ به‌دست آمد که این نتیجه را در فایل متنی ذخیره کرده‌ایم و در پوشه‌ی پروژه به نام 12.chain آورده‌ایم که با ویرایشگرهای متنی قابل مشاهده‌است.

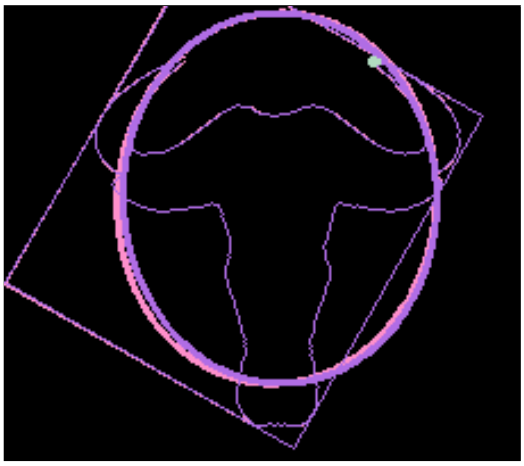
در روش فوریه، باید فوریه‌ی کانتور تصویر که یک لیست یک‌بعدی است را محاسبه نمود و اعداد به‌دست‌آمده را همان گونه که در درس اشاره شد، نسبت به چرخش و تغیر اندازه مقاوم نمود و این اعداد توصیف‌گر شکل خواهند بود.

[۷] Convex Hull

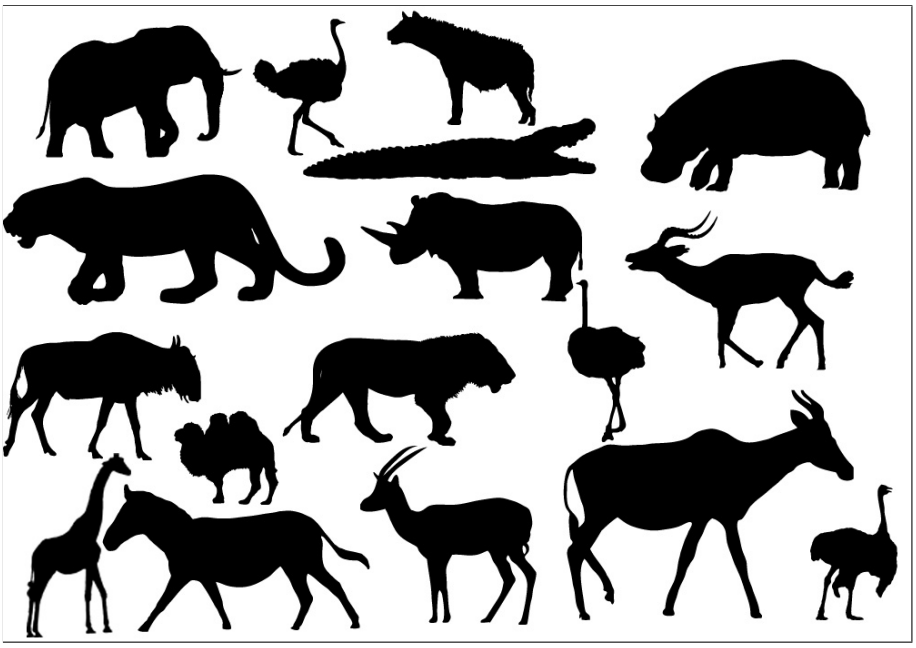
در روش برازش بیضی و مسطیل، از توابع معرفی‌شده در بخش اول همین تمرین استفاده نمودیم و نتیجه به شکل زیر شد:



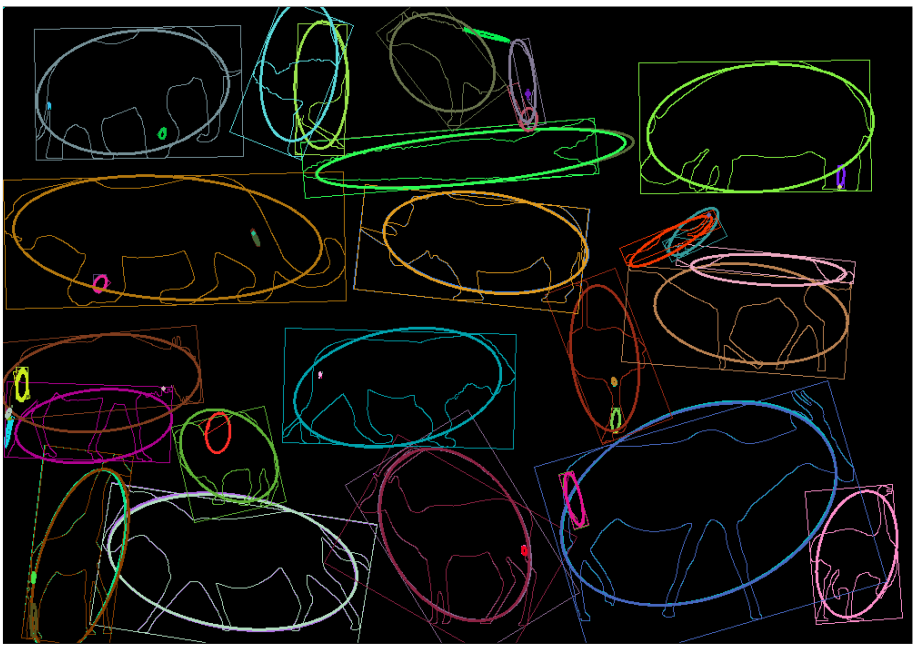
تصویر اصلی



تصویر برازش



تصویر اصلی



تصویر برازش

۲.۱ نازک‌سازی

پرسش:

برای محاسبه اسکلت اشیا روش های متعددی وجود دارد. با جستجوی روش ها در اینترنت یکی از این روش ها را با ارائه توضیح مختصر برای انجام تمرین زیر انتخاب کنید.

پاسخ: روش Zhang-Suen[۲] را انتخاب کردیم. در این روش

While points are deleted do

For all pixels p(i,j) do

if (a) 2 B(P1) 6

(b) A(P1) = 1

(c) Apply one of the following:

1. P2 x P4 x P6 = 0 in odd iterations
2. P2 x P4 x P8 = 0 in even iterations

(d) Apply one of the following:

1. P4 x P6 x P8 = 0 in odd iterations
2. P2 x P6 x P8 = 0 in even iterations

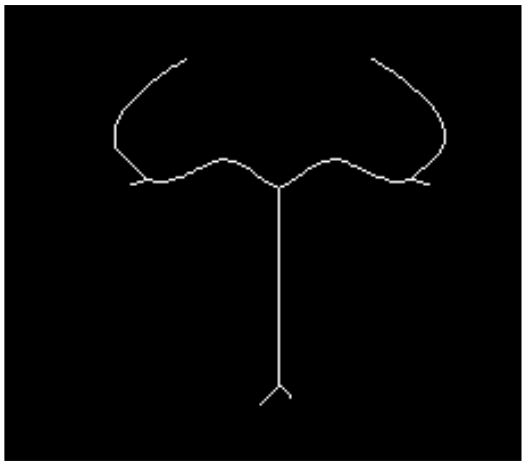
```
            then
                Delete pixel p(i,j)
            end if
        end for
end while
```

که در آن $A(P_1)$ تعداد گذر از صفر به یک در همسایگی در جهت ساعت تا P_9 و $B(P_1)$ تعداد همسایه‌های غیر صفر P_1 است.

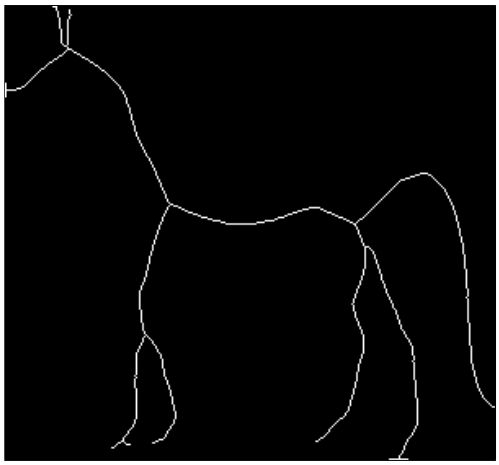
پرسش: روش ارائه با اسکلت را بر تصویر شماره ۱۲ اعمال کرده و آن را نمایش دهید. این تصویر از ساده ترین نمونه ها برای تشخیص اسکلت است. اگر روش انتخابی شما جواب مناسبی برای این تصویر پیدا نکرد روش را تغییر دهید.

پاسخ:

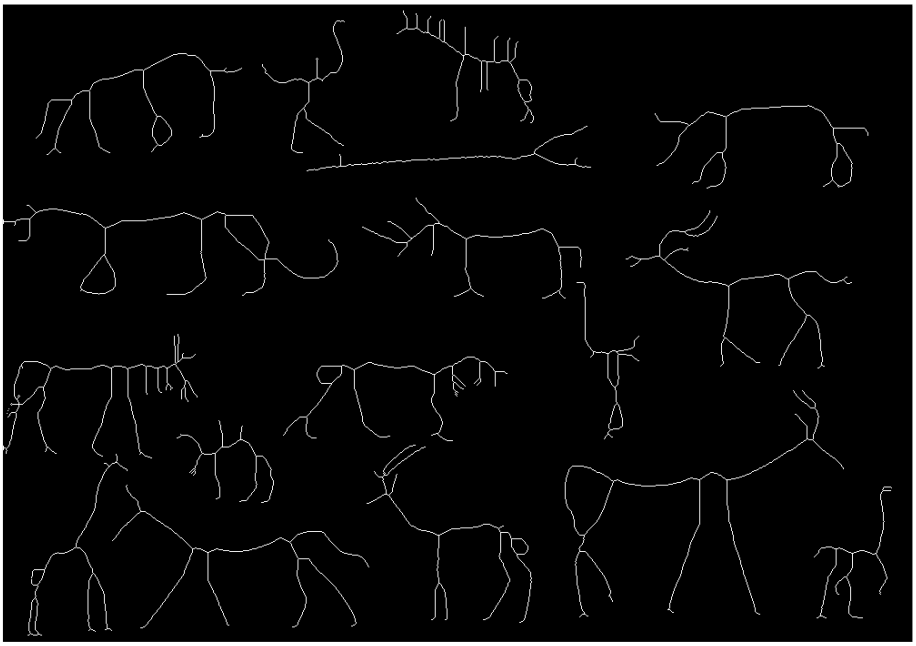
برای این تصویر باید عمل وارون کردن رنگ‌ها صورت گیرد که در رابط کاربری گنجانده شده‌است. نتایج به این شکل شد:



تصویر گوزن بخش قبل



تصویر ۱۲



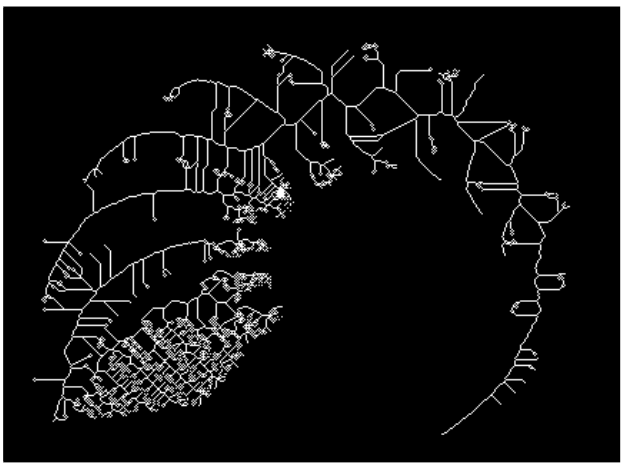
تصویر اسکلت حیوانات بخش قبل

پرسش:

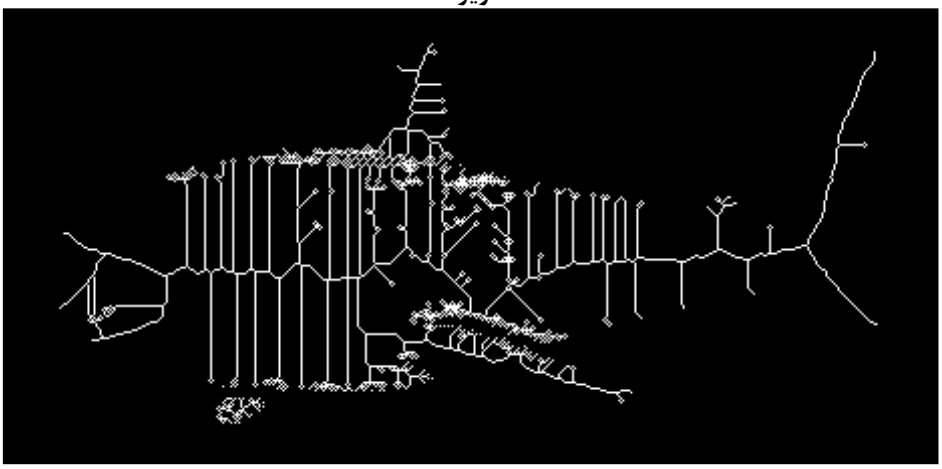
برای تصویر ۱۳ و ۱۴ نیز اسکلت شکل را تعیین کنید. آیا با تغییر پارامترهای روش به پاسخ متفاوتی میرسید؟ پاسخ خود را تحلیل کنید.

پاسخ:

نتایج متفاوت ترشلد کردن اثر گذار است. نتایج در زیر آمده‌است (ترشلد ۲۵۵-۱۰۰) :

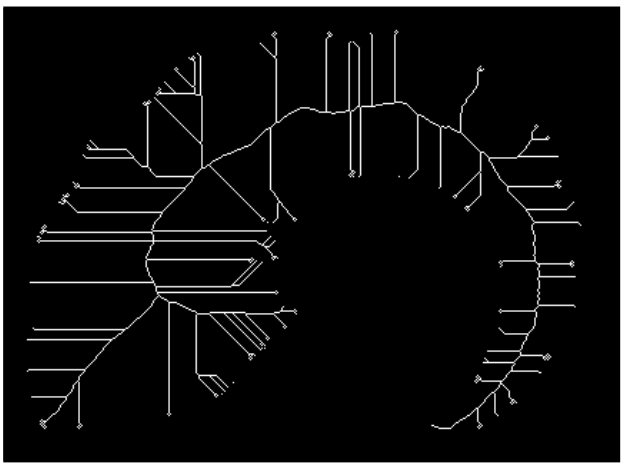


تصویر ۱۳

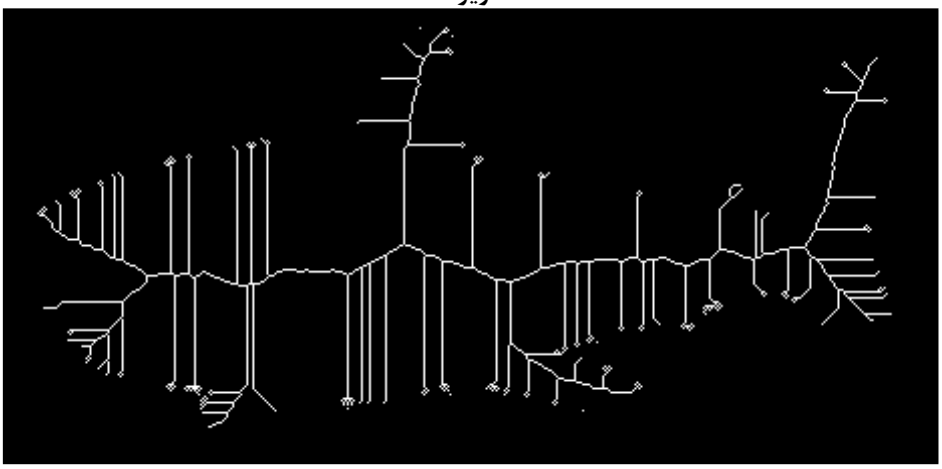


تصویر ۱۴

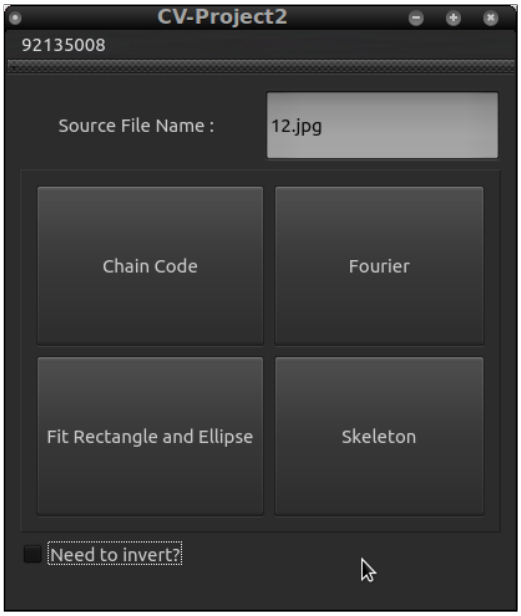
و نتیجه برای ترشلد بیشتر ۱۰ تا ۲۵۵:



تصویر ۱۳



تصویر ۱۴



نمای نرم‌افزار

مراجع

[1] M. Hu, “Visual Pattern Recognition by Moment Invariants,” IRE Transactions on Information Theory, vol. 8, no. 2, pp. 179-187, 1962.

[2] T. Y. Zhang, and C. Y. Suen, “A fast parallel algorithm for thinning digital patterns,” Communications of the ACM, vol. 27, no. 3, pp. 236-239, 1984.