



جزوه باما

دانلود جزوات، نمونه سوالات
و پروپونته‌های دانشگاهی

Jozvebama.ir



مقدمه‌ای بر
پردازش تصویر

Introduction to
Image Processing

انواع روش‌های تصویربرداری

یک تصویر به منظورهای متفاوتی تهیه می‌شود. از جمله

- ۱- ثبت خاطرات
- ۲- برای اوراق شناسایی
- ۳- برای موارد پزشکی
- ۴- برای مطالعات استخراج نفت و معادن
- ۵- برای مو غیره
- طالعات کیهان‌شناسی
- ۶-

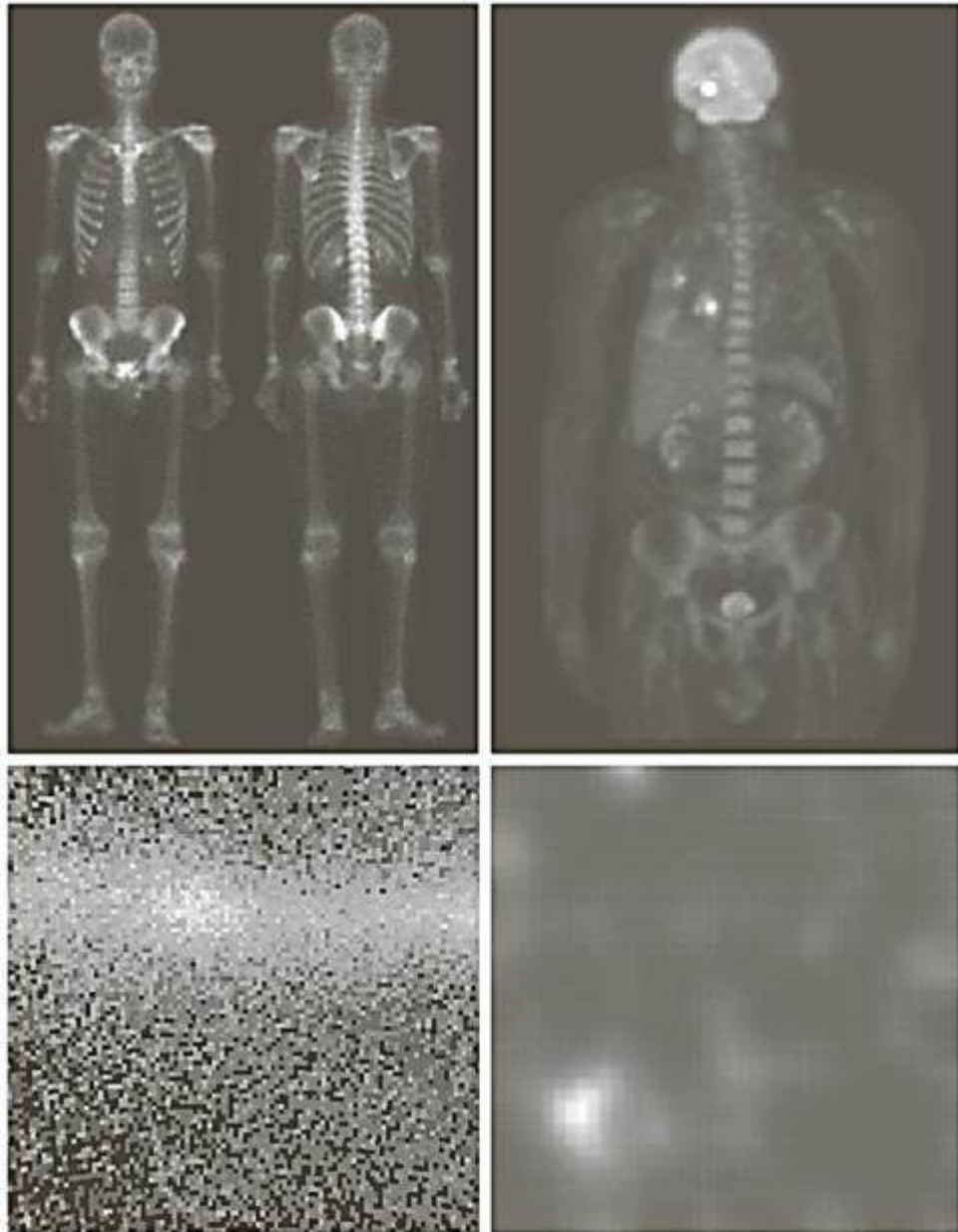
و البته از دستگاه‌ها و یا دوربین‌های متفاوتی برای عکس‌برداری استفاده می‌شود. از جمله:

1. تصویربرداری با دوربین‌های آنالوگ روی فیلم عکاسی
2. تصویربرداری با دوربین‌های دیجیتال
3. تصویر برداری اشعه گاما
4. تصویربرداری با اشعه ایکس
5. ماورا بنفش (UV) Ultra Violet
6. تصویر برداری نور مرئی و مادون قرمز
7. نور مرئی و مادون قرمز: میکروسکوپ نوری
8. تصویر برداری مایکروویو
9. تصویر برداری بوسیله امواج صوتی

انواع تصویربرداری

تصویر برداری اشعه گاما

نمونه هایی از تصویر برداری اشعه گاما. (الف) اسکن استخوانی، (ب) تصویر PET، (ج) ستاره ای که در صورت فلکی دجابه ۱۵۰۰۰ سال پیش منفجر شده تشکیل ابر گازی بسیار گرم به نام حلقه دجابه تولید کرده ، (د) تشعشع گاما از یک شیر فلکه رآکتور

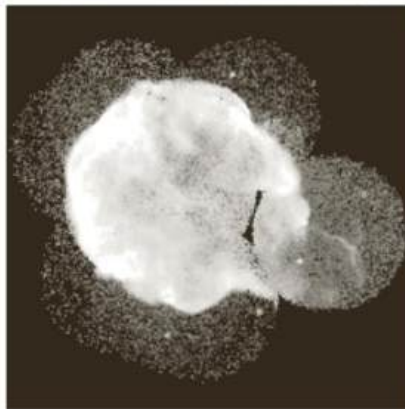
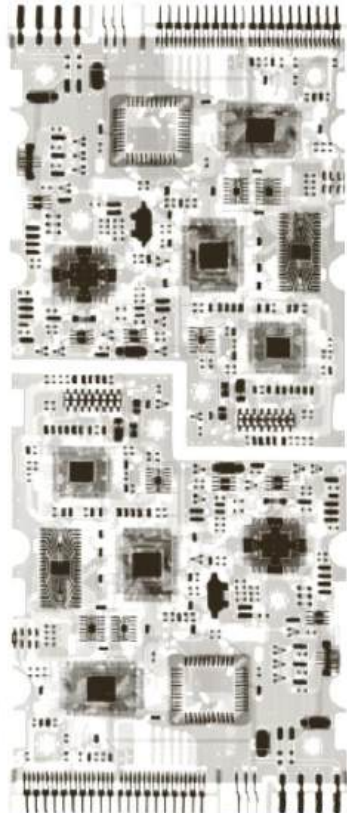


انواع تصویربرداری

تصویربرداری با اشعه ایکس

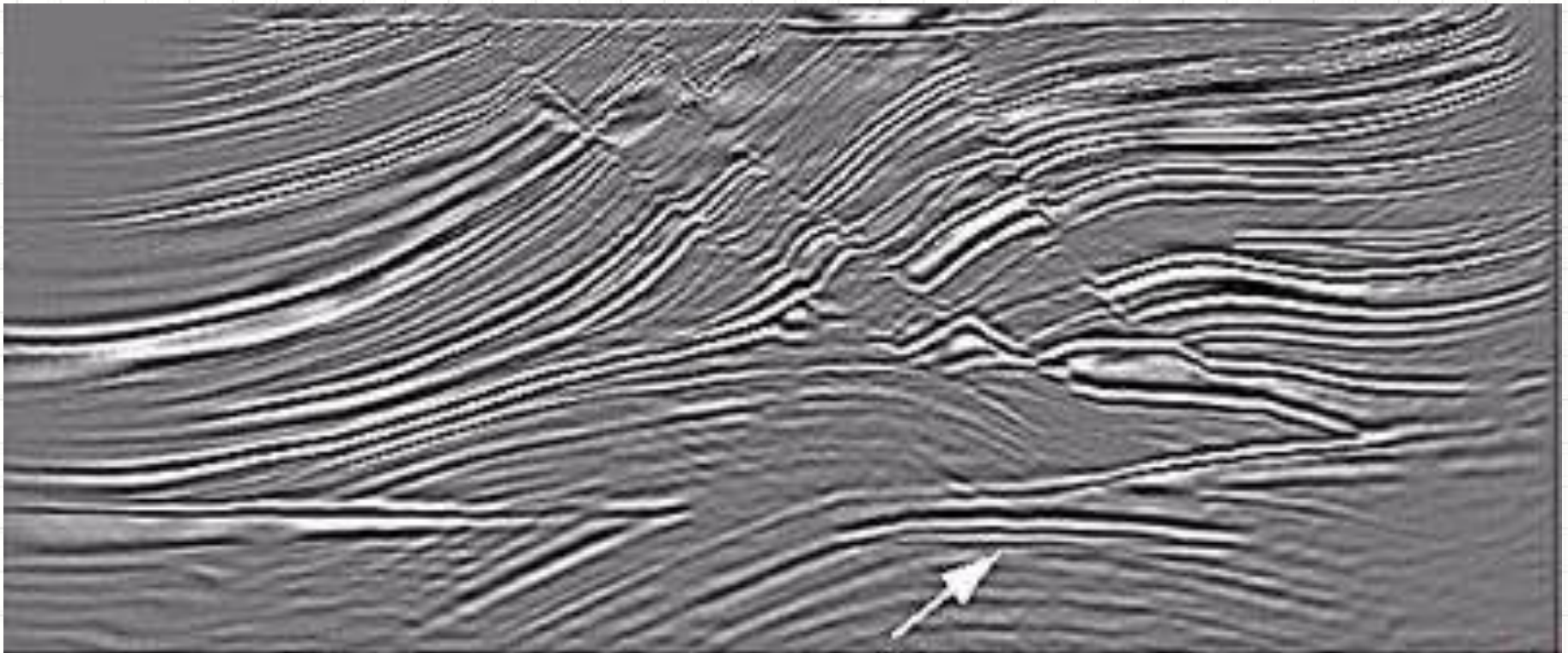
مثالهایی از تصویربرداری اشعه X: (الف)
اشعه X قفسه سینه. (ب) آنژیوگرام
سرخرگ. (ج) CAT از سر انسان. (د)
بردهای مدارات الکترونیکی. (ه) حلقه

دجانه



انواع تصویربرداری

تصویر برداری بوسیله امواج صوتی



یک تصویر از مقطع عرضی یک مدل لرزه ای. فلش اشاره به یک منبع نفت یا گاز دارد (که از وجود نقاط روشن در کنار حجم وسیع نقاط تیره تشخیص داده شده)

پردازش تصویر چیست؟

- پردازش تصویر به موارد زیر اطلاق می شود:
 - دست کاری تصاویر به منظور بهبود تصاویر، به طوری که تفسیر آن ها برای انسان بهتر گردد.
 - استخراج اطلاعاتی از تصویر که برای تفسیر و فهم تصویر توسط ماشین به کار می رود.

فهم تصویر توسط ماشین به بینایی ماشین (Machine Vision) موسوم است.
بینایی ماشین از مباحث پیشرفته در پردازش تصویر می باشد

سطوح پردازش تصاویر

پردازش های سطح پایین

شامل عملیات اولیه مانند پردازش تصویر برای کاهش نویز، ارتقای کنتراست و تیز کردن تصویر است. مشخصه این پردازش ها اینست که ورودی و خروجی، هر دو تصویر هستند.

پردازش های سطح متوسط

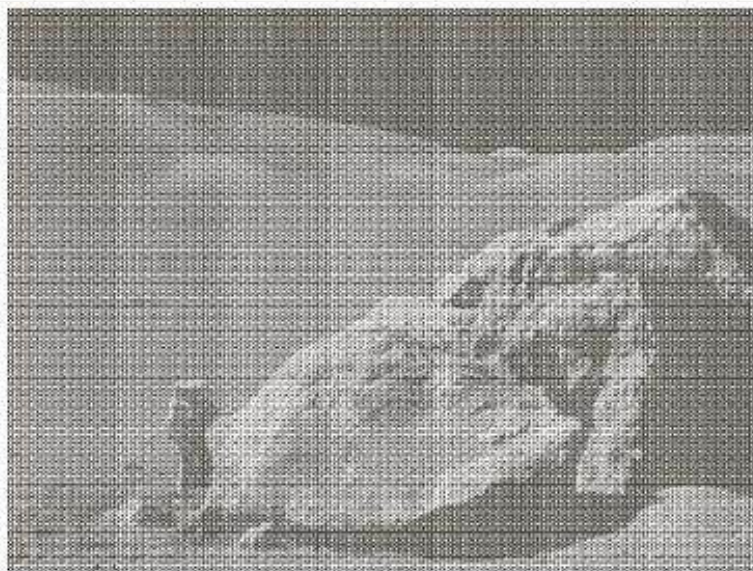
شامل عملیات قطعه بندی segmentation (تقسیم بندی تصویر به مناطق یا اشیا)، توصیف اشیا جهت ساده سازی وضعیت آنها به صورتی که برای پردازش توسط کامپیوتر مناسب شوند، و نیز طبقه بندی classification و شناسایی recognition اشیا می باشد. مشخصه این نوع پردازش اینست که ورودی ها عموماً تصویر بوده اما خروجی های آن توصیفاتی است که از تصویر (مانند لبه ها edges، منحنی میزان contours و مشخصه های identity شناسایی اشیا) استخراج شده اند.

پردازش های سطح بالا

شامل معنی دار کردن تصاویر می باشد.

سطوح پردازش تصاویر

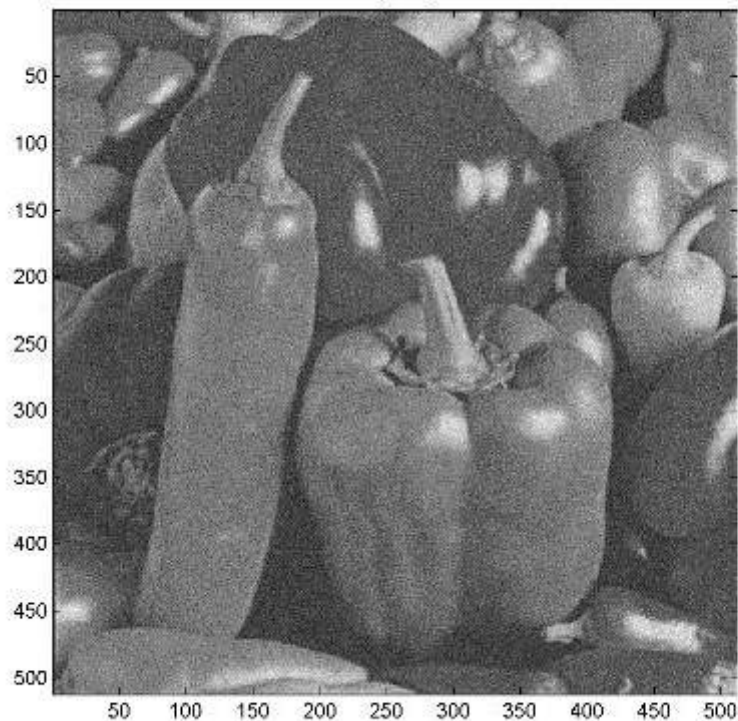
مثالی از پردازش های سطح پایین
کاهش نویز



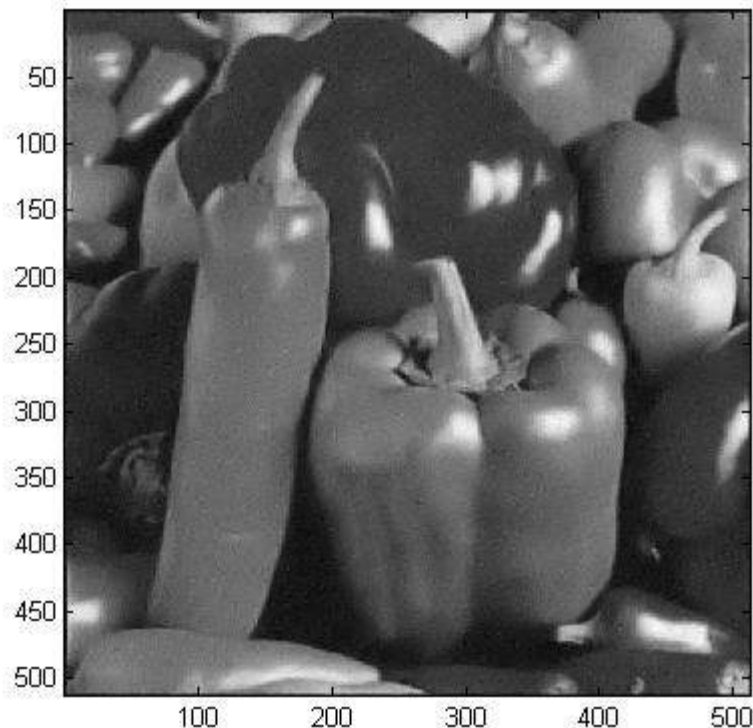
سطوح پردازش تصاویر

مثالی از پردازش های سطح پایین
کاهش نویز

Noisy Image

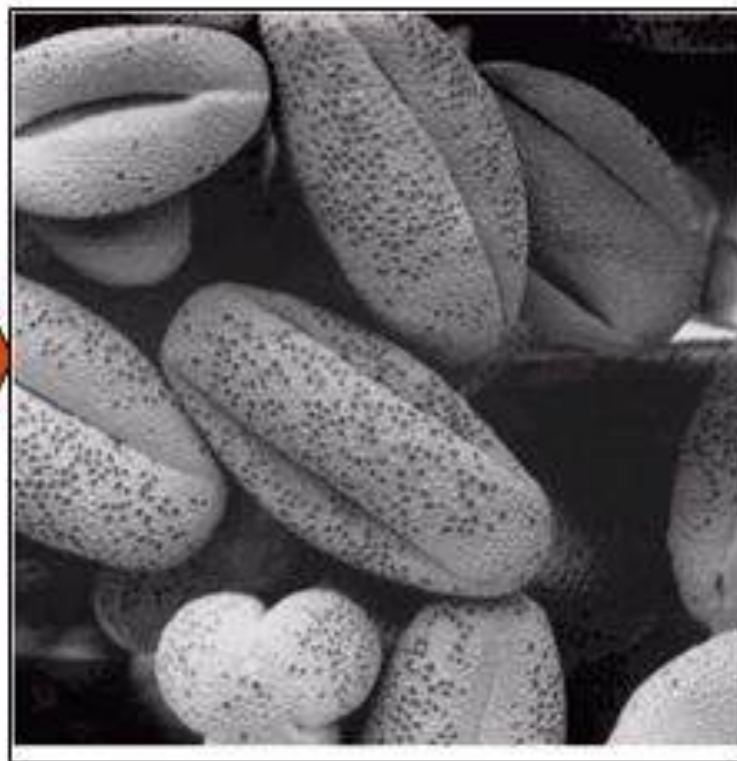
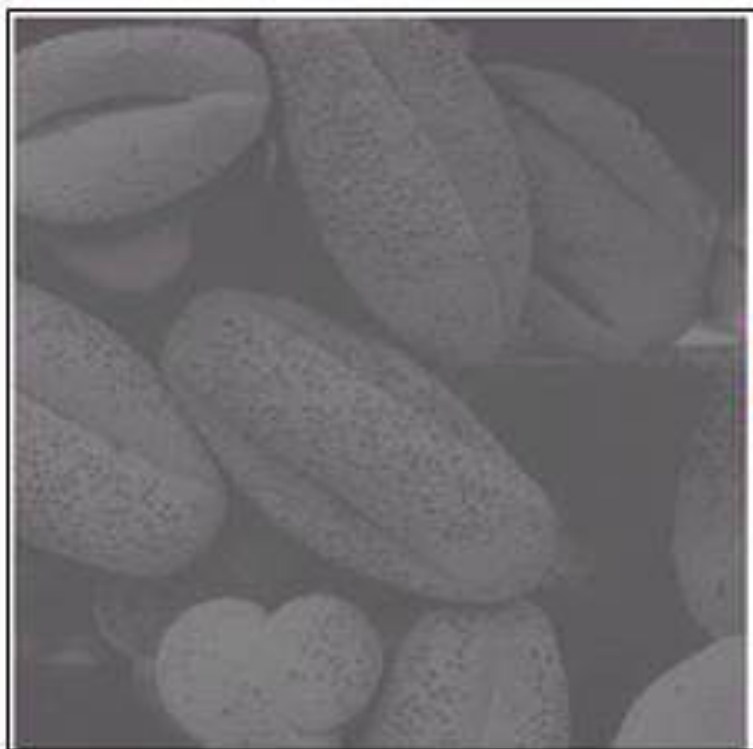


Denoised Image



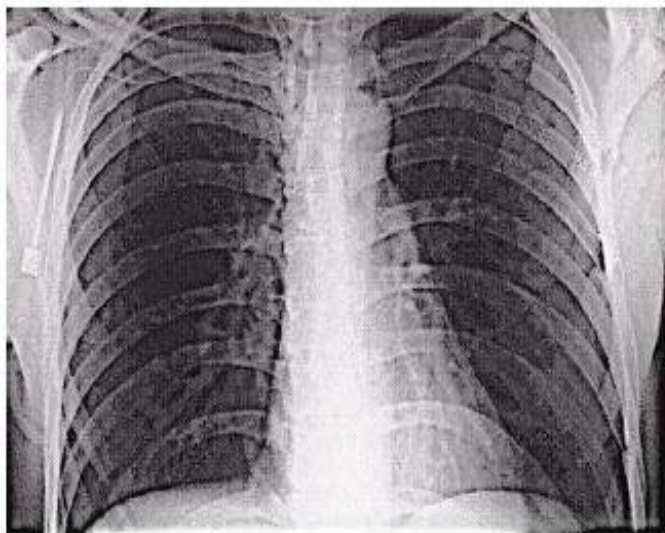
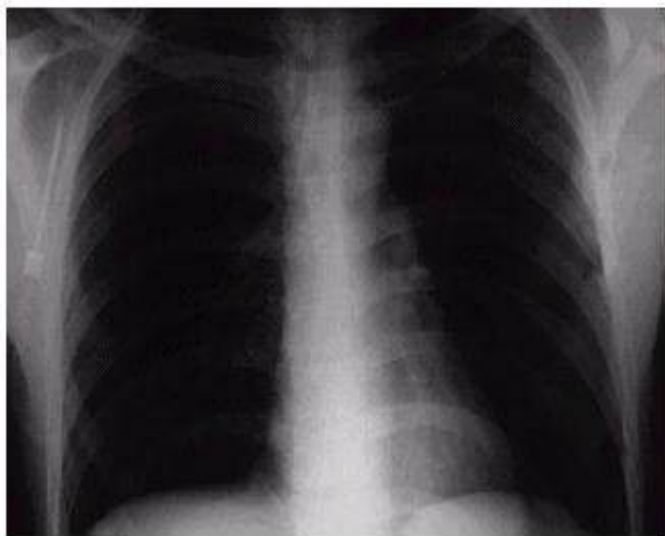
سطوح پردازش تصاویر

مثالی از پردازش‌های سطح پایین
ارتقای کنتراست



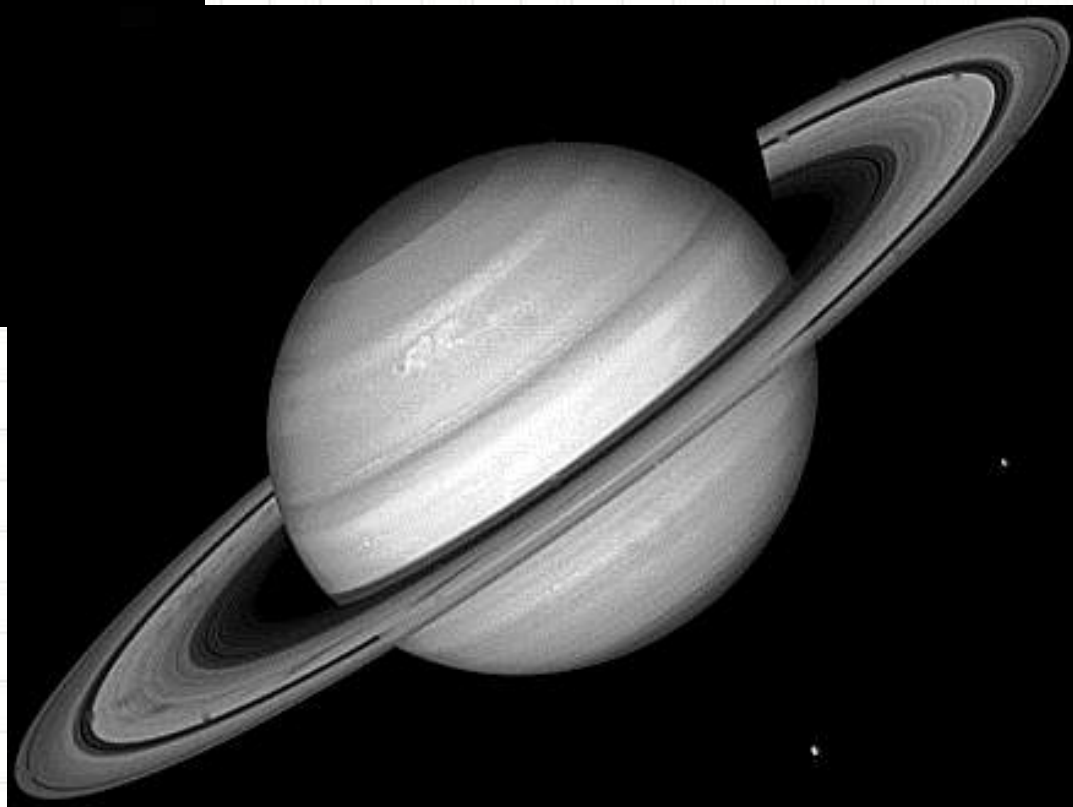
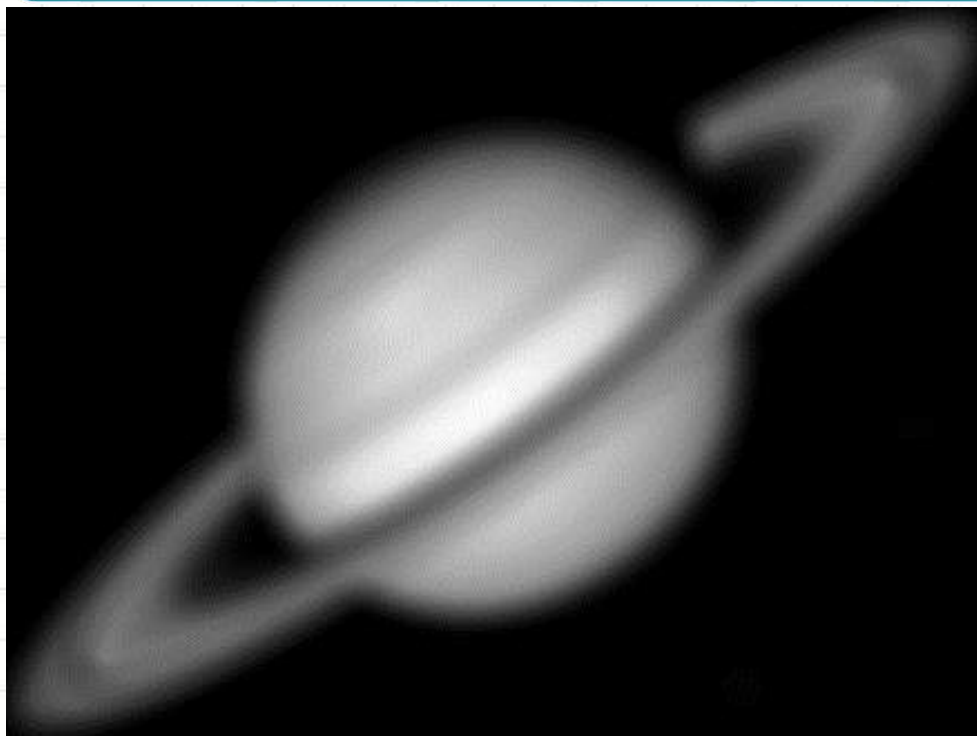
سطوح پردازش تصاویر

مثالی از پردازش‌های سطح پایین
ارتقای کنتراست



سطوح پردازش تصاویر

مثالی از پردازش‌های سطح پایین
تیز کردن تصویر



سطوح پردازش تصاویر

مثالی از پردازش‌های سطح پایین
اصلاح هندسی تصویر



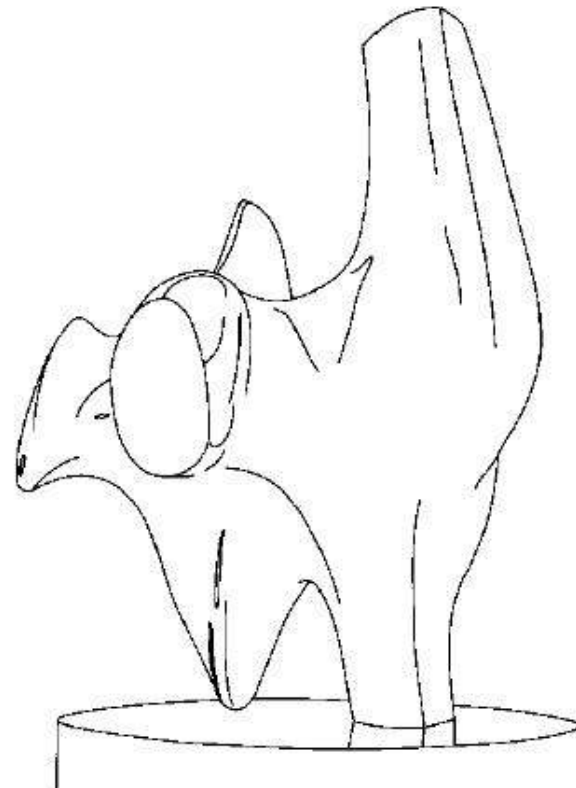
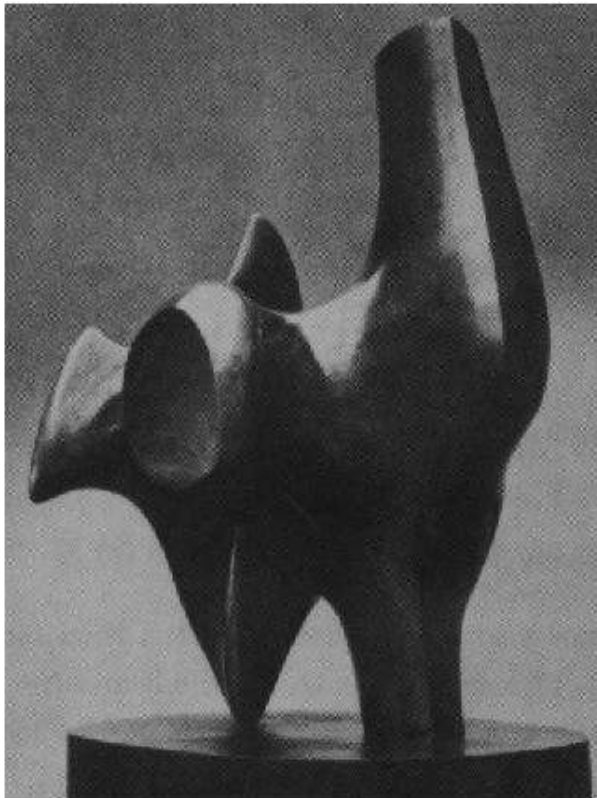
سطوح پردازش تصاویر



سطوح پردازش تصاویر

پردازش های سطح متوسط

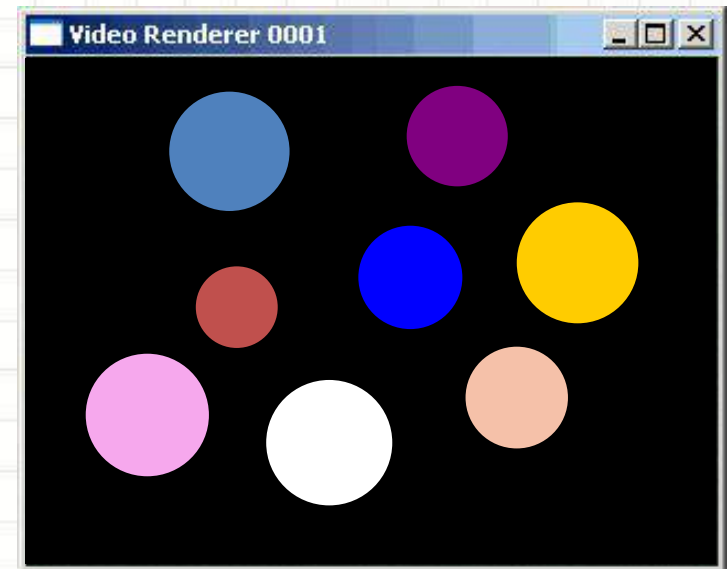
قطعه بندی segmentation (تقسیم بندی تصویر به مناطق یا اشیا)،
توصیف اشیا جهت ساده سازی وضعیت آنها به صورتی که برای پردازش توسط کامپیوتر
مناسب شوند.



سطوح پردازش تصاویر

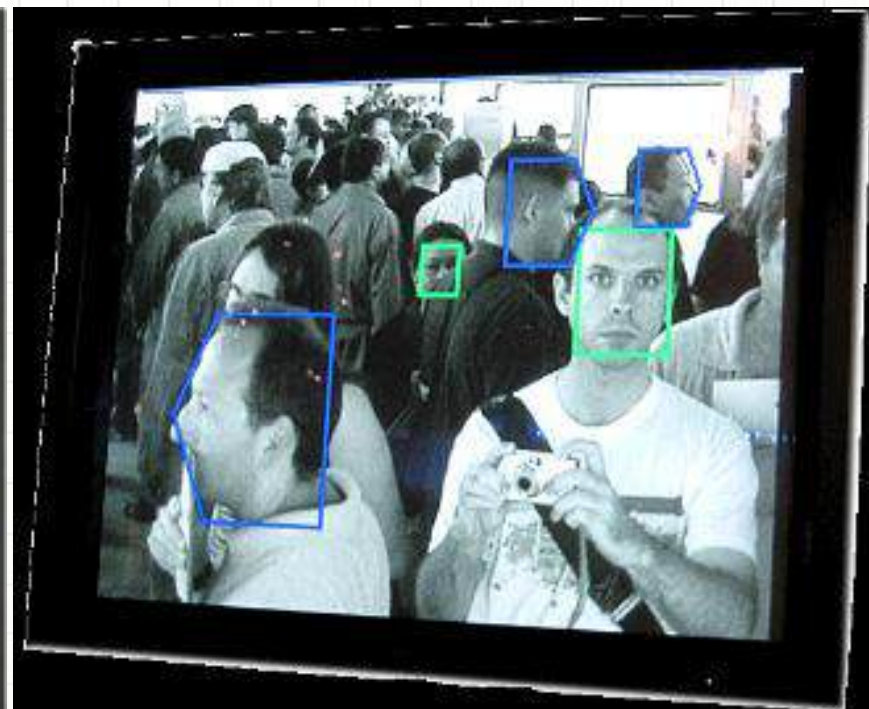
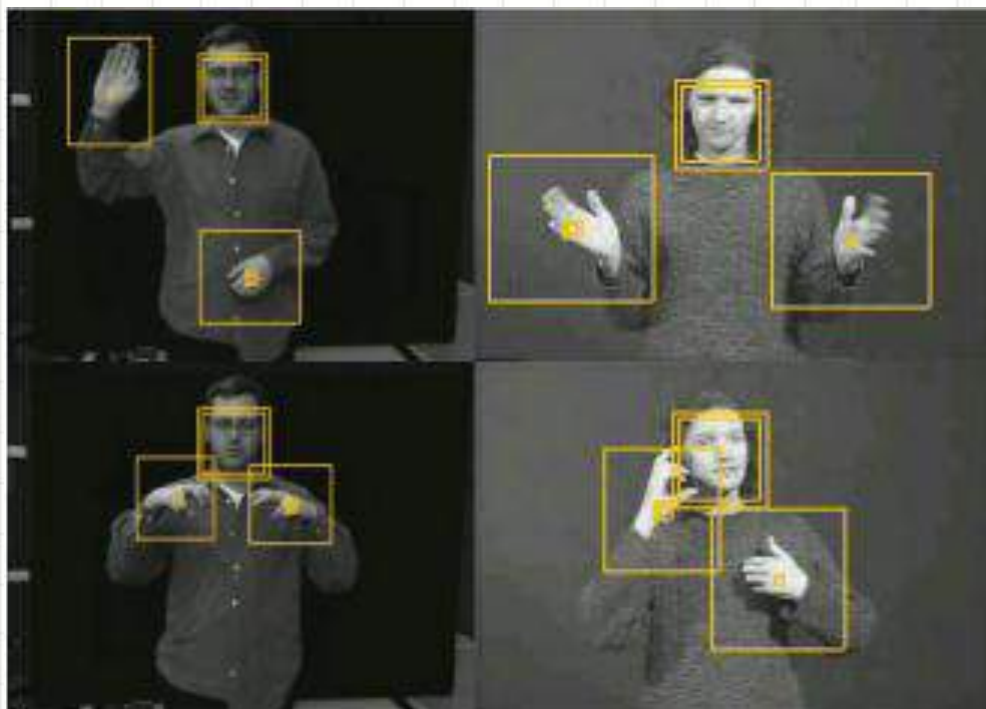
پردازش های سطح متوسط

قطعه بندی segmentation (تقسیم بندی تصویر به مناطق یا اشیا)،
توصیف اشیا جهت ساده سازی وضعیت آنها به صورتی که برای پردازش توسط کامپیوتر
مناسب شوند.



سطوح پردازش تصاویر

پردازش های سطح متوسط
شناسایی recognition اشیا



سطوح پردازش تصاویر

پردازش های سطح بالا

شامل معنی دار کردن اشیا تشخیص داده شده، همانطور که در تجزیه و تحلیل تصاویر است، و در انتهای طرف دیگر طیف، اجرای توابعی با قابلیت شناسایی است که معمولا با بینایی همراه است.

سطوح پردازش تصاویر

به طور کلی توافق چندانی بر روی اینکه گستره پردازش تصویر در کجا پایان می یابد و سایر حیطه های مرتبط مانند تحلیل تصاویر و بینایی ماشین از کجا آغاز می شوند، وجود ندارد.

Low Level Process

Input: Image
Output: Image

Examples: Noise removal, image sharpening

Mid Level Process

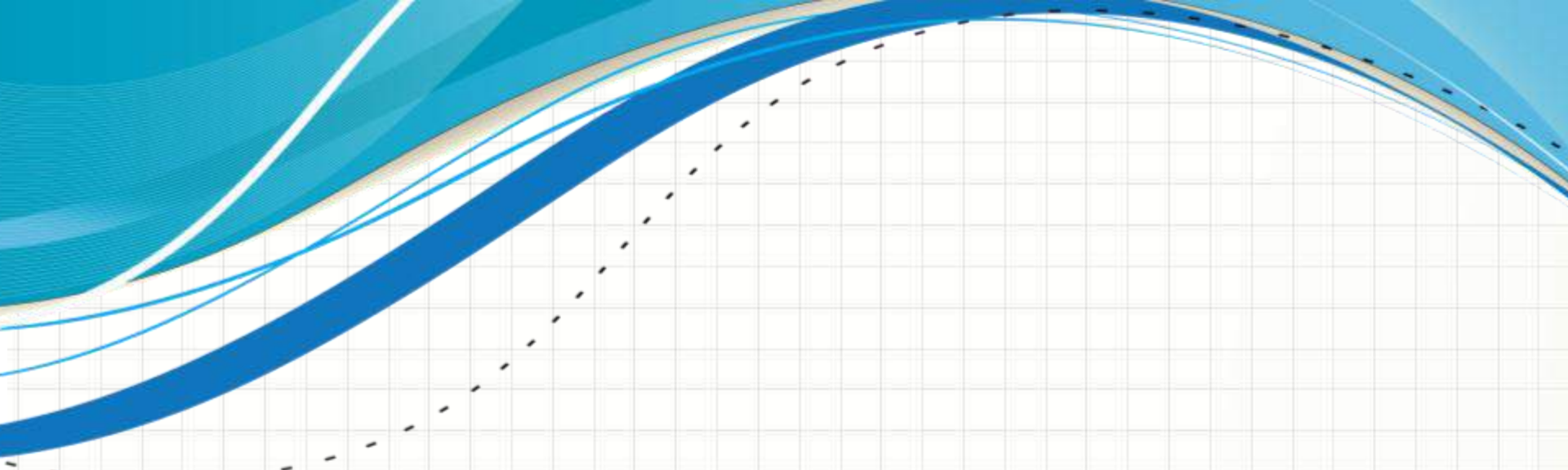
Input: Image
Output: Attributes

Examples: Object recognition, segmentation

High Level Process

Input: Attributes
Output: Understanding

Examples: Scene understanding, autonomous navigation



تصاویر دیجیتال

Digital Images

تصاویر دیجیتال و آنالوگ

کلیه داده‌ها از جمله تصاویر را می‌توان به فرم آنالوگ یا دیجیتال ذخیره و نگهداری کرد. از این دیدگاه، تصاویر به دو دسته زیر دسته‌بندی می‌شوند:

آنالوگ (Analog)

دیجیتال (Digital)

تصاویر آنالوگ روی رسانه‌های آنالوگ از قبیل **نگاتیو** ذخیره می‌شوند.



در صورتی که تصاویر دیجیتال برای ذخیره روی حافظه‌های کامپیوتری مناسب می‌باشند. به این صورت که یک نمایش عددی برای تصویر در نظر گرفته می‌شود و آن اعداد در حافظه کامپیوتر ذخیره می‌شوند.

تصاویر دیجیتال و آنالوگ

سوالی که مطرح می‌شود که تصاویر دیجیتال مطلوب‌تر می‌باشند و یا تصاویر آنالوگ؟

در پاسخ باید گفت که طبیعت دارای ساختار آنالوگ است ولی کامپیوترها و ماشین‌ها دارای ساختار دیجیتال می‌باشند. طبیعی است که اگر بخواهیم تصاویر را با کمک کامپیوترها

۱- ذخیره کنیم

۲- انتقال دهیم

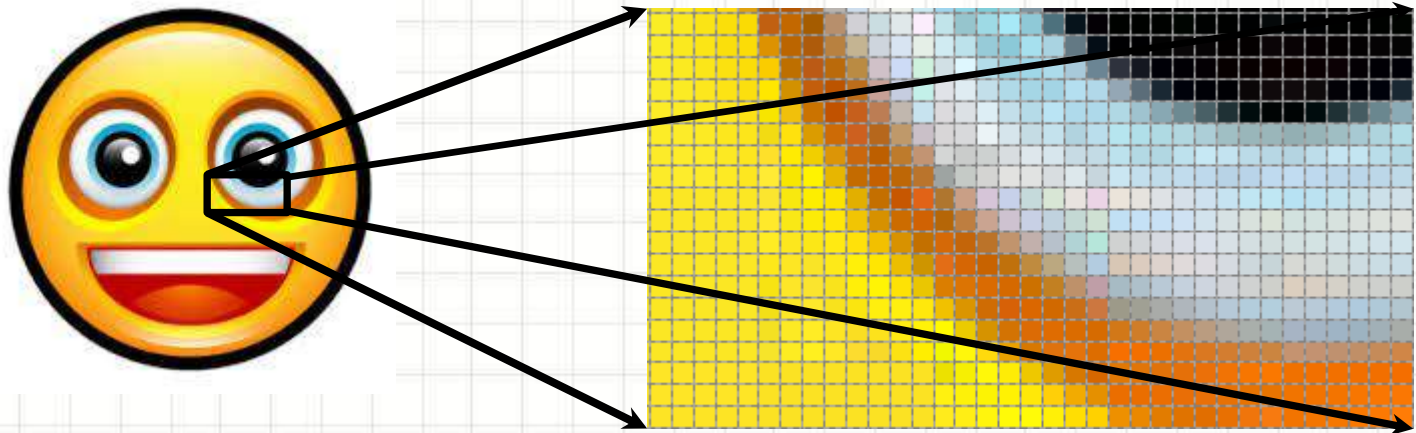
۳- پردازش کنیم

آن‌گاه بهتر است از رویکرد دیجیتال استفاده کنیم.

البته رویکرد آنالوگ هم دارای مزیت‌های خاص خود می‌باشد، ولی به خاطر محدودیت در ساختار کامپیوترها و وسایل الکترونیکی، امروزه رویکرد دیجیتال بیشتر مورد استفاده قرار می‌گیرد.

پیکسل در تصویر دیجیتال

یک تصویر دیجیتال از مجموعه‌ای از پیکسل‌ها (Pixel) تشکیل شده است. به طور دقیق‌تر هر تصویر دیجیتال عبارت است از آرایه‌ای از پیکسل‌ها که در کنار هم تصویر را تشکیل می‌دهند.



پیکسل که با واژه‌های Pel و Picture Element نیز نامگذاری می‌شود.

هر پیکسل از تصویر دارای دو ویژگی است:

- ۱- **موقعیت مکانی پیکسل:** که با شماره سطر و ستونی که پیکسل در آن قرار گرفته مشخص می‌شود. آدرس پیکسل با یک زوج مرتب به صورت (x, y) مشخص می‌شود.
- ۲- **رنگ پیکسل،** که گاهی به **شدت پیکسل (intensity)** نیز موسوم می‌باشد.

دسته‌بندی تصاویر دیجیتال

تصاویر دیجیتال بسته به این که رنگ پیکسل‌ها چگونه انتخاب شوند، به سه دسته زیر تقسیم می‌شوند

۱- تصاویر باینری (سیاه و سفید) (Binary Images)

۲- تصاویر تک‌رنگ (Monochrome)

تصاویر خاکستری (Grayscale Images)

۳- تصاویر رنگی (Color Image)

در تصاویر باینری هر پیکسل فقط دو رنگ (عموما سیاه یا سفید) می‌تواند اختیار کند. بنابراین هر پیکسل فقط دو حالت می‌تواند داشته باشد.

در تصاویر تک‌رنگ فقط از یک رنگ و طیف‌های متفاوت آن رنگ روی یک **رنگ زمینه** استفاده می‌شود. مثلاً یک تصویر که فقط از سبز و طیف‌های متفاوت سبز (از سبز پررنگ تا سبز کم‌رنگ) **روی زمینه سیاه** استفاده شده باشد، به یک تصویر تک‌رنگ موسوم است.

یک حالت خاص از تصاویر تک‌رنگ، عبارت است از تصاویر خاکستری. در تصاویر خاکستری فقط از رنگ‌های سیاه و طیف‌های خاکستری روی زمینه سفید برای هر پیکسل استفاده می‌شود.

در تصاویر رنگی، پیکسل‌ها می‌توانند رنگ‌های متفاوت را اختیار کنند.

دسته‌بندی تصاویر دیجیتال – مثال



تصویر دودویی
Binary



تصویر خاکستری
Gray scale

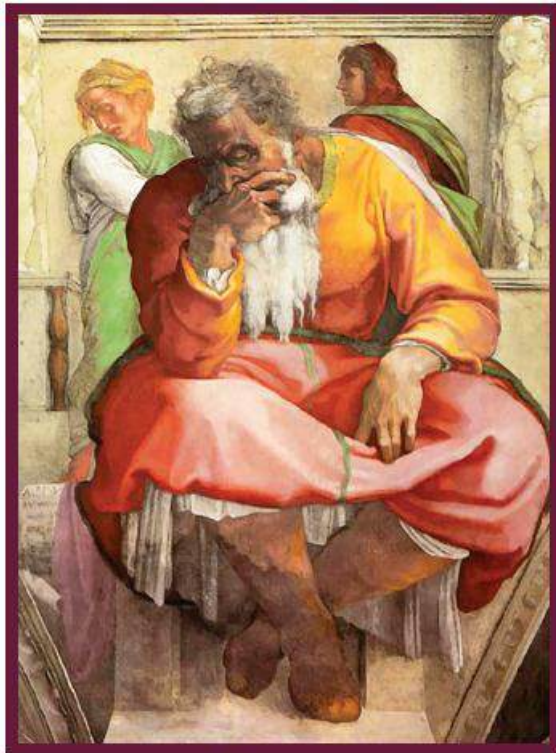


تصویر رنگی
Color

تصویر تک رنگ
monochrome



دسته‌بندی تصاویر دیجیتال – مثال



(a) Color image.



(b) Grayscale image.

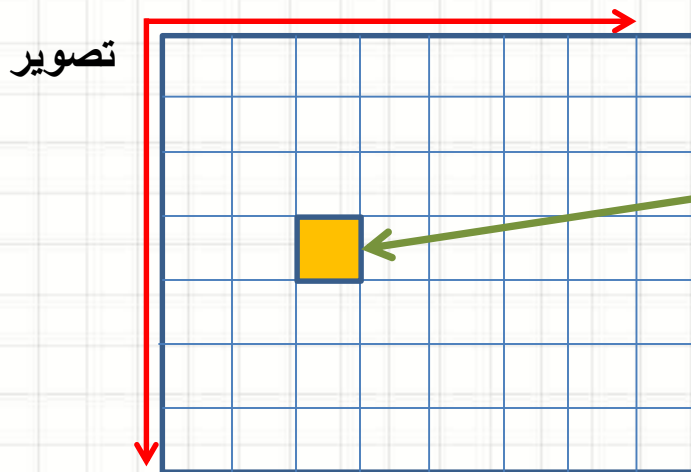


(c) Binary image.

ذخیره تصاویر دیجیتال

تصاویر دیجیتال آرایه‌ای از پیکسل‌ها می‌باشند. طبیعی است که برای ذخیره آن‌ها از ساختمان داده‌ای ماتریس استفاده کنیم.

برای ذخیره یک تصویر دیجیتال که دارای m ردیف سطری پیکسل و n ردیف ستونی پیکسل باشد، از یک ماتریس $m \times n$ استفاده می‌شود. مولفه (i, j) ام این ماتریس برای پیکسل ردیف i ام و ستون j ام تصویر در نظر گرفته می‌شود و اطلاعات رنگ پیکسل در آن ذخیره می‌شود.



مولفه $(4, 3)$ از ماتریس اطلاعات پیکسل سطر چهارم و ستون سوم را در خود ذخیره می‌کند.

تصاویر باینری

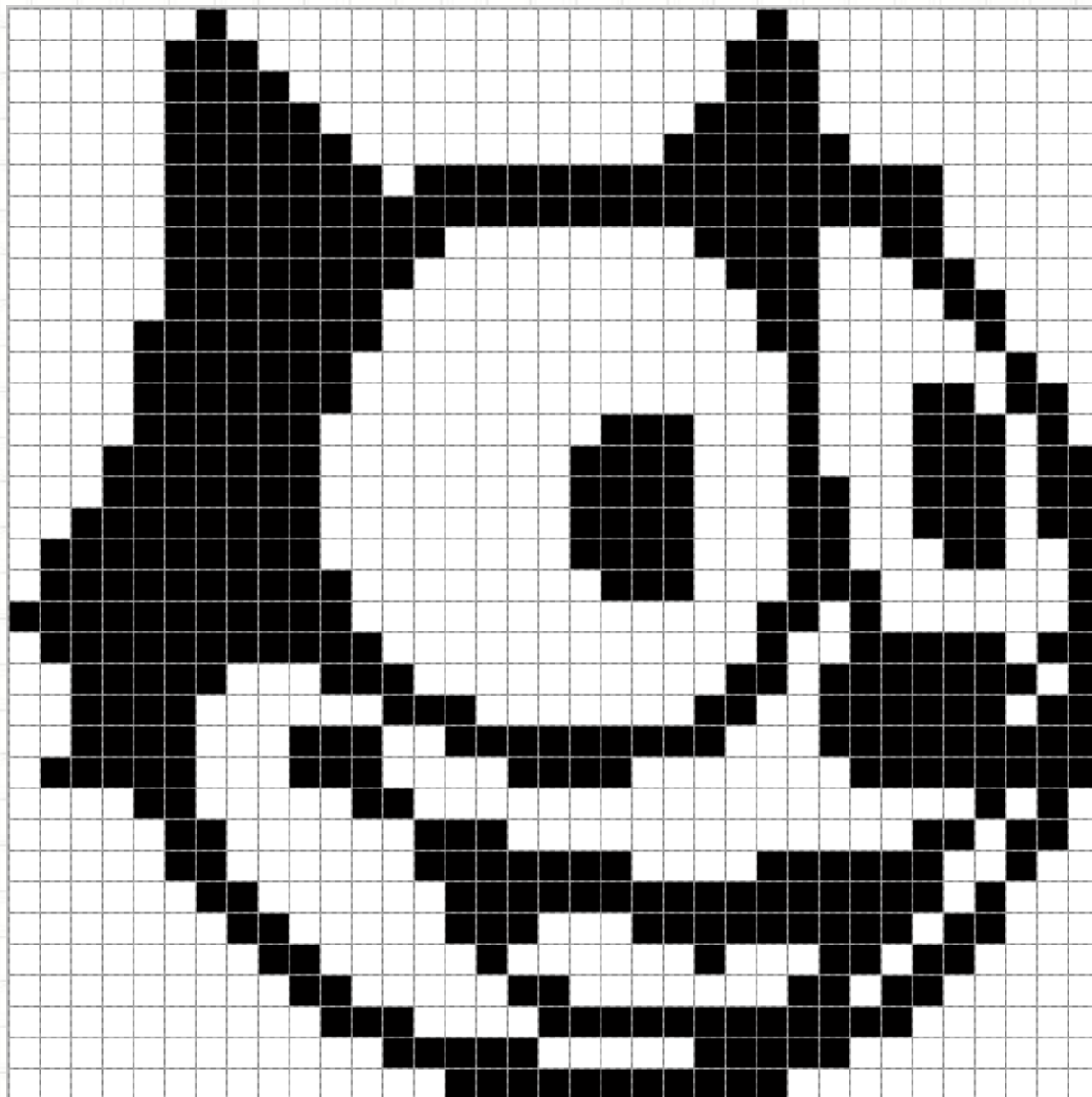
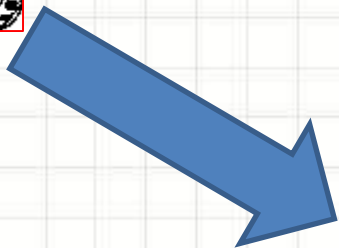
ذخیره تصاویر باینری

در تصاویر باینری، هر پیکسل دو حالت می‌تواند اختیار کند. بنابراین هر مولفه از ماتریس تصویر را می‌توان یک متغیر یک بیتی در نظر گرفت.

به عبارت دیگر ماتریس متناظر با یک تصویر باینری یک ماتریس است که مولفه‌های آن باینری (0 یا 1) می‌باشد.

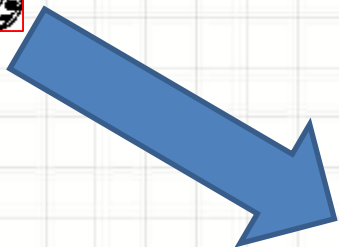
عدد صفر برای رنگ سیاه و عدد ۱ را برای رنگ سفید استفاده می‌شود.

ذخیره تصاویر باینری



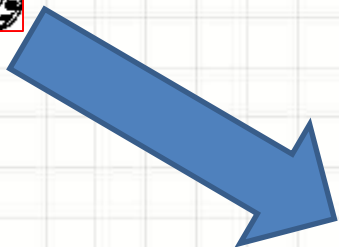
35 × 35

ذخیره تصاویر باینری



35 × 35

ذخیره تصاویر باینری



در کامپیوتر این
ماتریس ذخیره
می شود.



35×35

تصاویر خاکستری

عمق بیت

ذخیره تصاویر خاکستری

در تصاویر خاکستری، هر پیکسل بیش از دو حالت می‌تواند اختیار کند و این حالت‌ها برای مشخص کردن رنگ سیاه، سفید و برخی سطوح خاکستری استفاده می‌شود.

توجه شود که در دنیای واقعی بینهایت سطح خاکستری یا **بینهایت** رنگ خاکستری با بینهایت **شدت** (intensity) وجود دارد. امکان ذخیره تمامی این بینهایت شدت متفاوت وجود ندارد. لذا فقط تعداد متناهی از طیف خاکستری انتخاب می‌شود.

مثلا به تعداد L رنگ خاکستری (به همراه رنگ سفید و سیاه) برای هر پیکسل در نظر گرفته می‌شود. عدد L به **intensity resolution** موسوم است.

به خاطر ملاحظات ذخیره‌سازی غالبا d به صورت توانی از ۲ در نظر گرفته می‌شود. مثلا

$$L=2^d$$

که d به **عمق بیتی** (bit depth) یا **عمق رنگ** (Color depth) موسوم می‌باشد.

عمق بیتی در تصاویر خاکستری عبارت است از تعداد بیت‌های اختصاص داده شده برای نمایش سطوح خاکستری در هر پیکسل.

همینطور intensity resolution عبارت است از دو به توان عمق بیت.

ذخیره تصاویر خاکستری ادامه... ۲

بنابراین برای یک تصویر خاکستری با m سطر و n ستون پیکسل و با عمق بیت d به ماتریسی $m \times n$ نیاز داریم به طوری که هر مولفه این ماتریس از d بیت تشکیل شده است. در این صورت تعداد بیت‌های لازم برای ذخیره تصویر فوق عبارت است از

$$m \times n \times d$$

نتیجه: برای هر تصویر خاکستری سه پارامتر می‌توان در نظر گرفت

m : تعداد پیکسل‌ها در هر سطر

n : تعداد پیکسل‌ها در هر ستون

d : عمق بیت رنگ‌های خاکستری (تعداد بیت‌هایی که برای هر پیکسل در نظر گرفته می‌شود) توجه شود که تعداد رنگ‌های خاکستری در یک تصویر با عمق بیت d برابر است با $L = 2^d$.

معمولا تصاویر خاکستری با عمق رنگ $d=8$ بیشتر رایج می‌باشند. چرا که این تصاویر دارای وضوح کافی برای انسان می‌باشد.

عمق بیت در تصاویر خاکستری

عمق بیت ۱

0

1

عمق بیت ۲

0

1

2

3

یک تصویر خاکستری با **عمق بیت ۲**
عبارت است از یک تصویر با چهار
رنگ خاکستری. این چهار رنگ با
کدهای
0,1,2,3
مشخص می‌شوند

یک تصویر خاکستری با **عمق بیت ۱**
همان تصویر باینری می‌باشد که فقط از
دو رنگ سیاه و سفید تشکیل شده
است. این دو رنگ با کدهای
0,1
مشخص می‌شوند

عمق بیت در تصاویر خاکستری

عمق بیت ۱

0

1

عمق بیت ۲

0

1

2

3

عمق بیت ۳

0

1

2

3

4

5

6

7

عمق بیت ۴

0

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

یک تصویر خاکستری با **عمق بیت ۳** عبارت است از یک تصویر با **هشت** رنگ خاکستری. این هشت رنگ با کدهای 0,1,2,3,4,5,6,7 مشخص می‌شوند

یک تصویر خاکستری با **عمق بیت ۴** عبارت است از یک تصویر با **شانزده** رنگ خاکستری با کدهای صفر الی ۱۵.

نکته قابل توجه آن است که در عمق‌های ۱ الی ۴، چشم انسان می‌تواند کلیه رنگ‌های خاکستری را از یک‌دیگر تفکیک کند و تشخیص دهد.

عمق بیت در تصاویر خاکستری

عمق بیت ۱

در **عمق بیت ۵** به تعداد ۳۲ رنگ خاکستری وجود دارد که با شماره‌های صفر الی ۳۱ کد می‌شوند. لازم به ذکر است که چشم انسان قادر به تشخیص تمامی این ۳۲ رنگ خاکستری از یک‌دیگر نمی‌باشد.

البته مرز برخورد دو رنگ تیره‌تر به نظر می‌رسد و چشم ما مرز را تشخیص می‌دهد، اما رنگ داخل برخی **بلوک‌ها** را یکسان می‌بیند.

بیت ۲

بیت ۳

عمق بیت ۴

عمق بیت ۵

چشم انسان قادر به تشخیص فقط ۲۴ رنگ خاکستری متفاوت از یک‌دیگر می‌باشد.

این به این معنی نیست که تصاویر با عمق بیت ۲۴ و مثلاً ۶۴ به یک شکل دیده می‌شوند.

عمق بیت در تصاویر خاکستری

در **عمق‌های ۷ و ۸** به تعداد ۱۲۸ و ۲۵۶ رنگ خاکستری وجود دارد. همان‌گونه که ملاحظه می‌شود چشم انسان قادر به تفکیک بلوک‌های خاکستری از یکدیگر نمی‌باشد و این رنگ‌های خاکستری به صورت **پیوسته** از سفید تا سیاه دیده می‌شود.

عمق بیت ۱

بیت ۲

بیت ۳

عمق بیت ۴

عمق بیت ۵

عمق بیت ۶

عمق بیت ۷

عمق بیت ۸

عمق بیت در تصاویر خاکستری

عمق بیت ۱



عمق بیت ۲



عمق بیت ۳



عمق بیت ۴



عمق بیت ۵



عمق بیت ۶



عمق بیت ۷



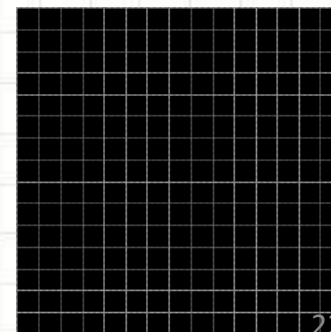
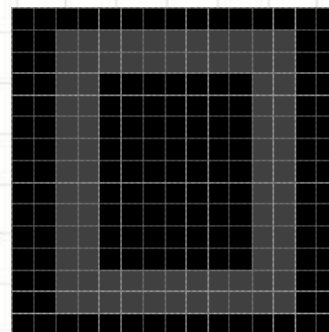
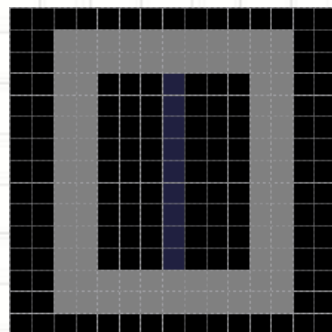
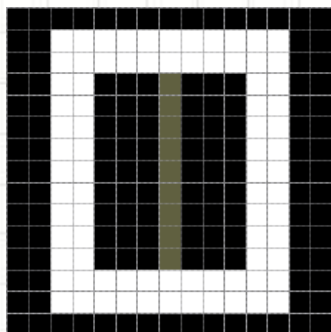
عمق بیت ۸



مثال: ماتریس یک تصویر خاکستری

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ماتریس مقابل متناظر با
کدام یک از تصویرهای
زیر می باشد؟



مثال: ماتریس یک تصویر خاکستری

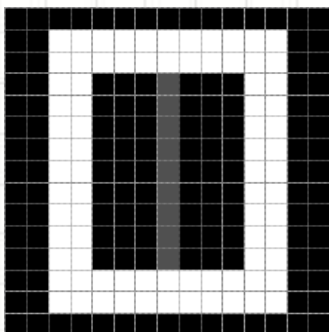
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	0	0	0	5	0	0	0	15	15	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	15	15	15	15	15	15	15	15	15	15	15	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

سوال فوق ناقص است. چرا که باید عمق بیت نیز مشخص شود.

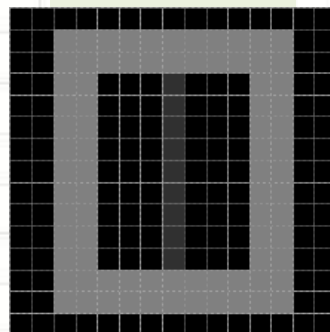
به عبارت دیگر باید مشخص شود که ماتریس روبرو در چه عمق بیتی داده شده است.

هر کدام از تصاویر زیر می‌تواند متناظر با ماتریس روبرو باشد، **منتهی در عمق‌های متفاوت.**

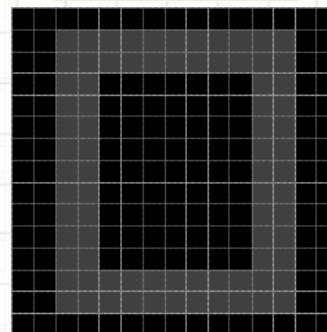
در عمق بیت ۴



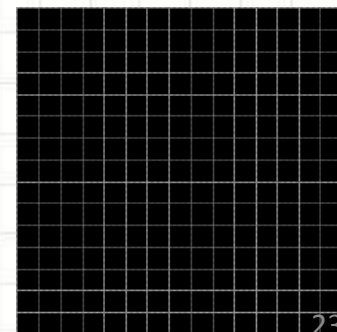
در عمق بیت ۵

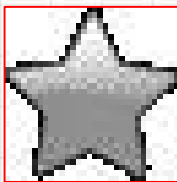


در عمق بیت ۶

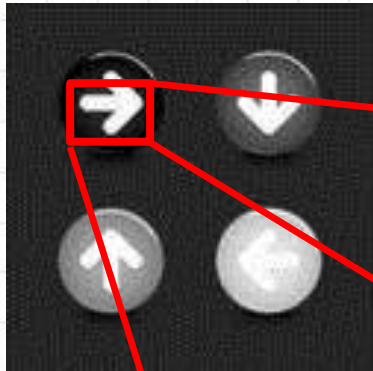


در عمق بیت ۸



[illegible]

ماتریس یک تصویر خاکستری



37	25	28	34	33	38	49	52	40	56	44	33	147	249	235	249	247	241	255	211	79	34	41	9
12	50	119	173	183	178	177	174	175	179	189	166	187	234	237	252	255	254	249	239	219	84	9	33
30	132	243	255	242	254	255	247	248	240	245	255	255	242	243	255	245	255	255	239	255	204	54	14
35	198	255	240	253	255	248	255	255	255	255	248	242	249	255	254	254	234	255	248	251	233	112	17
36	204	255	236	255	255	233	255	246	240	242	253	254	245	240	242	255	240	255	247	246	243	127	1
30	118	210	252	255	255	255	255	255	254	255	255	250	238	237	244	255	248	252	254	249	181	62	9
21	33	106	164	153	151	165	156	159	178	171	147	166	223	255	251	255	242	252	254	196	77	8	39
22	23	30	37	37	37	37	34	42	30	26	64	150	233	255	244	245	255	255	190	75	21	24	25
28	38	28	19	32	41	41	47	49	23	51	149	235	255	248	246	244	250	191	79	11	24	37	4
23	20	32	40	32	33	40	38	33	69	148	231	255	244	240	255	255	207	74	6	38	36	11	21
20	22	25	29	33	36	38	40	32	128	245	255	247	255	247	251	196	74	20	32	19	19	32	16
17	19	22	26	29	32	33	34	63	191	255	255	245	247	251	199	78	29	27	45	28	20	23	7
14	16	20	24	27	29	29	29	62	201	253	236	239	255	203	59	20	20	32	34	21	17	18	9
13	15	18	22	26	28	29	29	25	101	230	255	249	174	63	18	33	34	23	13	14	16	14	12
11	12	15	18	21	24	27	28	23	36	118	162	134	64	7	32	26	24	13	13	23	16	25	5

تصاویر رنگی

در آینده به طور مفصل توضیح داده خواهد شد.

تصویر برداری دیجیتال

نمونه برداری
کوانتیزاسیون
دقت مکانی
دقت شدت

تصویر برداری دیجیتال

یک تصویر دیجیتال ممکن است که از طریق یک دوربین دیجیتال، اسکنر و یا دستگاه‌های دیگر ایجاد شود. کلیه این دستگاه‌ها تصاویر آنالوگ را به تصاویر دیجیتال تبدیل می‌کنند.

یک منظره در طبیعت دارای ماهیت آنالوگ است و وقتی با یک دوربین دیجیتال از آن منظره عکس‌برداری می‌کنیم، در واقع کار تبدیل تصویر آنالوگ به تصویر دیجیتال صورت می‌گیرد. به همین ترتیب در اسکنرها نیز این تبدیل انجام می‌شود.

نمونه برداری و کوانتیزاسیون

در تبدیل یک تصویر آنالوگ به دیجیتال توسط دوربین دیجیتال، اسکنر و غیره دو کار انجام می‌شود.

۱- نمونه برداری (Sampling) ۲- کوانتیزاسیون (Quantizing)

در یک تصویر آنالوگ بی‌نهایت پیکسل وجود دارد. اما امکان لحاظ کردن بی‌نهایت پیکسل در تصاویر دیجیتال وجود ندارد. در واقع تعداد پیکسل‌ها در یک تصویر دیجیتال ممکن است بسیار زیاد باشد، اما به هر حال متناهی است. لذا در تبدیل یک تصویر آنالوگ به دیجیتال باید به تعداد متناهی پیکسل انتخاب شود و به اصطلاح تعداد متناهی پیکسل نمونه برداری شود. این عمل به نمونه برداری موسوم است.

در طبیعت و لذا در یک تصویر آنالوگ بی‌نهایت سطح رنگ خاکستری وجود دارد و پیکسل‌ها می‌توانند بی‌نهایت سطح رنگ خاکستری را اختیار کند. اما در تصاویر دیجیتال هر پیکسل می‌تواند تعداد متناهی سطح رنگ متفاوت را اختیار کند. لذا در تبدیل یک تصویر آنالوگ به دیجیتال باید از بی‌نهایت سطح رنگ خاکستری، فقط تعداد متناهی سطح رنگ انتخاب شود. این عمل به کوانتیزاسیون موسوم است.

نمونه برداری و کوانتیزاسیون

به طور خلاصه،

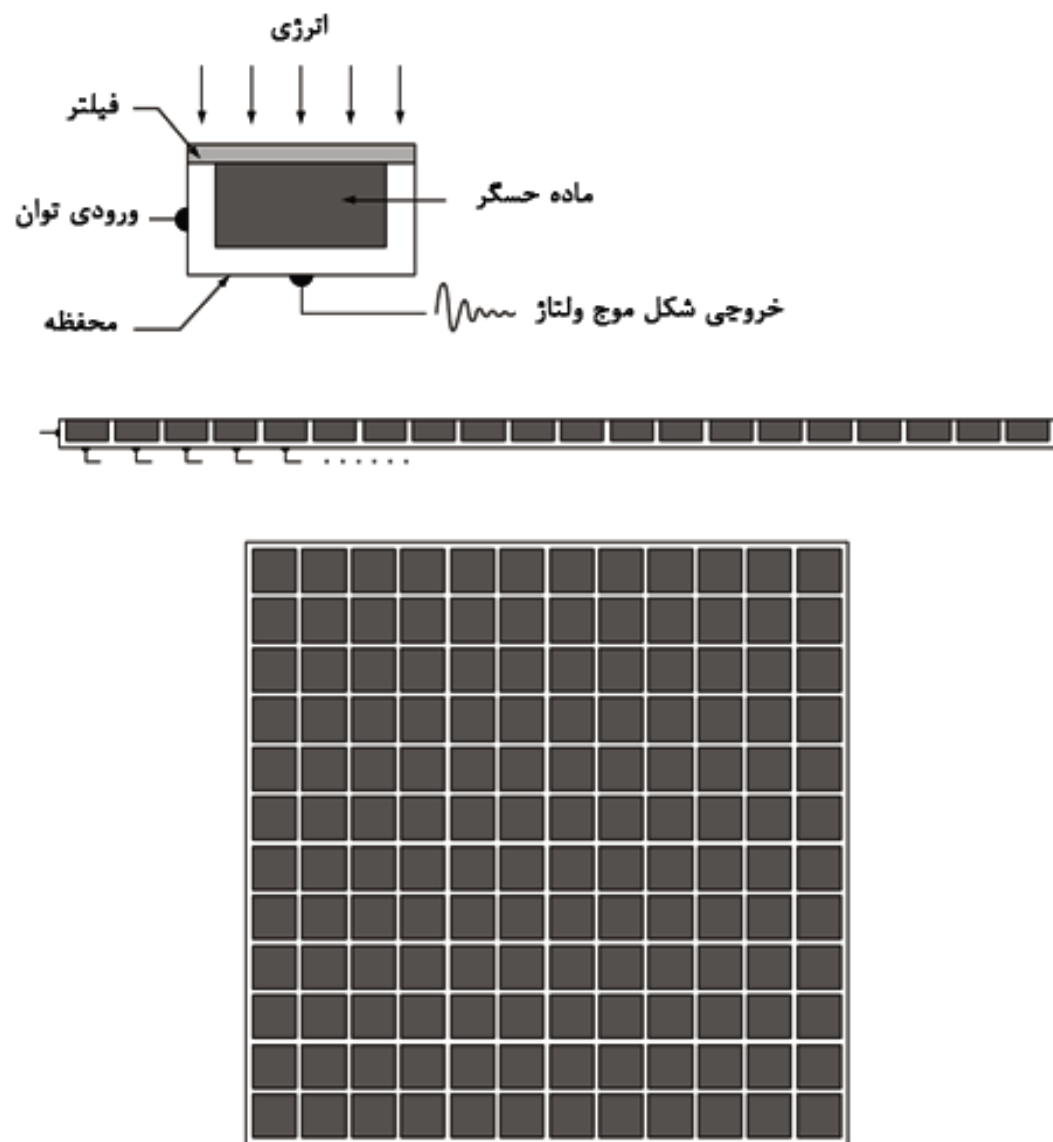
نمونه برداری عبارت است از انتخاب تعداد متناهی پیکسل از یک تصویر آنالوگ به منظور ذخیره سازی تصویر به صورت دیجیتال

کوانتیزاسیون عبارت است از انتخاب تعداد متناهی سطح رنگ از بی نهایت سطح رنگ برای اختصاص به پیکسل ها در یک تصویر دیجیتال

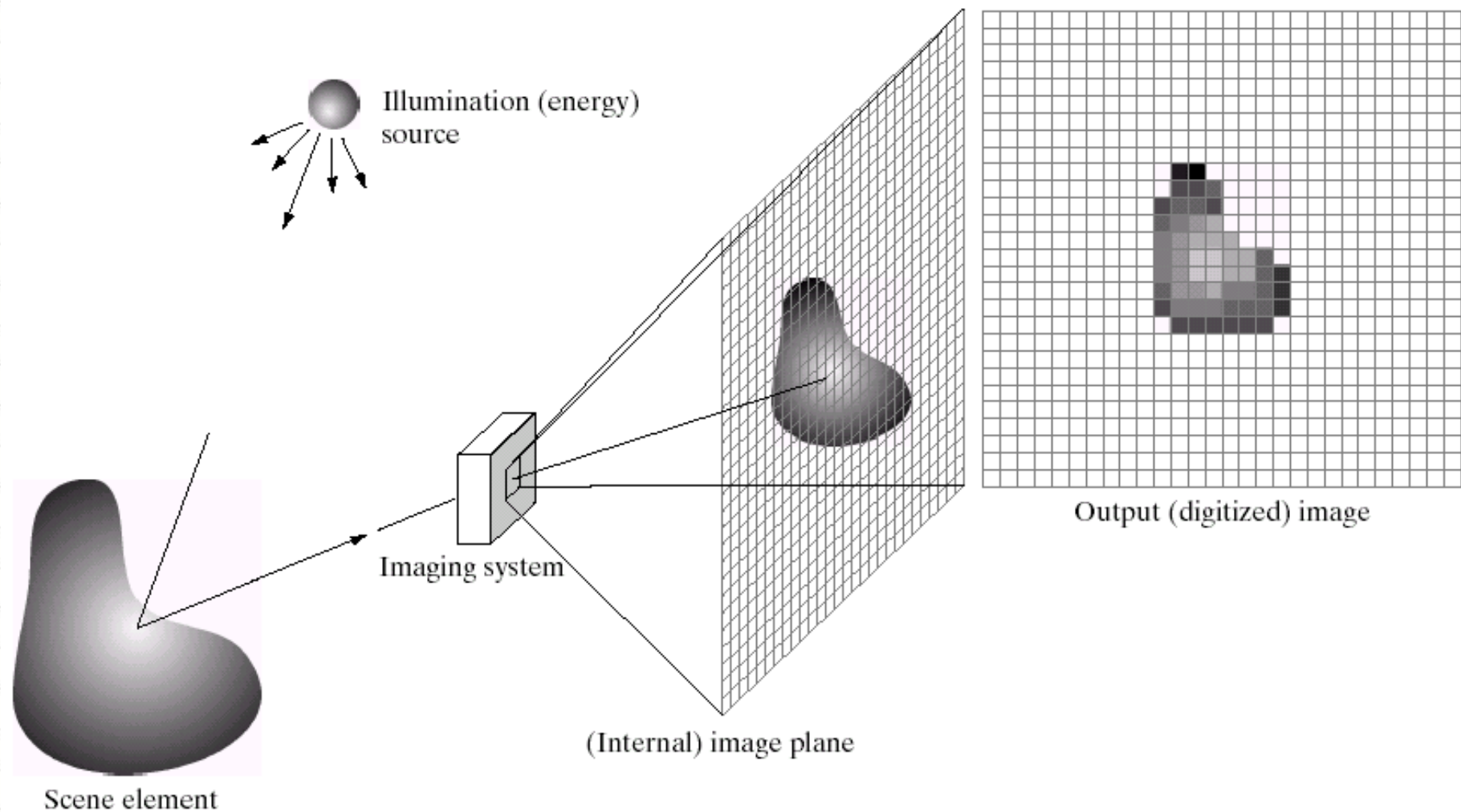
در ادامه دو مفهوم نمونه برداری و کوانتیزاسیون را در هنگام عکس برداری با دوربین دیجیتال توضیح می دهیم.



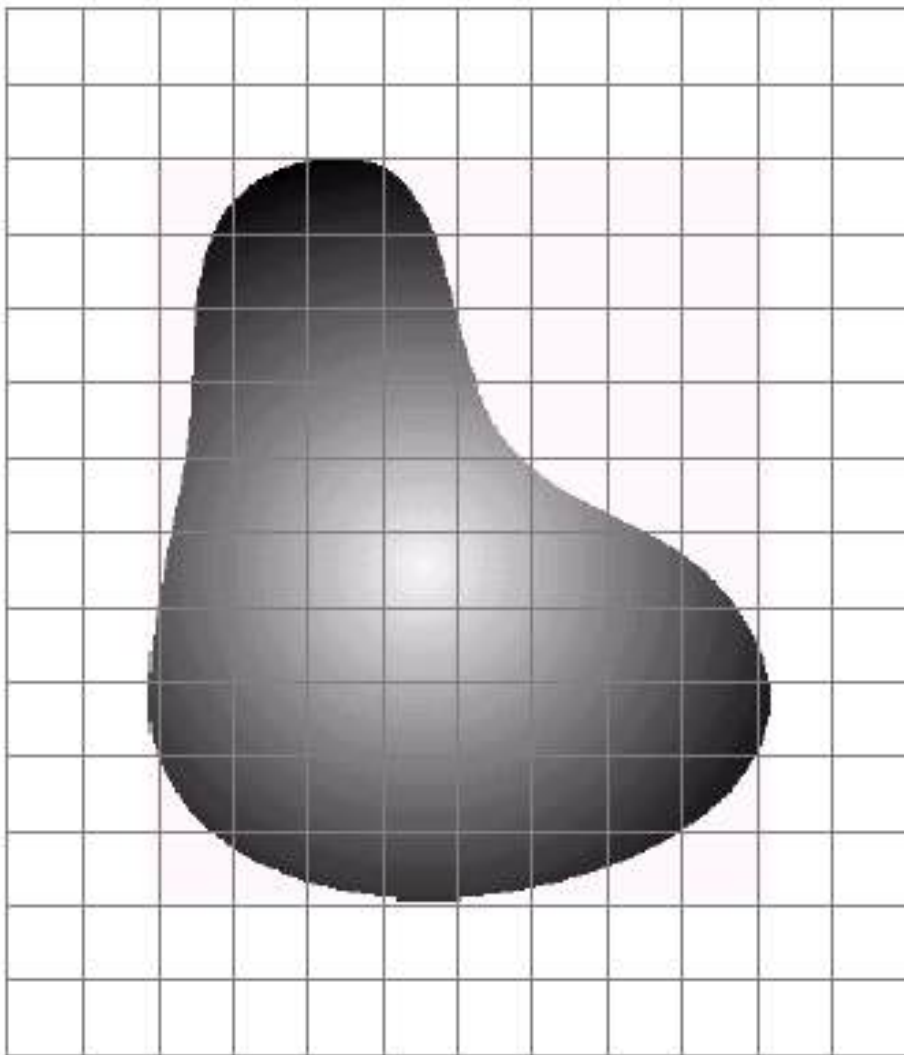
تصویر برداری با دوربین های دیجیتال (۱)



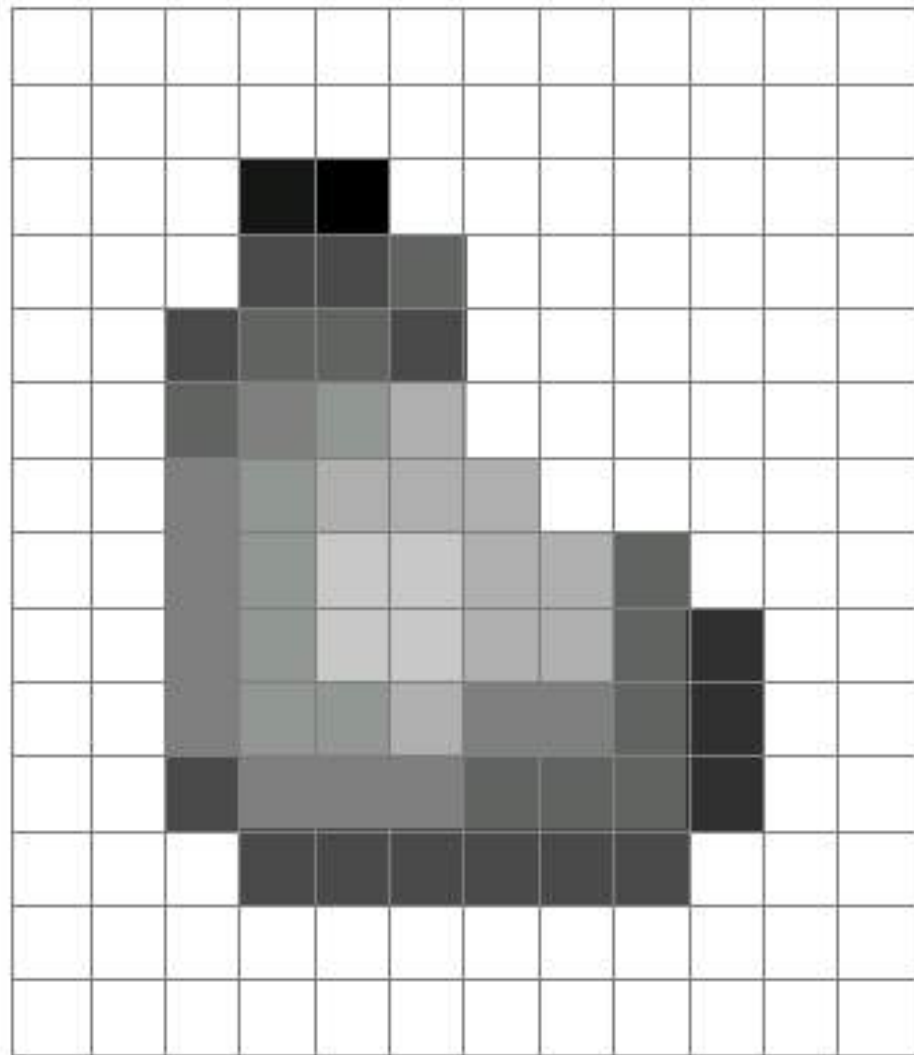
تصویر برداری با دوربین‌های دیجیتال (۲)



تصویر برداری با دوربین های دیجیتال (۳)



(الف) تصویر پیوسته،



نتیجه نمونه برداری و کوانتیزه-سازی
تصویر

دقت در نمونه برداری و کوانتیزاسیون

به دلیل انجام عمل‌های نمونه‌برداری و کوانتیزاسیون، یک تصویر دیجیتال یک **تقریب (Approximation)** از تصویر واقعی می‌باشد.

نمونه‌برداری و کوانتیزاسیون باعث **افت کیفیت و وضوح** تصویر می‌شود. هرچه تعداد نمونه‌برداری‌ها پیکسل‌ها و شدت‌های رنگ کمتر باشد، آن‌گاه وضوح و کیفیت تصویر دیجیتال کمتر خواهد بود.

برای **ارزیابی** کیفیت تصویر دیجیتال بعد از نمونه‌برداری و کوانتیزاسیون لازم است که پارامترهایی به صورت **کمی** گزارش شود که نشان دهنده **دقت** نمونه‌برداری و کوانتیزاسیون باشند.

این دو پارامتر عبارتند از

۱- دقت مکانی (Spatial Resolution)

۲- دقت شدت (Intensity Resolution)

دقت مکانی مشخص کننده آن است که عمل نمونه‌برداری با چه دقتی انجام شده است و دقت شدت مشخص کننده دقت در کوانتیزاسیون می‌باشد.

دقت مکانی

دقت مکانی به تعداد پیکسل‌های تصویر بستگی دارد. البته تعداد پیکسل‌ها به تنهایی مشخص کننده دقت مکانی نمی‌باشد، بلکه اندازه تصویر نیز مهم می‌باشد و باید علاوه بر تعداد پیکسل‌ها، اندازه تصویر را در تعیین دقت مکانی تاثیر داد.

غالبا **تعداد پیکسل‌ها در یک سطح واحد مانند یک اینچ مربع**، به عنوان دقت مکانی تصویر در نظر گرفته می‌شود.

- دقت مکانی با واحد bpi مشخص می‌شود که مخفف Bit Per Inche است.
- اگر یک اسکنر دارای دقت ۸ مگا پیکسل باشد، یعنی سنسورهای آن قادرند در هر اینچ مربع به تعداد ۸ میلیون پیکسل را نمونه برداری کند.
- به طور مشابهی دقت یک دوربین دیجیتال سنجیده می‌شود.
- دقت مکانی تصاویر ایجاد شده توسط دوربین دیجیتال و یا اسکنرها به حساسیت سنسورهای این دستگاه‌ها بستگی دارد.

مثال برای دقت مکانی

در زیر یک گل با تعداد نمونه برداری مکانی متفاوت تصویربرداری شده است. تصویر اول در هر سطر و ستون دارای ۱۰۲۴ پیکسل است و تصویر بعدی ۵۱۲ پیکسل در هر سطر و هر ستون دارد و ...



1024



512



256



128



64

32

هر چه تعداد پیکسل‌ها کمتر باشد،
آن‌گاه تصویر کوچک‌تر است و وضوح
کمتری دارد.

مثال برای دقت مکانی

اندازه تصویر 128×128 درست یک هشتم اندازه تصویر 1024×1024 است. حال اگر اندازه پیکسل‌ها در تصویر 128×128 را هشت برابر کنیم، آن‌گاه تصاویر هم‌اندازه می‌شوند. اما در این بین دقت تصویر کاهش می‌یابد.

1024×1024



128×128



دقت مکانی

1024x1024



512x512



256x256



128x128



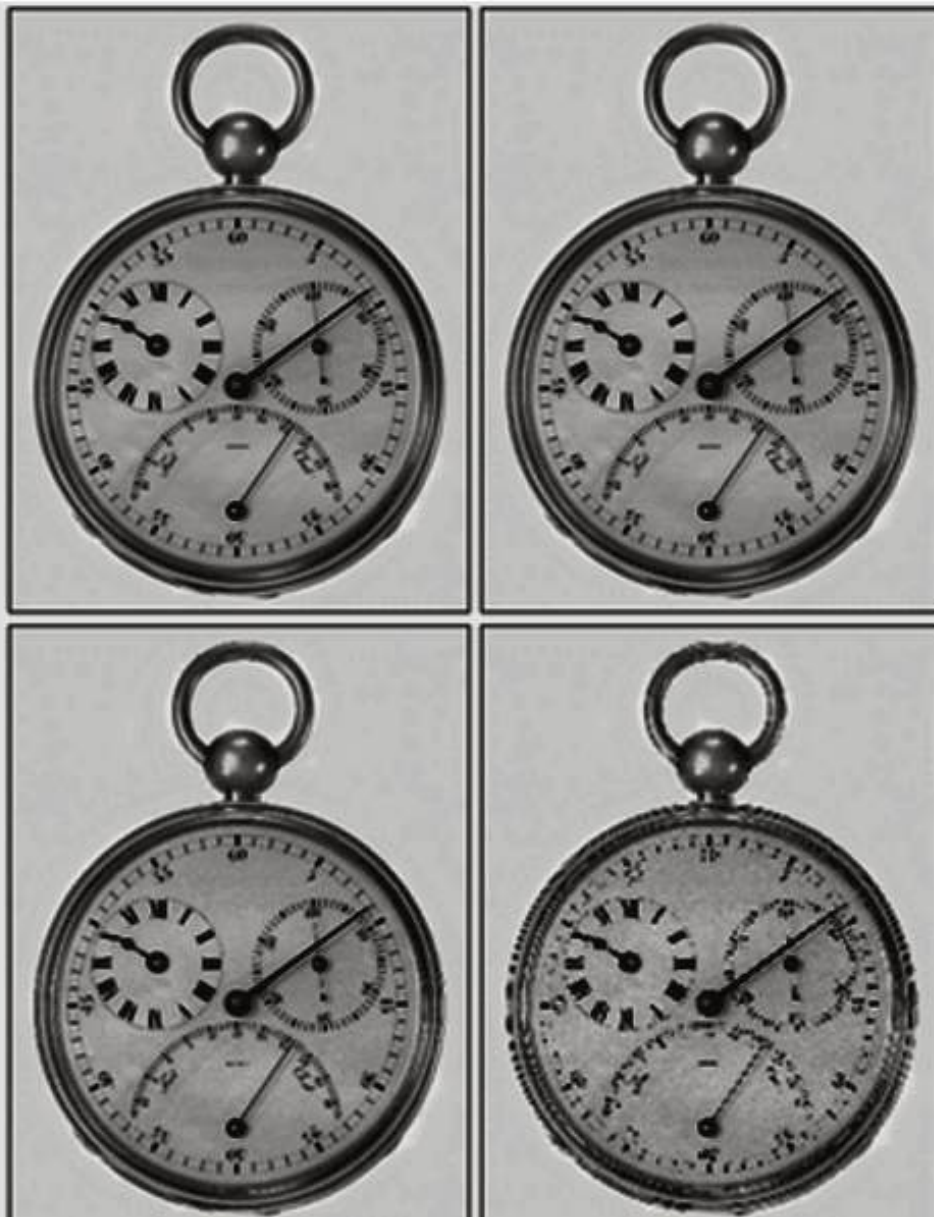
64x64



32x32



تأثیر رزولوشن در تصویر



اثرات کاهش رزولوشن مکانی.

تصاویر نشان داده شده در

(الف) 1250 dpi ،

(ب) 300 dpi ،

(ج) 150 dpi و

(د) 72 dpi .

مرزهای سیاه نازک برای جداسازی

اضافه شده اند و بخشی از داده ها

نیستند.

دقت شدت

دقت شدت همان تعداد سطوح خاکستری ممکن در تصویر دیجیتال است که برابر است با دو به توان عمق بیت.

بنابراین عمق بیت مبین دقت شدت در تصویر می باشد و هر چه عمق بیت بیشتر باشد، آن گاه دقت شدت تصویر نیز بالاتر است.

در اسلایدهای بعد تاثیر دقت شدت به نمایش در آورده شده است.

یک تصویر خاکستری با عمق‌های متفاوت

256 grey levels (8 bits per pixel)



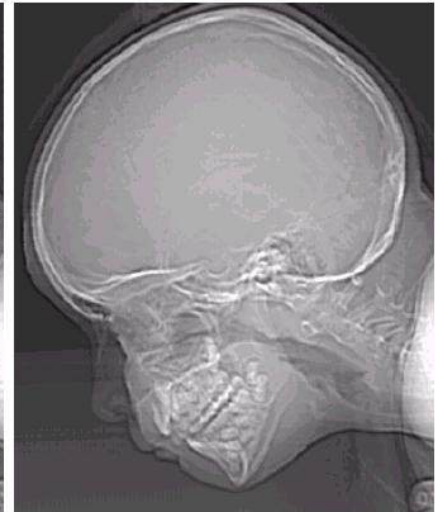
128 grey levels (7 bpp)



64 grey levels (6 bpp)



32 grey levels (5 bpp)



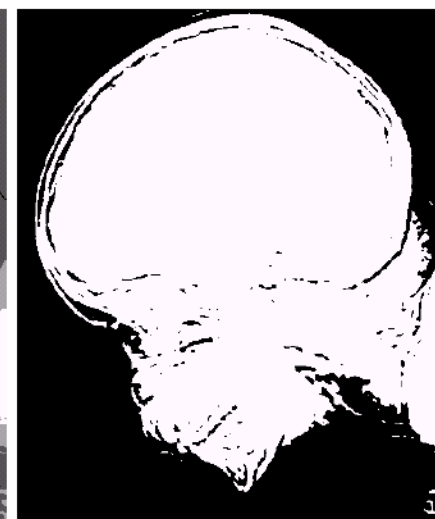
16 grey levels (4 bpp)



8 grey levels (3 bpp)



4 grey levels (2 bpp)



2 grey levels (1 bpp)

یک تصویر خاکستری با عمق‌های متفاوت

تصویر اصلی با عمق بیت ۸



تصویر با عمق بیت ۷



یک تصویر خاکستری با عمق‌های متفاوت

تصویر اصلی با عمق بیت ۸



تصویر با عمق بیت ۶



یک تصویر خاکستری با عمق‌های متفاوت

تصویر اصلی با عمق بیت ۸



تصویر با عمق بیت ۵



یک تصویر خاکستری با عمق‌های متفاوت

تصویر اصلی با عمق بیت ۸



تصویر با عمق بیت ۴



یک تصویر خاکستری با عمق‌های متفاوت

تصویر اصلی با عمق بیت ۸



تصویر با عمق بیت ۳

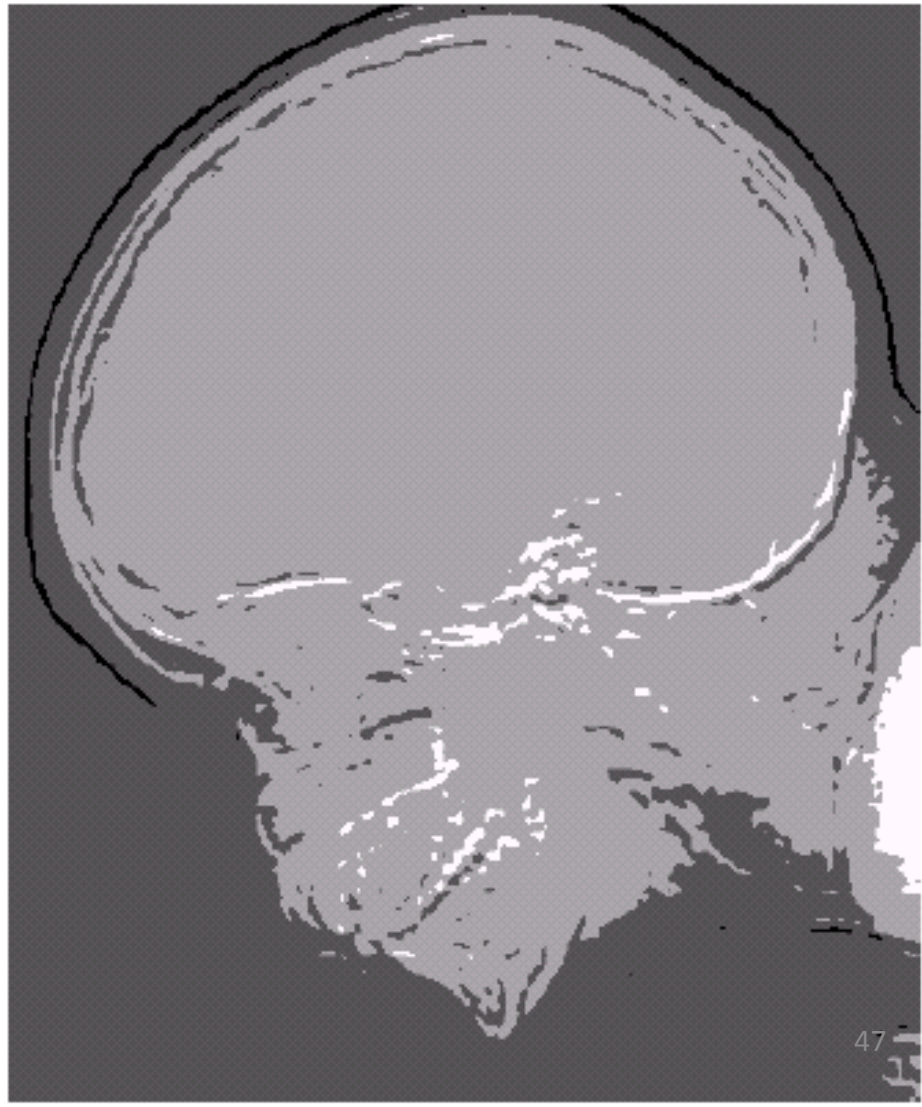


یک تصویر خاکستری با عمق‌های متفاوت

تصویر اصلی با عمق بیت ۸



تصویر با عمق بیت ۲



یک تصویر خاکستری با عمق‌های متفاوت

تصویر اصلی با عمق بیت ۸



تصویر با عمق بیت ۱



اهمیت دقت مکانی بیشتر است یا دقت شدت؟

در پاسخ به این سوال می‌توان گفت که هر دو نوع دقت در وضوح تصویر نقش دارند. اما در برخی تصاویر که دارای جزئیات زیادی می‌باشند و به اصطلاح شلوغ هستند، آن‌گاه دقت مکانی از اهمیت بیشتری برخوردار است. در تصاویری که سایه‌های خاکستری زیادی وجود دارد و جزئیات کمی دارد، آن‌گاه دقت شدت مناسب‌تر می‌باشد.



تصویر با جزئیات کم.



تصویر با جزئیات متوسط.



تصویر با جزئیات نسبتاً زیاد.

چه مقدار دقت مکانی و شدت مناسب است؟

طبیعی است که هرچه دقت مکانی و شدت تصویر بیشتر باشد، آن گاه تصویر مطلوب تر است. اما افزایش این دقت ها منجر به بالا رفتن حجم مورد نیاز برای ذخیره و افزایش هزینه می شود. لذا این دقت ها باید به نحو مناسبی انتخاب شوند. اندازه مناسب برای دقت مکانی و شدت به کاربری تصویر بستگی دارد. اگر جزئیات تصویر اهمیت داشته باشد، آن گاه لازم است که دقت مکانی بالا باشد. اما در صورتی که کلیت تصویر مهم باشد، آن گاه دقت چندان مهم نمی باشد.



?

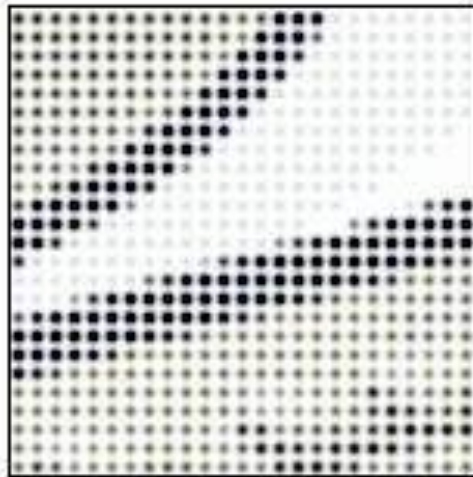


اگر هدف شمارش خودروها باشد، آن گاه تصویر سمت راست مناسب تر است. ولی اگر هدف خواندن پلاک خودروها باشد، آن گاه تصویر سمت چپ مطلوب تر است.

مطالب تکمیلی

رندر شدن تصاویر دیجیتال

از.

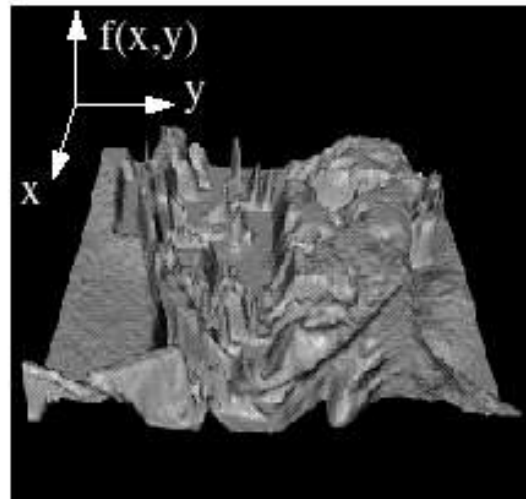
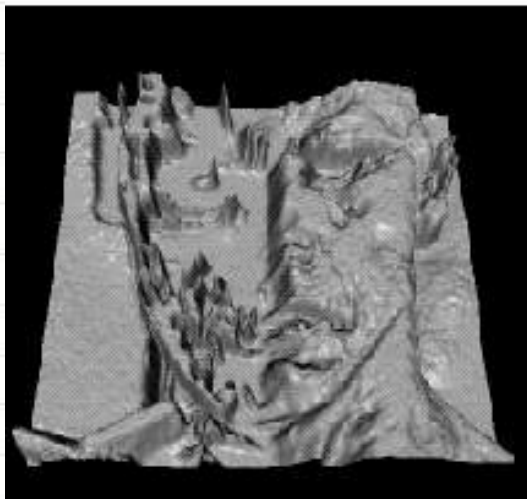


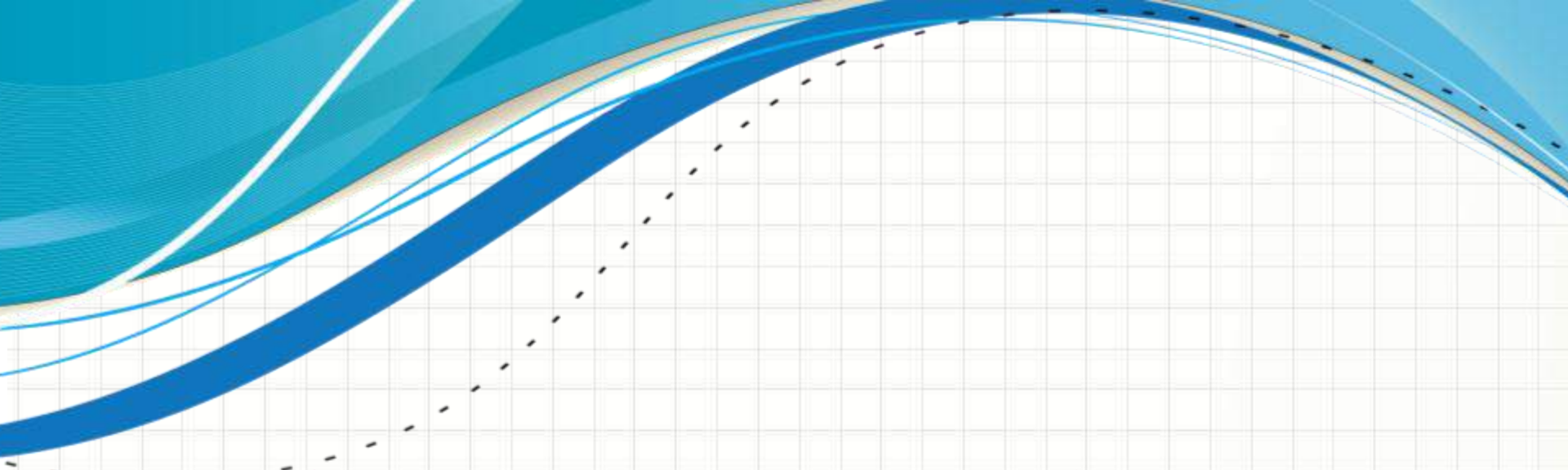
A pixel does not need to be rendered as a small square. This image shows alternative ways of reconstructing an image from a set of pixel values, using dots, lines, or smooth filtering.

تصاویر برداری (Vector Image)

از.

Image as a Function





مقدمه‌ای بر درونیابی

Introduction to
Interpolation

معرفی درونیابی

- درونیابی (Interpolation) به معنی یافتن تابعی که از نقاط مشخص و معلومی عبور کند.
- اگر مقدار یک تابع فقط در تعدادی نقطه مشخص باشد، آن گاه با درونیابی می توان مقدار تابع در نقاط دیگر را نیز تقریب زد.
- درونیابی انواع متفاوتی دارد. از جمله
 - درونیابی چند جمله ای (Polynomial Interpolation)
 - درونیابی اسپلاینی (Spline Interpolation)
 - درونیابی نزدیک ترین همسایگی (nearest)
 - درونیابی دو خطی (Bilinear Interpolation)
 - درونیابی دو مکعبی (Bicubic Interpolation)

در اینجا درونیابی توابع دو متغیره مدنظر می باشد. یعنی یک تابع دو متغیره $f(x,y)$ داریم که در چند نقطه $(x_i, y_j), i = 1, \dots, n, j = 1, \dots, m$ مقدار آن را می دانیم و می خواهیم مقدار تابع را در نقاط دیگر به دست آوریم.

معرفی درونیابی

- در پردازش تصویر بیشتر از
 - درونیابی نزدیکترین همسایگی،
 - درونیابی دوخطی و
 - درونیابی دوماکعبی
- استفاده می شود.

- این نوع درونیابی ها برای داده های دوبعدی استفاده می شوند.
- در این نوع درونیابی ها مقدار تابع در یک **مش منظم (Regular Mesh)** مشخص است و می خواهیم مقدار تابع در برخی نقاط واقع در بین مش ها را تقریب بزنیم.

درونیابی-مثال

به عنوان مثال فرض کنید که مقدار تابع $f(x,y)$ به صورت زیر در برخی نقاط مشخص و داده شده است.

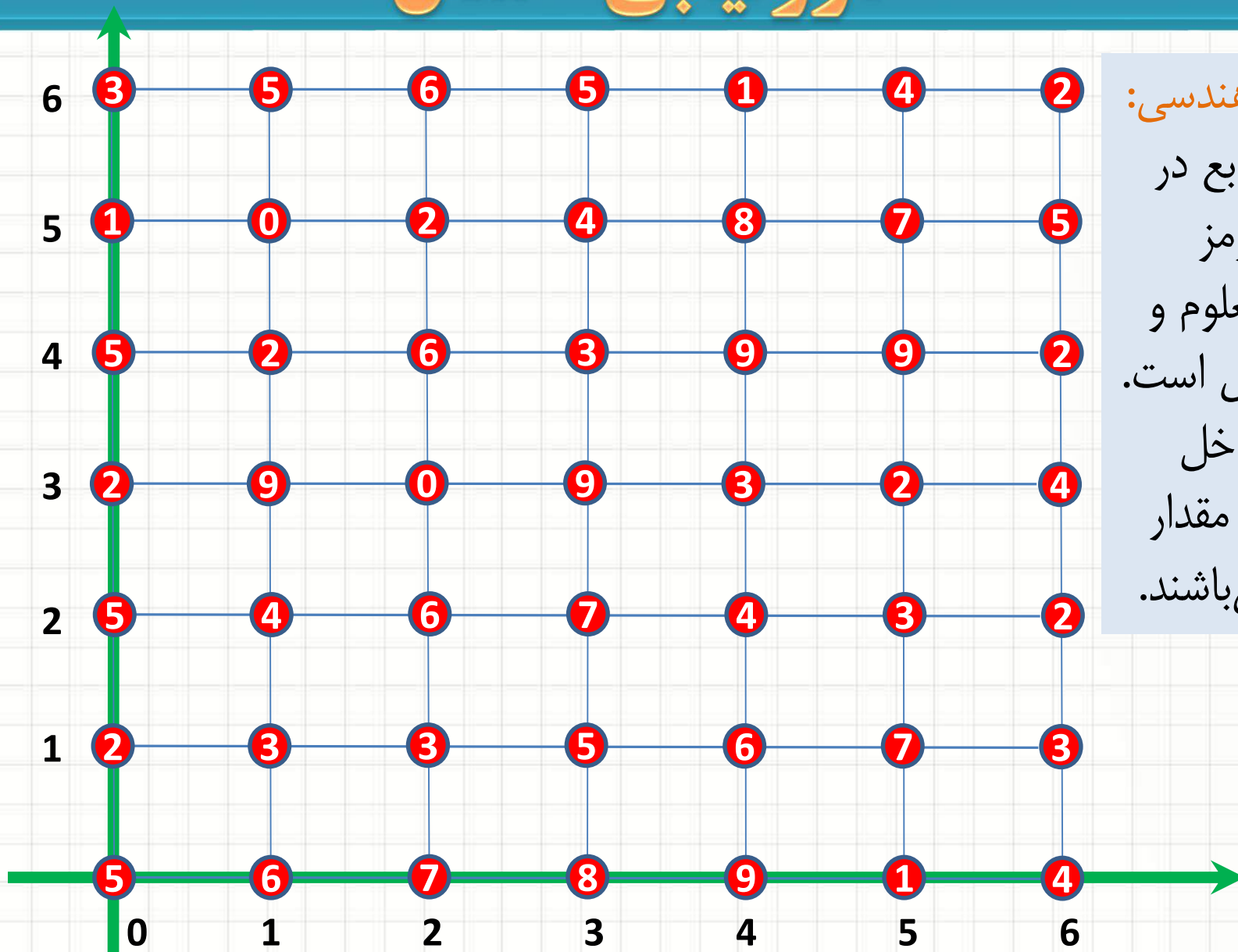
$y_j \backslash x_i$	0	1	2	3	4	5	6
0	3	5	6	5	1	4	2
1	1	0	2	4	8	7	5
2	5	2	6	3	9	9	2
3	2	9	0	5	3	2	4
4	5	4	6	7	4	3	2
5	2	3	3	5	6	7	3
6	5	6	7	8	9	1	5

مقدار تابع f به ازای
 $x = 1$ و $y = 3$
برابر 4 است. یعنی
 $f(1,3) = 4$

مقدار تابع f به ازای
 $x = 4$ و $y = 5$
برابر 3 است. یعنی
 $f(4,5) = 3$

حال می‌خواهیم مقدار تابع f را در نقاط دیگر مثلاً نقطه $(0.4, 5.5)$ و ... به دست آوریم.

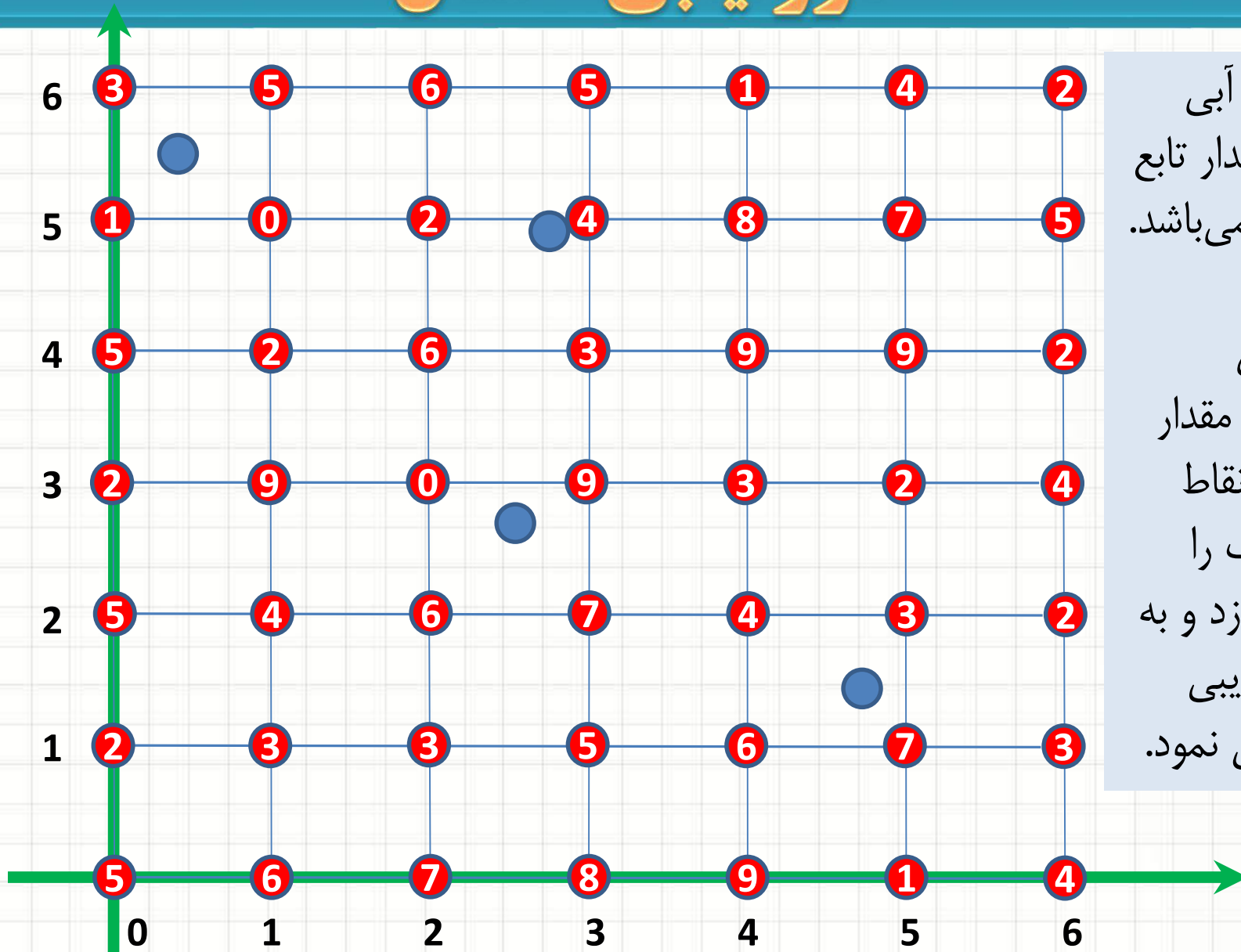
درونیابی-مثال



تجسم هندسی:

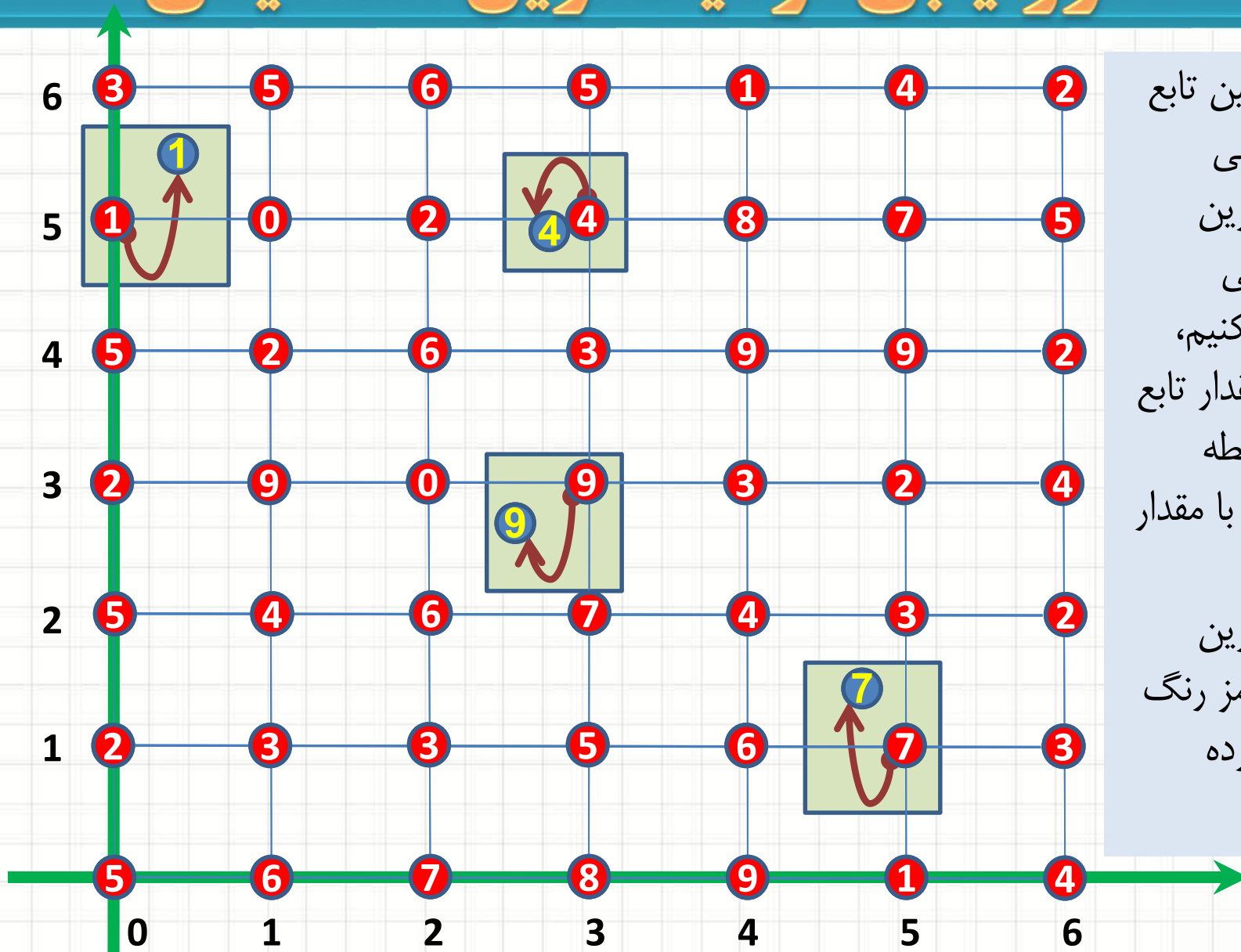
مقدار تابع در
نقاط قرمز
رنگ معلوم و
مشخص است.
اعداد داخل
دایره‌ها مقدار
تابع می‌باشند.

درونیابی-مثال



در نقاط آبی
رنگ مقدار تابع
معلوم نمی باشد.
باکمک
درونیابی
می توان مقدار
تابع در نقاط
آبی رنگ را
تخمین زد و به
طور تقریبی
مشخص نمود.

درونیابی نزدیکترین همسایگی



اگر تخمین تابع
از درونیابی
نزدیکترین
همسایگی
استفاده کنیم،
آن گاه مقدار تابع
در هر نقطه
آبی رنگ با مقدار
تابع در
نزدیکترین
نقطه قرمز رنگ
تخمین زده
می شود.

درونیابی نزدیکترین همسایگی

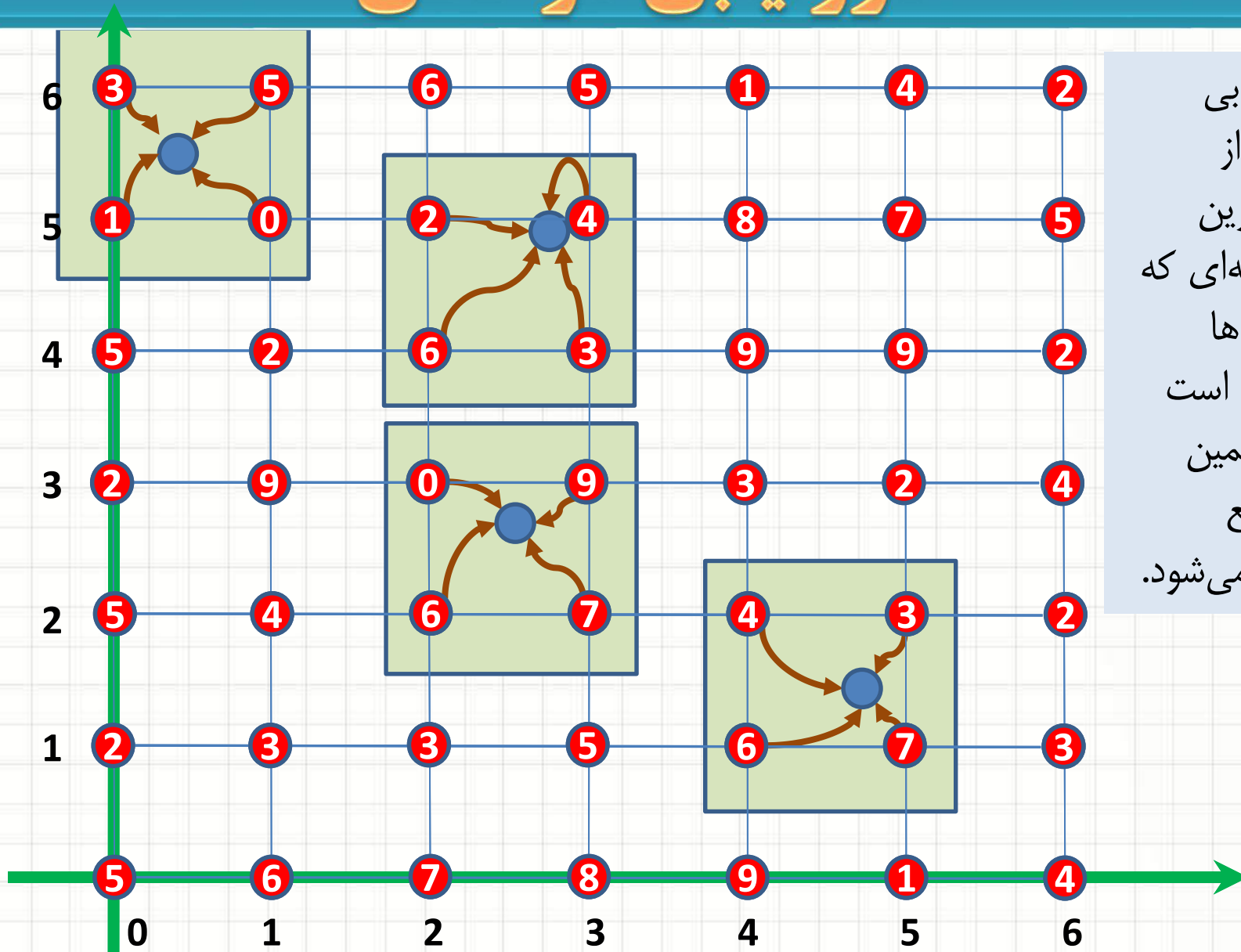
جمع‌بندی:

در درونیابی نزدیکترین همسایگی، اگر مقدار تابع f در نقاط مش منظم $(x_i, y_i), i = 1, \dots, n, j = 1, \dots, m$

داده شده باشد و بخواهیم مقدار f را در نقطه (p, q) به دست آوریم، آن گاه

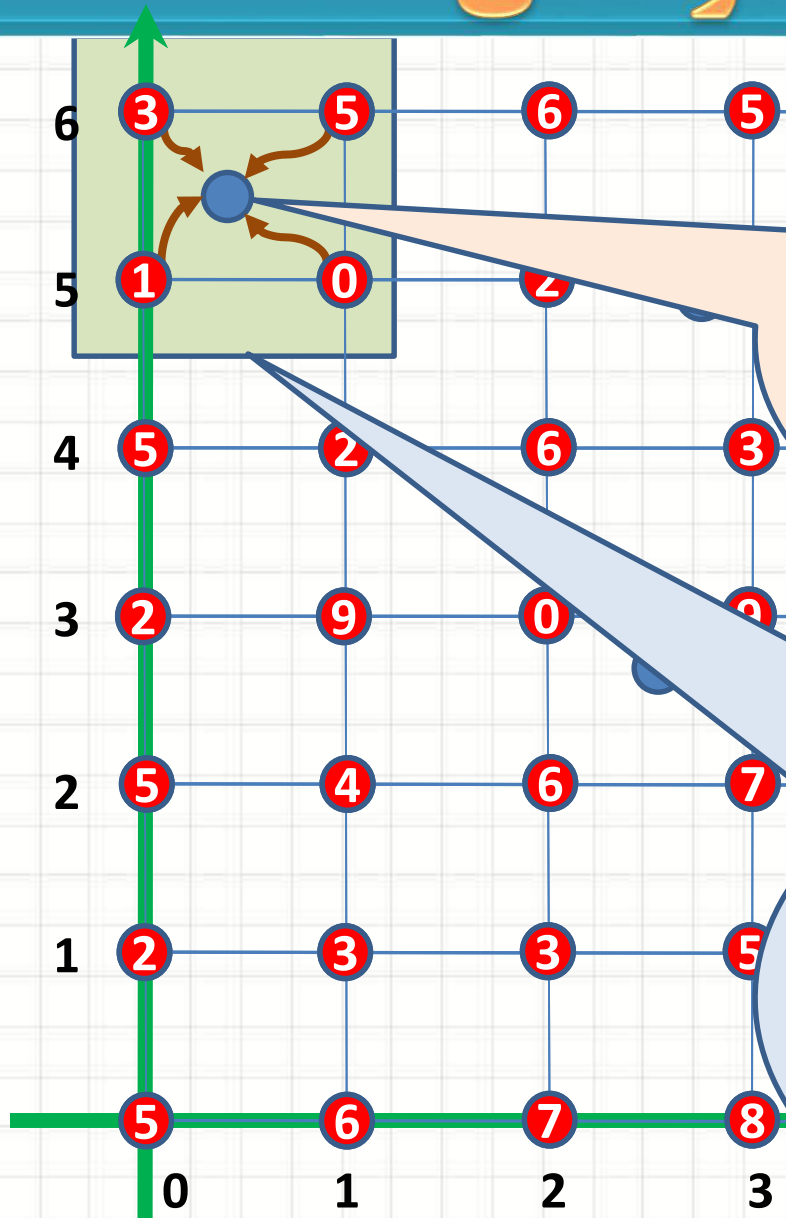
- در ابتدا باید مشخص کنیم که این نقطه در بین کدام چهار نقطه مش قرار می‌گیرد
- سپس مشخص کنیم که به کدامیک از این چهار نقطه نزدیک‌تر است
- در نهایت مقدار تابع در نزدیک‌ترین نقطه را به عنوان مقدار تقریبی تابع در (p, q) معرفی می‌کنیم.

درونیابی دوخطی



در درونیابی
دوخطی از
نزدیک‌ترین
چهار نقطه‌ای که
مقدار آن‌ها
مشخص است
برای تخمین
مقدار تابع
استفاده می‌شود.

درونیابی دوخطی



مختصات این نقطه

$(0.4, 5.5)$

است و می‌خواهیم مقدار این
نقطه را با استفاده از درونیابی
دوخطی تقریب بزنیم.

در درونیابی دوخطی از چهار
نزدیک‌ترین نقطه استفاده می‌شود.

این نقاط عبارتند از

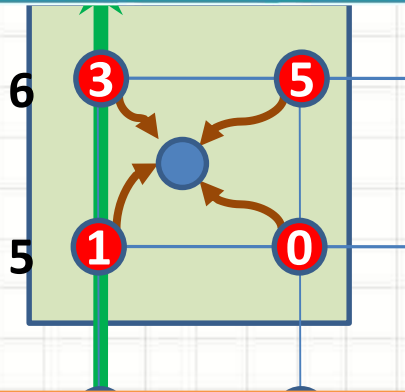
نقطه $(0, 5)$ با مقدار 1

نقطه $(0, 6)$ با مقدار 3

نقطه $(1, 5)$ با مقدار 0

نقطه $(1, 6)$ با مقدار 5

انجام درونیابی دوخطی



در درونیابی دوخطی تابع درونیاب به صورت زیر فرض می شود

$$F(x, y) = ax + by + cxy + d$$

حال ضرایب از حل دستگاه زیر به دست می آیند

$$\begin{cases} f(0,5) = 1 \\ f(0,6) = 3 \\ f(1,5) = 0 \\ f(1,6) = 5 \end{cases} \Rightarrow \begin{cases} 0a + 5b + 0c + d = 1 \\ 0a + 6b + 0c + d = 3 \\ 1a + 5b + 5c + d = 0 \\ 1a + 6b + 6c + d = 5 \end{cases} \Rightarrow \begin{cases} a = -16 \\ b = 2 \\ c = 3 \\ d = -9 \end{cases}$$

لذا تابع درونیاب به صورت زیر است

$$F(x, y) = -16x + 2y + 3xy - 9$$

حال مقدار تابع درونیاب در نقطه مورد نظر (0.4, 5.5) عبارت است از:

$$F(0.4, 5.5) = -16(0.4) + 2(5.5) + 3(0.4)(5.5) - 9 = 2.2$$

درونیابی دوخطی

جمع‌بندی:

در درونیابی دوخطی، اگر مقدار تابع f در نقاط مش منظم

$$(x_i, y_j), i = 1, \dots, n, j = 1, \dots, m$$

داده شده باشد و بخواهیم مقدار f را در نقطه (p, q) به دست آوریم، آن‌گاه در ابتدا باید مشخص کنیم که این نقطه در بین کدام چهار نقطه مش قرار می‌گیرد و سپس با توجه به مقدار تابع در این چهار نقطه مقدار تابع در نقطه (p, q) را تقریب می‌زنیم. در این درونیابی، تابع درونیاب به صورت دوخطی زیر در نظر گرفته می‌شود:

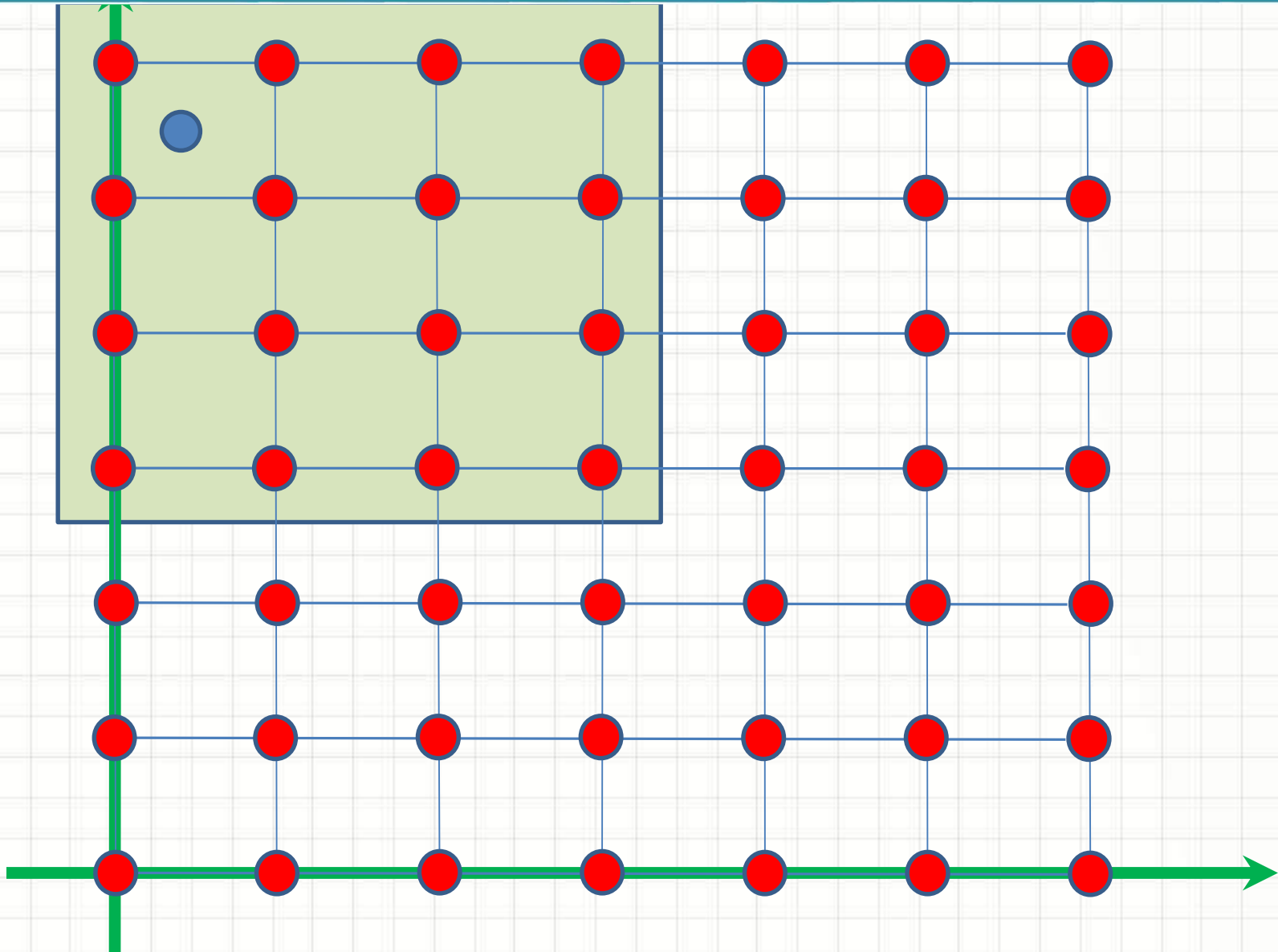
$$F(x, y) = ax + by + cxy + d$$

درونیابی دومربعی

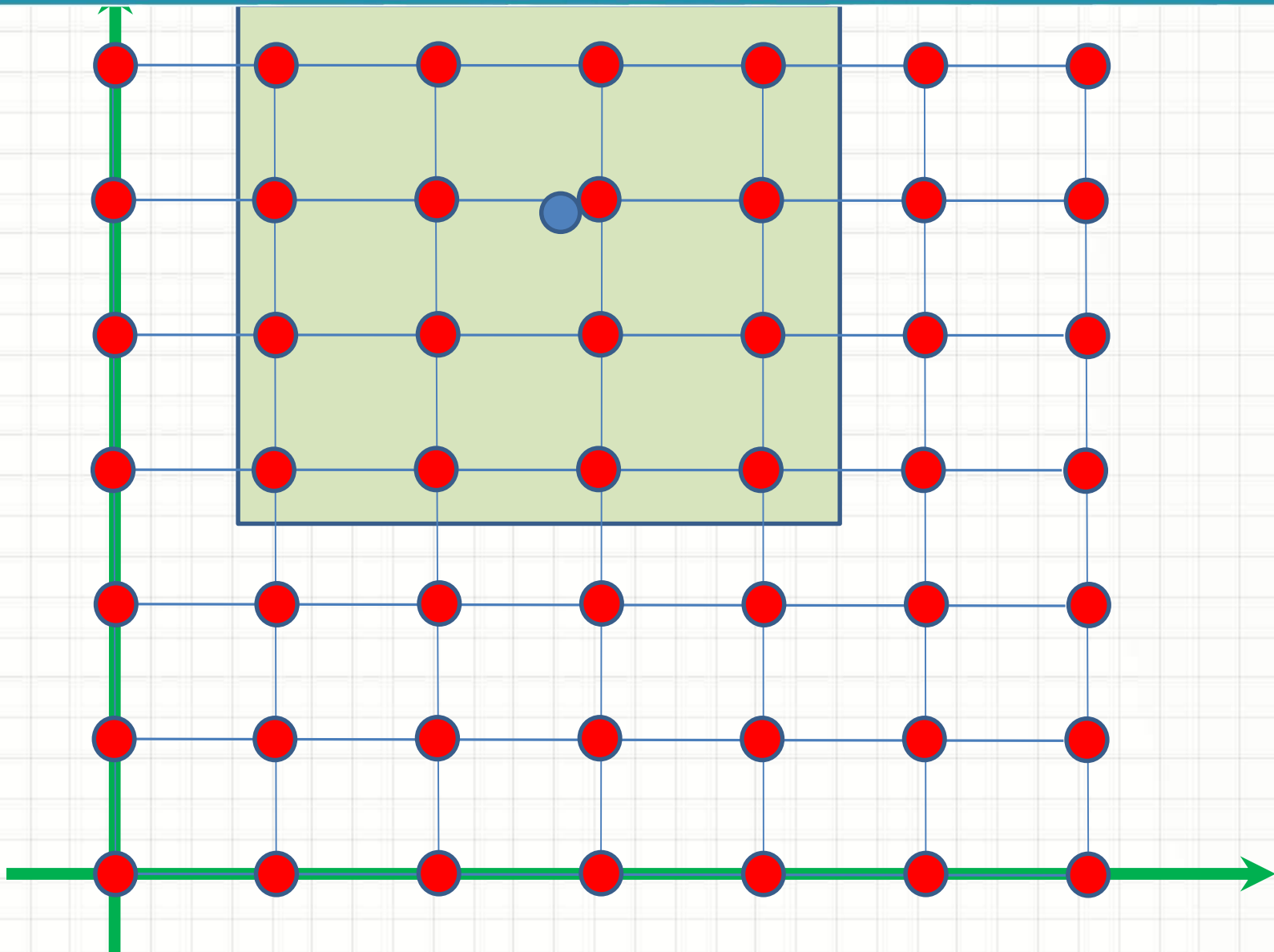
- درونیابی دومربعی (Bi-Cubic) همانند درونیابی دوخطی است، با این تفاوت که به جای استفاده از چهار نقطه همسایگی از **شانزده** نقطه همسایگی استفاده می‌کند.
- در این درونیابی، تابع درونیاب به صورت زیر در نظر گرفته می‌شود:

$$F(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

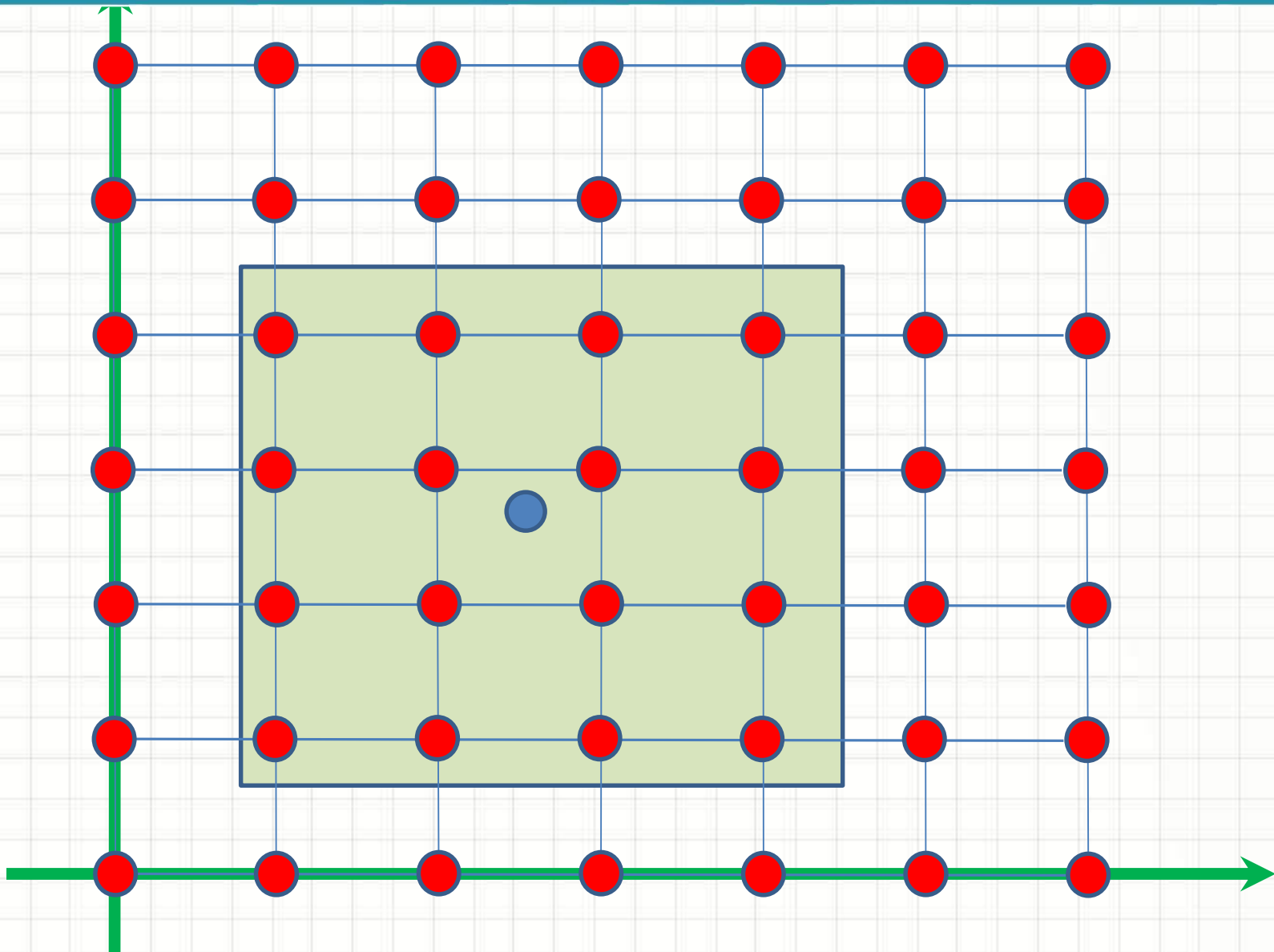
درون‌یابی دو مربعی



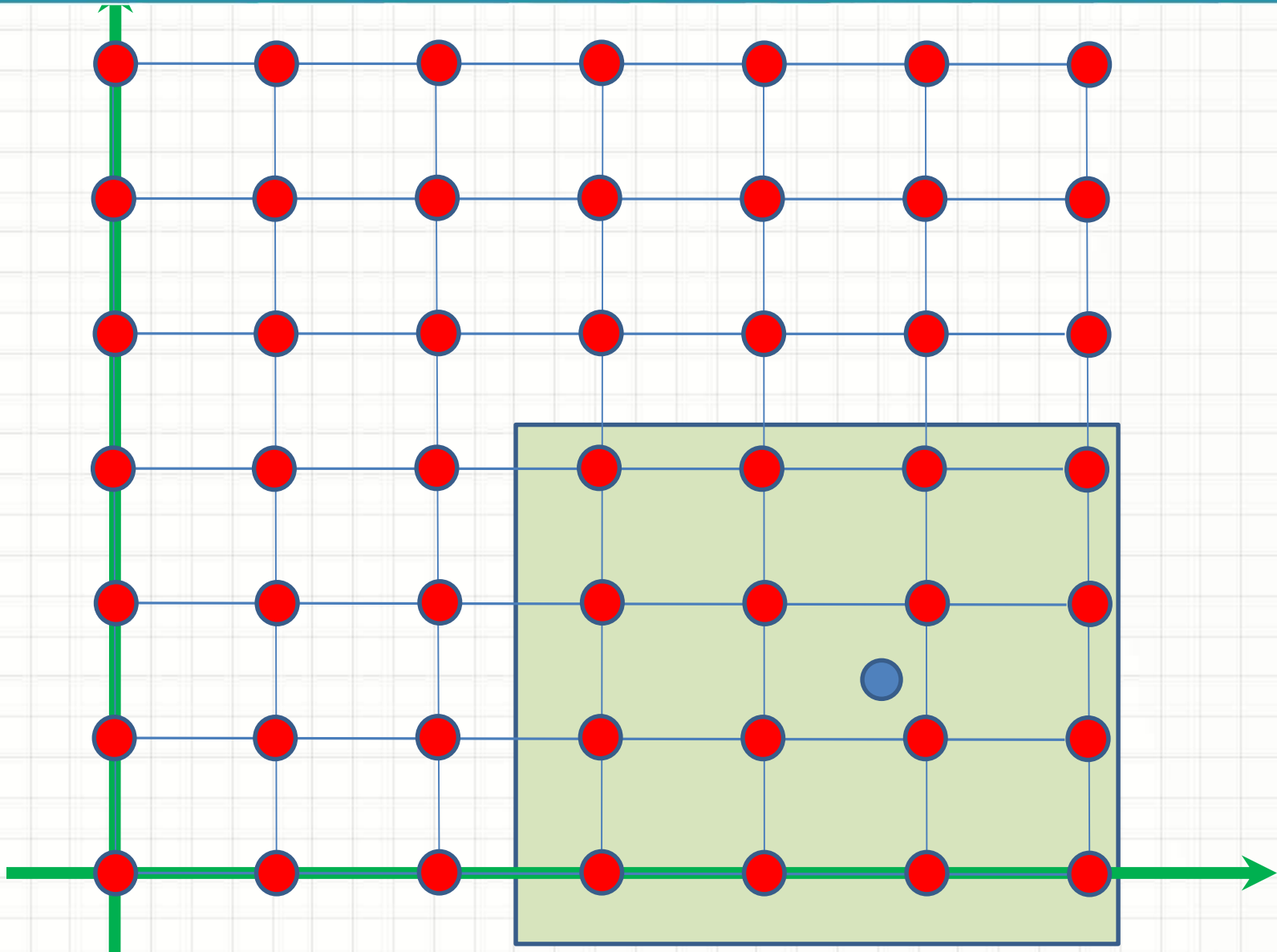
درون‌یابی دو مربعی



درونیابی دومربعی

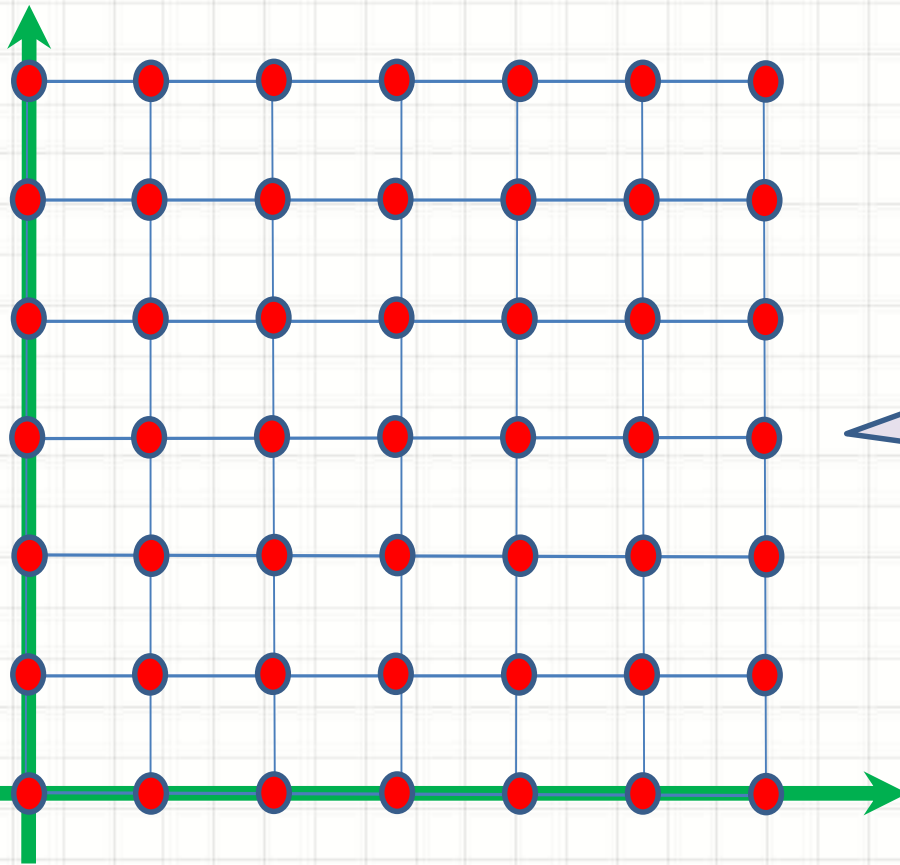


درونیابی دومربعی



درونیابی مبتنی بر مش نامنظم

در درونیابی‌های گفته شده در قبل فرض کردیم که مقدار تابع در یک مش منظم مستطیلی مشخص است. با این فرض بحث درونیابی پیش رفت.

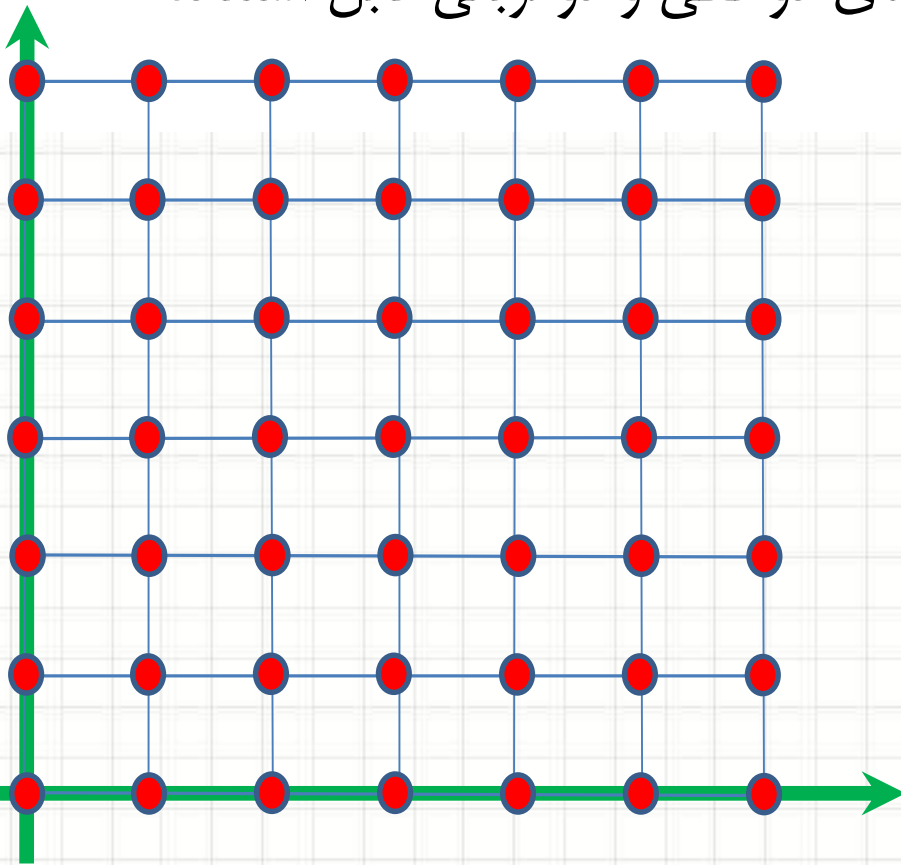


در مش منظم نقاطی که مقدار تابع در آن‌ها معلوم است به صورت یک صف منظم می‌باشند

درونیابی مبتنی بر مش نامنظم

در درونیابی‌های گفته شده در قبل فرض کردیم که مقدار تابع در یک مش منظم مستطیلی مشخص است. با این فرض بحث درونیابی پیش رفت.

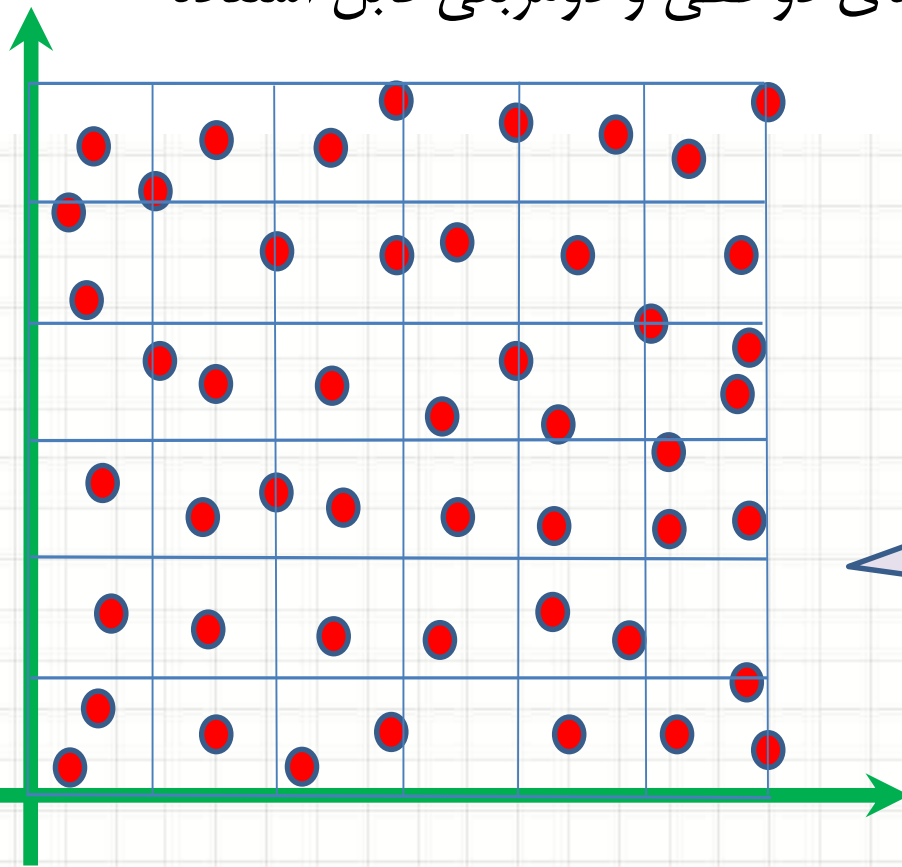
اما سوالی که مطرح است آن است که اگر مقدار تابع در نقاطی مشخص باشد که تشکیل یک مش منظم ندهند، آیا باز هم درونیابی‌های دوخطی و دومربعی قابل استفاده می‌باشند؟



درونیابی مبتنی بر مش نامنظم

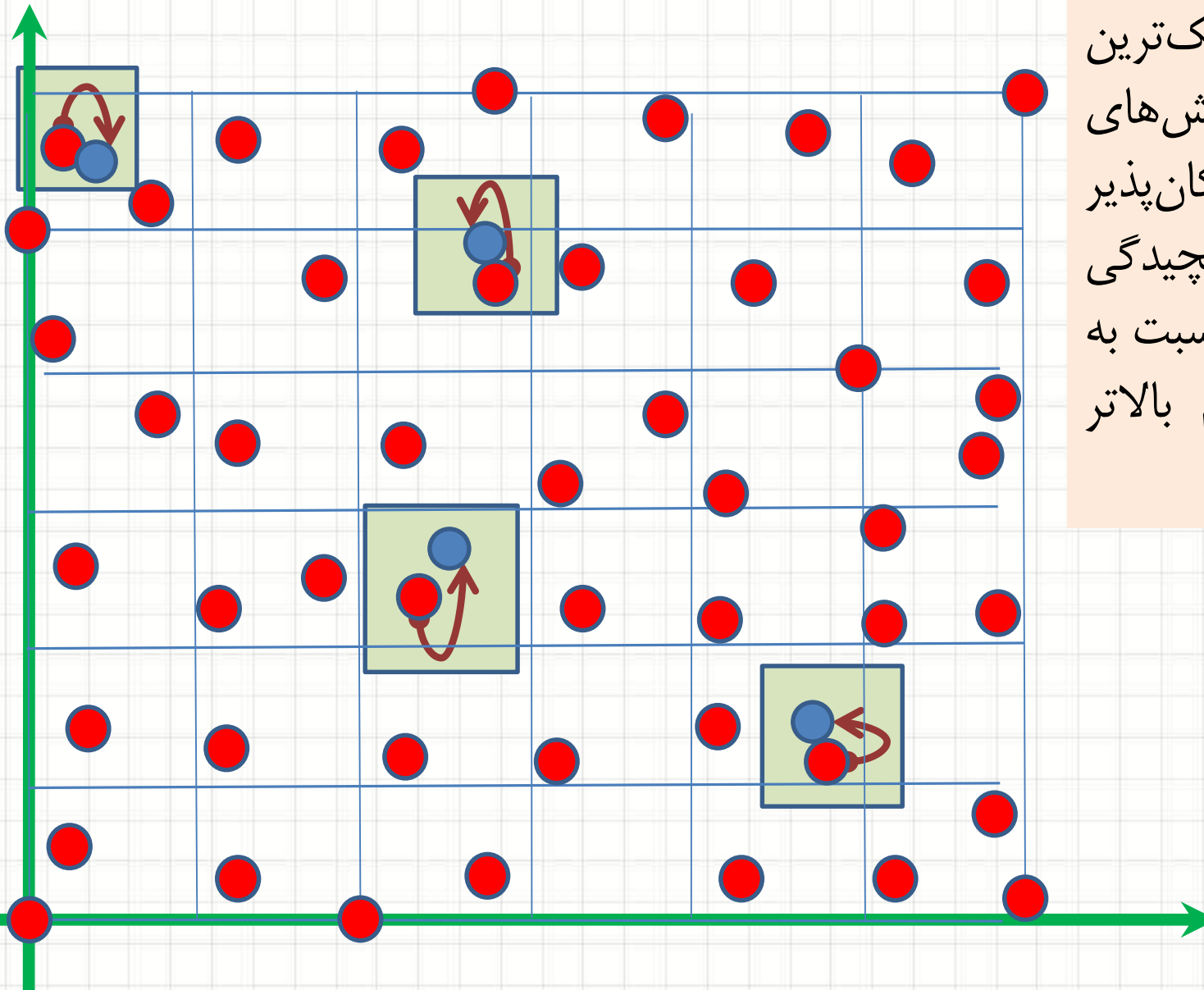
در درونیابی‌های گفته شده در قبل فرض کردیم که مقدار تابع در یک مش منظم مستطیلی مشخص است. با این فرض بحث درونیابی پیش رفت.

اما سوالی که مطرح است آن است که اگر مقدار تابع در نقاطی مشخص باشد که تشکیل یک مش منظم ندهند، آیا باز هم درونیابی‌های دوخطی و دومربعی قابل استفاده می‌باشند؟



در مش نامنظم نقاطی که مقدار تابع در آن‌ها معلوم است به صورت پراکنده‌اند و نظم خاصی ندارند

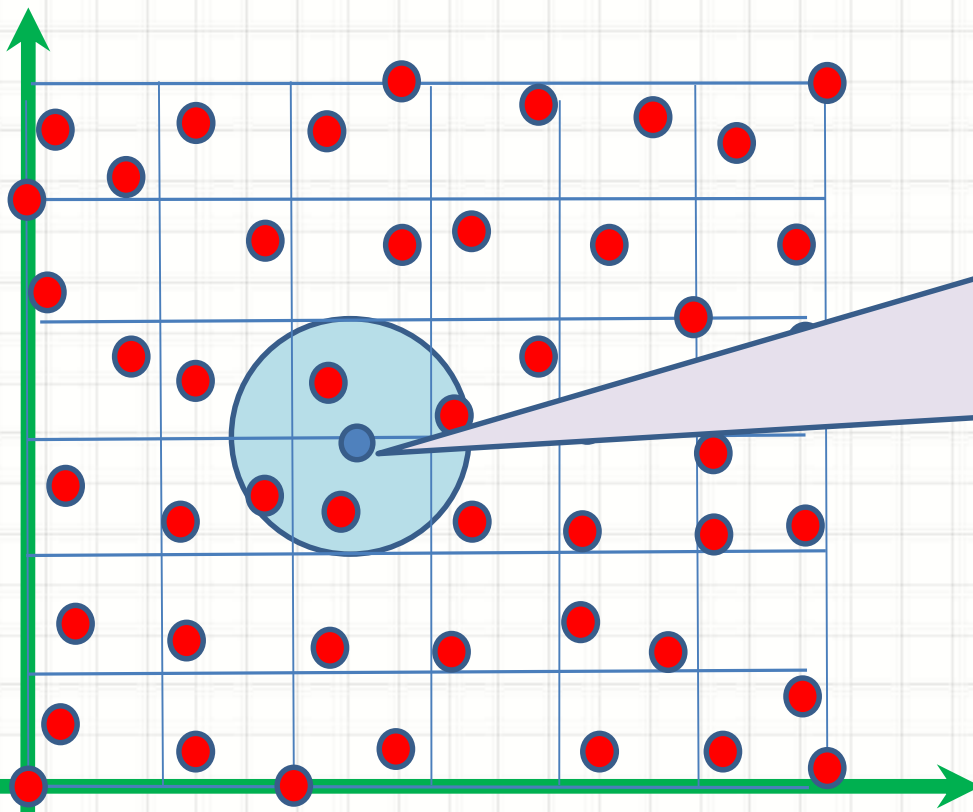
درونیابی نزدیکترین همسایگی در مش نامنظم



درونیابی نزدیکترین همسایگی در مش های نامنظم امکان پذیر است، البته پیچیدگی محاسباتی آن نسبت به مش های منظم بالاتر می باشد.

درونیابی دوخطی و دو مکعبی در مش نامنظم

در مش‌های نامنظم وجود و یکتایی درونیاب دوخطی و یا دو مربعی برقرار نمی‌باشد. به این معنی که ممکن است دستگاه‌های مربوطه دارای جواب نباشند و یا بی‌نهایت جواب داشته باشد. علاوه بر آن یافتن **نقاط همسایگی** در مش‌های نامنظم بسیار پیچیده‌تر می‌شود.



تضمینی وجود ندارد که بتوان مقدار تابع در نقطه آبی‌رنگ را بر حسب مقدار تابع در چهار نقطه اطرافش به روش درونیابی دوخطی به دست آورد.

عدم استفاده از درونیابی مبتنی بر مش نامنظم

- به دلایل فوق معمولاً از درونیابی مبتنی بر مش‌های نامنظم اجتناب می‌شود.

- البته نوع دیگری از درونیابی موسوم به

درونیابی توابع پایه شعاعی (Radial Basis Function)

وجود دارد که به برای درونیابی مبتنی بر مش‌های نامنظم مناسب می‌باشد.

مقایسه بین درونیابی‌ها

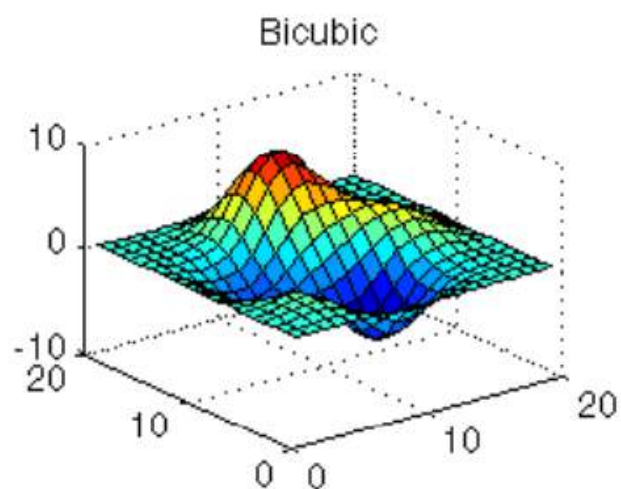
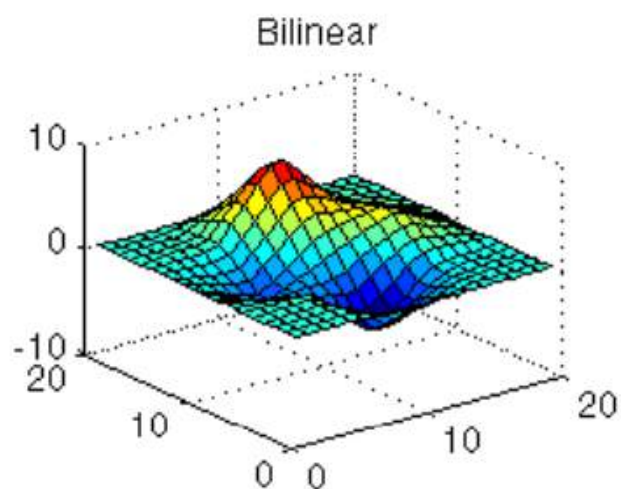
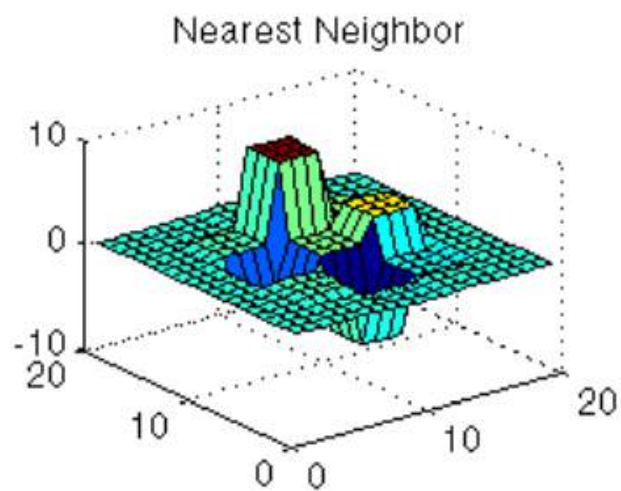
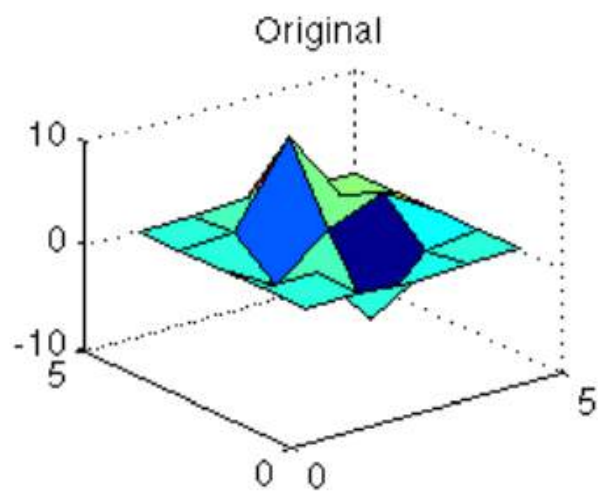
درونیابی نزدیک‌ترین همسایگی از دو درونیابی دیگر ساده‌تر است و پیچیدگی محاسباتی آن پایین‌تر است. همچنین برای مش‌های نامنظم نیز قابل اعمال است. لذا به لحاظ سادگی و حجم محاسبات درونیابی نزدیک‌ترین همسایگی در رتبه اول قرار دارد.

همین‌طور واضح است که درونیابی دوخطی دارای محاسبات کمتری نسبت به درونیابی دومربعی می‌باشد. در درونیابی دوخطی یک دستگاه چهارتایی حل می‌شود و در درونیابی دومربعی یک دستگاه شانزده‌تایی. لذا حجم محاسبات در درونیابی دومربعی نسبت به درونیابی دوخطی چشمگیر می‌باشد.

درونیابی یک نوع تقریب است و درونیابی‌های متفاوت منجر به نتایج متفاوت می‌گردد. در بین سه درونیابی گفته شده در فوق، درونیابی دومربعی منجر به نتایج دقیق‌تر می‌شود و درونیابی دوخطی نیز از درونیابی نزدیک‌ترین همسایگی دقیق‌تر است.

اگر به دنبال تعادل بین دقت و حجم محاسبات باشیم، آن‌گاه درونیابی دوخطی از بقیه مناسب‌تر است. به همین لحاظ این نوع درونیابی در عمل بیشتر از دو درونیابی دیگر استفاده می‌شود.

مقایسه بین درونیابی‌ها



تبدیلات مکانی هندسی در
پردازش تصویر

Spatial Geometric Transforms in
Image Processing

معرفی تبدیلات هندسی

در تبدیلات هندسی با **جابجایی پیکسل‌های** یک تصویر به تصویر دیگری تبدیل می‌شود. در واقع در تبدیلات هندسی یک تصویر به عنوان ورودی گرفته می‌شود و با جابجا کردن مکان پیکسل‌های آن، یک تصویر دیگر ایجاد می‌شود. توجه شود که در تبدیلات هندسی، شدت رنگ خاکستری پیکسل‌ها تغییری نمی‌کند.

به عنوان نمونه‌ای از تبدیلات هندسی می‌توان به قرینه‌سازی تصویر اشاره کرد

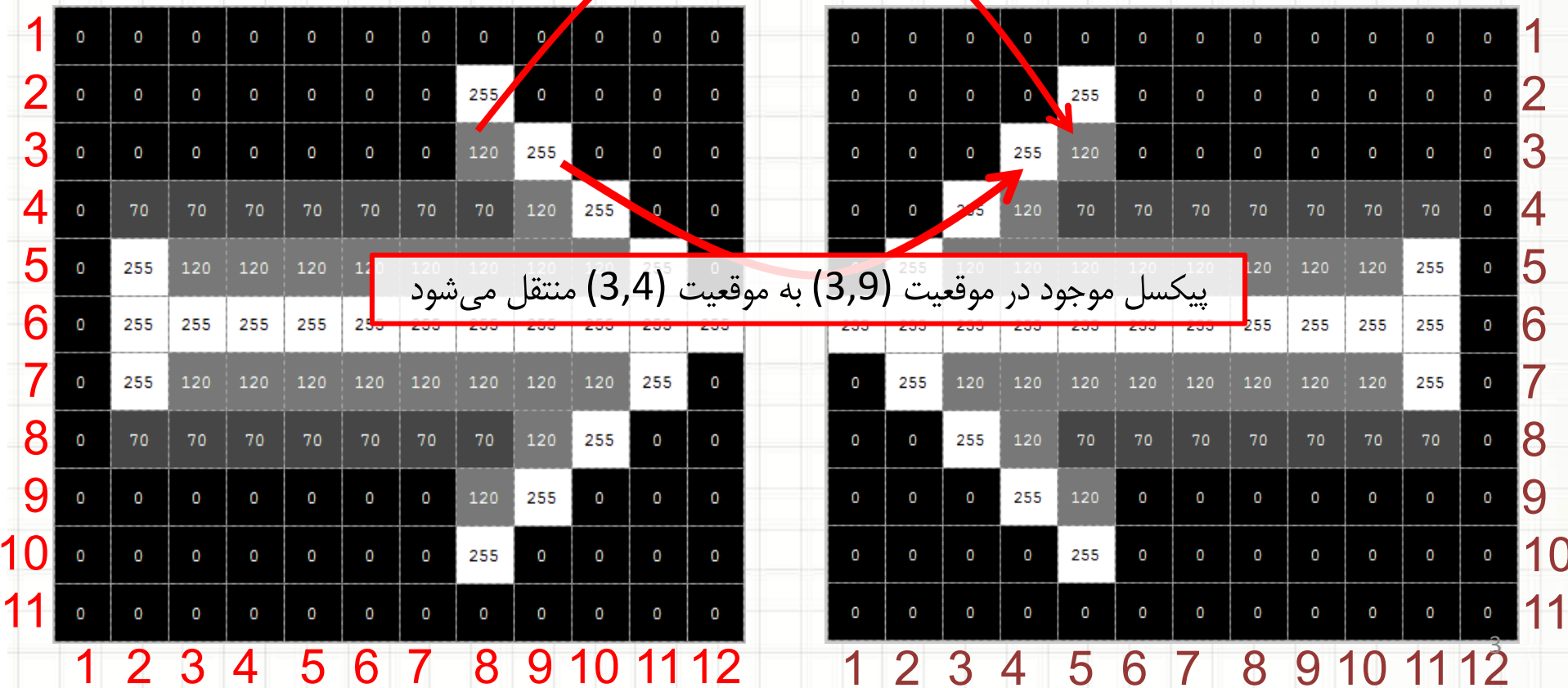


تبدیل
هندسی

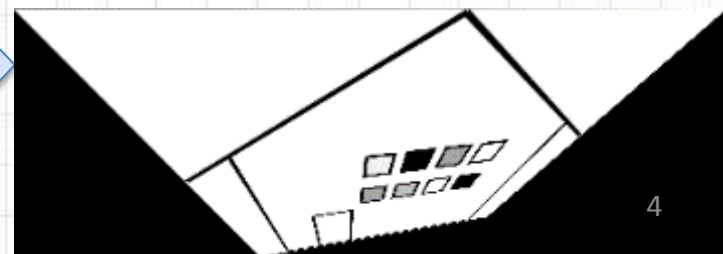
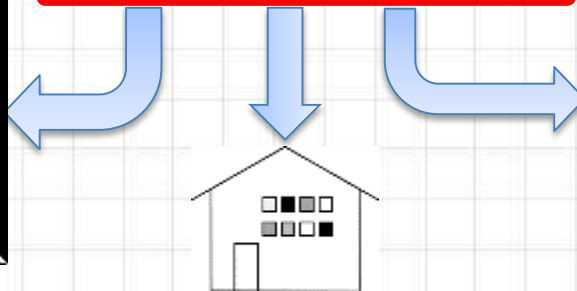
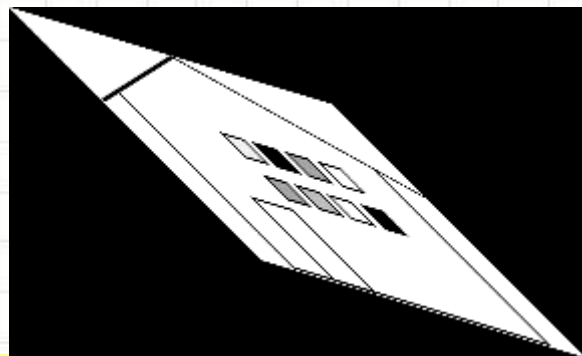
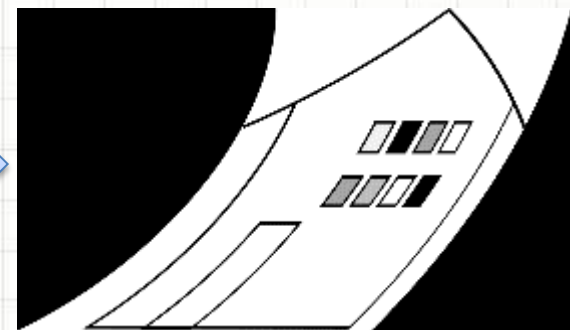
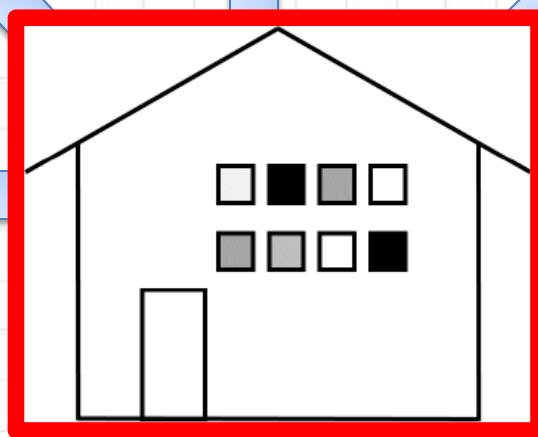
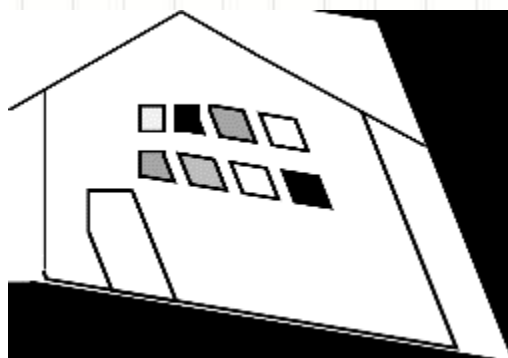
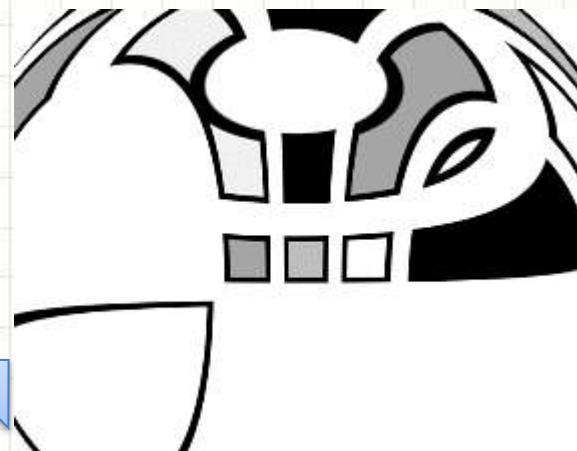
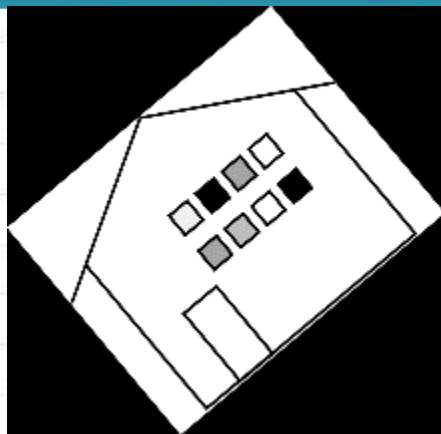
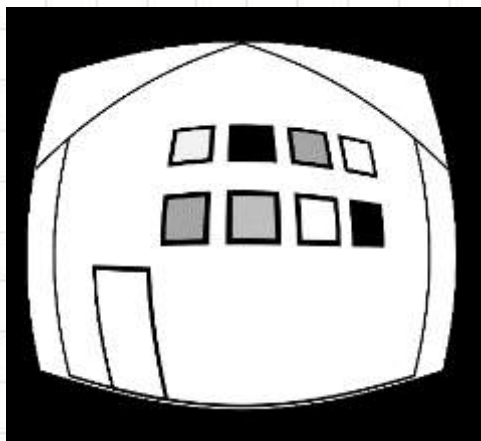


در تبدیلات مکانی فقط جای پیکسل‌ها عوض می‌شود ولی رنگ پیکسل‌ها هیچ تغییری نمی‌کند.

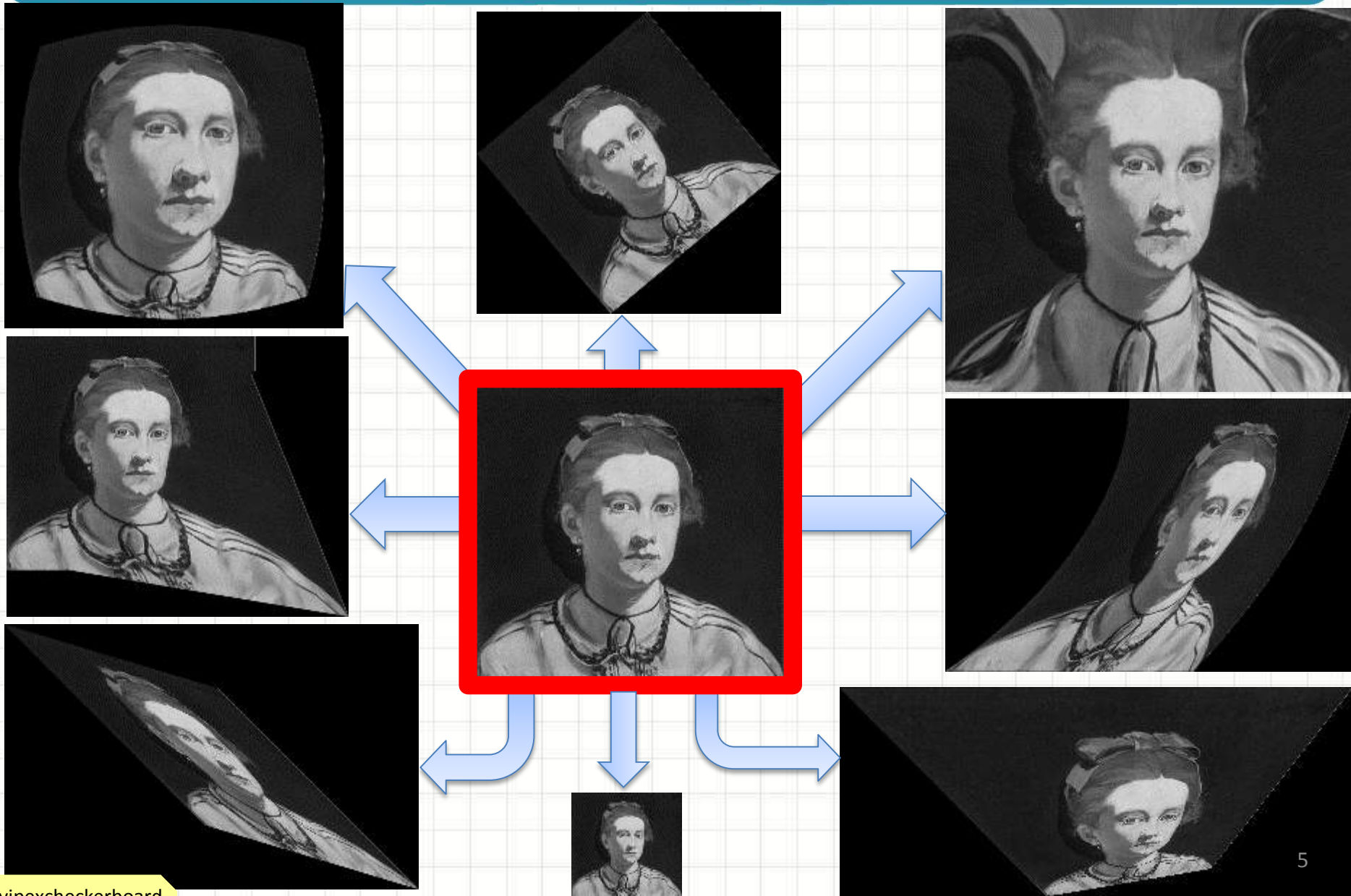
پیکسل موجود در موقعیت (3,9) به موقعیت (3,4) منتقل می‌شود



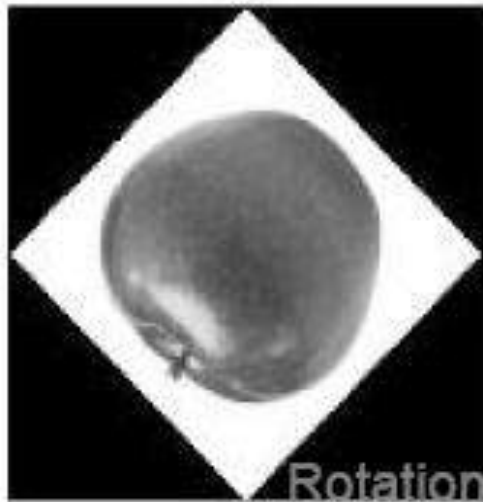
نمونه‌هایی از تبدیلات هندسی



نمونه‌هایی از تبدیلات هندسی



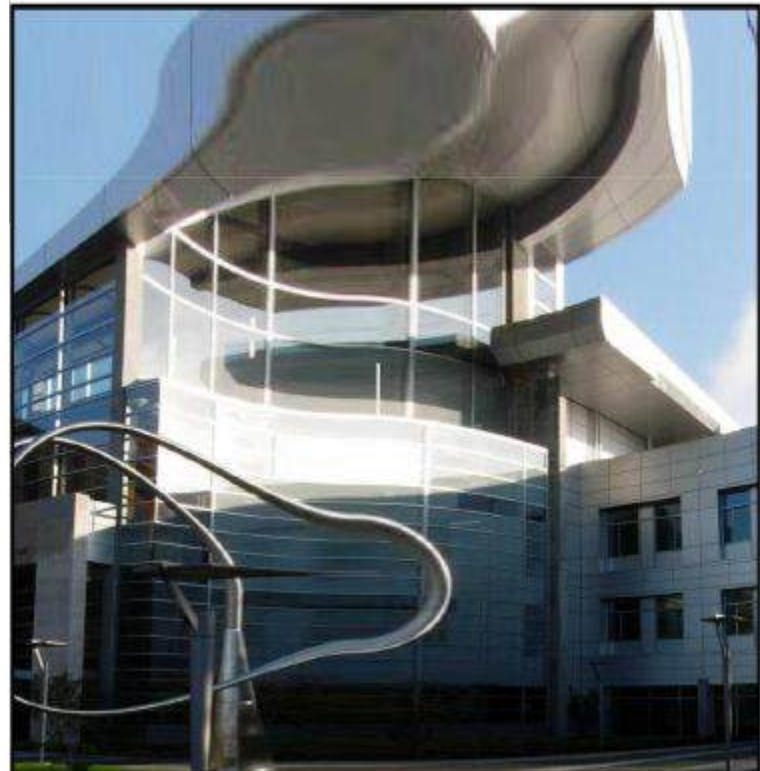
نمونه‌هایی از تبدیلات هندسی



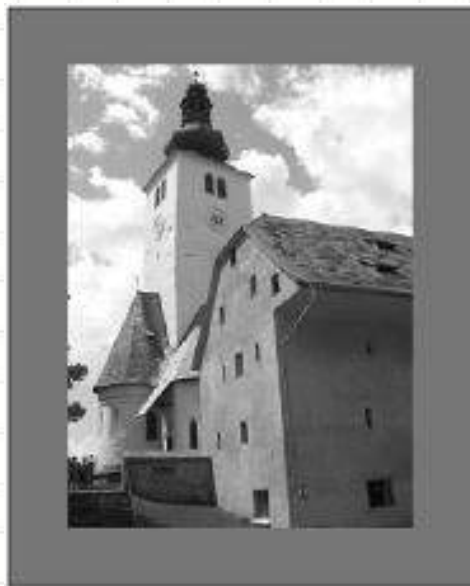
نمونه‌هایی از تبدیلات هندسی



نمونه‌هایی از تبدیلات هندسی



نمونه‌هایی از تبدیلات هندسی



نمونه‌هایی از تبدیلات هندسی



نمونه‌هایی از تبدیلات هندسی

A professor



A warped professor



اصول ریاضی و دسته‌بندی تبدیلات مکانی

تابع تبدیل متناظر با تبدیلات هندسی

از دیدگاه ریاضی تبدیل هندسی یک **تابع** موسوم به **تابع تبدیل** است که آدرس پیکسل‌ها در تصویر مبداء را می‌گیرد و آدرس آن پیکسل در تصویر مقصد را مشخص می‌کند. به عبارت دقیق‌تر اگر (x, y) آدرس یک پیکسل در تصویر مبداء باشد، آن‌گاه یک تبدیل هندسی تابعی مانند T است که (x, y) را می‌گیرد و یک آدرس دیگر مانند (x', y') را پس می‌دهد.

از دیدگاه دیگر **تابع تبدیل** T یک تابع ریاضی است که دو عدد را می‌گیرد و دو عدد را پس می‌دهد. در واقع T یک تابع برداری دومتغیره به صورت زیر است

$$(x', y') = T(x, y)$$

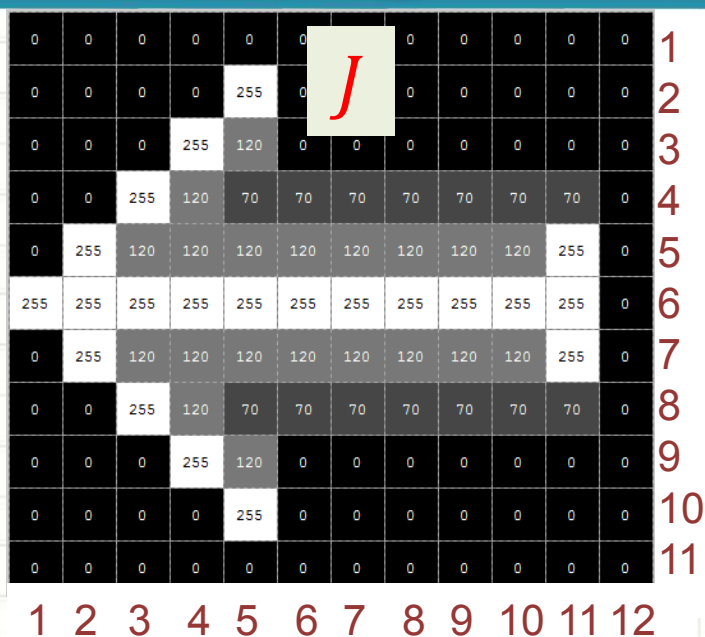
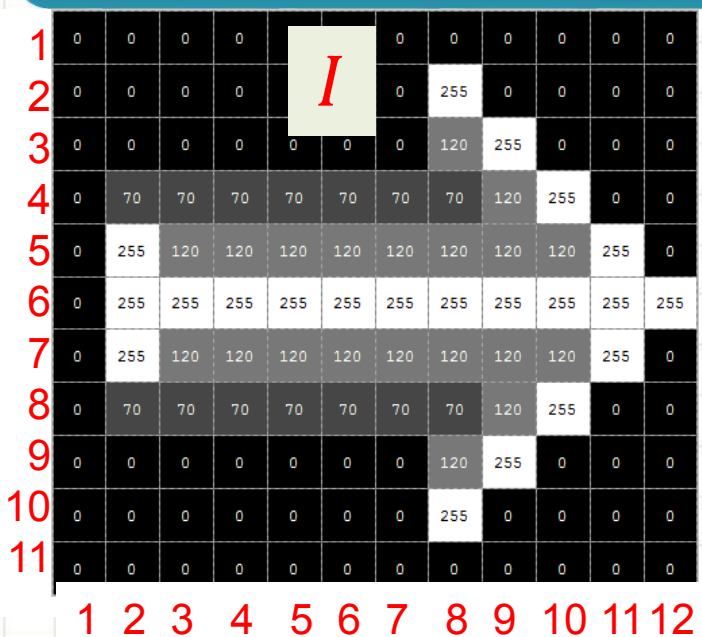
تابع T از دو تابع دیگر ساخته شده آن‌ها را T_x و T_y می‌نامیم. یعنی

$$T(x, y) = (T_x(x, y), T_y(x, y))$$

بنابراین داریم

$$\begin{cases} x' = T_x(x, y), \\ y' = T_y(x, y). \end{cases}$$

تابع تبدیل – مثال



تبدیل مکانی
موجود در شکل
روبرو را در نظر
بگیرید.

تابع تبدیل T به صورت $T(x, y) = (T_x(x, y), T_y(x, y))$ است که

$$\begin{cases} x' = T_x(x, y) = 13 - x, \\ y' = T_y(x, y) = y. \end{cases}$$

توجه شود که در این تبدیل داریم:

$$J(T_x(x, y), T_y(x, y)) = I(x, y)$$

معکوس تابع تبدیل

معکوس تابع تبدیل T را با T^{-1} نشان می‌دهیم. تابع T^{-1} متغیرهای (x', y') را به (x, y) بر می‌گرداند. به عبارت دیگر داریم:

$$(x', y') = T(x, y) \quad \leftrightarrow \quad (x, y) = T^{-1}(x', y')$$

تابع T^{-1} نیز از دو تابع دیگر ساخته شده آن‌ها را T_x^{-1} و T_y^{-1} می‌نامیم. یعنی

$$T^{-1}(x', y') = (T_x^{-1}(x', y'), T_y^{-1}(x', y'))$$

بنابراین داریم

$$\begin{cases} x = T_x^{-1}(x', y'), \\ y = T_y^{-1}(x', y'). \end{cases}$$

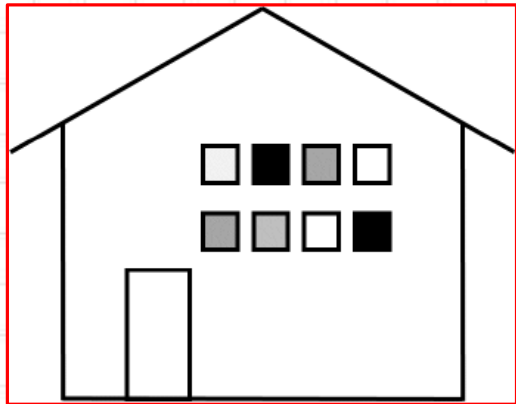
همان‌گونه که خواهیم دید، معکوس تابع تبدیل از اهمیت بالایی برخوردار است و حتی در برخی موارد از خود تابع تبدیل با اهمیت‌تر می‌باشد.

تابع تبدیل

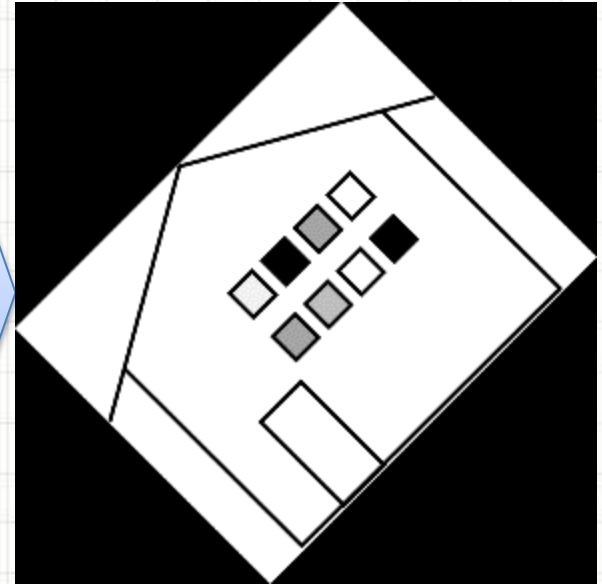
- هر تابعی که دو عدد را بگیرد و دو عدد را پس دهد را می‌توان به عنوان تابع تبدیل در نظر گرفت (البته ملاحظات لازم است که اشاره خواهد شد).
- توابع تبدیل متفاوت باعث تاثیر و تغییر متفاوتی روی تصویر می‌شوند و تصویر را به صورت متفاوتی تغییر می‌دهند.
- لازم است که برای اعمال یک تغییر هندسی در تصویر یک تابع تبدیل مناسب **انتخاب** شود.
- مطلوب‌تر است که تابع تبدیل یک تابع **یک به یک** باشد و معکوس آن نیز به سادگی محاسبه شود.
- اگر تابع تبدیل انتخابی را T بنامیم و T_x و T_y مولفه‌های آن باشند و اگر I تصویر اولیه و J تصویر تبدیل شده باشد، آن‌گاه سعی می‌شود که رابطه زیر برقرار گردد

$$J(T_x(x, y), T_y(x, y)) = I(x, y)$$

تابع تبدیل-مثال



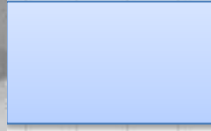
$$\begin{cases} x' = T_x(x, y) = \frac{\sqrt{2}}{2}x - \frac{\sqrt{2}}{2}y \\ y' = T_y(x, y) = \frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y \end{cases}$$



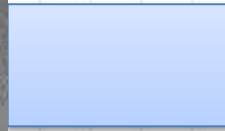
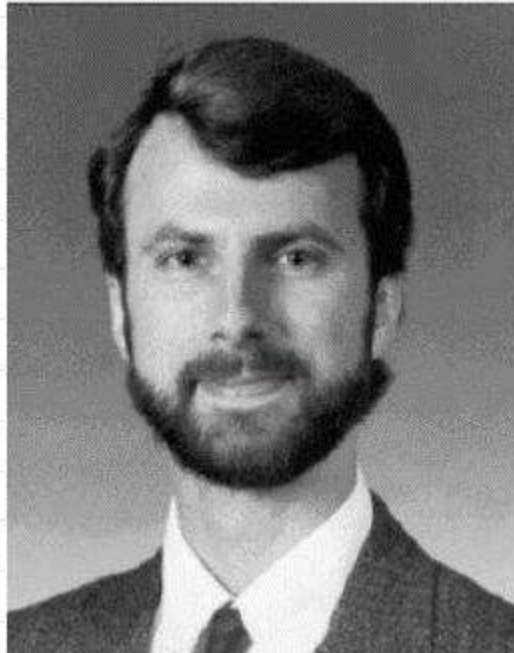
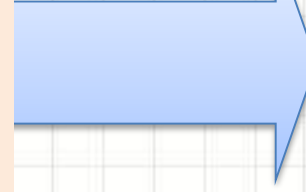
$$\begin{cases} x' = T_x(x, y) = x + 20\sin(\frac{2\pi x}{30}) \\ y' = T_y(x, y) = y \end{cases}$$



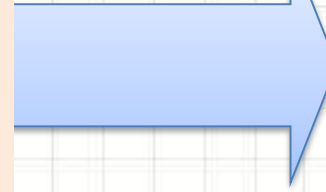
تابع تبدیل - مثال



?



?



دسته‌بندی تبدیلات هندسی

- دسته‌بندی تبدیلات هندسی بر حسب نوع تابع تبدیل متناظر می‌باشد.
- تابع تبدیل T با مولفه‌های T_x و T_y را می‌توان به صورت خطی، آفینی، غیرخطی، چندجمله‌ای، کسری و ... انتخاب کرد و بر حسب این انتخاب‌ها تبدیل را خطی، غیرخطی و ... نامگذاری می‌شود.

در این راستا تبدیلات مکانی به صورت زیر دسته‌بندی می‌شوند

- تبدیلات آفینی (Affine)
- تبدیلات خطی (آفینی) هم‌مدیس (Linear Conformal)
- تبدیلات تکه‌ای خطی (Piecewise Linear)
- تبدیلات تصویری (Projective)
- تبدیلات چندجمله‌ای (Polynomial)
- تبدیلات غیرخطی (Non-Linear)

البته تبدیلات خطی حالت خاصی از تبدیلات آفینی می‌باشند و تبدیلات تصویری و چندجمله‌ای حالات خاصی از تبدیلات غیرخطی می‌باشند.

تبدیلات آفینی و تبدیلات آفینی همدیس

تبدیلات آفینی

در این نوع تبدیلات، تابع تبدیل T به صورت زیر است:

$$\begin{cases} x' = T_x(x, y) = a_{11}x + a_{12}y + b_1 \\ y' = T_y(x, y) = a_{21}x + a_{22}y + b_2 \end{cases}$$

که a_{ij} ها و b_1 و b_2 اعدادی حقیقی می باشند.

اگر a_{ij} ها و b_1 و b_2 در یک ماتریس به صورت زیر قرار داده شود

$$A = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

آن گاه تبدیل آفینی را می توان به فرم ماتریسی زیر نوشت

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

ماتریس A به ماتریس تبدیل آفینی موسوم است.

تبدیلات آفینی

با استفاده از هندسه تحلیلی ثابت می شود که

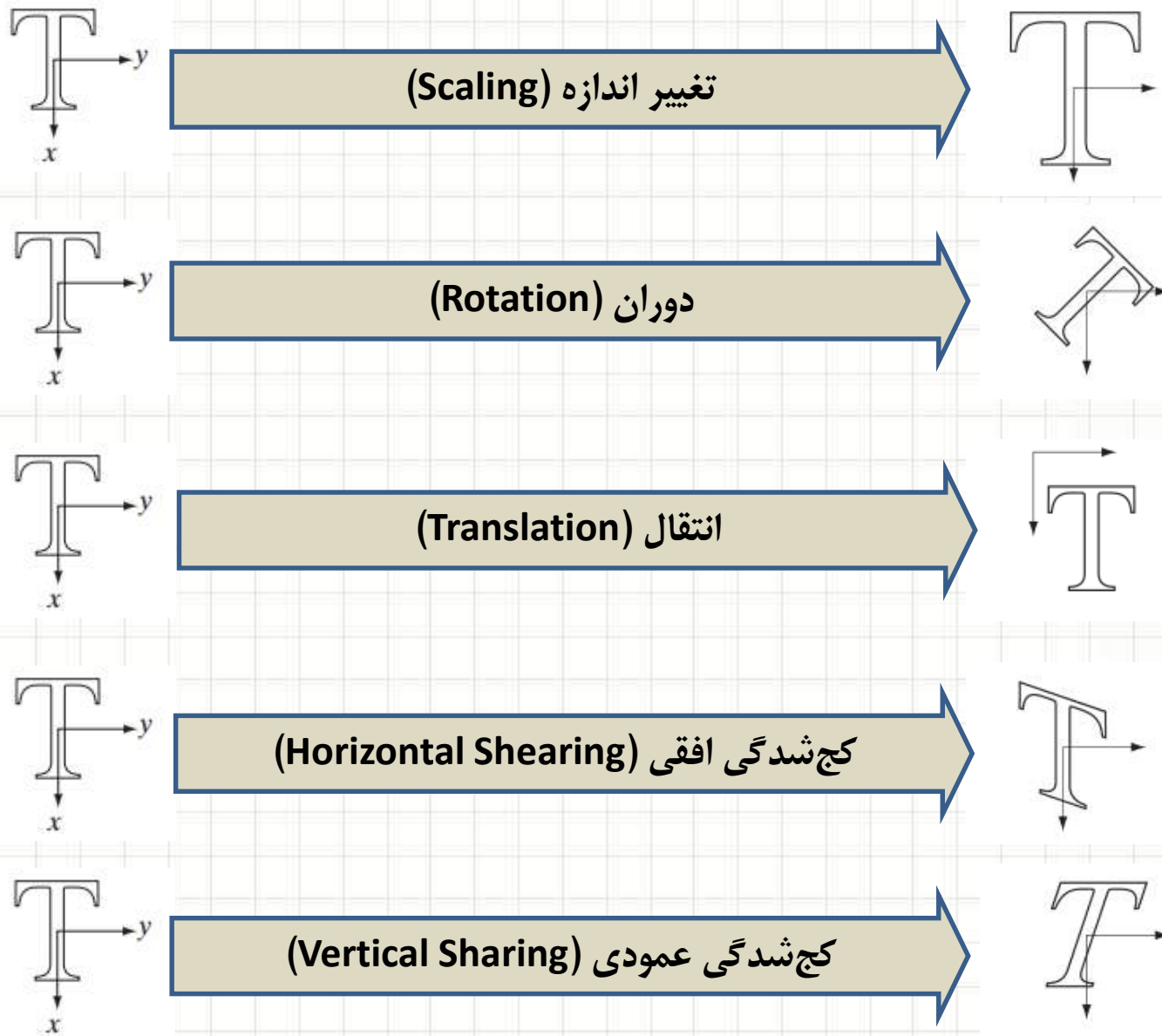
- تبدیلات آفینی یک خط را به خط تبدیل می کنند.
- موازی بودن خطوط را به هم نمی زنند. تبدیل آفینی دو خط موازی باز هم دو خط موازی است.
- اندازه زاویه ها تحت تبدیل آفینی ممکن است تغییر کند و یا ثابت بماند.

بسته به انتخاب a_{ij} ها و b_1 و b_2 تبدیلات آفینی متفاوتی وجود دارد. از جمله تبدیلات آفینی مشهور می توان موارد زیر را نام برد:

- تغییر اندازه (بزرگ یا کوچک کردن) تصویر (Scaling)
- دوران تصویر (Rotation)
- انتقال تصویر (Translation)
- کج کردن عمودی و افقی تصویر (Vertical and Horizontal Shear)

در ادامه تبدیلات فوق توضیح داده می شود.

توصیف تصویری تبدیلات آفینی رایج



تبدیل آفینی تغییر اندازه

تابع و ماتریس تبدیل تغییر اندازه به صورت زیر است:

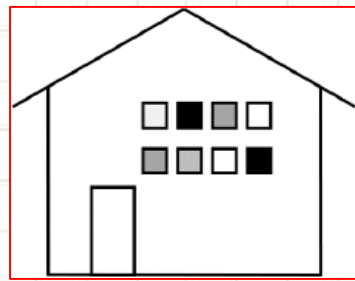
$$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T: \begin{cases} x' = T_x(x, y) = c_x x \\ y' = T_y(x, y) = c_y y \end{cases}$$

به طوری که c_x و c_y اعداد حقیقی مثبت می‌باشند که به ترتیب مبین ضریب تغییر اندازه در راستای x و y می‌باشند.

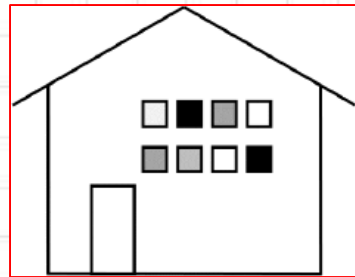
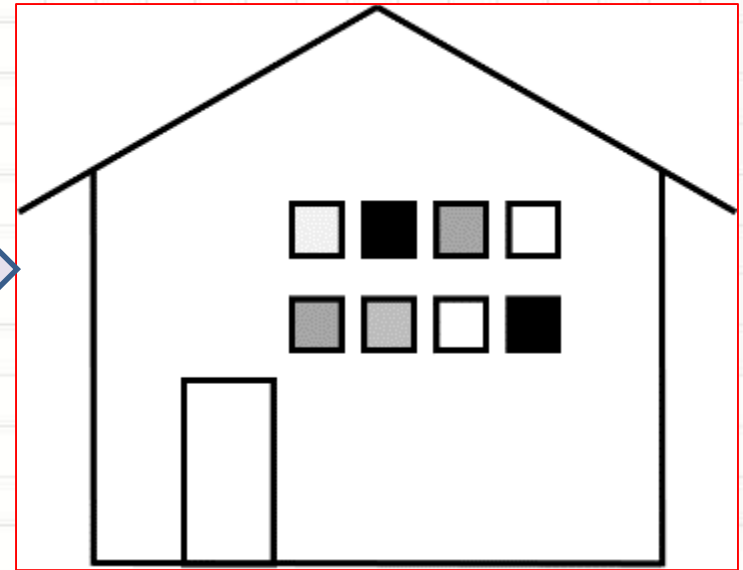
- در حالتی که $c_x = 1$ ، آن‌گاه در راستای x تغییر اندازه نداریم.
- در حالتی که $c_x < 1$ ، آن‌گاه در راستای x کاهش اندازه داریم و این کاهش اندازه با مقدار c_x مشخص می‌شود.
- در حالتی که $c_x > 1$ ، آن‌گاه در راستای x متناسب با مقدار c_x افزایش اندازه داریم.

به طور مشابه روی مقدار c_y می‌توان اظهار نظر کرد.

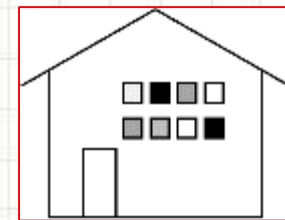
تبدیل آفینی تغییر اندازه - مثال



$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

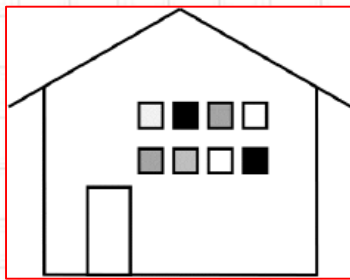


$$\begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

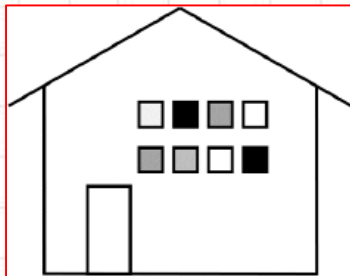
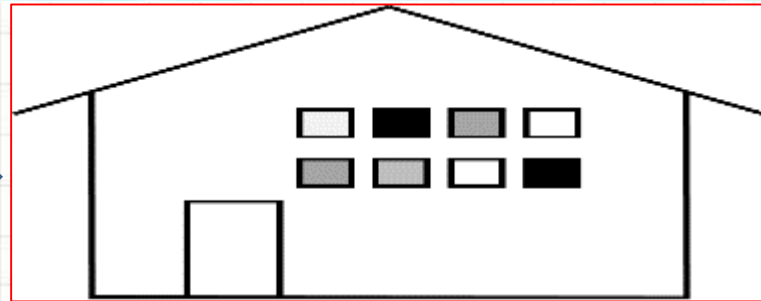
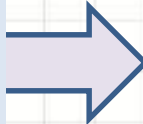


تغییر اندازه اندازه‌های فوق **متقارن** هستند. یعنی تغییر اندازه در راستای x و y یکسان است. در این نوع تغییر اندازه، خطوط موازی بودن خطوط حفظ می‌شود و همچنین اندازه زاویه بین خطوط تغییر نمی‌کند (تبدیل کانفرمال).

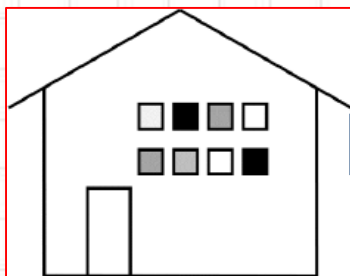
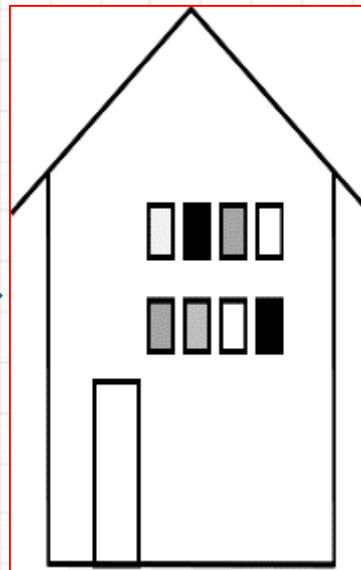
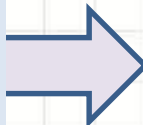
تبدیل آفینی تغییر اندازه - مثال



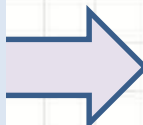
$$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



تغییر اندازه غیرمتقارن:
خطوط موازی باز هم موازی باقی می ماند.
اما اندازه زاویه ها به هم می خورد.

تبدیل آفینی دوران

تابع و ماتریس تبدیل دوران به اندازه θ به صورت زیر است:

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T: \begin{cases} x' = +x \cos(\theta) + y \sin(\theta) \\ y' = -x \sin(\theta) + y \cos(\theta) \end{cases}$$

در تبدیل دوران، تمامی شکل حول مبدا مختصات به اندازه θ دوران داده می‌شود. واضح است که در این تبدیل توازی خطوط به هم نمی‌خورد و همچنین اندازه زاویه‌ها نیز تغییر نمی‌کند.

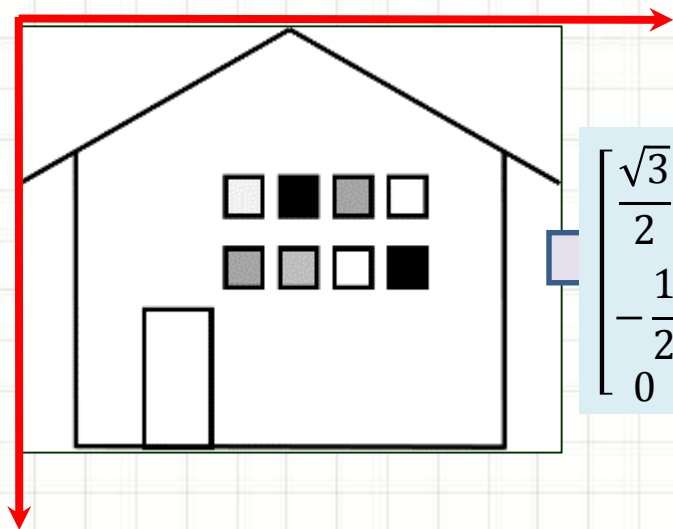
البته اندازه و چهارچوب تصویر به هم می‌خورد.

تبدیل آفینی دوران-مثال

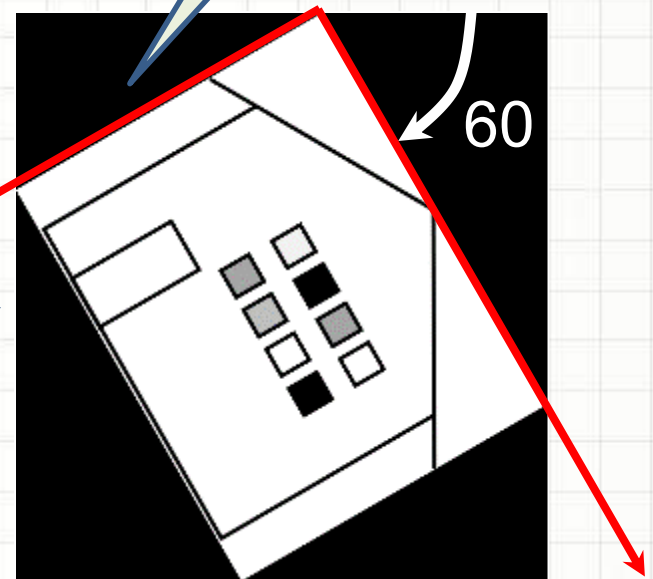
ماتریس تبدیل دوران به ازای $\theta = 60^\circ$ به صورت زیر است

$$\begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

همان گونه که ملاحظه می شود قسمت های سیاه رنگ در تصویر دوران یافته ایجاد می شود.

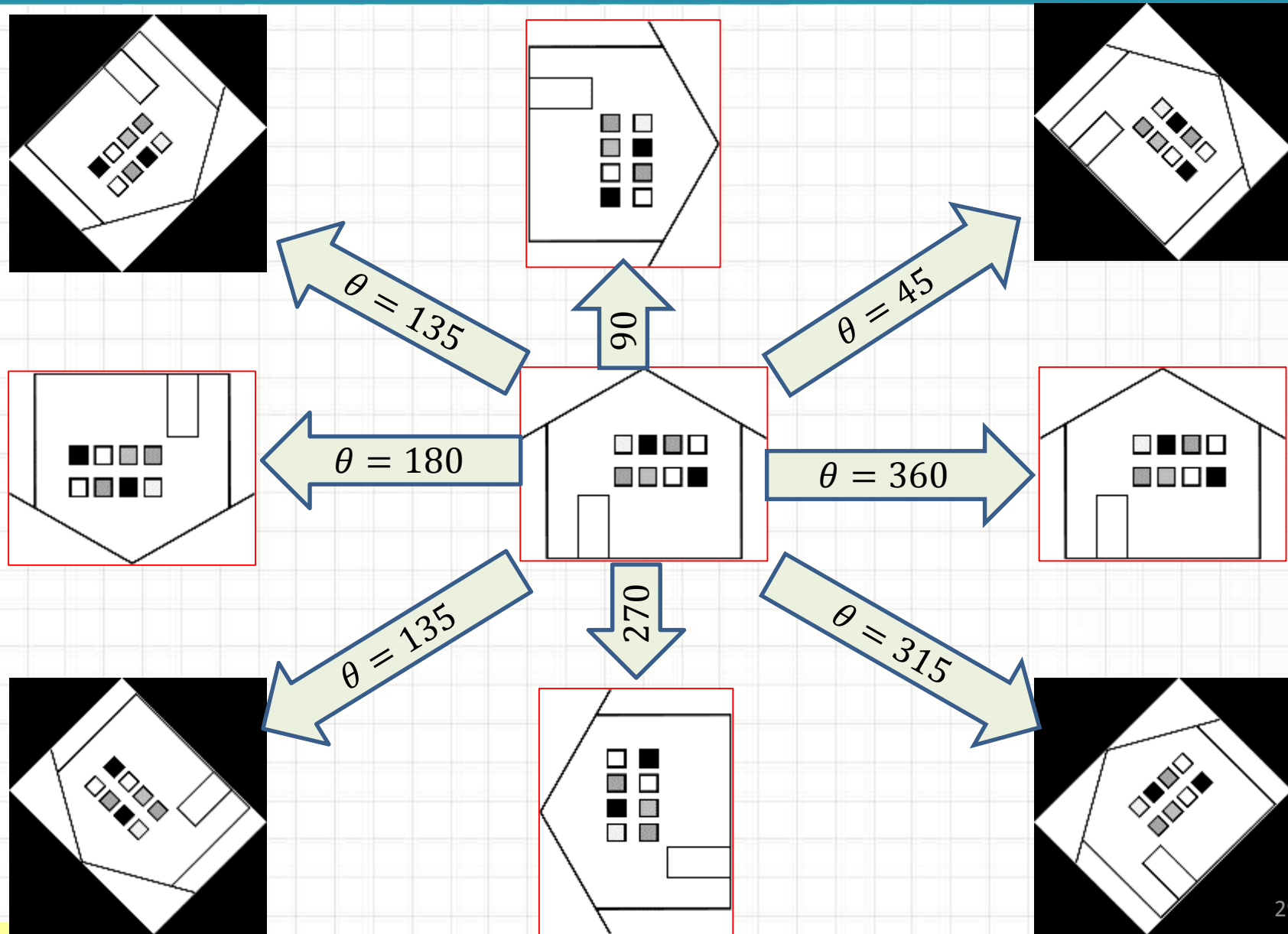


$$\begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



در دوران، امکان ظهور قسمت های سیاه و تغییر اندازه چهارچوب تصویر وجود دارد.

تبدیل آفینی دوران-مثال



تبدیل آفینی انتقال

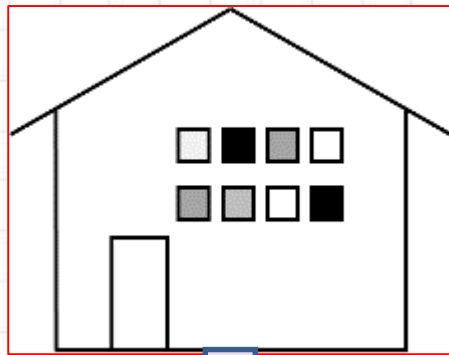
این تبدیل باعث جابجایی تصویر می‌شود. ماتریس متناظر با این انتقال به صورت زیر است:

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \quad T: \begin{cases} x' = T_x(x, y) = x + t_x \\ y' = T_y(x, y) = y + t_y \end{cases}$$

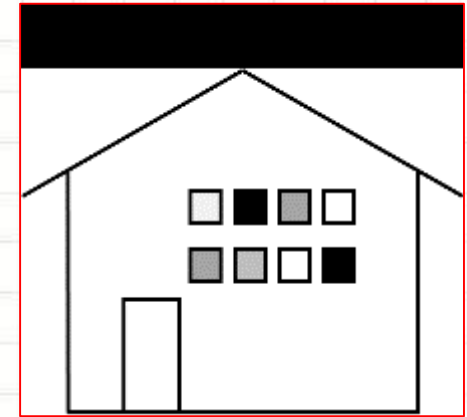
به طوری که t_x و t_y اعداد حقیقی می‌باشند که به ترتیب مبین انتقال تصویر در راستای x و y می‌باشند.

توجه شود که واحد t_x و t_y عبارت است از پیکسل. در واقع t_x و t_y تعداد پیکسل‌های انتقال داده شده می‌باشند. بنابراین منطقی است که t_x و t_y به صورت اعدادی صحیح انتخاب شوند.

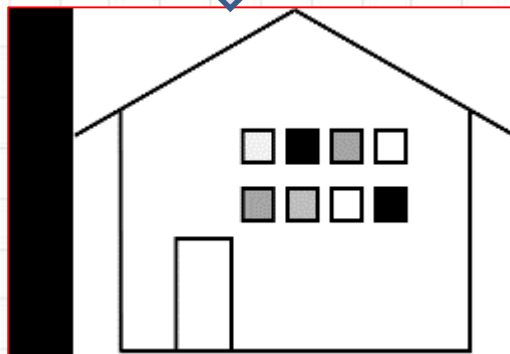
تبدیل آفینی انتقال-مثال



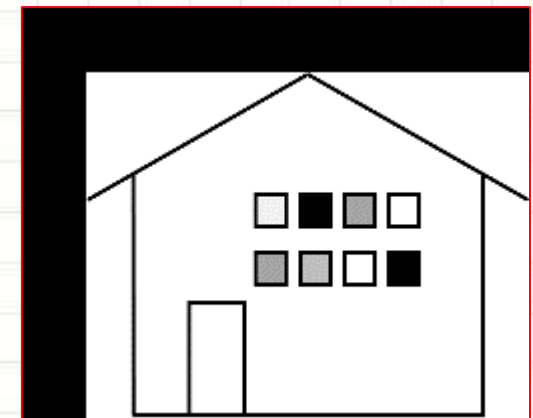
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 100 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 100 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix}$$



تبدیل آفینی کج کردن عمودی و افقی

تابع و ماتریس تبدیل **کج کردن افقی** به صورت زیر است:

$$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T: \begin{cases} x' = x + y \times s_h \\ y' = y \end{cases}$$

به طوری که s_h یک عدد حقیقی می باشد که علامت آن جهت کج کردن افقی را مشخص می کند.

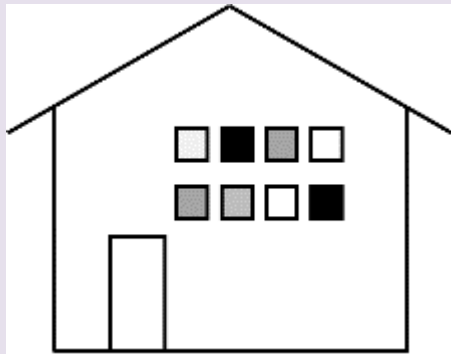
تابع و ماتریس تبدیل **کج کردن عمودی** به صورت زیر است:

$$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad T: \begin{cases} x' = x \\ y' = x \times s_v + y \end{cases}$$

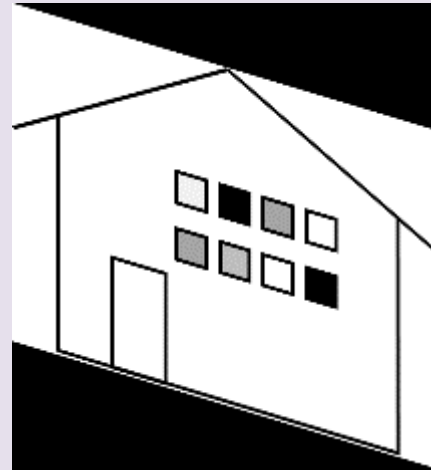
به طوری که s_v یک عدد حقیقی می باشد که علامت آن جهت کج کردن عمودی را مشخص می کند.

تبدیل آفینی کج کردن افقی-مثال

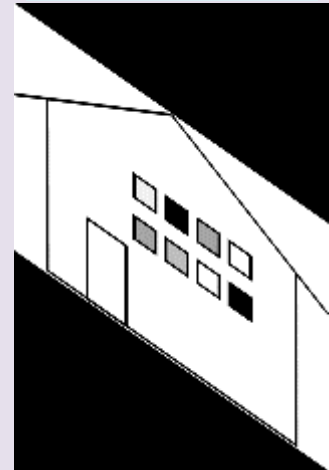
$$\begin{bmatrix} 1 & 0.1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



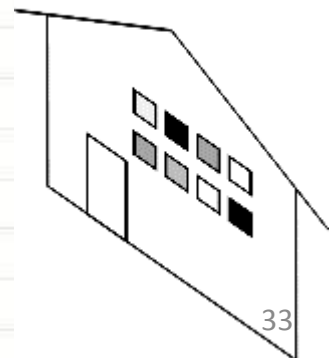
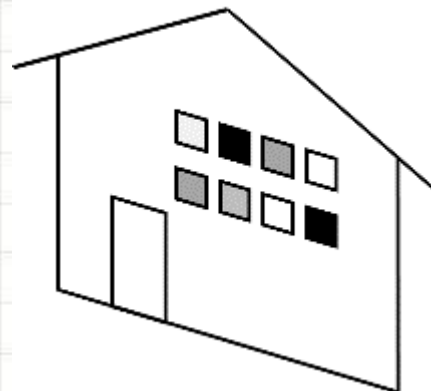
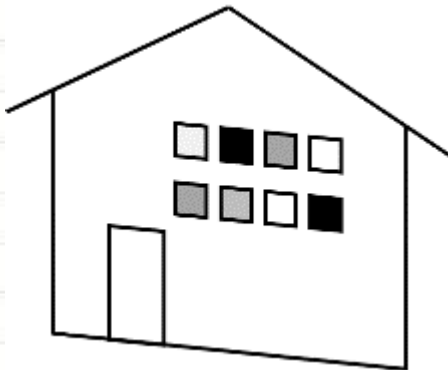
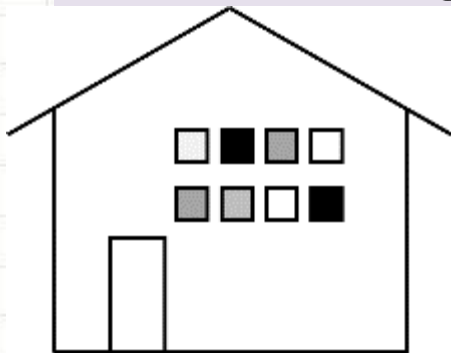
$$\begin{bmatrix} 1 & 0.3 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0.7 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

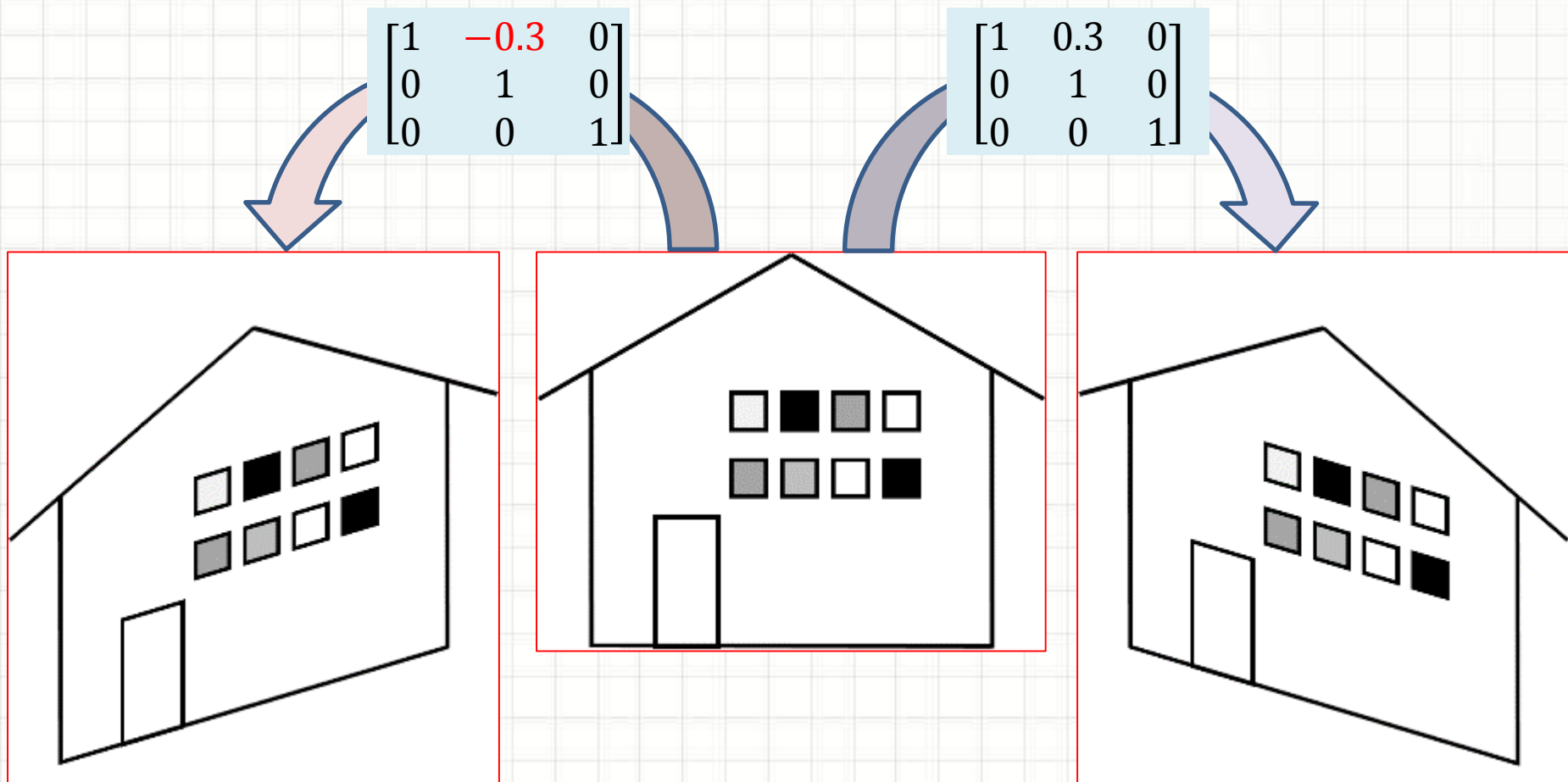


در تصاویر زیر، برای وضوح بهتر، رنگ سیاه به رنگ زمینه اصلی تصویر تبدیل شده.

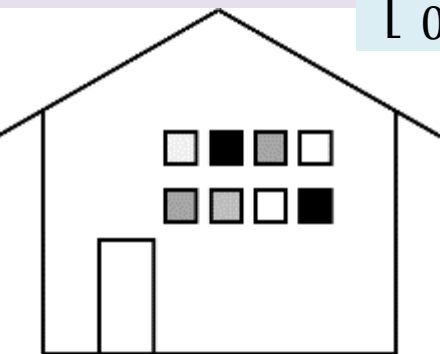


تبدیل آفینی کج کردن افقی-مثال

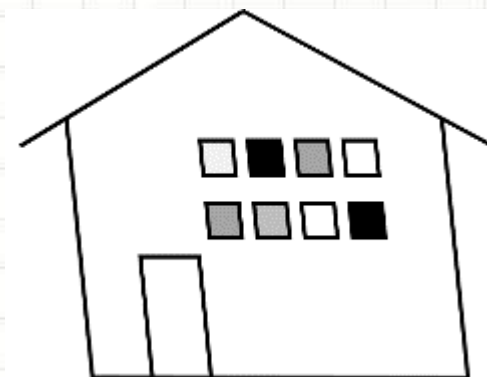
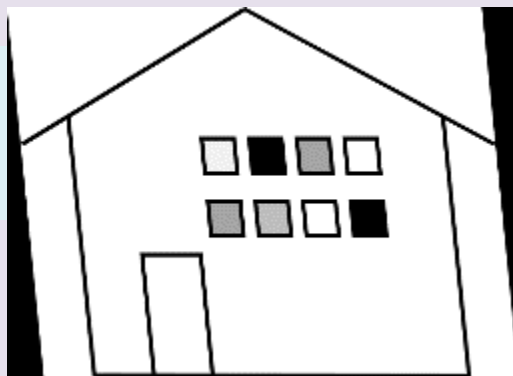
در صورتی که پارامتر S_h در ماتریس تبدیل افقی منفی اختیار شود، آن گاه جهت کج شدن تغییر می کند.



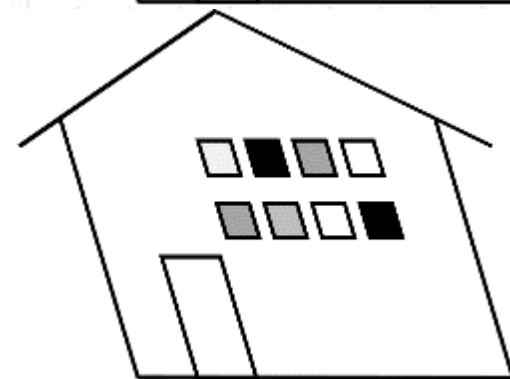
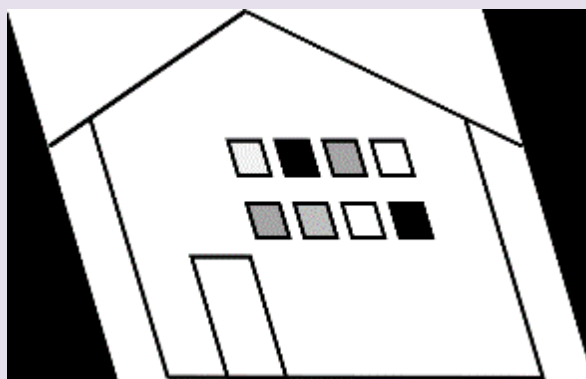
تبدیل آفینی کج کردن عمودی-مثال



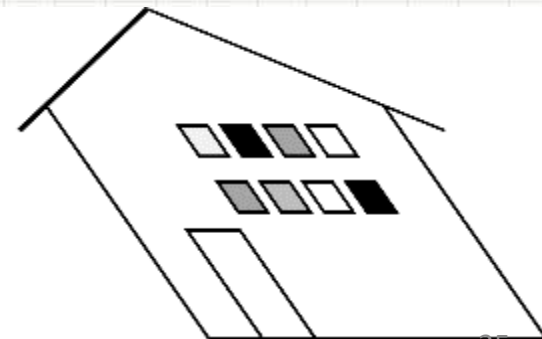
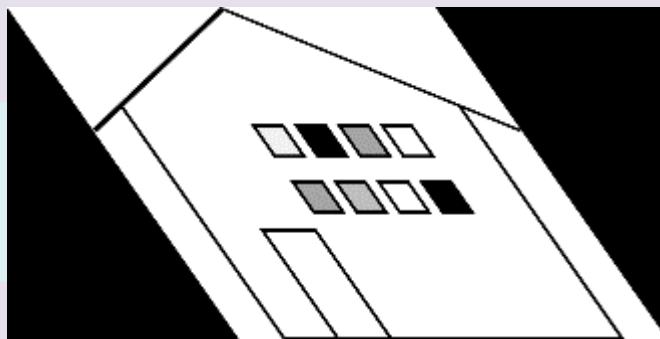
$$\begin{bmatrix} 1 & 0 & 0 \\ 0.1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 0 \\ 0.3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



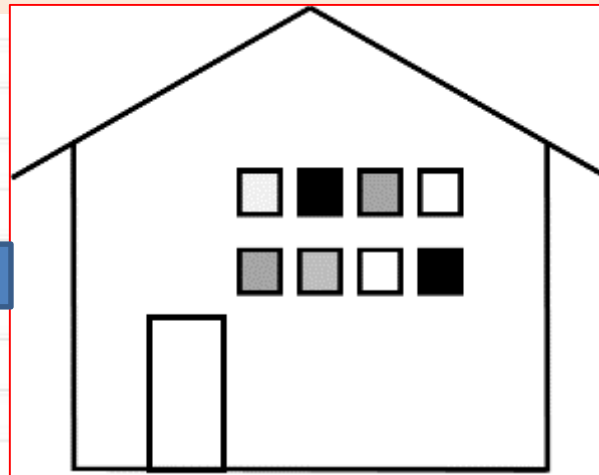
$$\begin{bmatrix} 1 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



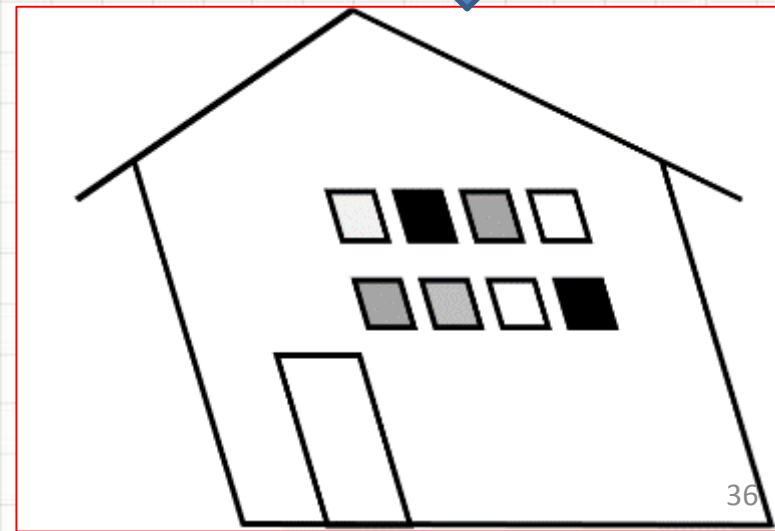
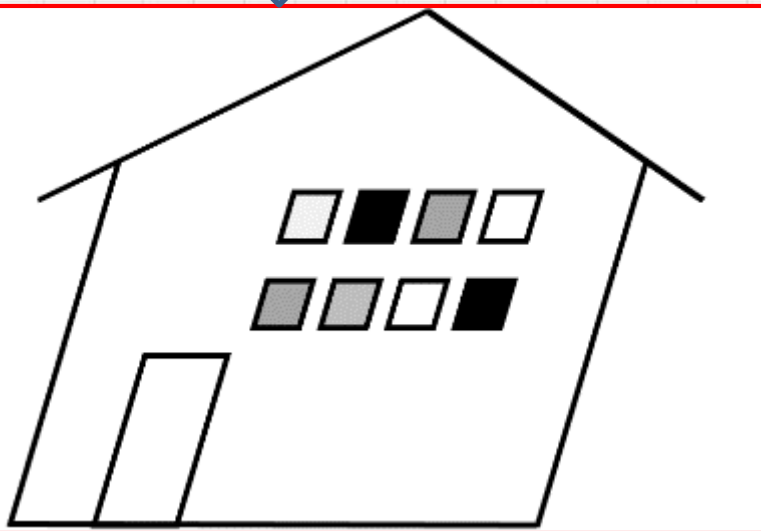
تبدیل آفینی کج کردن عمودی-مثال

در صورتی که پارامتر S_h در ماتریس تبدیل عمودی منفی اختیار شود، آن گاه جهت کج شدن تغییر می کند و به سمت جلو می شود.

$$\begin{bmatrix} 1 & 0 & 0 \\ -0.3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 0 & 0 \\ +0.3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



تبدیلات آفینی همدیس

همان گونه که ذکر شد، تبدیلات همدیس خط راست را به خط راست تبدیل می کنند. حال یک تبدیل آفینی همدیس (Conformal) نامیده می شود، هرگاه اندازه زاویه بین دو خط را تغییر ندهد. به عبارت دیگر اگر در تصویر اولیه، دو خط راست باشد که همدیگر را با زاویه 60° درجه قطع کنند، آن گاه تحت یک تبدیل آفینی همدیس، این دو خط باز هم خط باقی می مانند و زاویه بین آنها نیز 60° باقی می ماند.

اگر a_{ij} ها و b_1 و b_2 در یک ماتریس به صورت زیر قرار داده شود

$$A = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

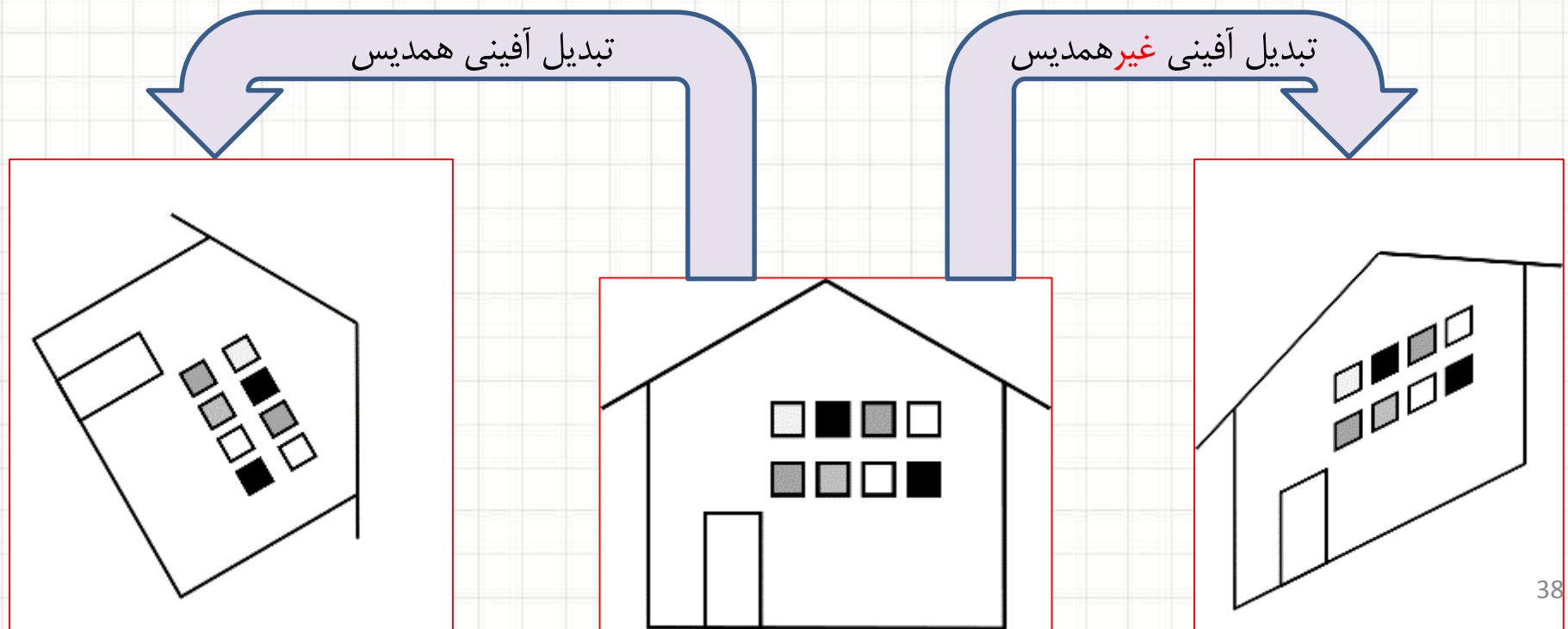
آن گاه تبدیل آفینی را می توان به فرم ماتریسی زیر نوشت

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

ماتریس A به ماتریس تبدیل آفینی موسوم است.

تبدیلات آفینی همدیس

همان گونه که ذکر شد، تبدیلات همدیس خط راست را به خط راست تبدیل می کنند. حال یک تبدیل آفینی همدیس (Conformal) نامیده می شود، هرگاه اندازه زاویه بین دو خط را تغییر ندهد. به عبارت دیگر اگر در تصویر اولیه، دو خط راست باشد که همدیگر را با زاویه 60° درجه قطع کنند، آن گاه تحت یک تبدیل آفینی همدیس، این دو خط باز هم خط باقی می مانند و زاویه بین آنها نیز 60° باقی می ماند.



تبدیلات آفینی همدیس

در قضیه زیر شرط کافی برای همدیس بودن یک تبدیل آفینی آورده شده است.
قضیه: یک تبدیل آفینی با ماتریس زیر

$$A = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

یک تبدیل همدیس است، اگر و فقط اگر

$$a_{11} = a_{22} \text{ (الف)}$$

$$a_{12} = -a_{21} \text{ (ب)}$$

با توجه به قضیه فوق و یا با مشاهده تبدیلات، می‌توان نتیجه گرفت که

(الف) تبدیل بزرگنمایی متقارن، تبدیل دوران و تبدیل انتقال از نوع همدیس می‌باشند.

(ب) تبدیل بزرگنمایی غیرمتقارن، تبدیل کج‌کردن مایل و عمودی از نوع غیرهمدیس می‌باشند.

معکوس تبدیلات آفینی

همان گونه که خواهیم دید، **معکوس تابع تبدیل** نقش مهمی در انجام تبدیل بازی می کند. لذا در ادامه، تبدیل معکوس و تابع تبدیل معکوس در تبدیلات آفینی به طور دقیق بررسی می گردد.

در یک تبدیل آفینی T که (x, y) را به (x', y') تبدیل می کند، تابع تبدیل به صورت زیر است:

$$T: \begin{cases} x' = T_x(x, y) = a_{11}x + a_{12}y + b_1 \\ y' = T_y(x, y) = a_{21}x + a_{22}y + b_2 \end{cases}$$

حال معکوس تابع فوق که (x', y') را به (x, y) برمی گرداند، به صورت زیر است:

$$T^{-1}: \begin{cases} x = T_x^{-1}(x', y') = +\frac{a_{22}}{d}x' - \frac{a_{12}}{d}y' - \frac{a_{22}b_1 - a_{12}b_2}{d} \\ y = T_y^{-1}(x', y') = -\frac{a_{21}}{d}x' + \frac{a_{11}}{d}y' + \frac{a_{21}b_1 - a_{11}b_2}{d} \end{cases}$$

به طوری که $d = a_{11}a_{22} - a_{12}a_{21}$

معکوس تبدیلات آفینی

همان گونه که ملاحظه شد، تابع تبدیل را می توان به فرم ماتریسی زیر نوشت:

$$T: \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

حال برای تبدیل معکوس نیز می توان یک فرم ماتریسی به صورت زیر نوشت:

$$T^{-1}: \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \frac{1}{d} \begin{bmatrix} a_{22} & -a_{12} & -a_{22}b_1 + a_{12}b_2 \\ -a_{21} & a_{11} & a_{21}b_1 - a_{11}b_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

به طوری که $d = a_{11}a_{22} - a_{12}a_{21}$. توجه شود که ماتریس تبدیل معکوس به صورت زیر است:

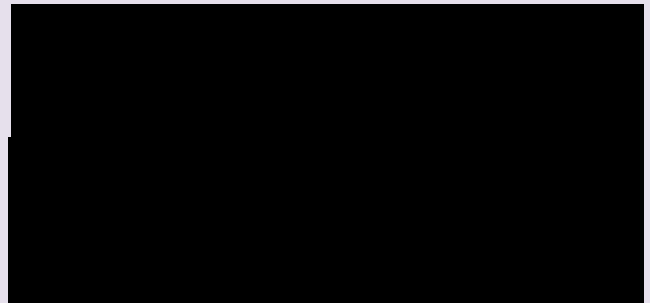
$$\frac{1}{d} \begin{bmatrix} a_{22} & -a_{12} & -a_{22}b_1 + a_{12}b_2 \\ -a_{21} & a_{11} & a_{21}b_1 - a_{11}b_2 \\ 0 & 0 & 1 \end{bmatrix}$$

تبدیلات غیر خطی

تبدیلات هندسی غیر خطی

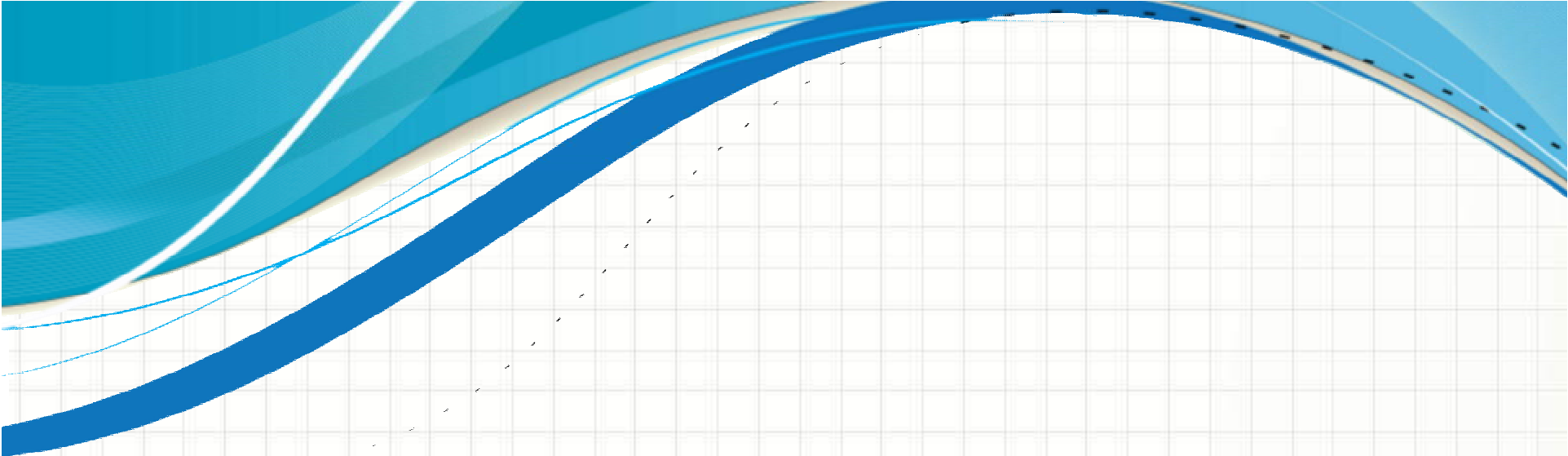
در تبدیلات هندسی غیرخطی تابع تبدیل به فرم غیرخطی رخ می دهد. به عنوان نمونه تابع تبدیل می تواند به صورت زیر تعریف شود.

$$(x', y') = T(x, y) : \begin{cases} x' \\ y' \end{cases}$$



تنوع تبدیلات غیرخطی بیشتر می باشد و البته کارکردن با آنها نیز پیچیده تر است.

تبدیلات هندسی غیرخطی خطوط را تغییر داده و به منحنی های خمیده تبدیل می کنند. همچنین موازی بودن را به هم می زنند و غیره.



نحوه انجام تبدیلات مکانی

Spatial Geometric Transforms in Image Processing

Forward Mapping



اعمال تبدیلات هندسی

به نظر می آید که اعمال تبدیلات هندسی ساده و سر راست است. به این صورت که کافی است که فقط آدرس پیکسل ها را به تابع تبدیل بدهیم و سپس آدرس پیکسل در فایل مقصد را استخراج کنیم و مقدار شدت رنگ آن را مشخص کنیم. اما در عمل مشکلاتی رخ می دهد که نیاز به ملاحظات خاصی نیاز دارد. از جمله

- 1- ممکن است که اندازه تصویر مقصد با اندازه تصویر اولیه متفاوت باشد.
- 2- پیکسل های تصویر اولیه دقیقا به روی پیکسل های تصویر مقصد منتقل نمی شود.
- 3- برخی پیکسل ها در تصویر مقصد مقدار دهی نمی شوند. یعنی هیچ پیکسلی از تصویر اولیه به آن جا منتقل نمی شود.
- 4- ممکن است بیش از یک پیکسل در تصویر اولیه به یک پیکسل در تصویر مقصد منتقل شود.

درونیابی در بزرگنمایی تصویر



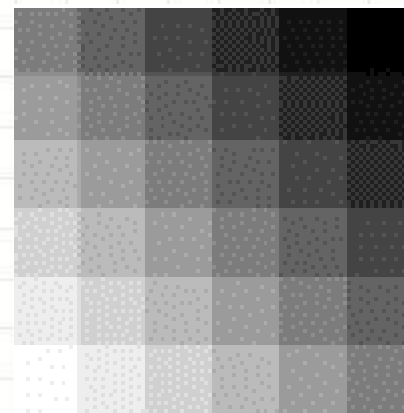
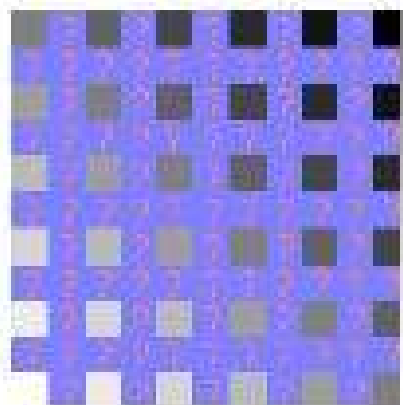
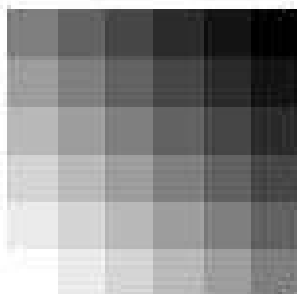
$$\begin{bmatrix} 125 & 170 & 129 \\ 172 & 170 & 175 \\ 125 & 128 & 128 \end{bmatrix}$$

$$\begin{bmatrix} ? & ? & ? & ? & ? & ? \\ ? & 125 & ? & 170 & ? & 129 \\ ? & ? & ? & ? & ? & ? \\ ? & 172 & ? & 170 & ? & 175 \\ ? & ? & ? & ? & ? & ? \\ ? & 125 & ? & 128 & ? & 128 \end{bmatrix}$$

درونیابی نزدیکترین همسایگی

?	?	?	?	?	?
?	125	?	170	?	129
?	?	?	?	?	?
?	172	?	170	?	175
?	?	?	?	?	?
?	125	?	128	?	128

125	125	?	?	?	?
125	125	?	170	?	129
?	?	?	?	?	?
?	172	?	170	?	175
?	?	?	?	?	?
?	125	?	128	?	128



درونیابی دوخطی

۱	?	?	?	?	?
۲	?	125	?	170	?
۳	?	?	?	?	?
۴	?	172	?	170	?
	?	?	?	?	?
	?	125	?	128	?

BiLinear Interpolation:

$$F(x,y) = ax + by + cxy + d$$

$$\begin{cases} f(2,2) = 125 \\ f(2,4) = 170 \\ f(4,2) = 172 \\ f(4,4) = 170 \end{cases} \Rightarrow \begin{cases} 2a + 2b + 4c + d = 125 \\ 2a + 4b + 8c + d = 170 \\ 4a + 2b + 8c + d = 172 \\ 4a + 4b + 16c + d = 170 \end{cases} \Rightarrow \begin{cases} a = 47 \\ b = 46 \\ c = -11.75 \\ d = -14 \end{cases}$$

$$F(x,y) = 47x + 46y - 11.75xy - 14$$

درونیابی دوخطی

?	?	?	?	?	?
?	125	?	170	?	129
?	?	?	?	?	?
?	172	?	170	?	175
?	?	?	?	?	?
?	125	?	128	?	128

$$F(x,y) = 47x + 46y - 11.75xy - 14$$

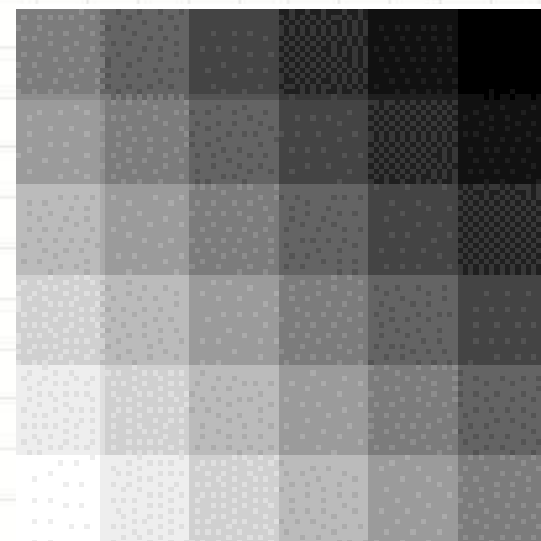
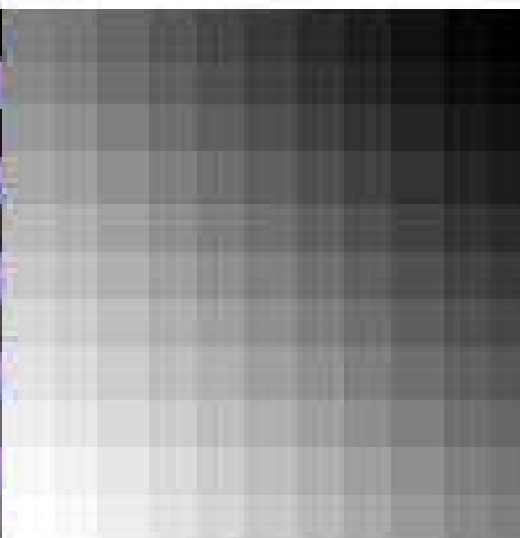
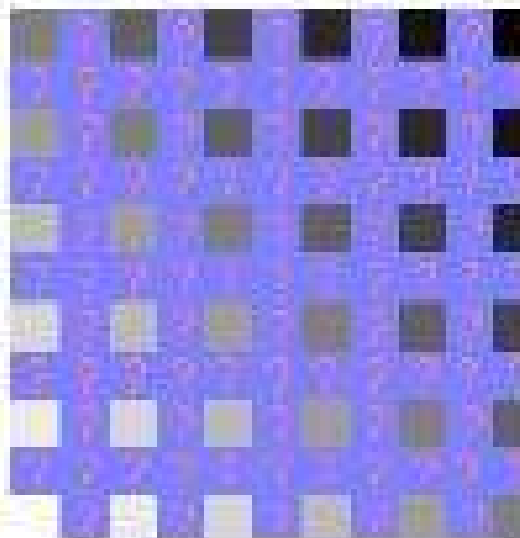
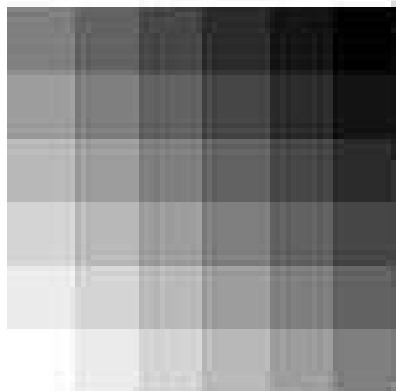
$$F(1,1) = 47 + 46 - 11.75 - 14$$

$$= 67.25$$

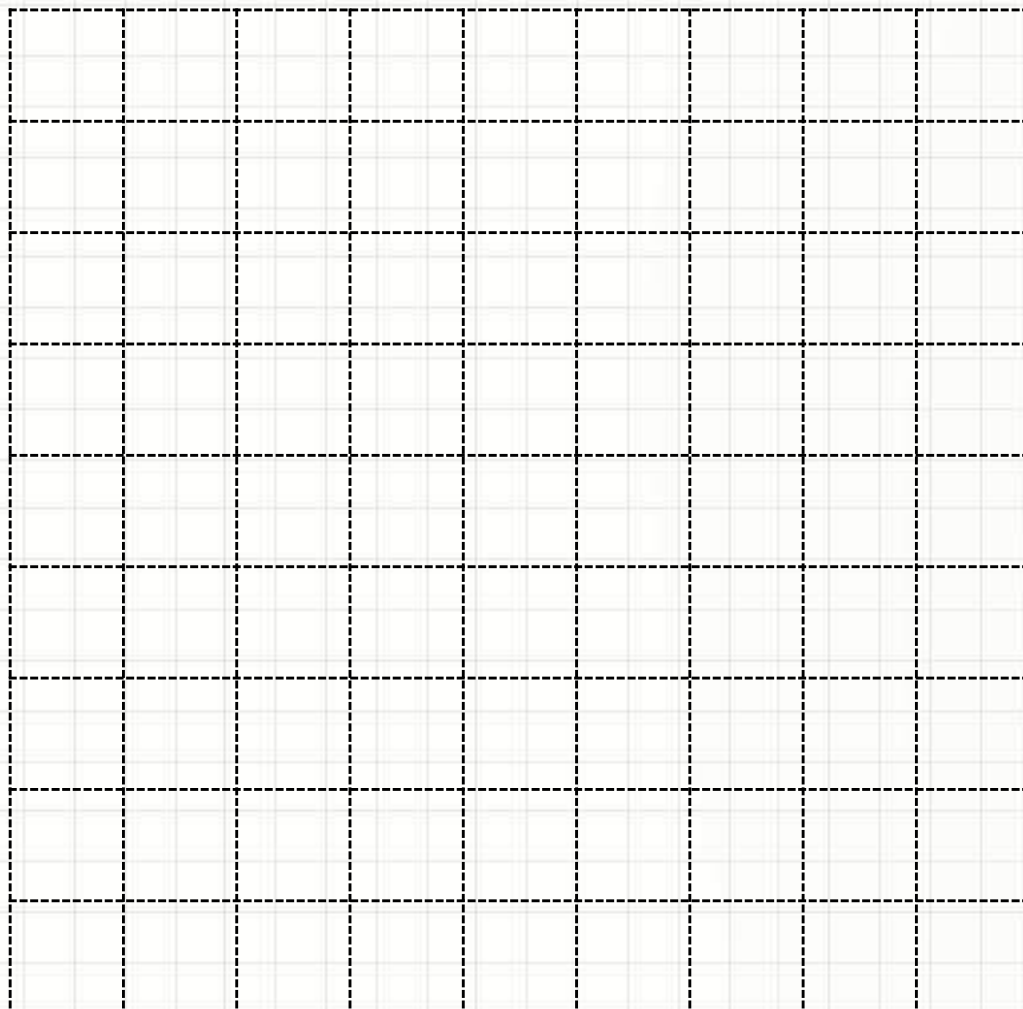
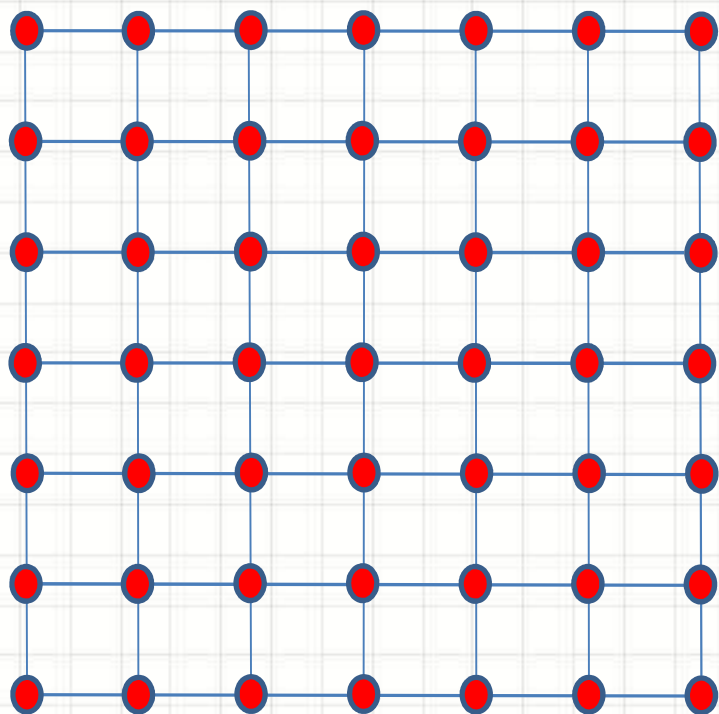
67

67	?	?	?	?	?
?	125	?	170	?	129
?	?	?	?	?	?
?	172	?	170	?	175
?	?	?	?	?	?
?	125	?	128	?	128

درونیابی دوخطی

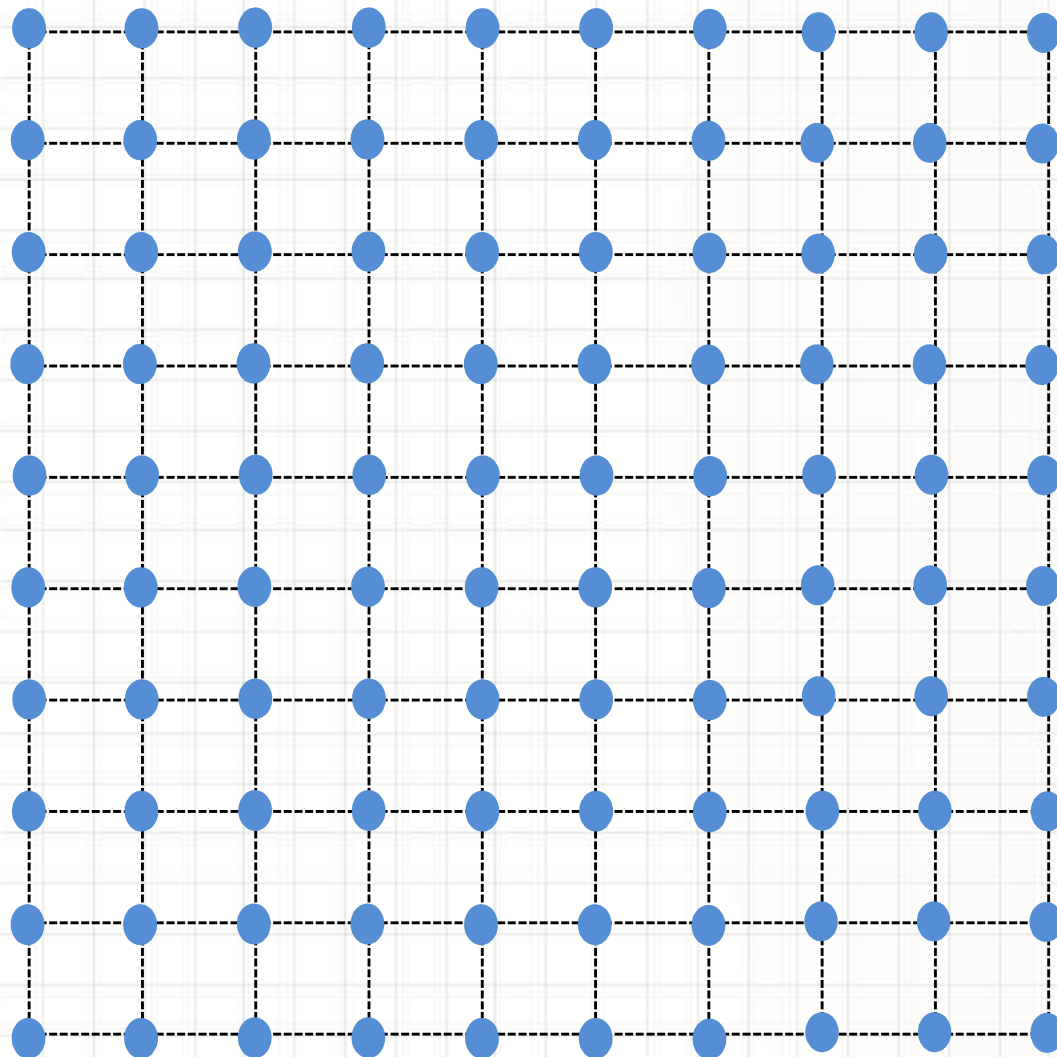
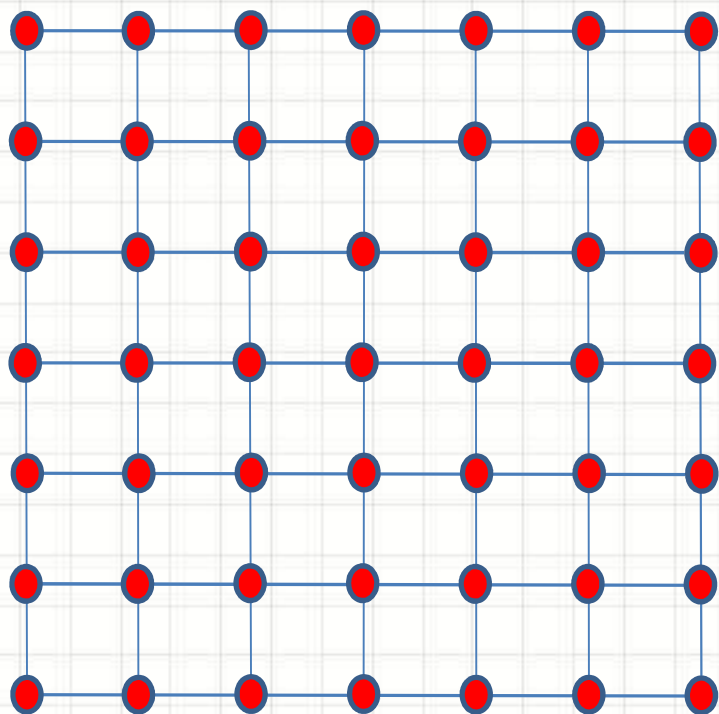


اعمال تبدیل هندسی خطی روی یک تصویر



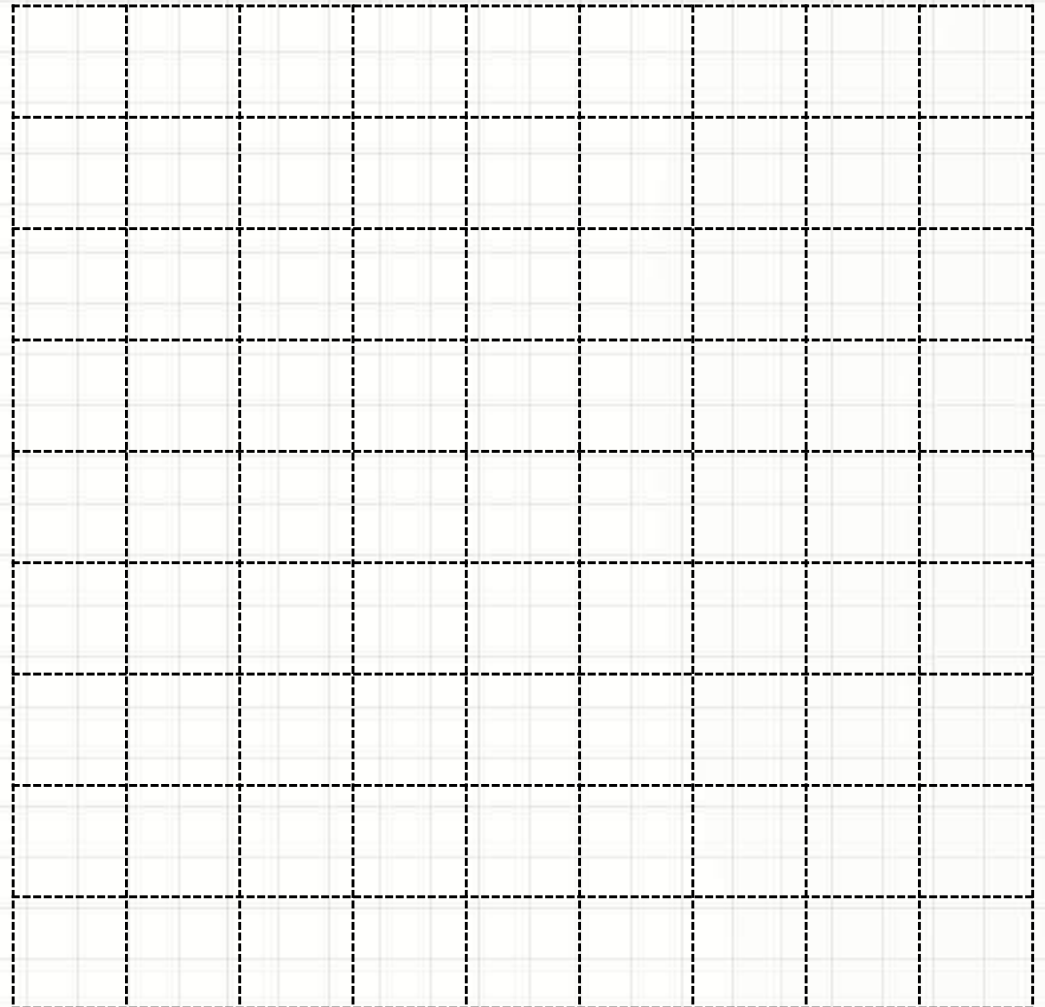
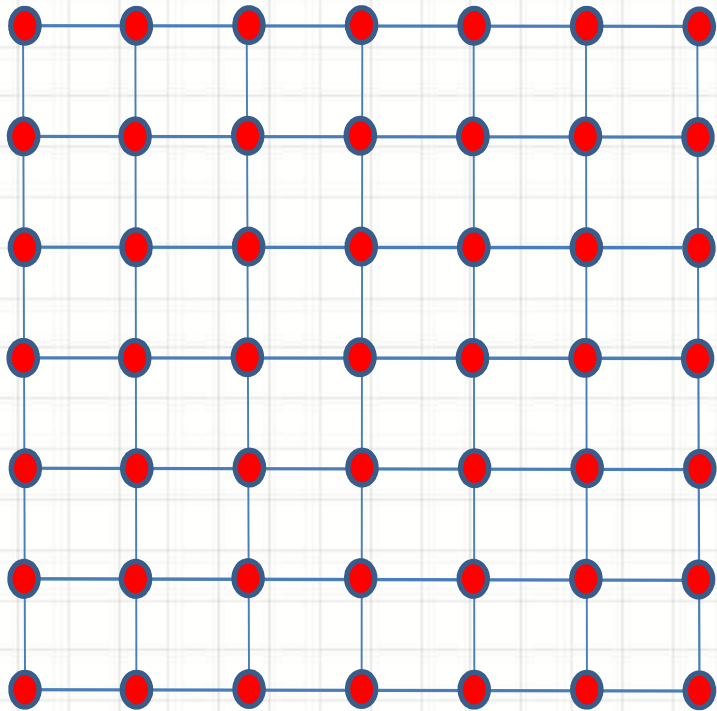
فرض کنید که می‌خواهیم یک تصویر 7×7 را به اندازه 45 درجه دوران دهیم

اعمال تبدیل هندسی خطی روی یک تصویر



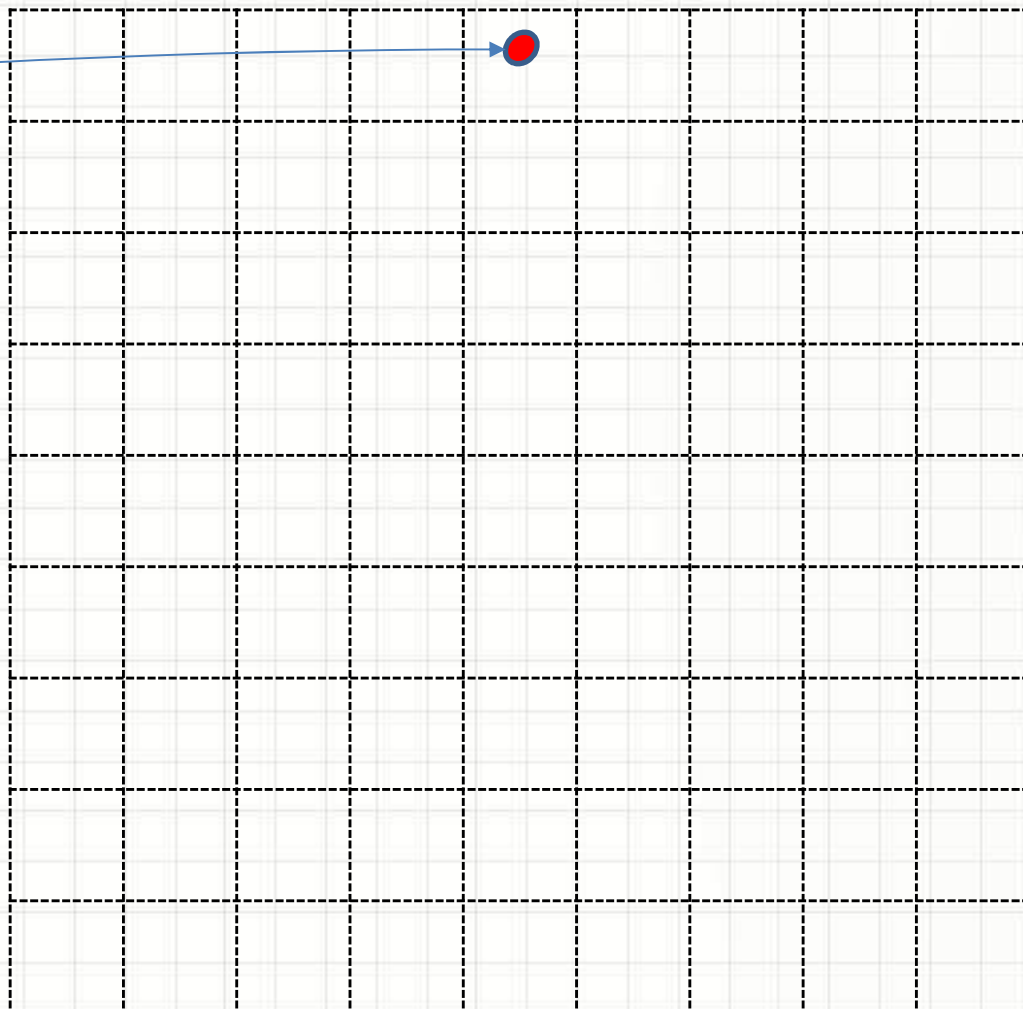
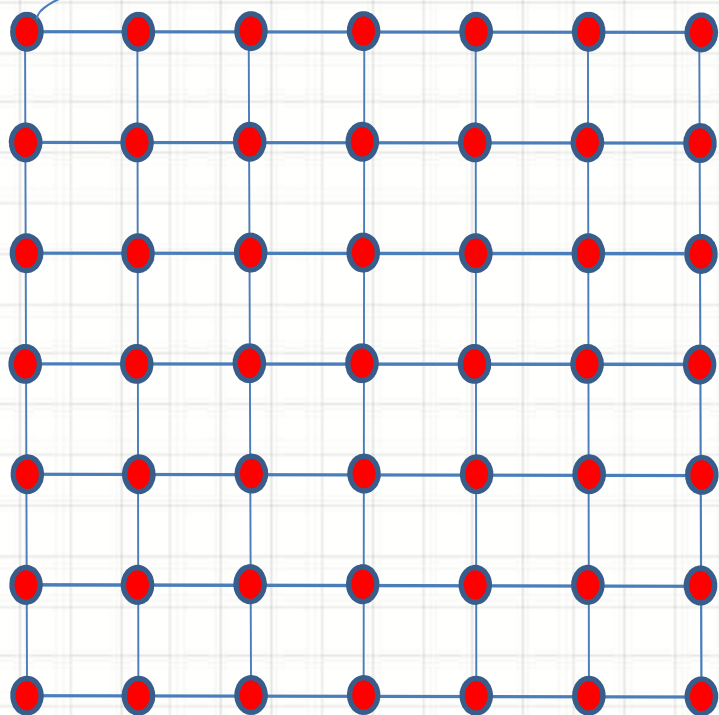
لذا باید در تصویر جدید مقدار پیکسل‌ها را مشخص کنیم.

اعمال تبدیل هندسی خطی روی یک تصویر



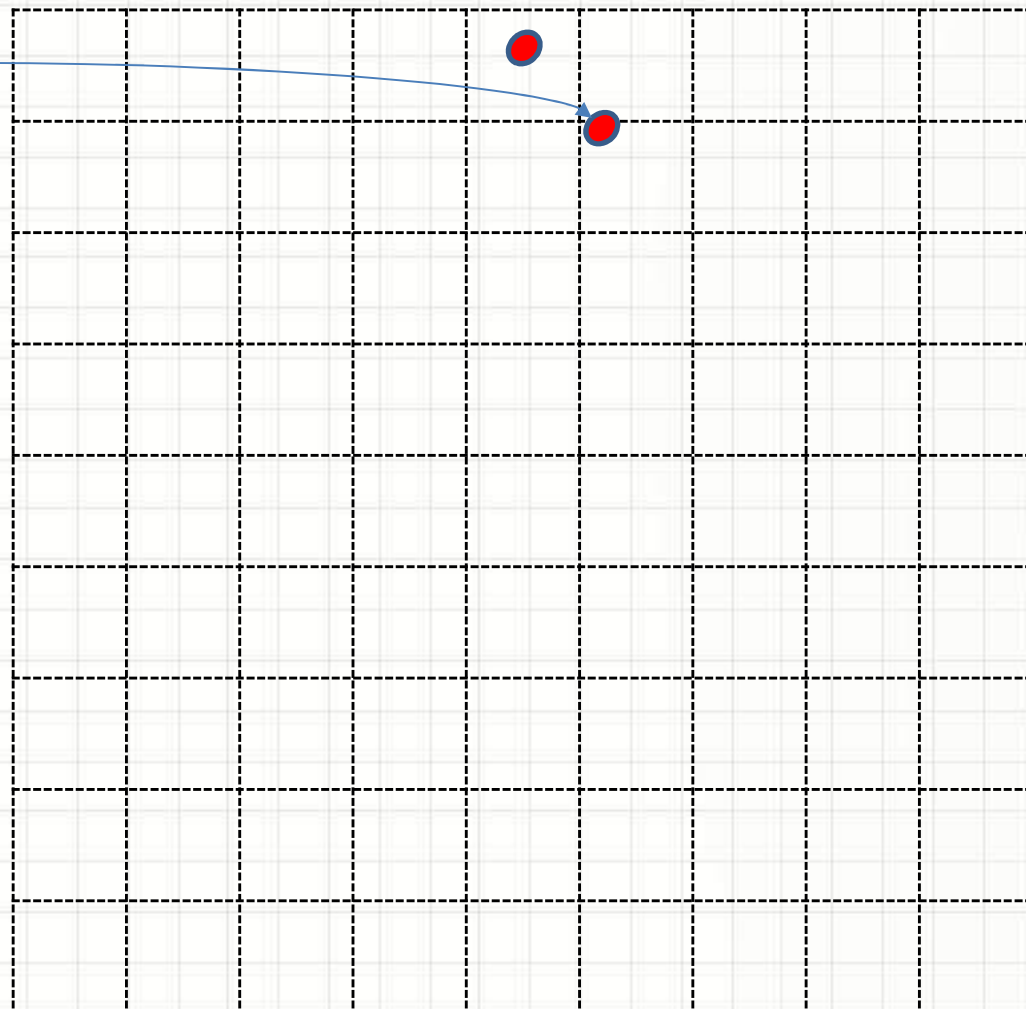
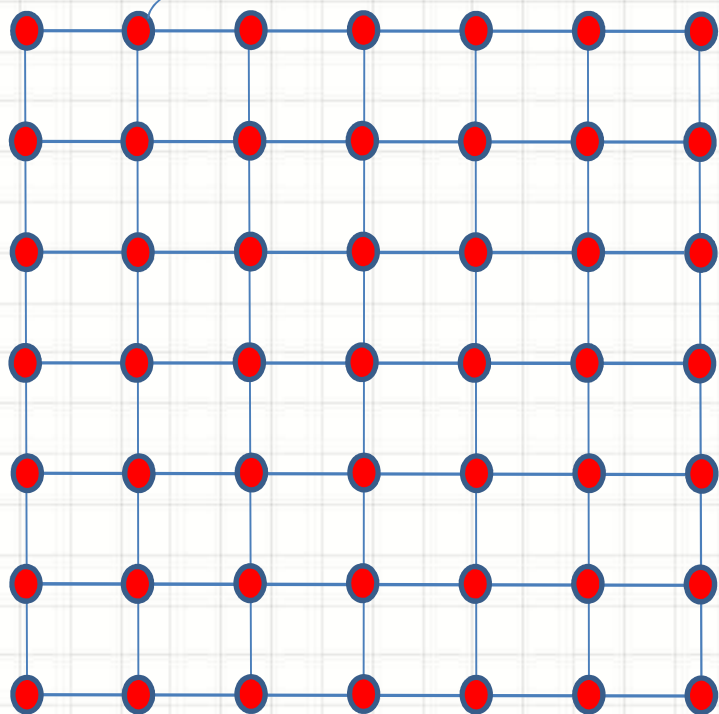
تک تک پیکسل‌ها را منتقل می‌کنیم.

اعمال تبدیل هندسی خطی روی یک تصویر



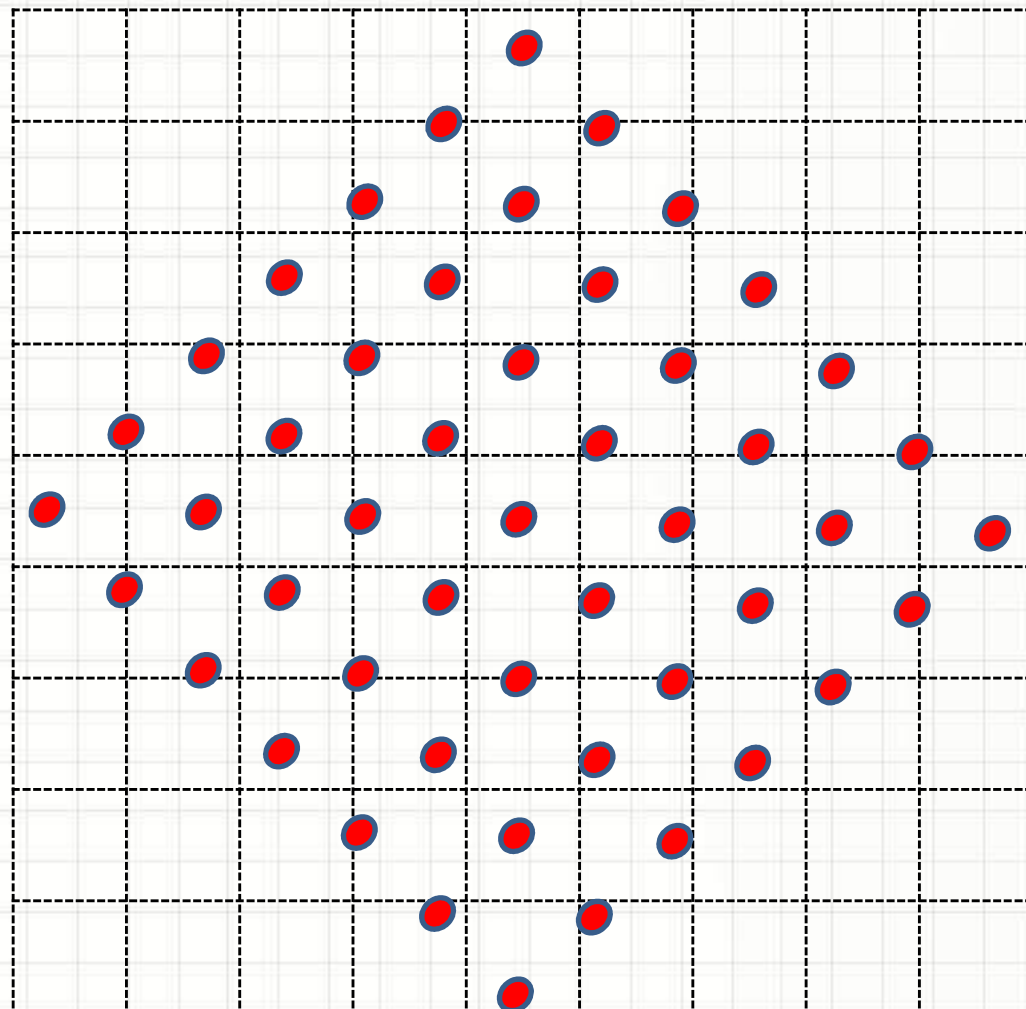
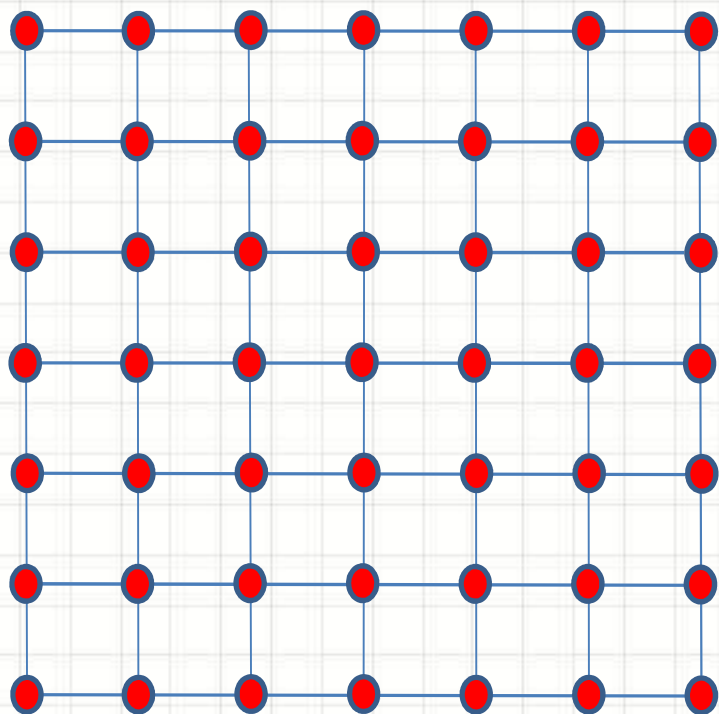
تک تک پیکسل‌ها را منتقل می‌کنیم.

اعمال تبدیل هندسی خطی روی یک تصویر



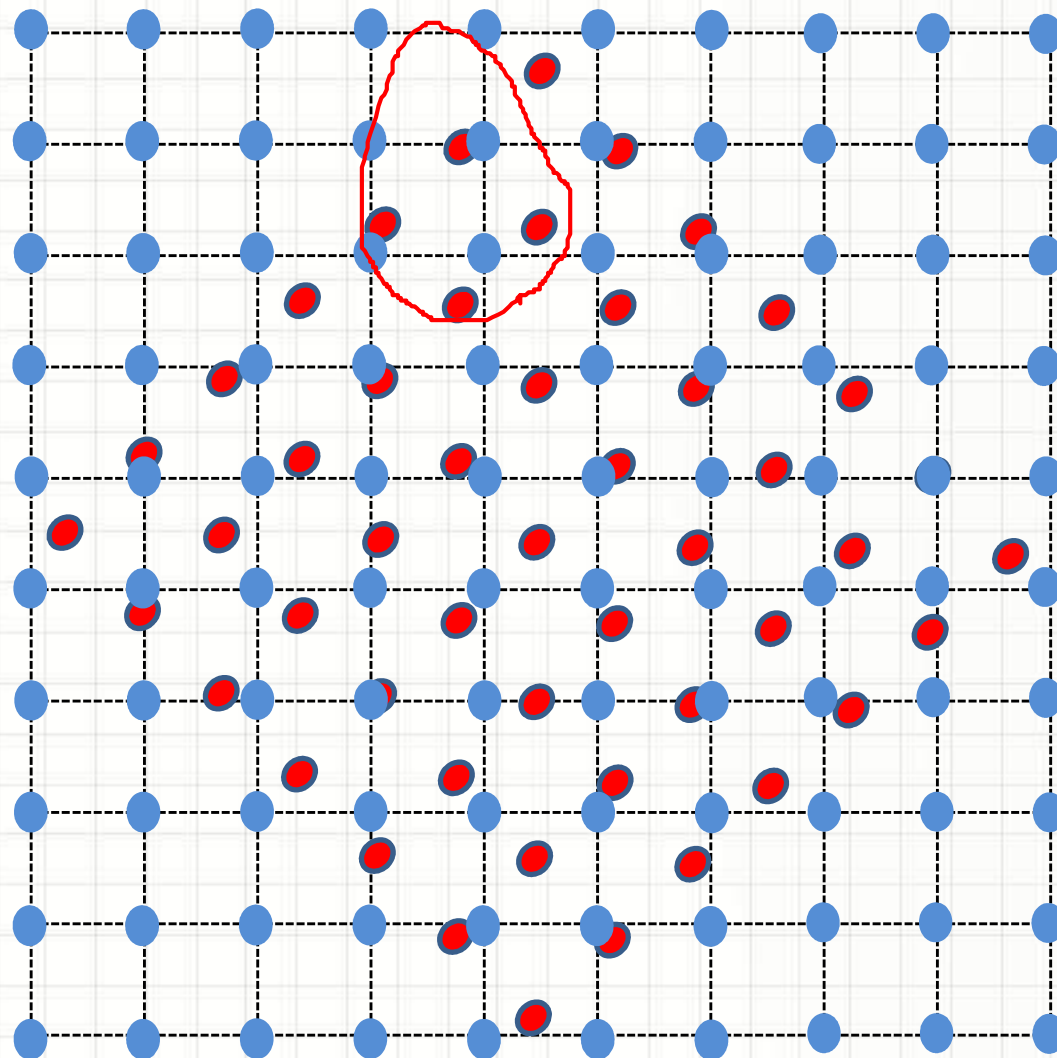
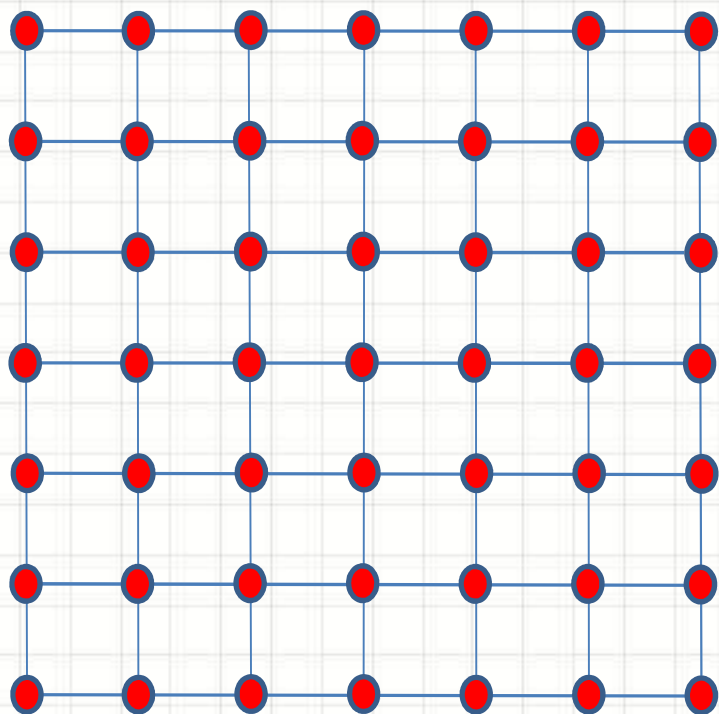
تک تک پیکسل‌ها را منتقل می‌کنیم.

اعمال تبدیل هندسی خطی روی یک تصویر



تا این که کلیه پیکسل ها تصویر اولیه به محیط تصویر دوم منتقل شوند.

اعمال تبدیل هندسی خطی روی یک تصویر



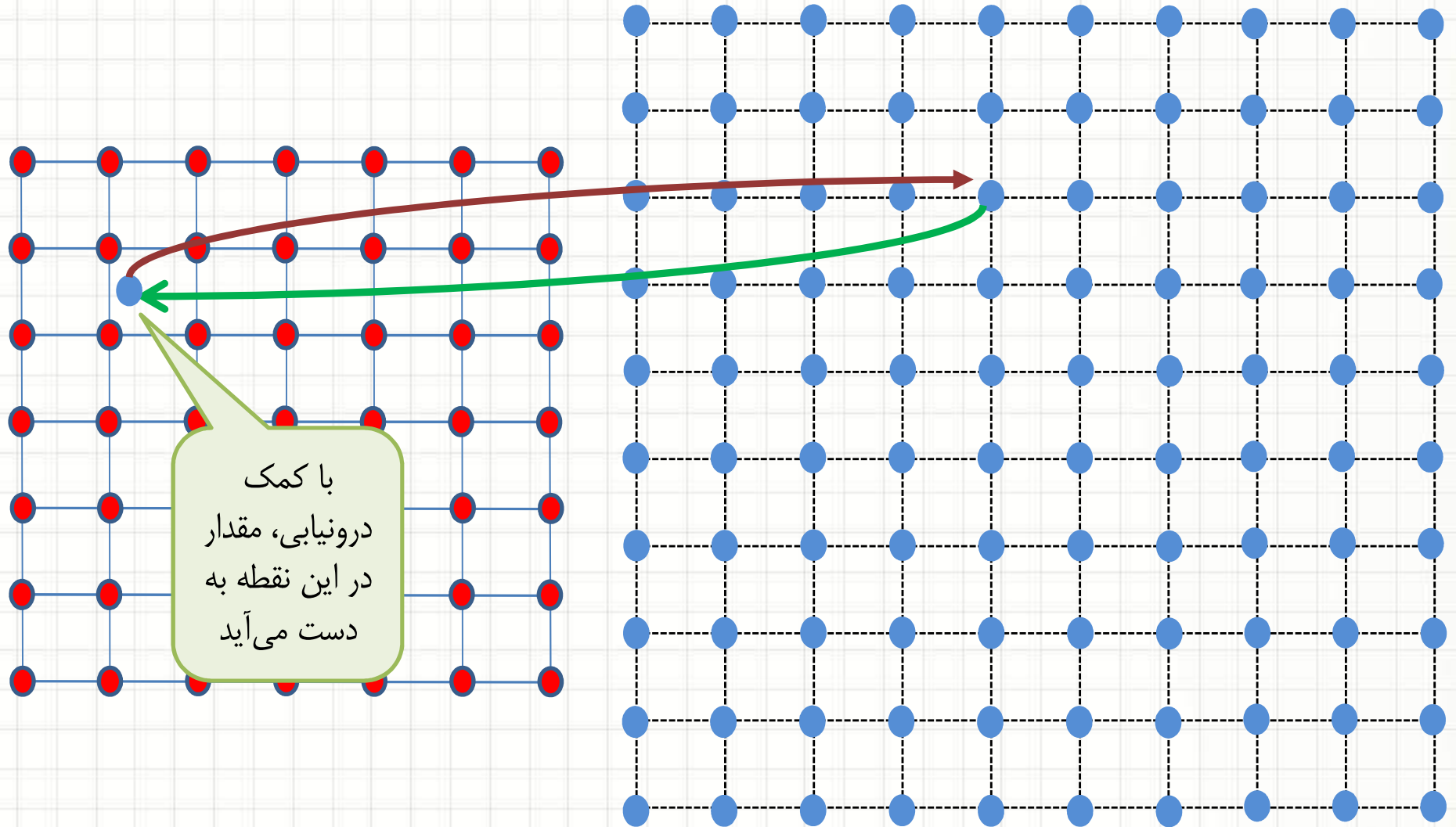
اما پیکسل‌های منتقل شده دقیقاً بر روی پیکسل‌های فایل مقصد قرار نمی‌گیرند.

اعمال تبدیل هندسی خطی روی یک تصویر

استفاده از درونیابی دوخطی و دومتعبی پیشنهاد می شود، اما یک اشکال وجود دارد!!!!

راه حل استفاده از درونیابی معکوس است.

درونیابی معکوس



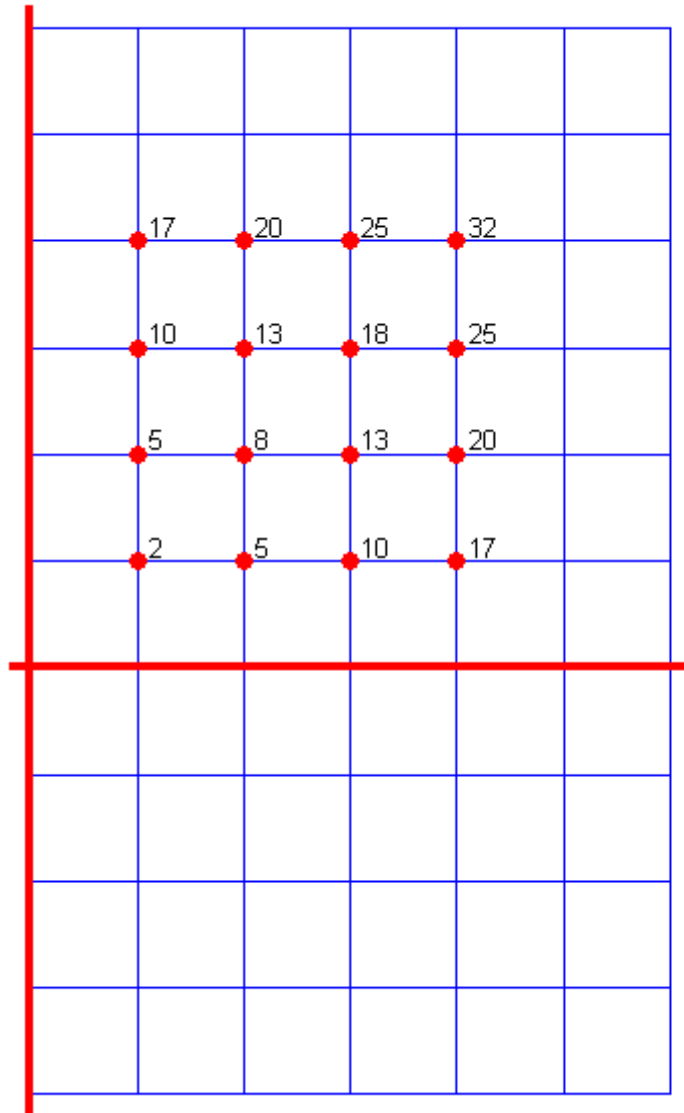
مثال: درونیابی معکوس در دوران یک تصویر

فرض کنید که تصویری به صورت زیر داریم و می‌خواهیم این تصویر را 45 درجه بچرخانیم.

17	20	25	32
10	13	18	25
5	8	13	20
2	5	10	17

برای این منظور، تصویر را در یک محور مختصات دکارتی قرار می‌دهیم. اسلاید بعد...

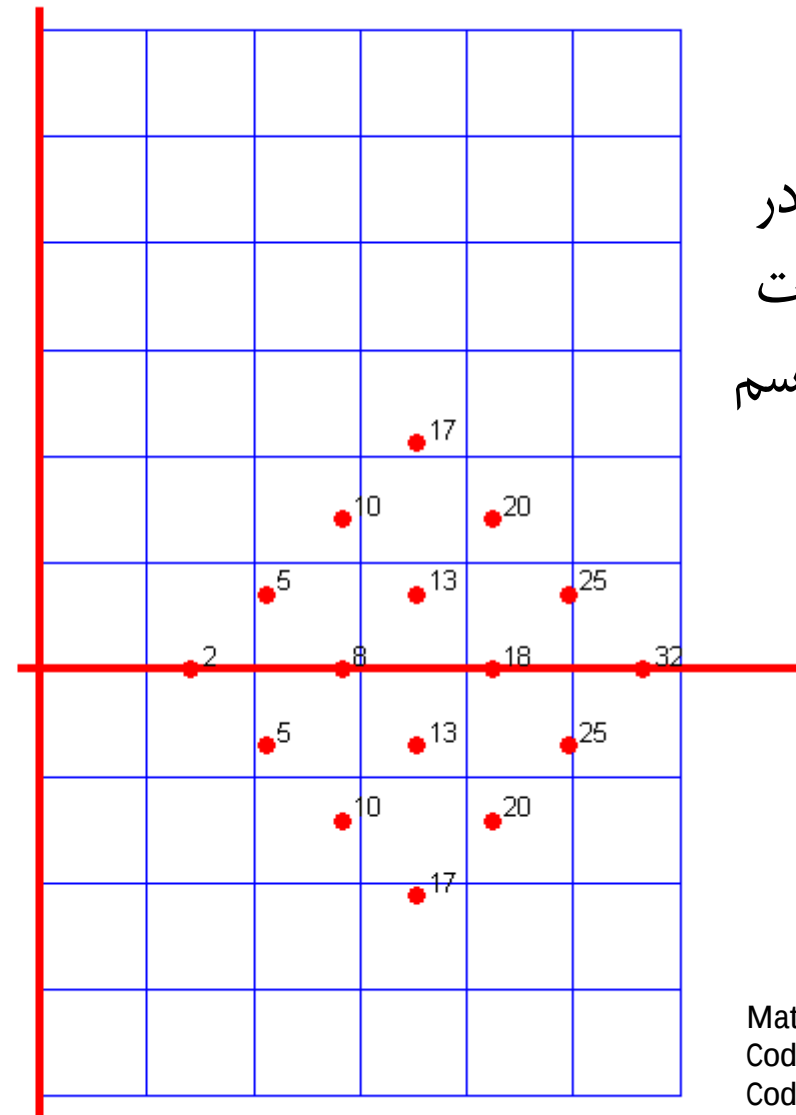
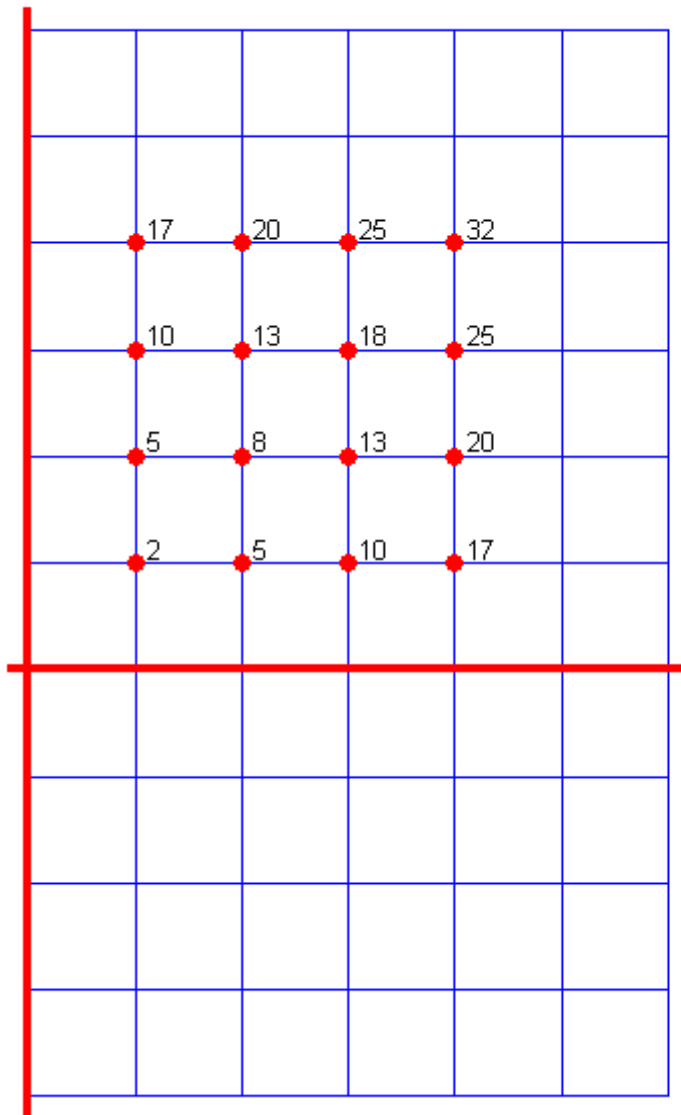
مثال: درونیابی معکوس در دوران یک تصویر



دوران تصویر با کمک تبدیل هندسی
زیر بر روی آدرس پیکسل‌ها انجام
می‌شود.

$$\begin{cases} x' = x \cos(\theta) - y \sin(\theta) \\ y' = x \sin(\theta) + y \cos(\theta) \end{cases}$$

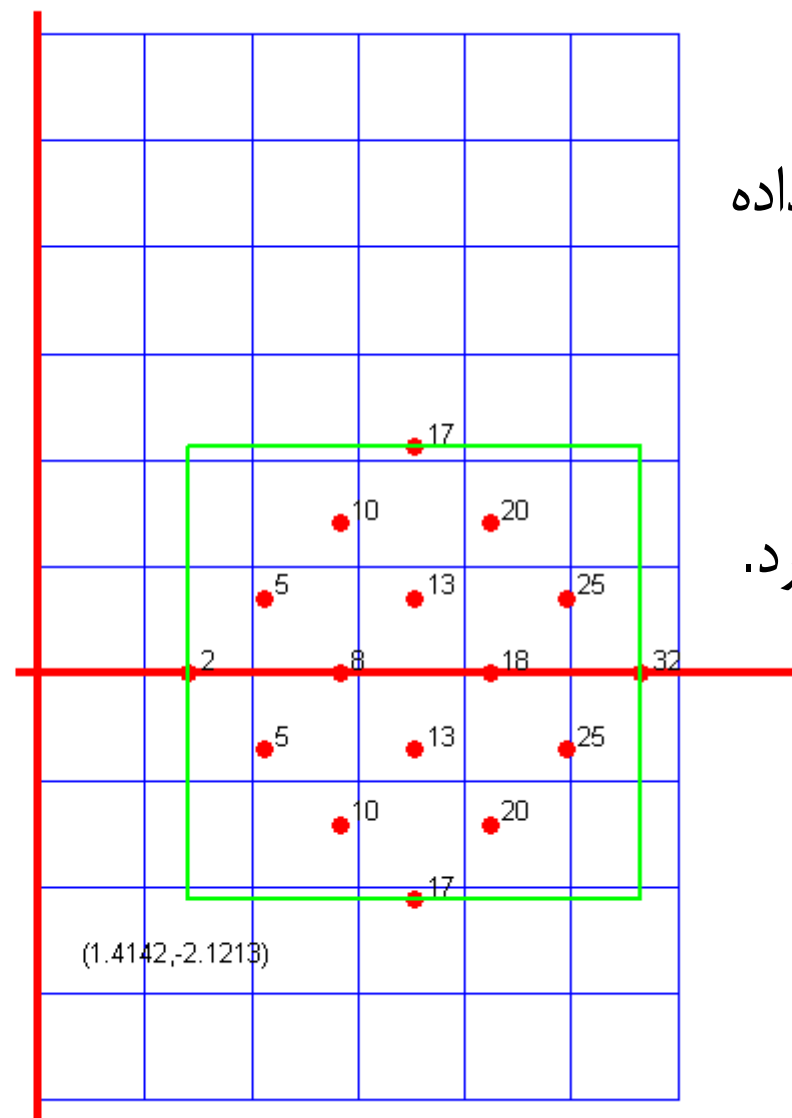
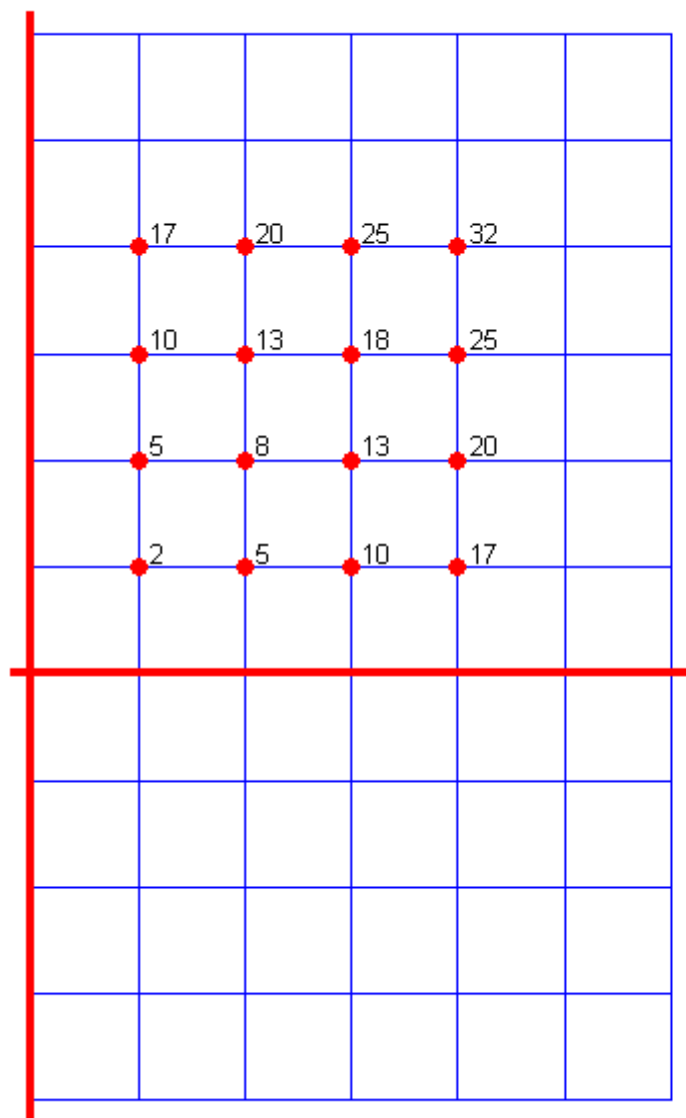
مثال: درونیابی معکوس در دوران یک تصویر



دوران
یافته
تصویر در
مختصات
جدید رسم
شده.

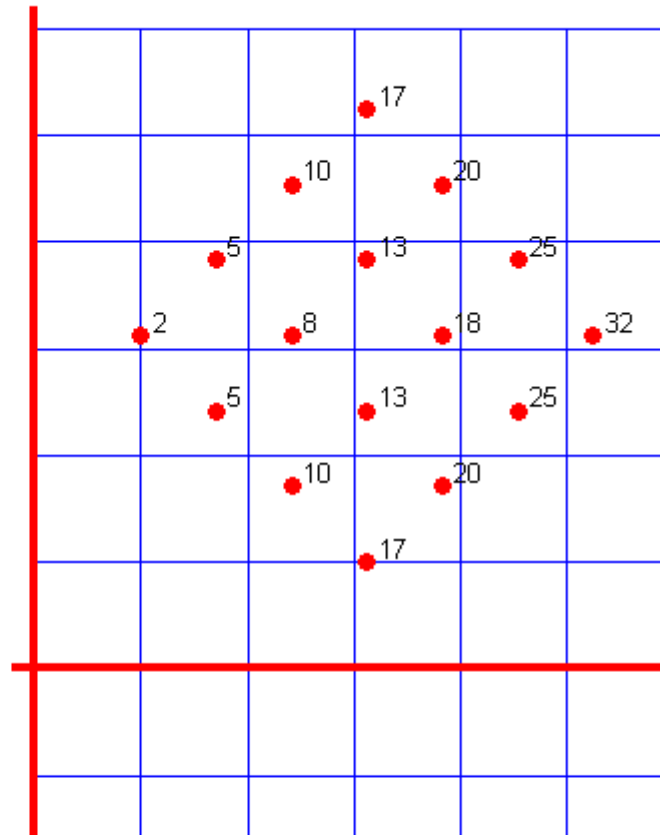
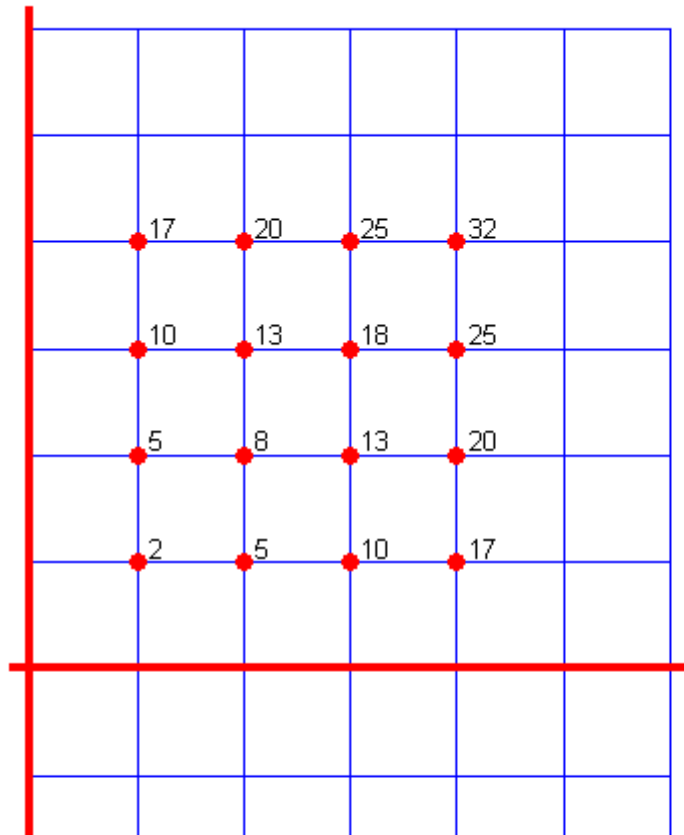
Matlab Codes
Code1.m
Code2.m

مثال: درونیابی معکوس در دوران یک تصویر



تصویر
جدید را
انتقال داده
تا در
مکان
مناسب
قرار گیرد.

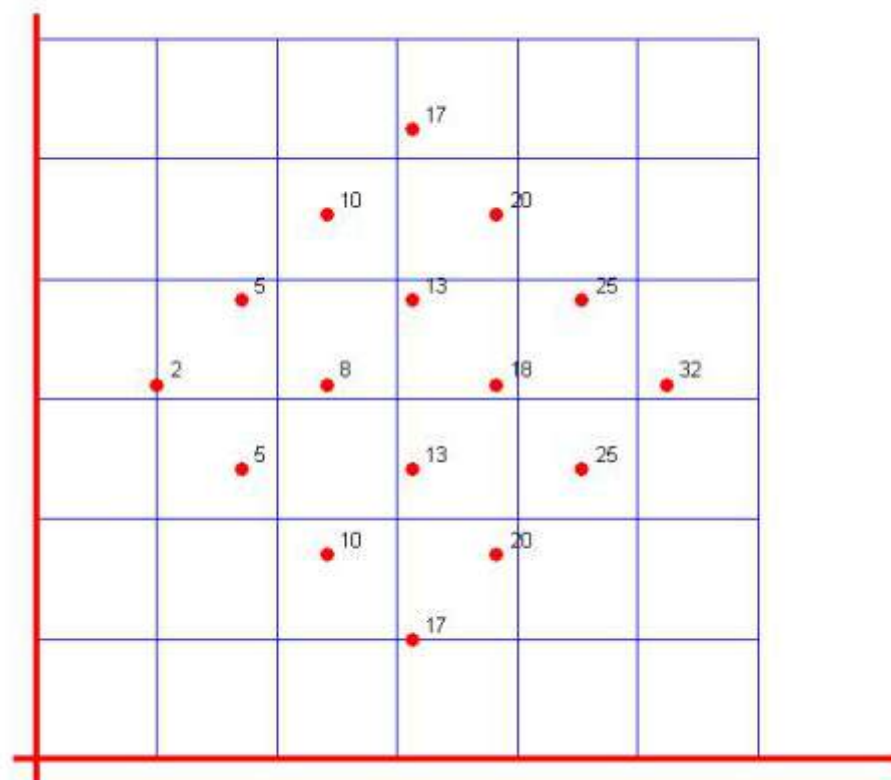
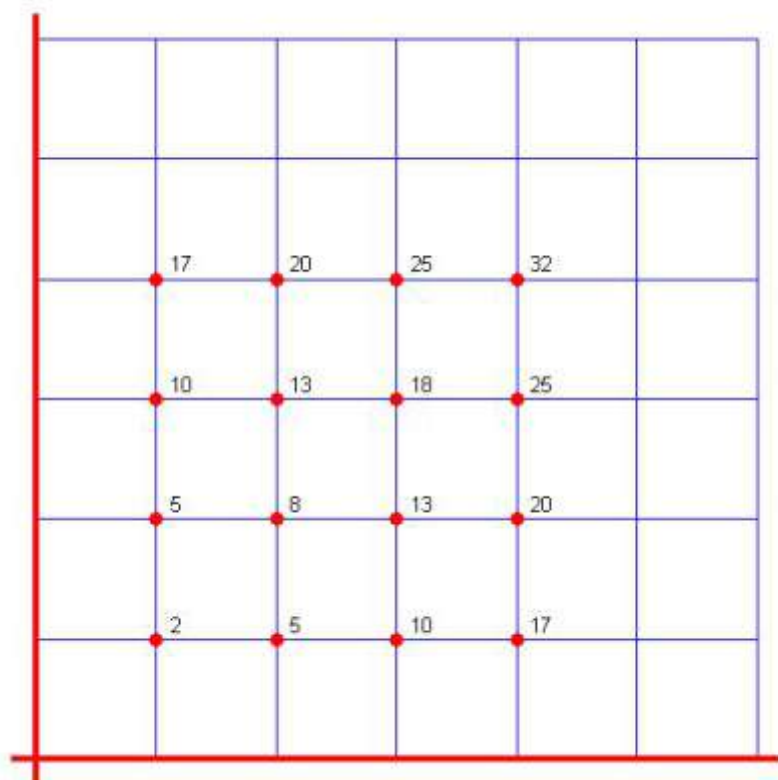
مثال: درونیابی معکوس در دوران یک تصویر



تابع تبدیل
را با توجه
به انتقال
نقاط
به‌هنگام
می‌کنیم.

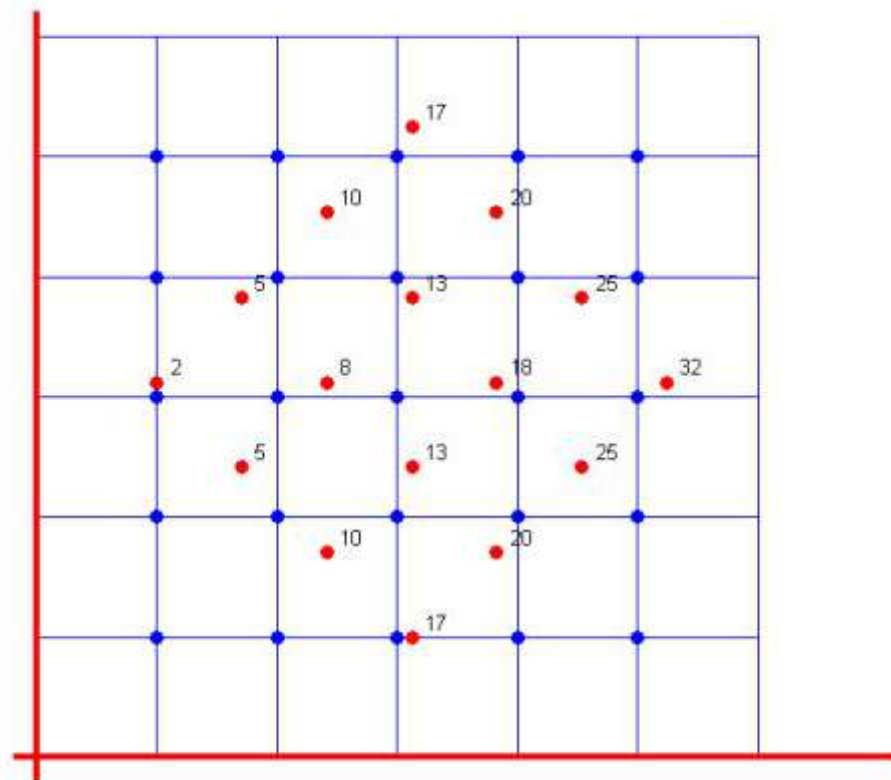
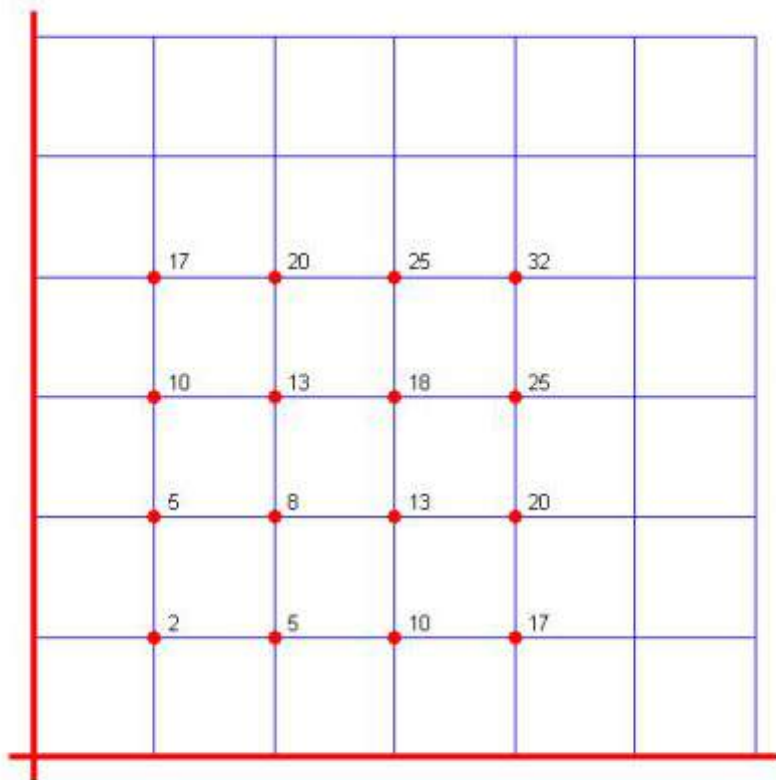
$$\begin{cases} x' = x \cos(\theta) - y \sin(\theta) - 1.4142 \\ y' = x \sin(\theta) + y \cos(\theta) - (-2.1213) \end{cases}$$

مثال: درونیابی معکوس در دوران یک تصویر



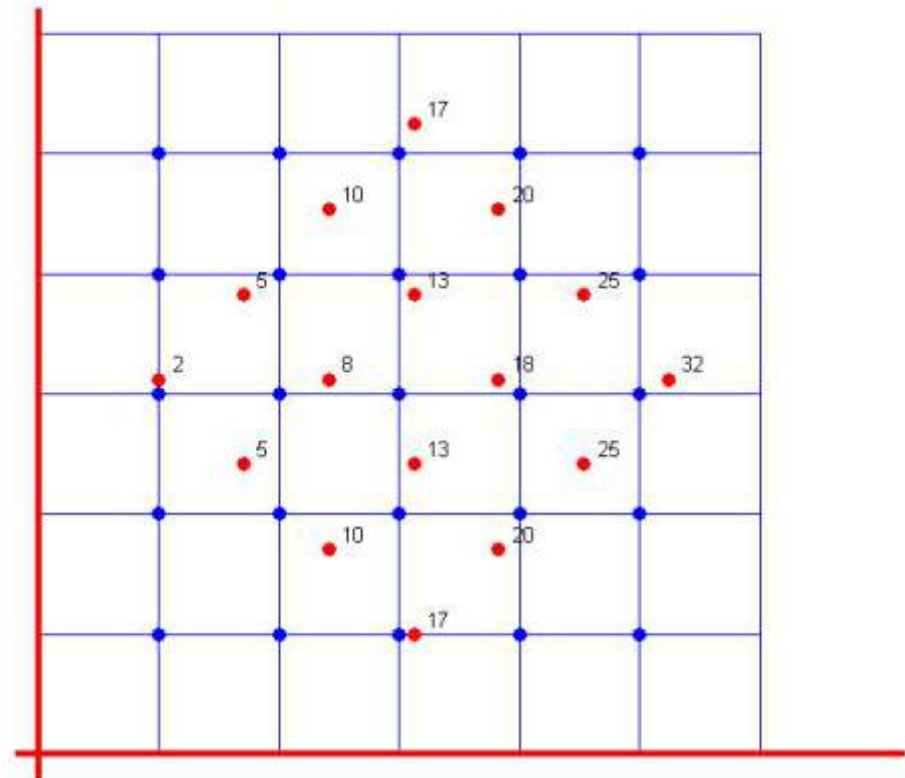
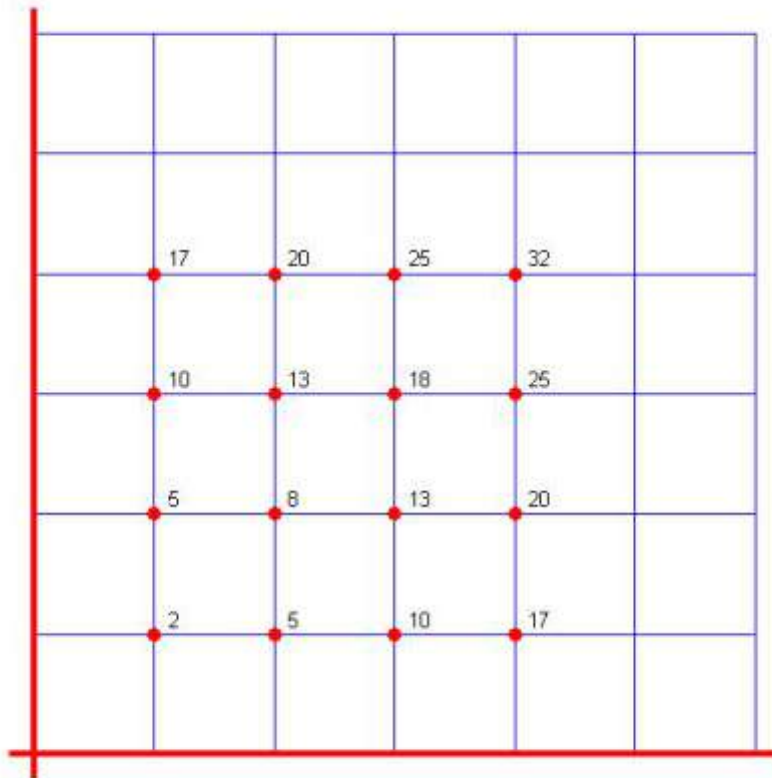
پیکسل‌های منتقل شده در یک مش منظم قرار ندارد.
(مقدارها در یک مش نامنظم مشخص است)

مثال: درونیابی معکوس در دوران یک تصویر



مقدر نقاط در یک مش منظم (نقاط آبی رنگ) باید مشخص گردد

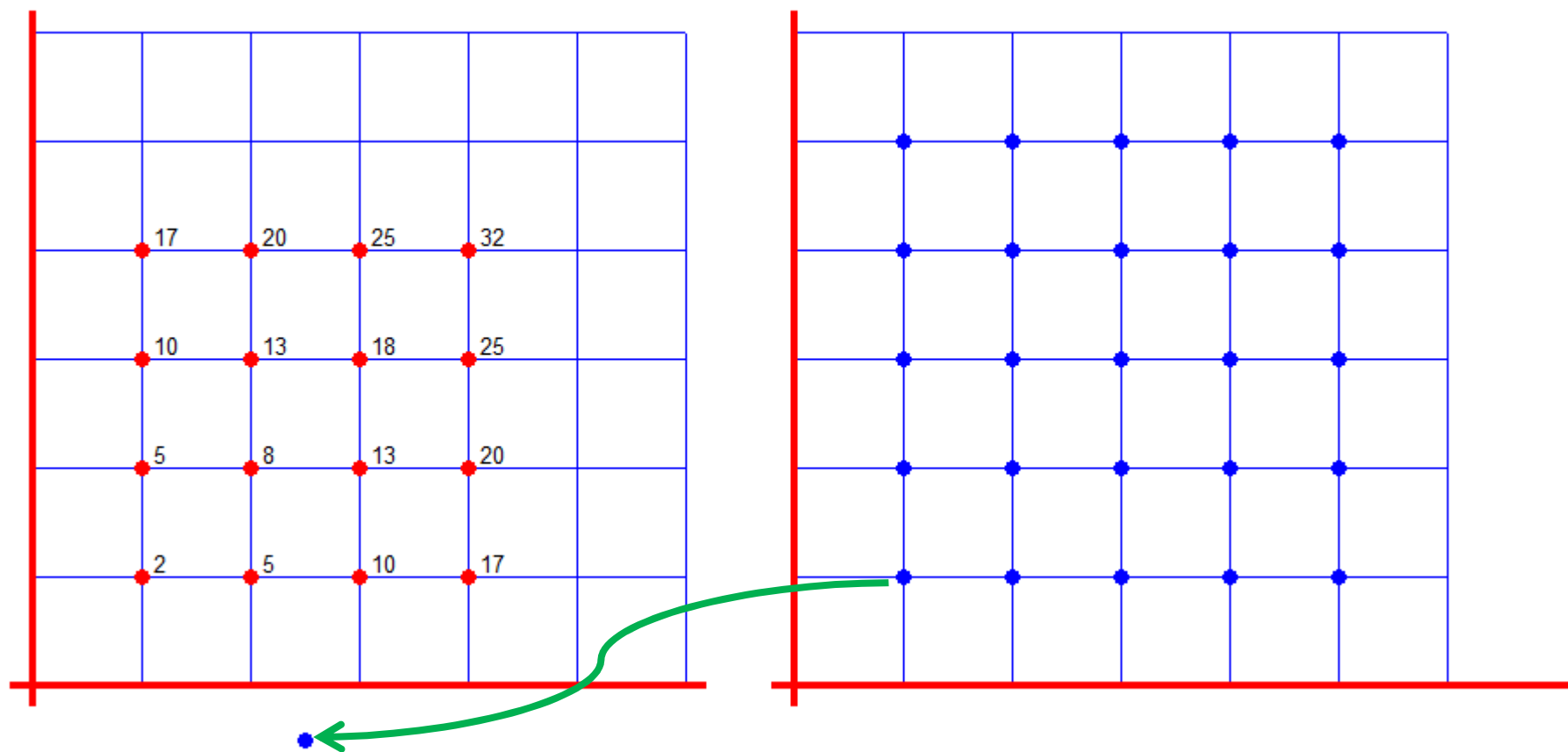
مثال: درونیابی معکوس در دوران یک تصویر



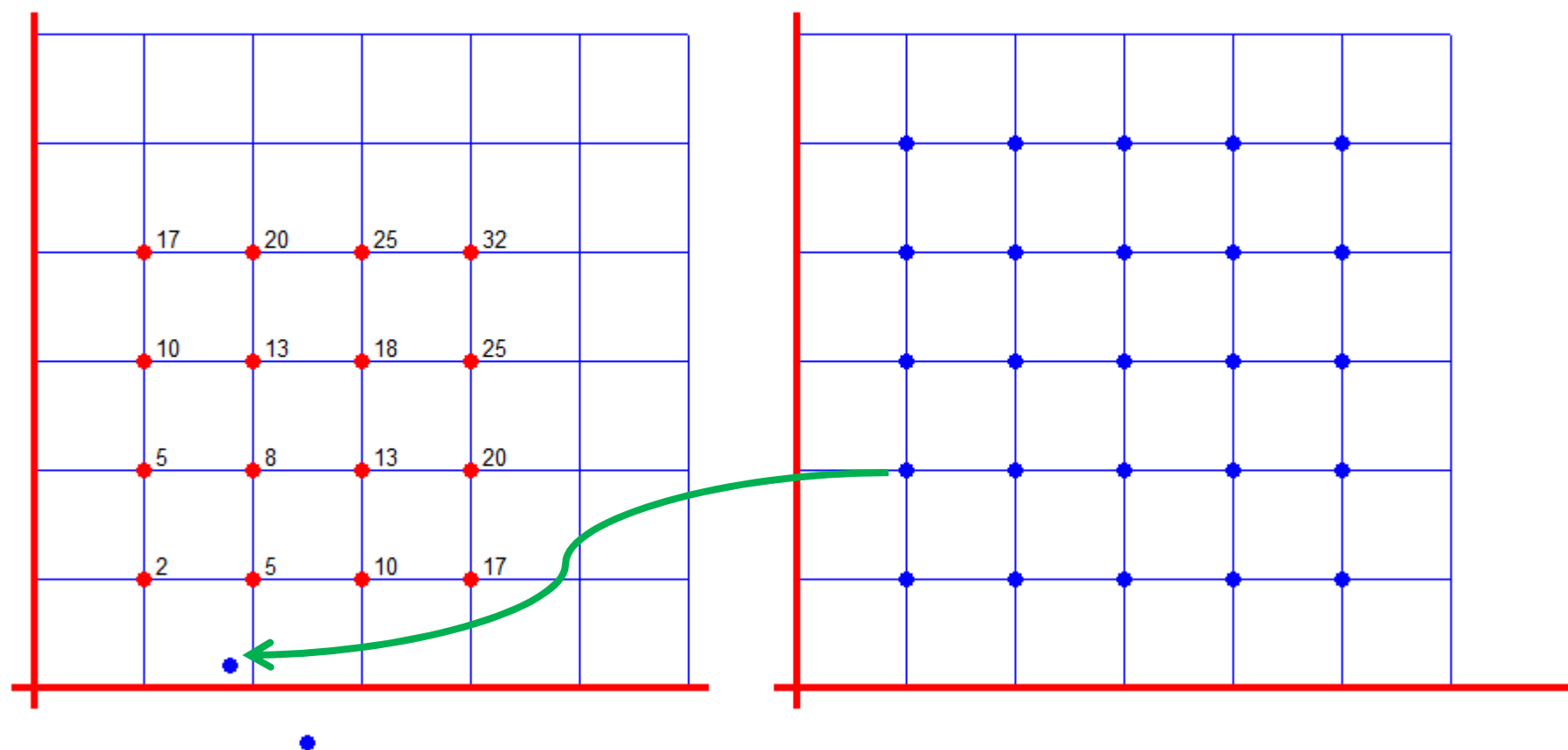
$$\begin{cases} x = +x' \cos(\theta) + y' \sin(\theta) + 1.4142 \\ y = -x' \sin(\theta) + y' \cos(\theta) + (-2.1213) \end{cases}$$

تبدیل
معکوس به
صورت زیر
است.

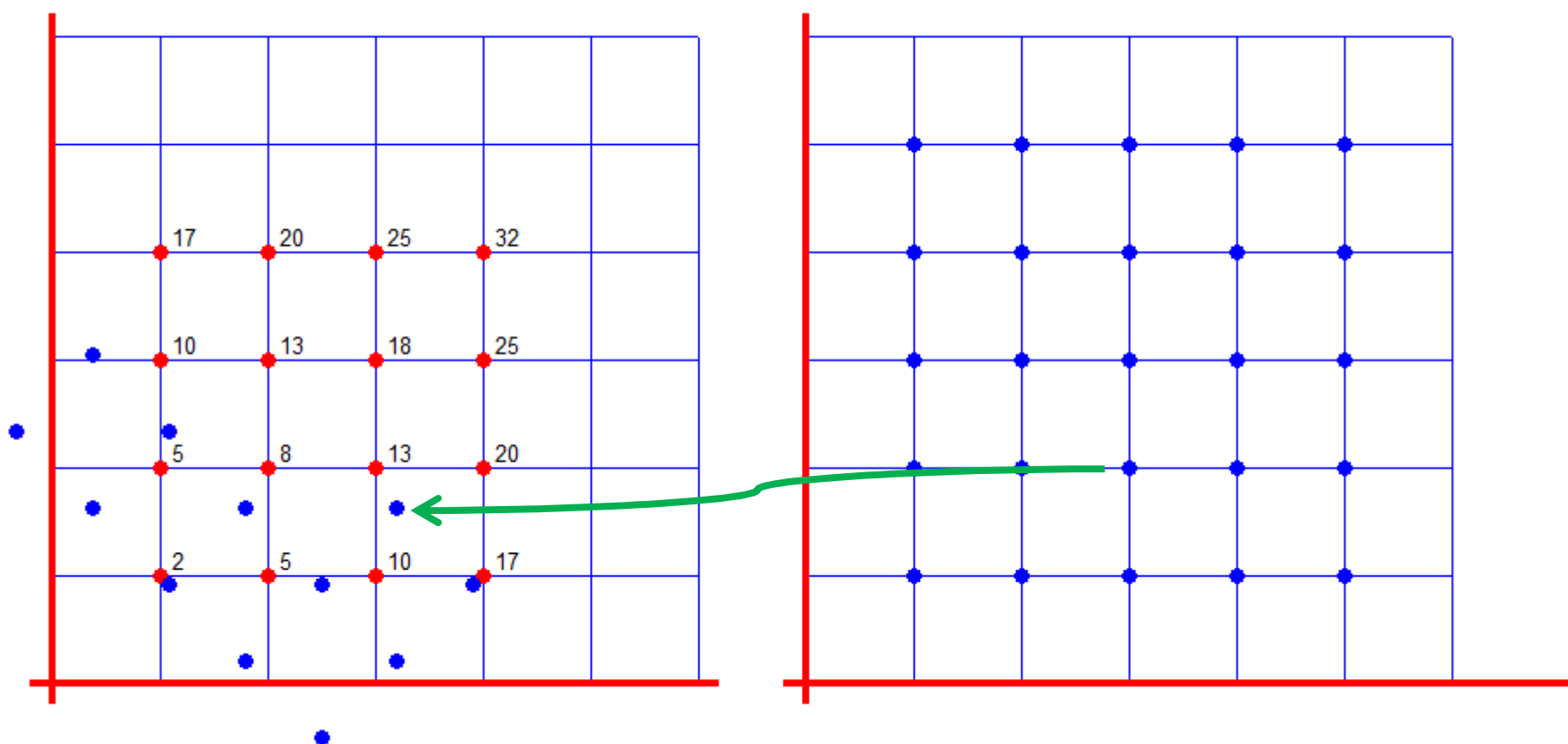
مثال: درونیابی معکوس در دوران یک تصویر



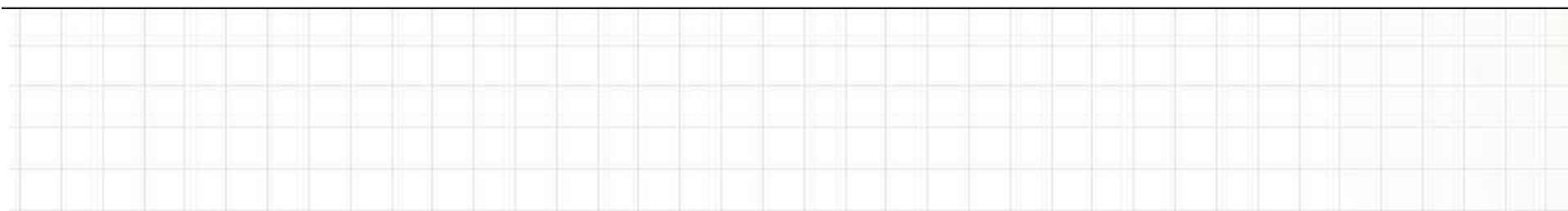
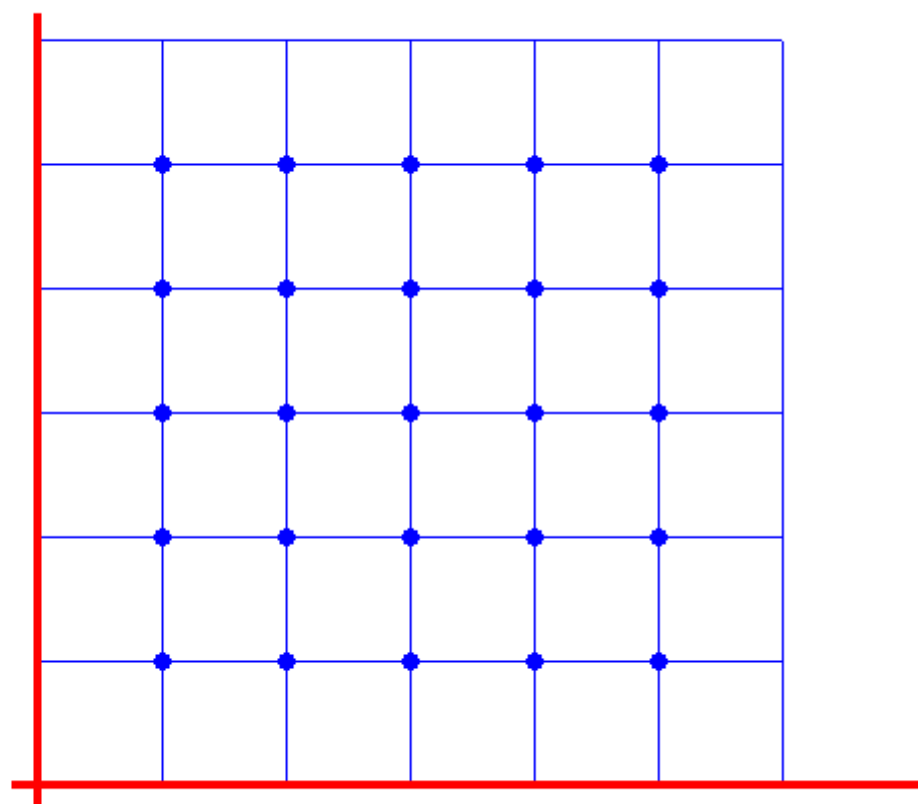
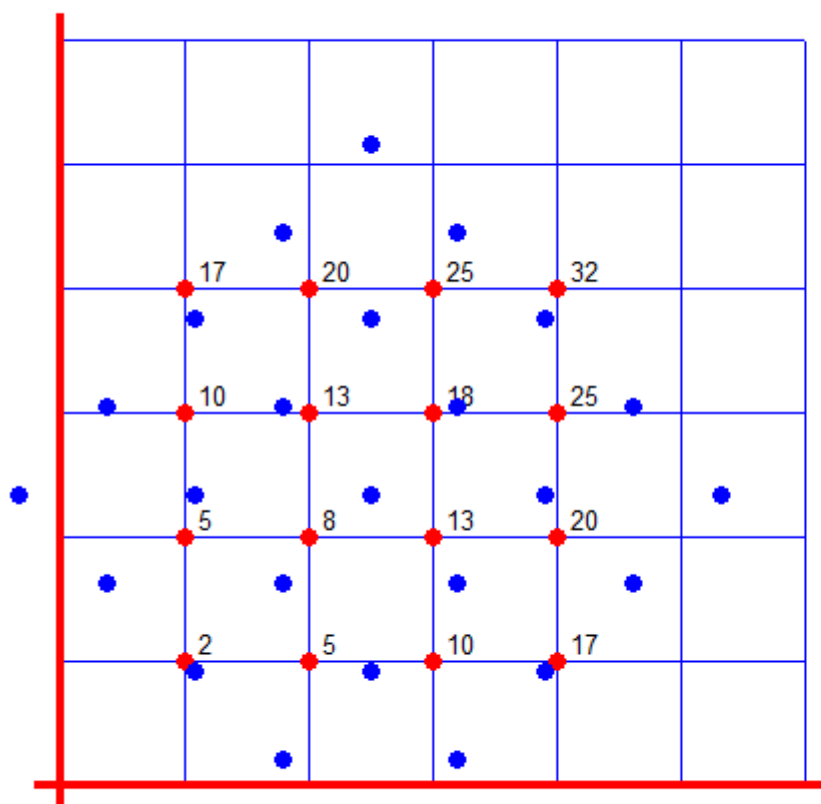
مثال: درونیابی معکوس در دوران یک تصویر



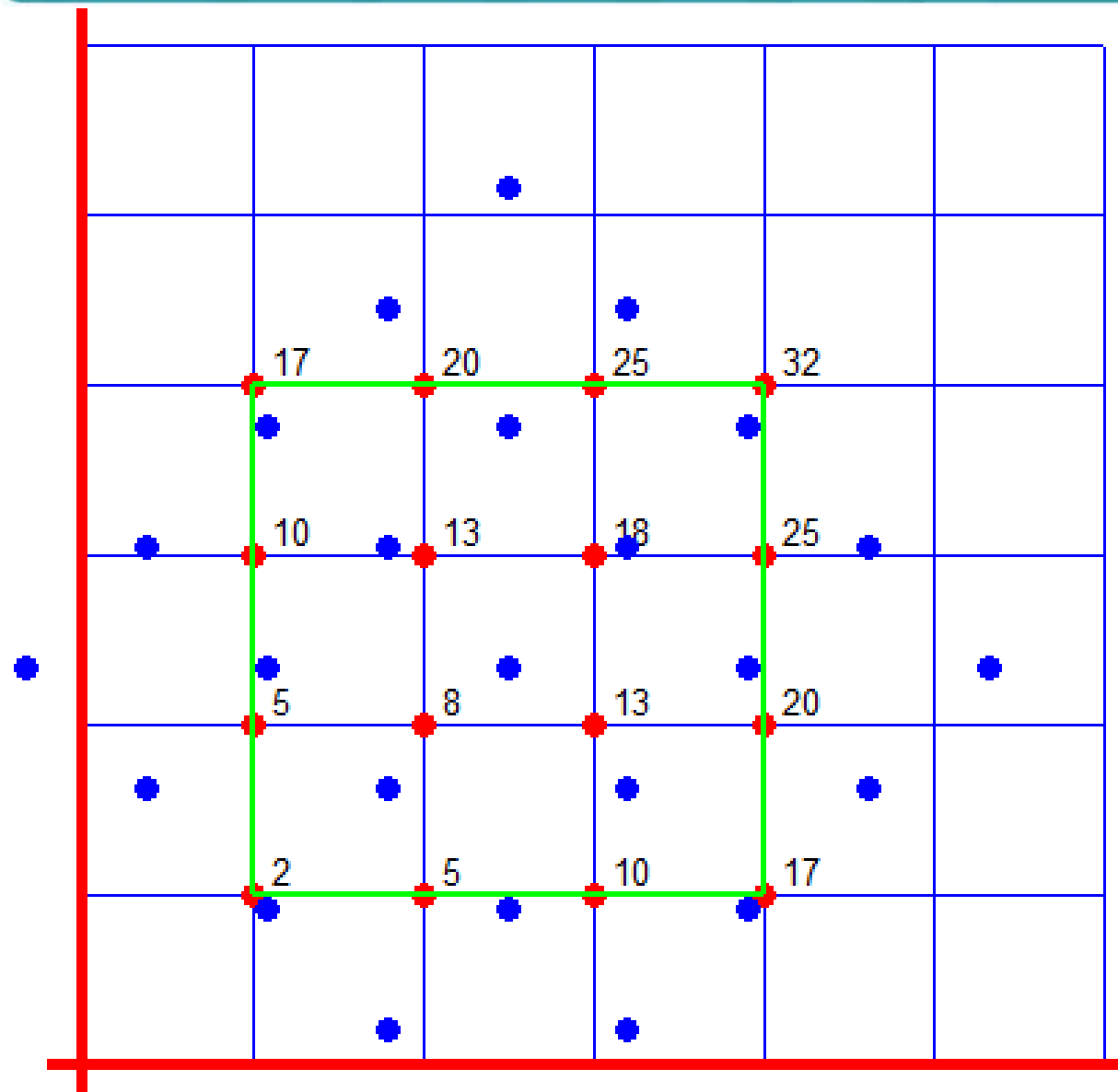
مثال: درونیابی معکوس در دوران یک تصویر



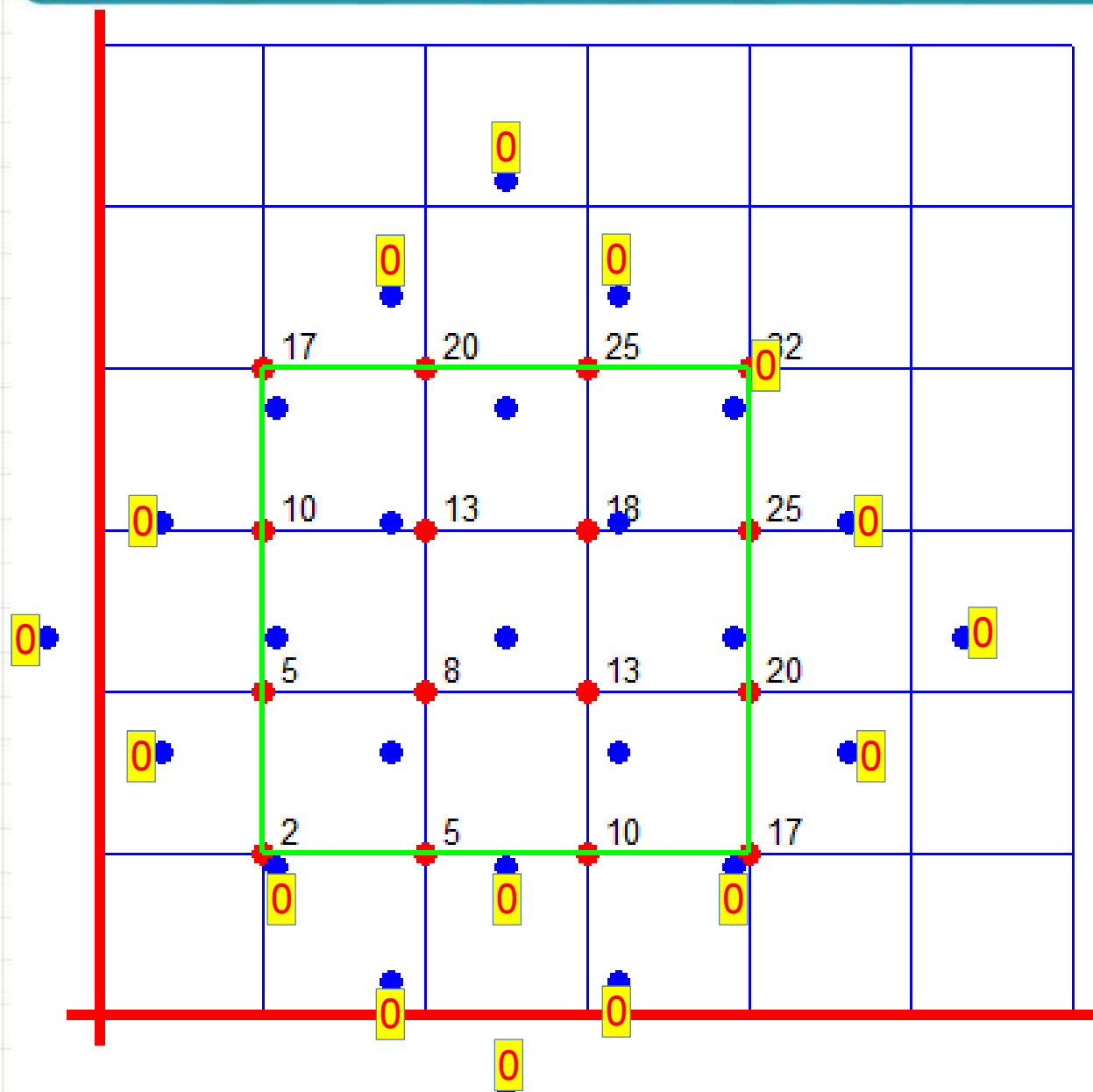
مثال: درونیابی معکوس در دوران یک تصویر



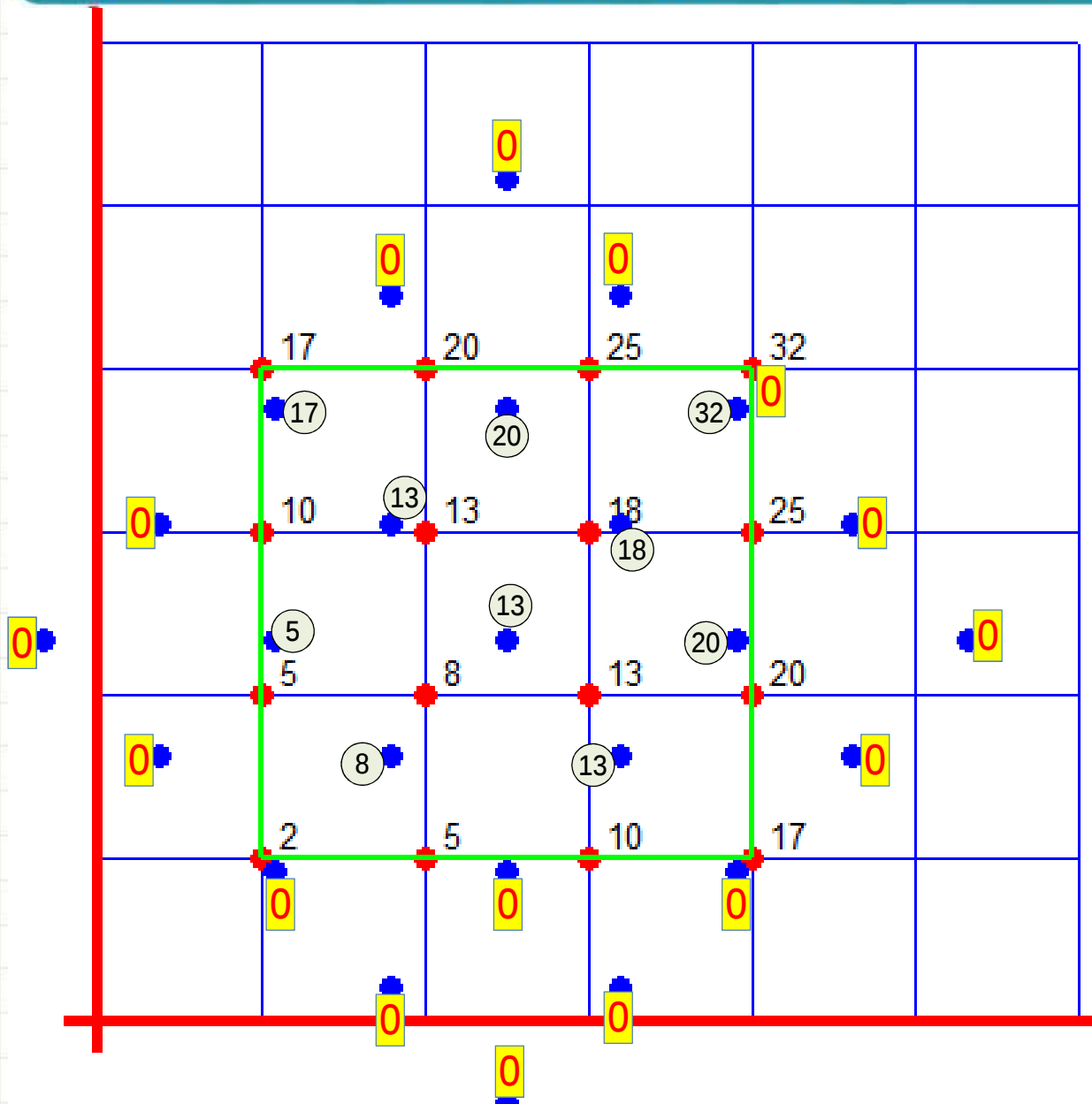
مثال: درونیابی معکوس در دوران یک تصویر



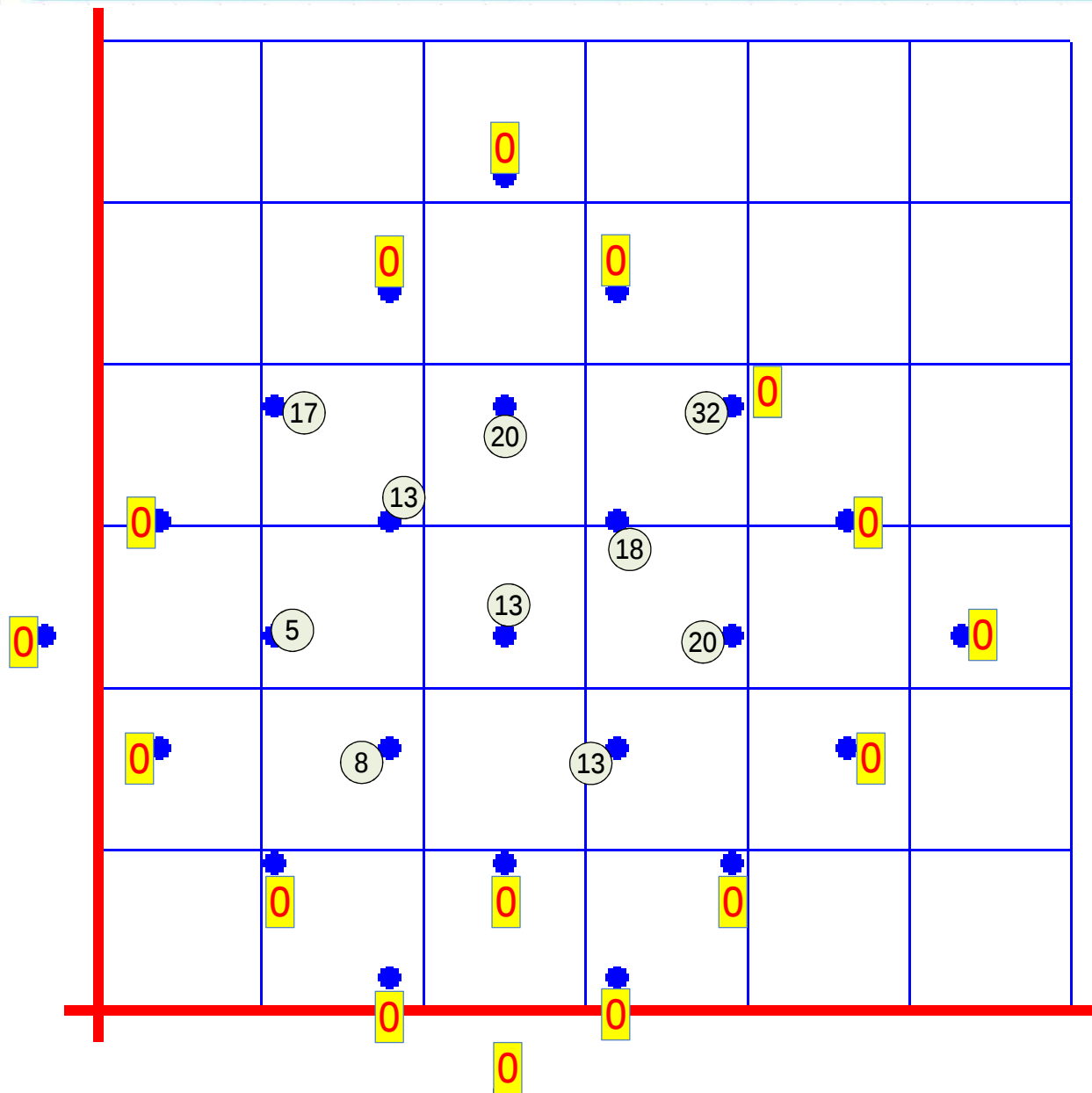
مثال: درونیابی معکوس در دوران یک تصویر



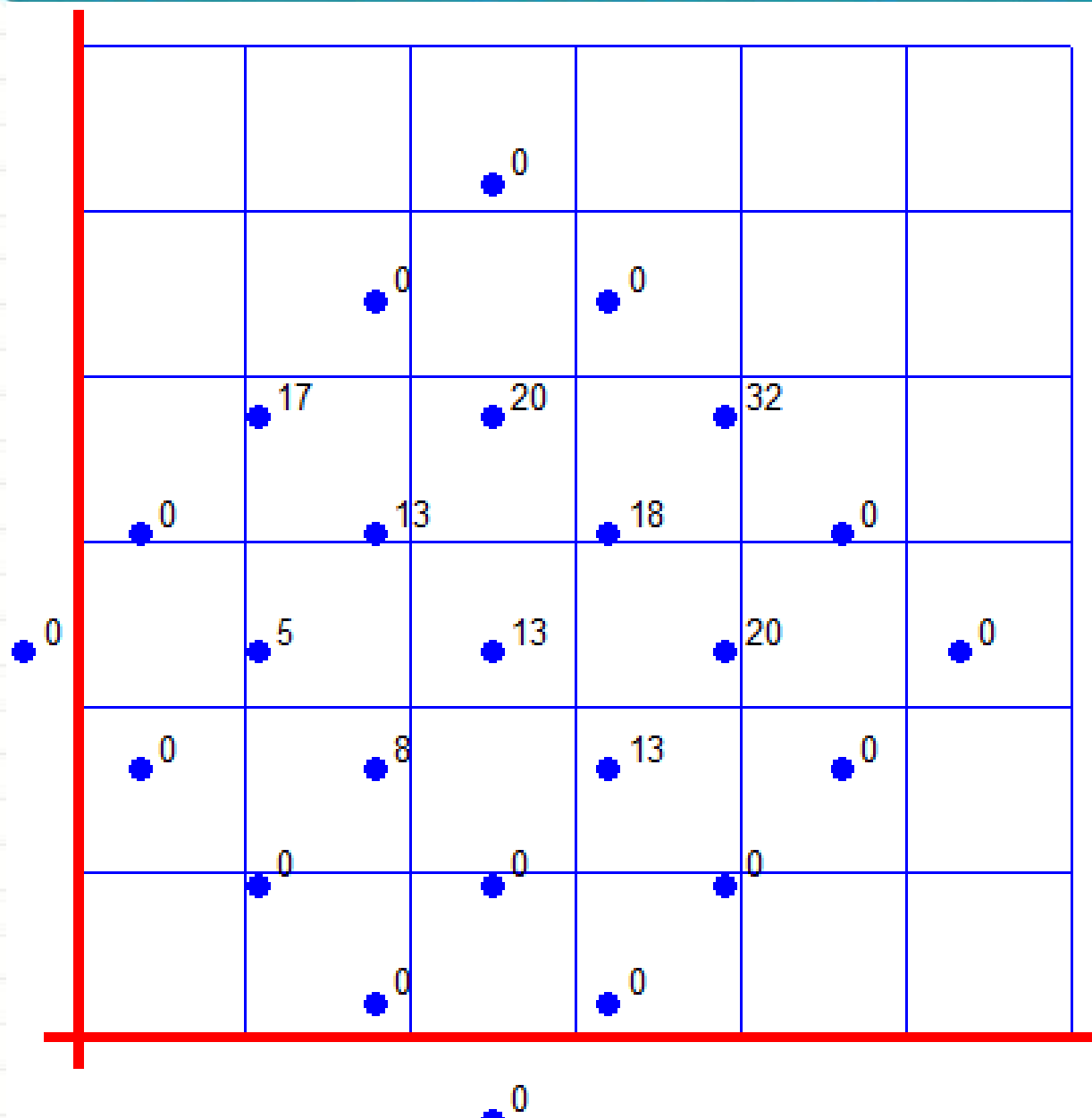
مثال: درونیابی معکوس در دوران یک تصویر



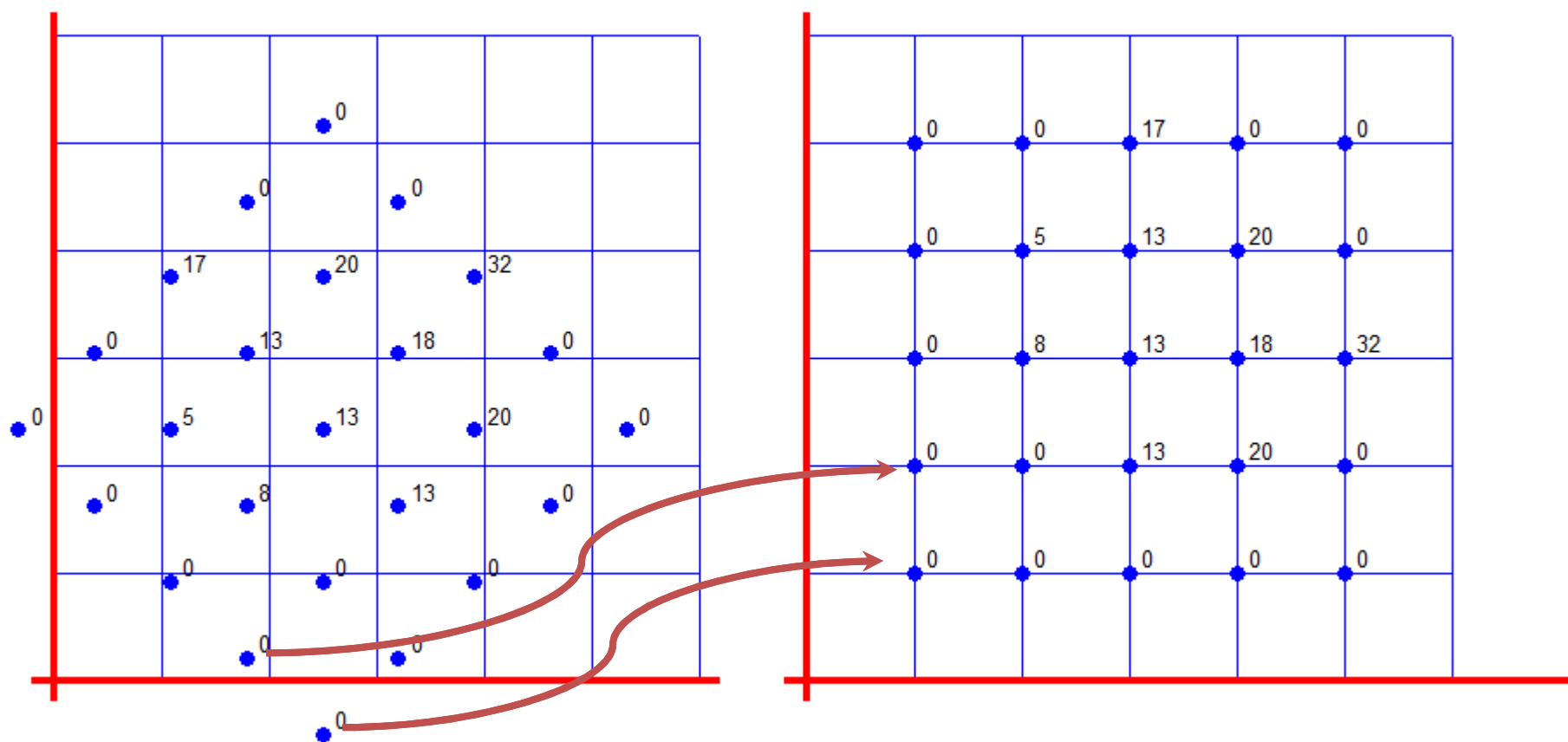
مثال: درونیابی معکوس در دوران یک تصویر



مثال: درونیابی معکوس در دوران یک تصویر



مثال: درونیابی معکوس در دوران یک تصویر



مثال: بزرگ‌نمایی ۴ برابری تصویر



نزدیک‌ترین همسایگی

دوخطی

دومکعبی



مثال: بزرگ‌نمایی ۵ برابری تصویر



نزدیک‌ترین همسایگی

دوخطی

دومکعبی



بهبود شدت در
تصاویر دیجیتال
Intensity
Enhancement

انگیزه



تصویر مقابل یک
تصویر با کنتراست
پایین است. که تقریباً
غیرقابل تشخیص است.

سوال: آیا امکان دارد که این تصویر را طوری دستکاری کرد که وضوح آن بیشتر شود؟

انگیزه

برای پاسخ سوال در ابتدا تصویر را دقیق تر مطالعه می کنیم. برای این منظور تکه ای از تصویر را بزرگنمایی می کنیم.

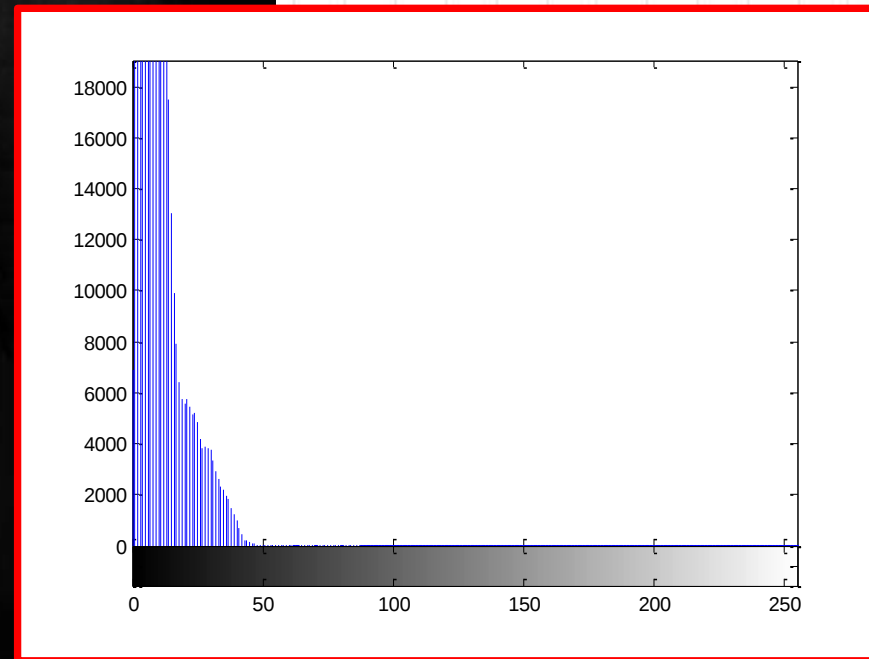


16	15	16	17	18	23	29	30	29	29	28	29	27	28	30	29	28	28	29
17	18	18	18	17	21	29	30	29	29	29	29	28	28	29	29	28	29	28
17	18	17	15	13	19	28	30	29	28	29	30	29	28	30	30	29	30	29
15	15	12	10	9	16	27	30	28	27	30	30	29	28	30	30	29	31	30
12	12	10	8	8	16	27	30	28	27	30	30	29	28	30	30	29	32	31
9	11	12	11	10	16	25	28	28	29	29	29	29	29	30	29	30	30	29
11	13	14	14	14	16	23	28	28	29	30	30	29	29	28	27	29	30	30
12	14	13	14	15	15	21	27	28	30	31	30	29	28	28	28	29	30	30
12	13	11	12	14	14	20	26	28	29	30	30	29	29	28	29	30	30	29
12	12	11	12	14	15	18	26	29	29	29	29	29	29	29	30	29	29	29
9	11	13	14	15	16	18	24	28	29	30	30	30	30	30	29	29	30	30
8	10	12	13	13	15	18	24	30	30	30	30	30	30	29	29	29	30	29
8	11	12	13	14	15	16	20	28	29	29	30	30	29	29	30	29	30	30

		16	17	18	23	29	30	29	29	28	29	27	28	30	29	28	28	29
					21	29	30	29	29	29	29	28	28	29	29	28	29	28
17	18	17	15	13	19	28	30	29	28	29	30	29	28	30	30	29	30	29
15	15	12	10	9	16	27	30	28	27	30	30	29	28	30	30	29	31	30
12	12	10	8	8	16	27	30	28	27	30	30	29	28	30	30	29	32	31
11				10	16	25	28	28	29	29	29	29	29	30	29	30	30	29
	14	14	14	16	23	28	28	29	30	30	29	29	28	27	29	30	30	
14	13	14	15	15	21	27	28	30	31	30	29	28	28	28	29	30	30	
13	11	12	14	14	20	26	28	29	30	30	29	29	28	29	30	30	29	
12	11	12	14	15	18	26	29	29	29	29	29	29	29	30	29	29	29	
11	13	14	15	16	18	24	28	29	30	30	30	30	30	29	29	30	30	
10	12	13	13	15	18	24	30	30	30	30	30	30	29	29	29	30	29	
8	11	12	13	14	15	16	20	28	29	29	30	30	29	29	30	29	30	30

شدت رنگ خاکستری در این دو پیکسل برابر ۲۷ و ۱۶ می باشد. رنگ این دو پیکسل توسط چشم انسان از یکدیگر قابل تفکیک می باشد.

انگیزه



از هیستوگرام تصویر مشاهده می‌شود که شدت رنگ پیکسل‌ها در بازه $[0, 50]$ متمرکز شده است.

تبدیل



به نظر می‌رسد که اگر شدت رنگ پیکسل‌ها را طوری تغییر دهیم که فاصله بین شدت رنگ پیکسل‌ها افزایش یابد، آن‌گاه وضوح تصویر بهتر می‌شود.

مثلاً شدت پیکسل‌ها را پنج برابر کنیم. در این صورت دو پیکسل با شدت ۲۹ و ۳۰ به شدت‌های ۱۴۵ و ۱۵۰ تبدیل می‌شوند که توسط چشم انسان قابل تفکیک می‌باشند. توجه شود که در این صورت محدوده شدت از $[0, 50]$ به $[0, 250]$ تغییر می‌یابد.

تبدیل و تابع تبدیل

در فوق شدت رنگ پیکسل‌ها را در ۵ ضرب کردیم. به عبارتی: تصویر اولیه به تصویر دیگر **تبدیل** شد که مکان و تعداد پیکسل‌ها تغییر نکرد و فقط شدت رنگ پیکسل‌ها طبق یک ضابطه تغییر کرد.

با توجه به اینکه هر تصویر یک ماتریس است می‌توان گفت که: تبدیل **تابعی** است که یک ماتریس را به ماتریس دیگر تبدیل می‌کند.
تابع مذکور به **تابع تبدیل** یا **تابع تبدیل شدت** موسوم می‌باشد.

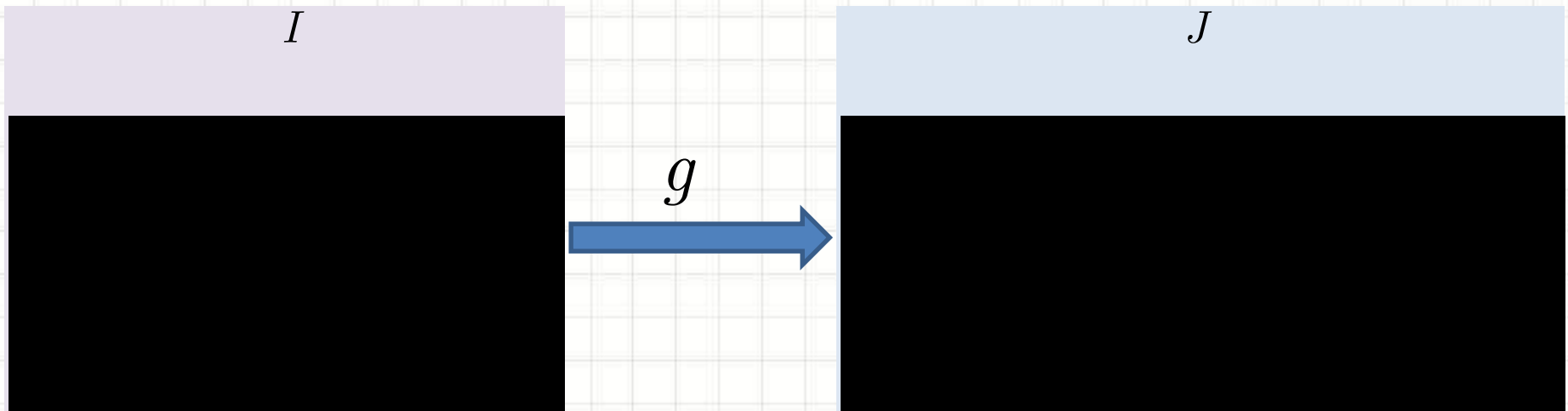
در مثال فوق تابع تبدیل تابعی است که یک عدد را می‌گیرد و حاصلضرب آن عدد در ۵ را برمی‌گرداند. به عبارت دیگر تابع تبدیل عبارت است از

$$g(i) = 5i$$

در واقع اگر تصویر اولیه I باشد و شدت پیکسل (x,y) برابر $I(x,y)$ باشد، آن‌گاه تبدیل g تصویر I را به تصویر J تبدیل می‌کند، به طوری که شدت پیکسل (x,y) در تصویر J عبارت است از $g(I(x,y))$.

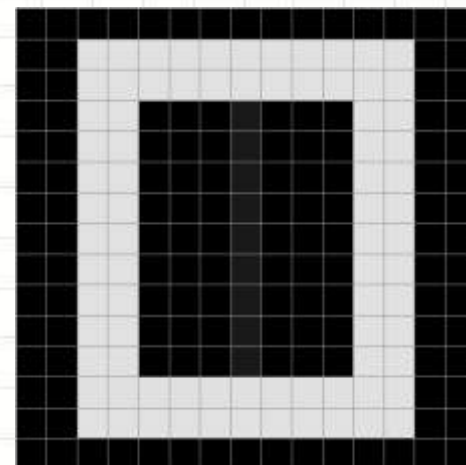
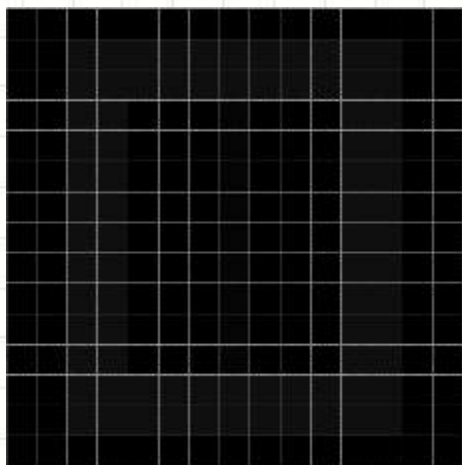
تابع تبدیل شدت

همان گونه که ذکر شد، تبدیل یک تصویر چیزی نیست به جزء اعمال یک تابع موسوم به تابع تبدیل شدت، بر روی مولفه‌های ماتریس تصویر.



[illegible]

$$g : a \longrightarrow a^2$$

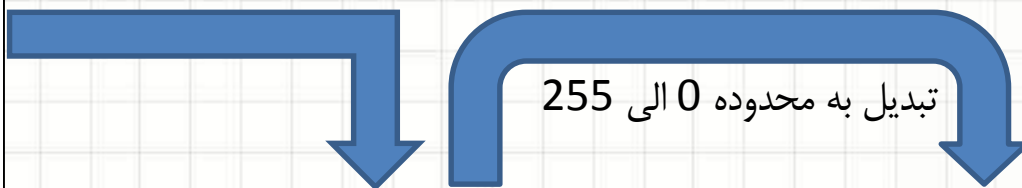
[illegible]

تابع تبدیل شدت-مثال

اگر عمق بیت برابر ۸ باشد، آن گاه تابع تبدیل روی اعداد ۰ الی ۲۵۵ عمل می کند و باید اعداد صحیح در این محدوده نیز برگرداند. اگر ضابطه تابع تبدیل به گونه ای باشد که اعدادی خارج از محدوده صفر الی ۲۵۵ تولید کرد، آن گاه باید اعداد بزرگتر از ۲۵۵ به ۲۵۵ تبدیل شوند و اعداد کوچکتر از صفر به صفر تبدیل شوند.

0	0	0	0	0	0	0	0	0	0	0	0	0
0	25	25	25	25	25	25	25	25	25	25	25	0
0	25	15	15	15	15	15	15	15	15	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	0	0	0	5	0	0	0	15	25	0
0	25	15	15	15	15	15	15	15	15	15	25	0
0	25	25	25	25	25	25	25	25	25	25	25	0
0	0	0	0	0	0	0	0	0	0	0	0	0

$$g(a) = a^2 - 10a$$



0	0	0	0	0	0	0	0	0	0	0	0	0
0	375	375	375	375	375	375	375	375	375	375	375	0
0	375	75	75	75	75	75	75	75	75	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	0	0	0	-25	0	0	0	75	375	0
0	375	75	75	75	75	75	75	75	75	75	375	0
0	375	375	375	375	375	375	375	375	375	375	375	0
0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0
0	255	255	255	255	255	255	255	255	255	255	255	0
0	255	75	75	75	75	75	75	75	75	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	0	0	0	0	0	0	0	75	255	0
0	255	75	75	75	75	75	75	75	75	75	255	0
0	255	255	255	255	255	255	255	255	255	255	255	0
0	0	0	0	0	0	0	0	0	0	0	0	0

تابع تبدیل شدت

بنابراین در تصاویر با عمق بیت ۸، تابع تبدیل تابعی به صورت زیر است

$$g: \{0,1, \dots, 255\} \rightarrow \{0,1, \dots, 255\}$$

هر تابعی دلخواهی به صورت فوق را می توان به عنوان تابع تبدیل در نظر گرفت.

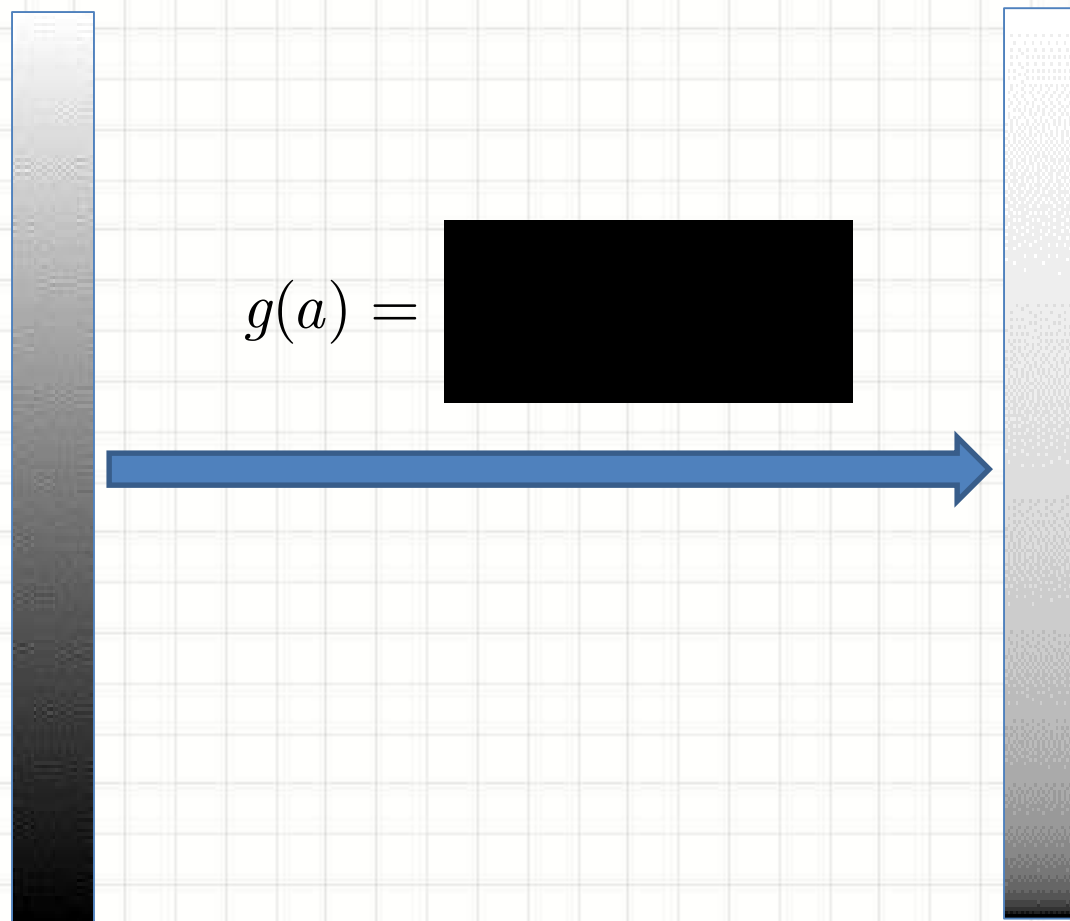
سوالی که مطرح می شود آن است که آیا می توان هر تابع دلخواه را به عنوان تابع تبدیل در نظر گرفت؟ جواب مثبت است مشروط بر آن که

- تابع روی مجموعه $\{0,1, \dots, 255\}$ قابل تعریف باشد.
- بتوان ضابطه آن را با دستکارهای جزئی مانند روند کردن و غیره طوری تغییر داد که حاصل آن به ازای اعداد 0 الی 255 باز هم در مجموعه $\{0,1, \dots, 255\}$ قرار گیرد.

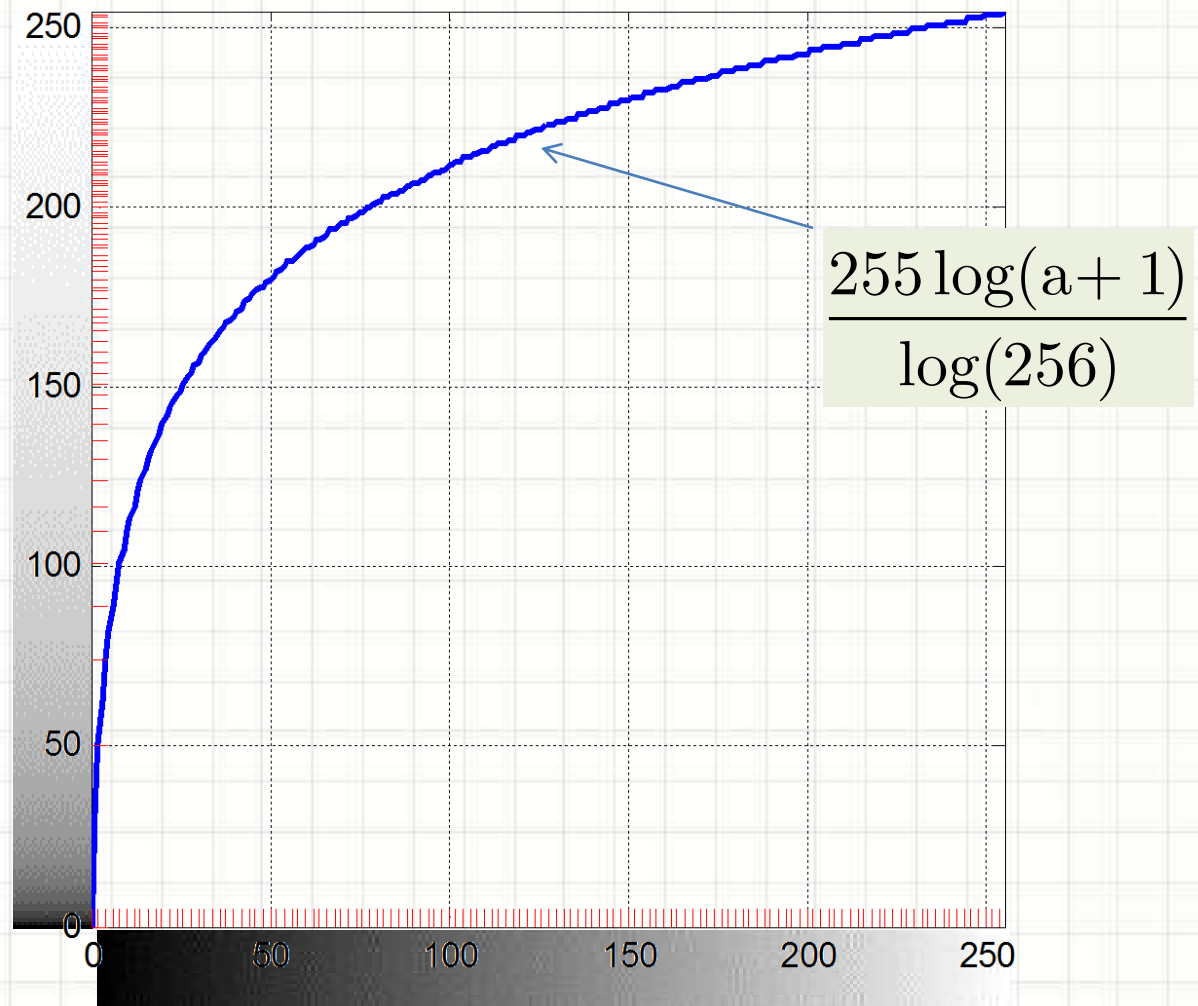
بنابراین توابع تبدیل بسیار زیادی می توان تعریف کرد. هر کدام از این توابع تبدیل تصویر را به گونه ای متفاوت تغییر می دهند. لذا مفید است که تجسمی از تغییرات حاصل از تبدیل داشته باشیم.

تجسم تابع تبدیل

یک راه ساده برای تجسم تابع تبدیل آن است که رنگ‌های خاکستری از 0 الی 255 را به صورت یک نوار رسم کرده و سپس تحت تاثیر تبدیل قرار دهیم و نتیجه را مقایسه کنیم.



تجسم تابع تبدیل

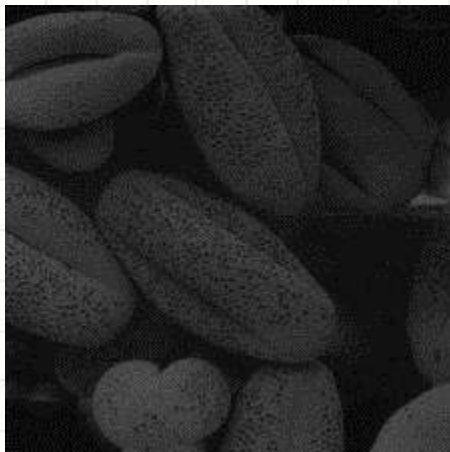
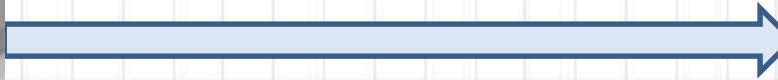


تجسم تابع تبدیل

شاید بهترین راه برای تجسم تابع تبدیل اعمال آن روی چند شکل باشد.



$$g(a) = \left\lfloor \frac{255 \log(a+1)}{\log(256)} \right\rfloor$$



$$g(a) = \left\lfloor \frac{255 \log(a+1)}{\log(256)} \right\rfloor$$



تجسيم تابع تبديل



$$g(a) = \left\lfloor \frac{255 \log(a+1)}{\log(256)} \right\rfloor$$



$$g(a) = 5a$$



دسته‌بندی تبدیلات شدت

هر تبدیل شدت دارای یک تابع تبدیل شدت می‌باشد. دسته‌بندی تبدیلهای تصویر بر مبنای نوع تابع تبدیل شدت می‌باشد. اگر تابع تبدیل شدت خطی باشد، آن‌گاه تبدیل را خطی گوئیم و ...

در زیر برخی توابع تبدیل شدت آورده شده است.

- تبدیلات خطی
- تبدیل لگاریتمی
- تبدیلات توانی یا تبدیل گاما
- تبدیلات تکه‌ای خطی

تبدیلات خطی

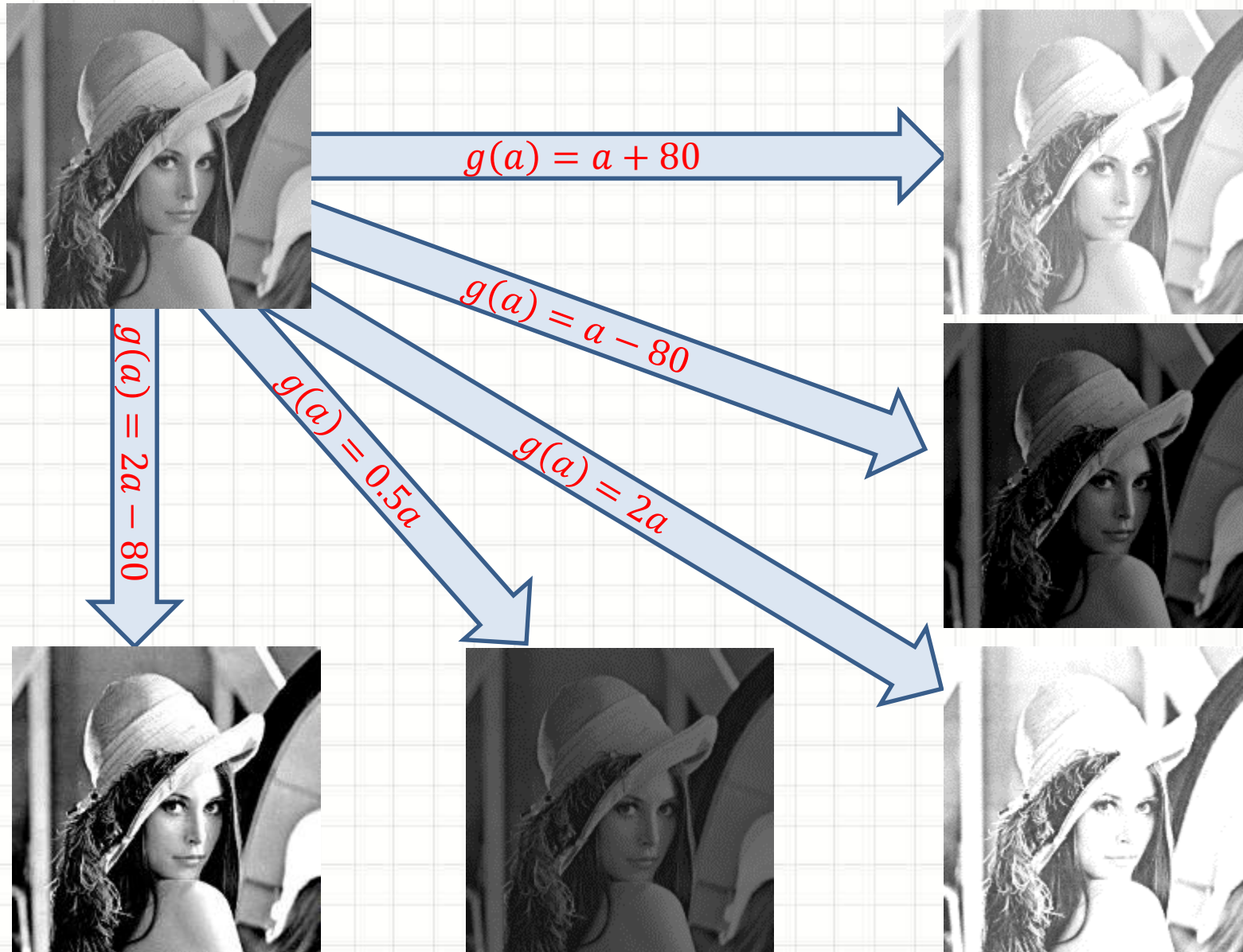
در تبدیلات خطی، تابع تبدیل شدت به صورت زیر است

$$g(a) = \alpha a + \beta$$

یعنی اگر تصویر اولیه I باشد و شدت رنگ پیکسل (x,y) برابر $I(x,y)$ باشد، آن گاه این تبدیل تصویر را به تصویر J تبدیل می کند به طوری که شدت رنگ پیکسل (x,y) در تصویر J برابر است با

$$\alpha I(x,y) + \beta$$

تبدیلات خطی - مثال



تبدیلات خطی - تبدیل قرینه (نگاتیو)



$$g(a) = -a + 255$$



تبدیل لگاریتمی

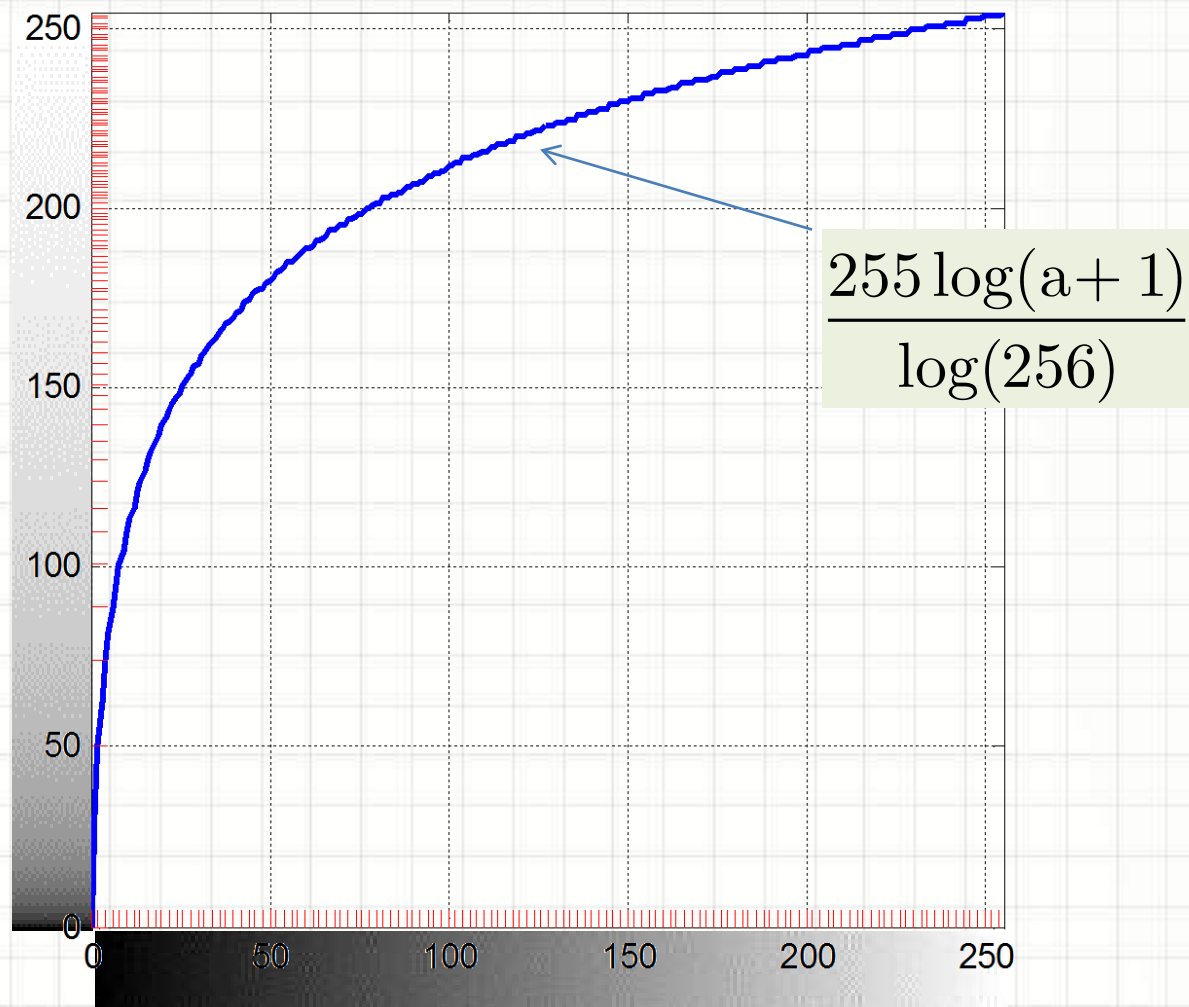
تبدیلات لگاریتمی، تابع تبدیل شدت برحسب تابع لگاریتم تعریف می‌شود. البته این تابع را باید طوری دستکاری کنیم که مقادیر صحیح و متناسب با عمق بیت تصویر برگرداند. در حالت کلی تابع تبدیل شدت لگاریتمی به صورت زیر است

$$g(a) = [c \log(a + 1)]$$

درحالتی که عمق بیت برابر ۸ است، این تابع باید بتواند اعداد صحیح ۰ الی ۲۵۵ را تولید کند. لذا در این حالت پارامتر c برابر $255/\log(256)$ انتخاب می‌شود. یعنی تابع تبدیل به صورت زیر است

$$g(a) = \left[\frac{255}{\log(256)} \log(a + 1) \right]$$

تجسم تابع تبدیل لگاریتمی



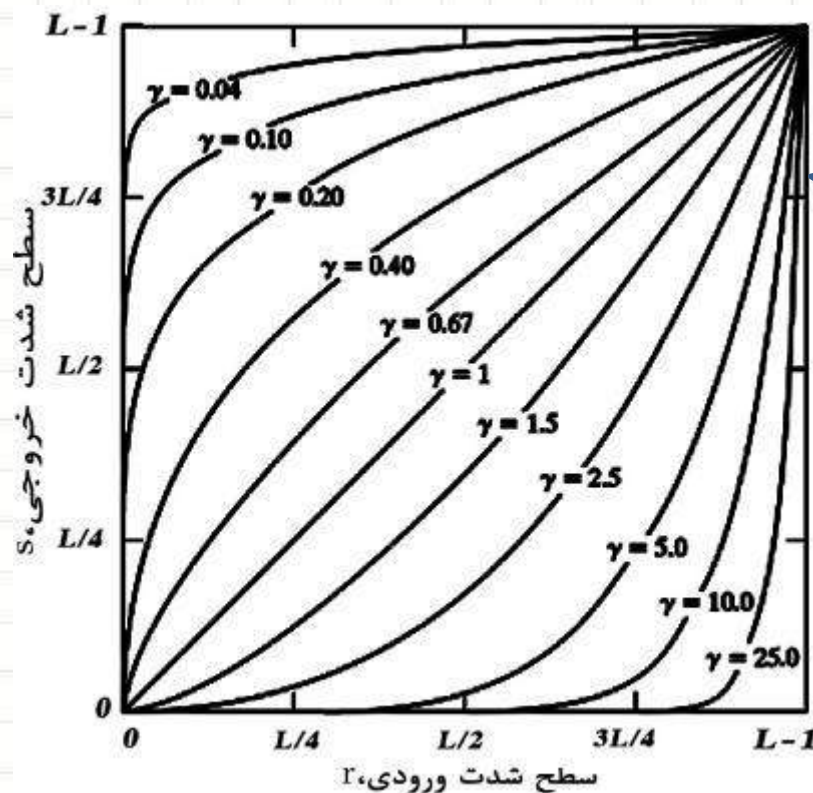
تبدیل لگاریتمی، فاصله‌ها را معنی دار تر می‌کند.

تبدیلات توانی (تبدیل گاما)

در تبدیلات توانی، تابع تبدیل شدت به صورت زیر است

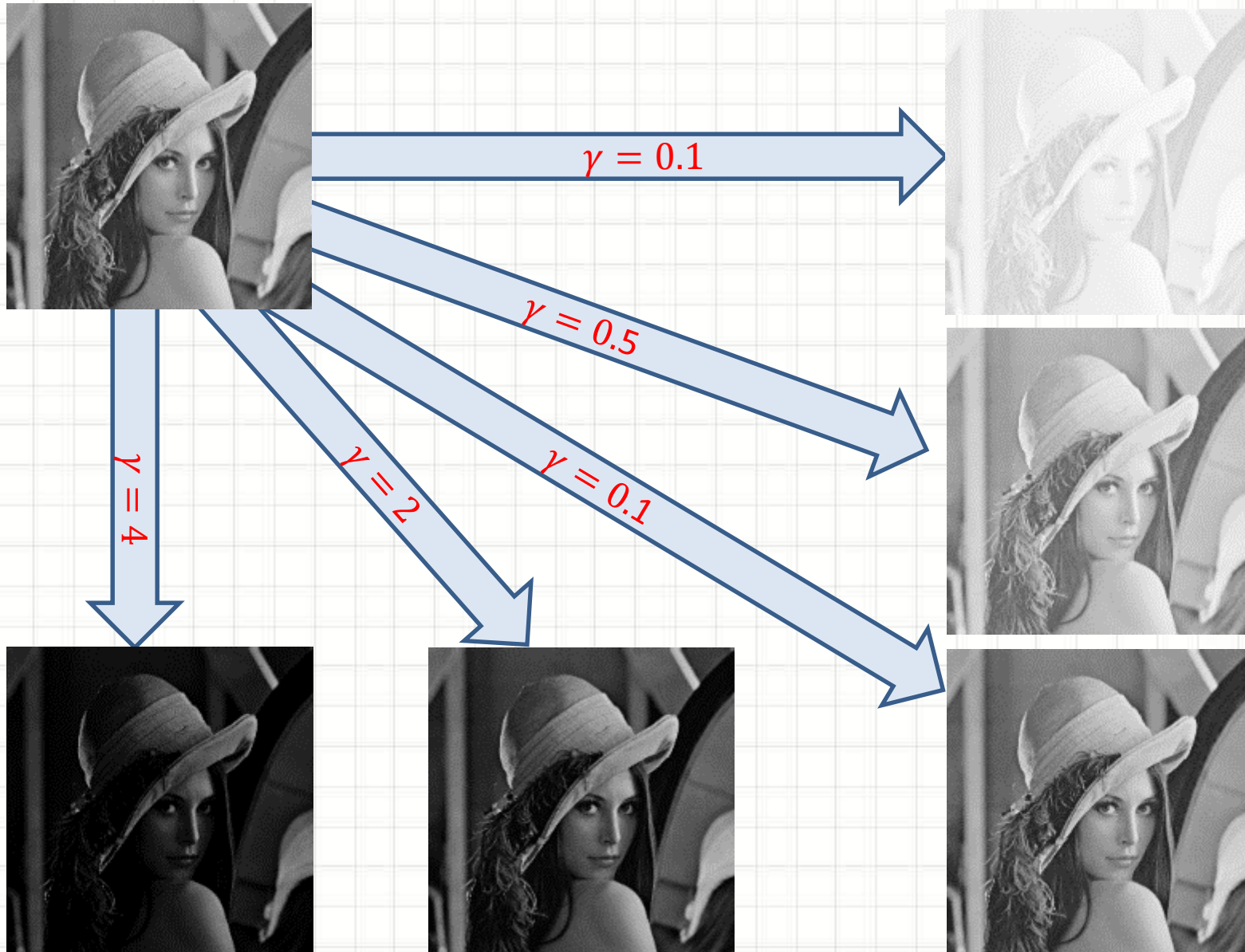
$$g(a) = [c a^\gamma]$$

به طوری که γ عددی مثبت است و پارامتر c ، متناسب با عمق بیت تصویر مشخص می‌شود.



در این شکل تابع تبدیل گاما به ازای مقادیر متفاوت γ رسم شده. توجه شود که پارامتر c به گونه‌ای انتخاب شده است که $g(L-1) = L-1$ به طوری که L دقت شدت یا همان تعداد رنگ‌های خاکستری است.

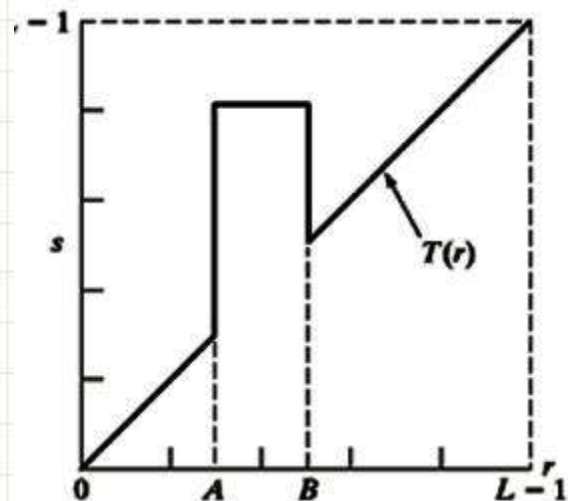
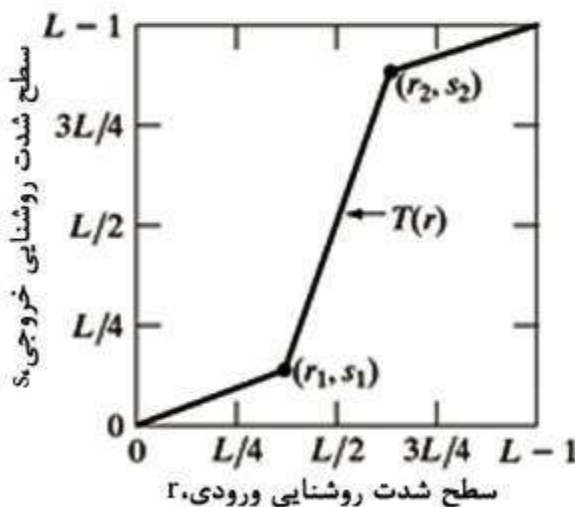
تبدیلات توانی - مثال



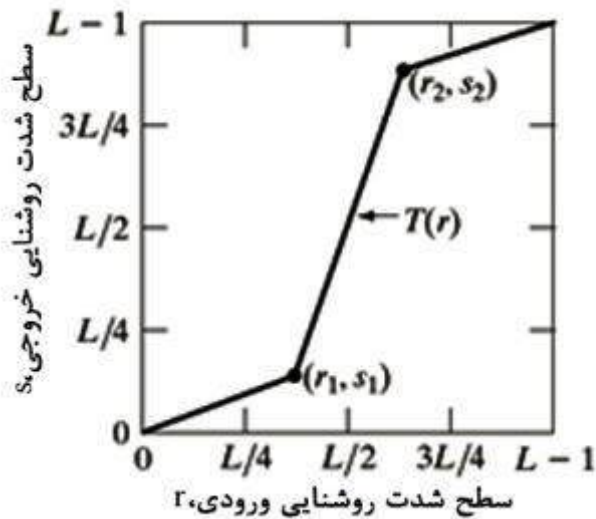
تبدیلات تکه‌ای خطی

در تبدیلات تکه‌ای خطی، تابع تبدیل شدت به صورت زیر است

$$g(a) = f(x) = \begin{cases} a_1 a + \beta_1, & 0 < a < p_1, \\ a_2 a + \beta_2, & p_1 < a < p_2, \\ \vdots & \vdots \\ a_k a + \beta_k, & p_{k-1} < a < p_k, \end{cases}$$



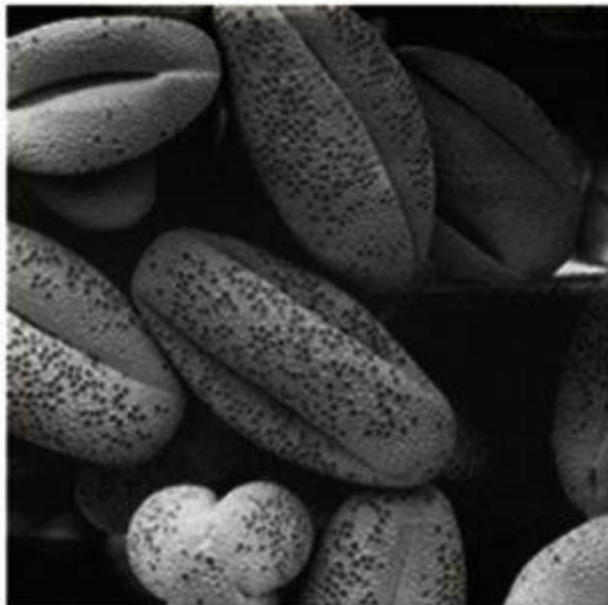
تبدیل تکه‌ای خطی-مثال



الف	ب
ج	د

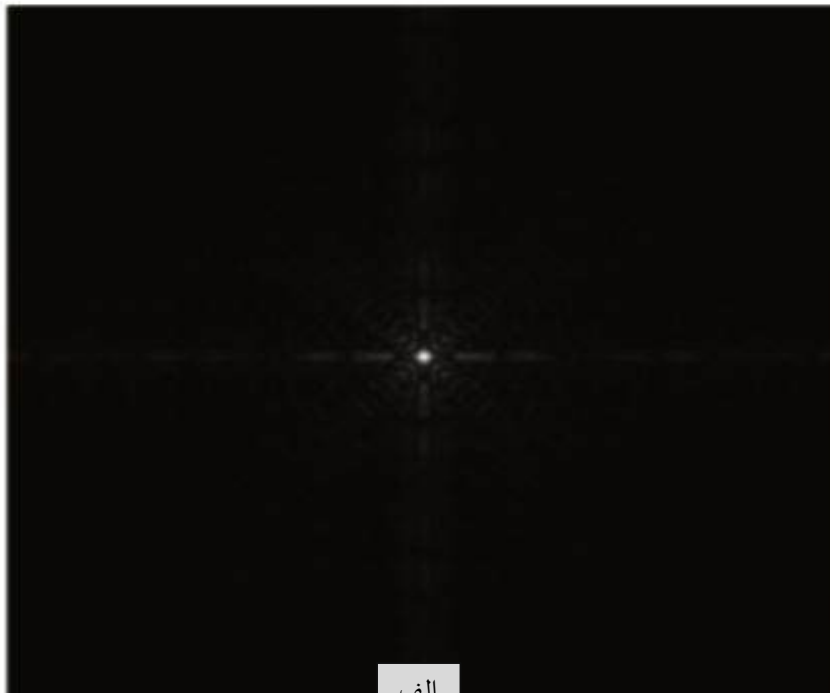
شکل 10-3

کش آوردن تباین. (الف) شکل
تابع تبدیل. (ب) یک تصویر
تباین پایین. (ج) نتیجه گسترش
تباین (د) نتیجه آستانه گذاری.

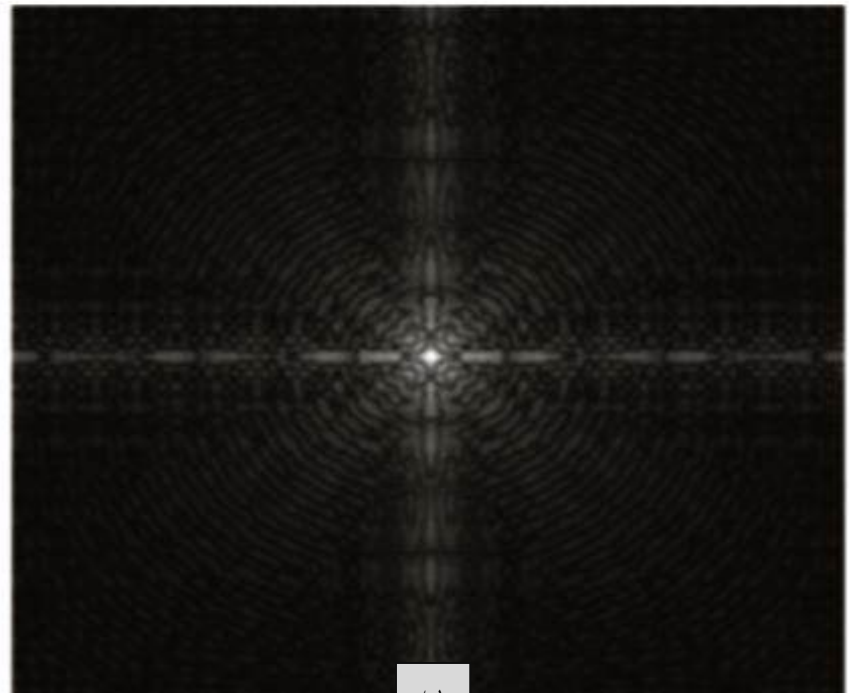


مثال‌های اضافی

تبدیل نقطه‌ای لگاریتمی



الف



ب

شکل 3-5

(الف) تصویر اصلی، (ب) نتیجه اعمال تبدیل لگاریتمی معادله $s = \log(1 + r)$

تبدیل نقطه‌ای توانی



الف	ب
ج	د

با استفاده از تصحیح گاما می‌توان
کنتراست تصویر را نیز بهبود داد

شکل 8-3

(الف) یک تصویر MRI از
نخاع شکسته شده انسان.
(ب) تا (د) نتایج اعمال
تبدیل در معادله (3-2-3) با
به ترتیب $c=1$ و $y=0.6$,
0.4 و 0.3

تبدیل توانی-مثال

الف	ب
ج	د

شکل 3-9

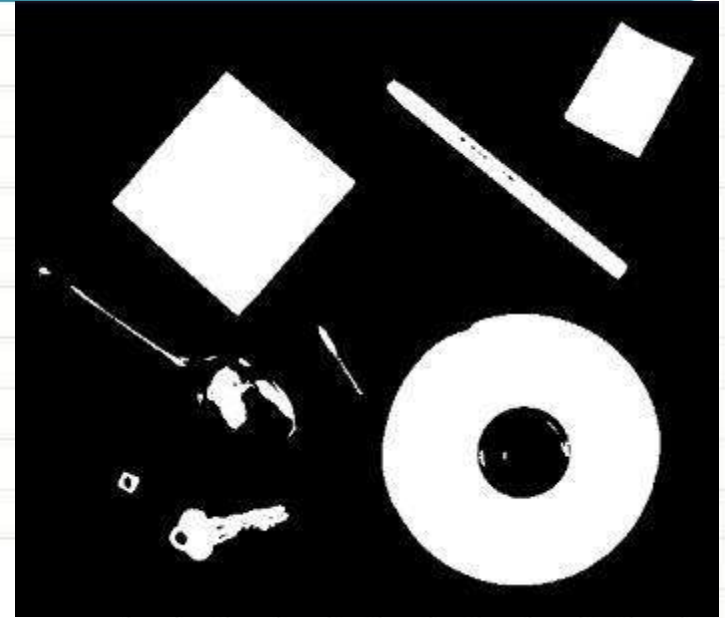
(الف) تصویر هوایی.

(ب) تا (د) نتایج اعمال تبدیل در

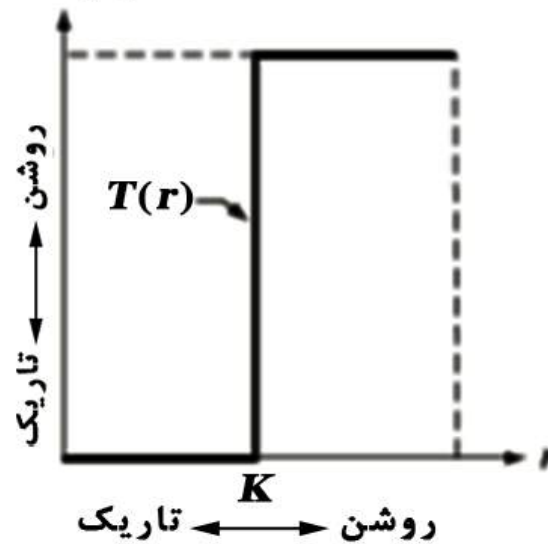
معادله (3-2-3) با $c=1$ و به ترتیب

$y=0.3, 0.4, 0.5$ و 0.5

مثال

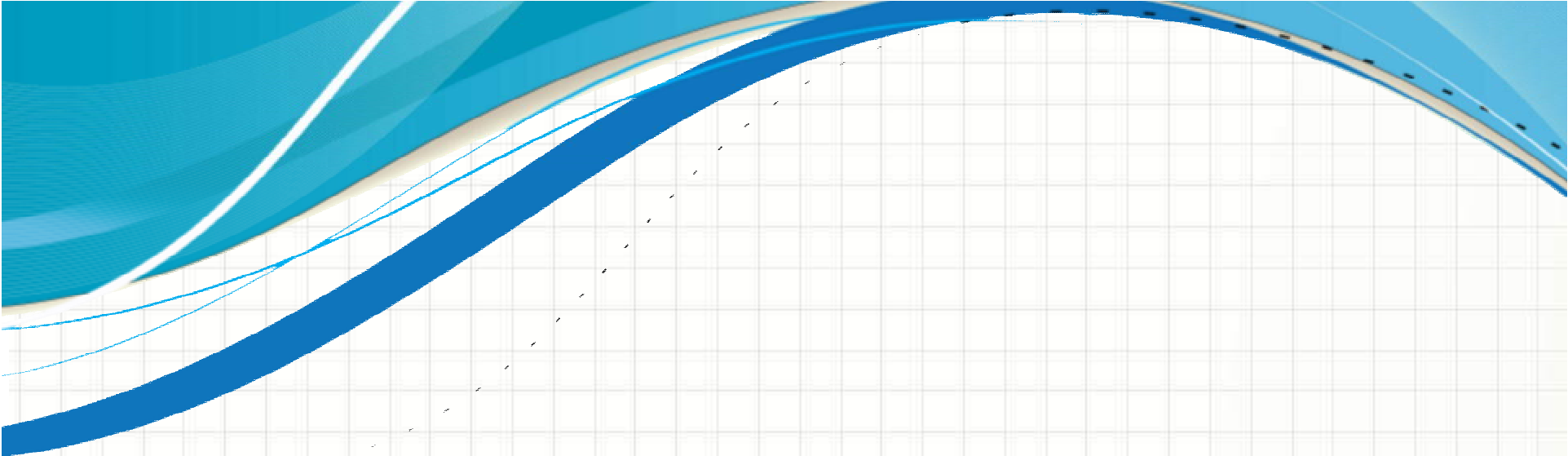


$$S = T(r)$$



سوال

برای بهبود کنتراست یک تصویر دلخواه، چه تبدیلی مناسب است؟



بهبود شدت در تصاویر دیجیتال به وسیله هیستوگرام

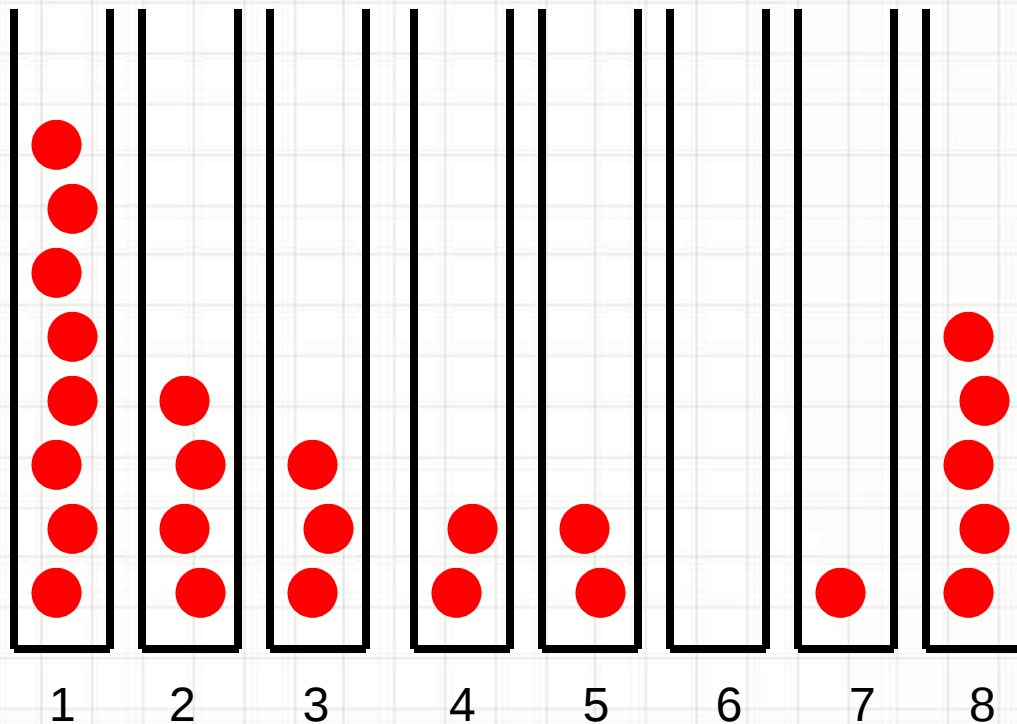
Intensity Enhancement by Histogram

هیسٹوگرام تصویر

هیستوگرام

هیستوگرام نمایشی تصویری از توزیع عبارت است از اطلاعات مناسبی از یک تصویر در اختیار می‌گذارد که به وسیله آن می‌توان ویژگی‌های تصویر را مشخص کرد.

1	8	4	3	4
1	1	1	7	8
8	8	3	3	1
2	2	1	5	2
1	1	8	5	2



هیستوگرام

مثال

$$h(r_1) = 8$$

$$h(r_2) = 4$$

$$h(r_3) = 3$$

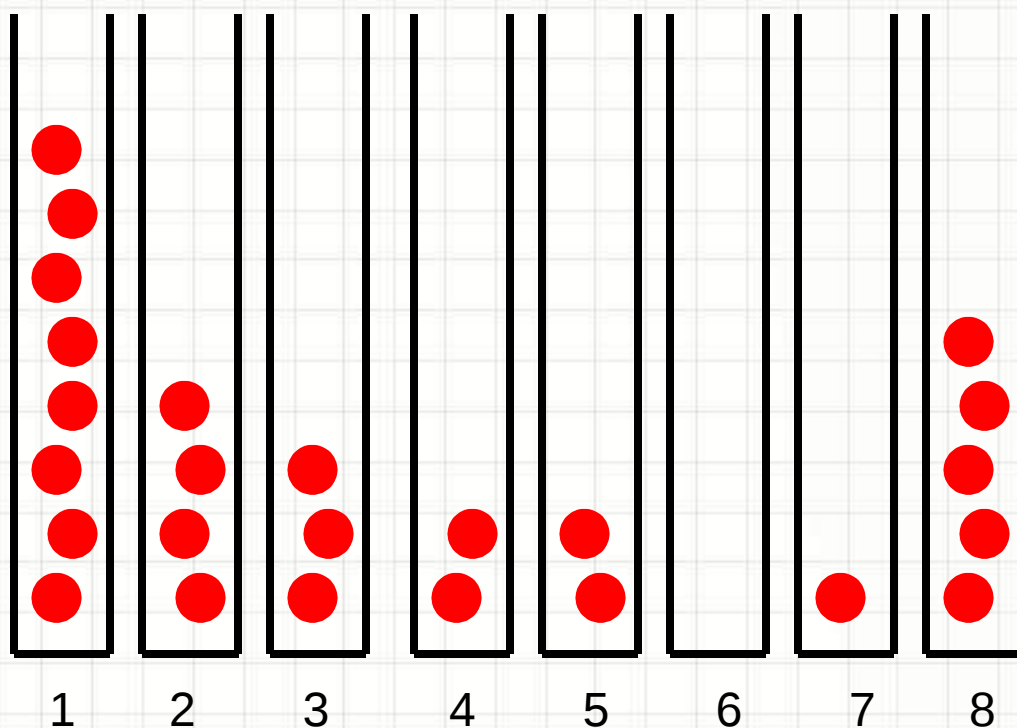
$$h(r_4) = 3$$

$$h(r_5) = 2$$

$$h(r_6) = 0$$

$$h(r_7) = 1$$

$$h(r_8) = 5$$



هیستوگرام نرمال شده

برای نرمال سازی مقادیر هیستوگرام به مجموع کل مقادیر تقسیم می شود.

$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}$$

$$h(r_1) = 8$$

$$p(r_1) = 8 / 25 = 0.32$$

$$h(r_2) = 4$$

$$p(r_2) = 4 / 25 = 0.16$$

$$h(r_3) = 3$$

$$p(r_3) = 3 / 25 = 0.12$$

$$h(r_4) = 3$$

$$p(r_4) = 3 / 25 = 0.08$$

$$h(r_5) = 2$$

$$p(r_5) = 2 / 25 = 0.08$$

$$h(r_6) = 0$$

$$p(r_6) = 0 / 25 = 0.00$$

$$h(r_7) = 1$$

$$p(r_7) = 1 / 25 = 0.04$$

$$h(r_8) = 5$$

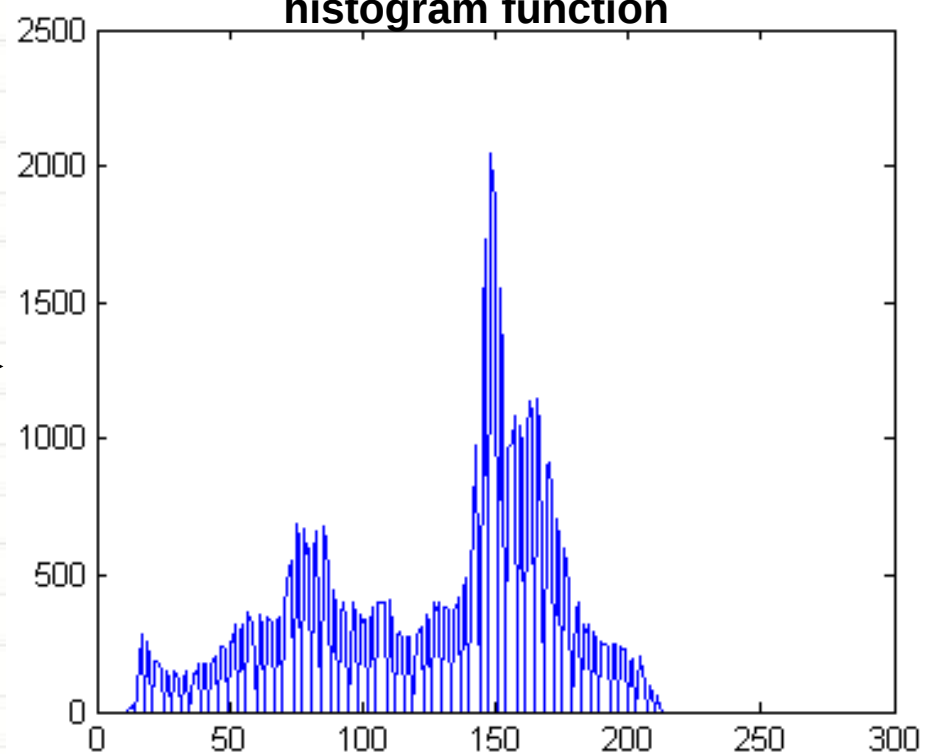
$$p(r_8) = 5 / 25 = 0.20$$

هیسٹوگرام یک تصویر

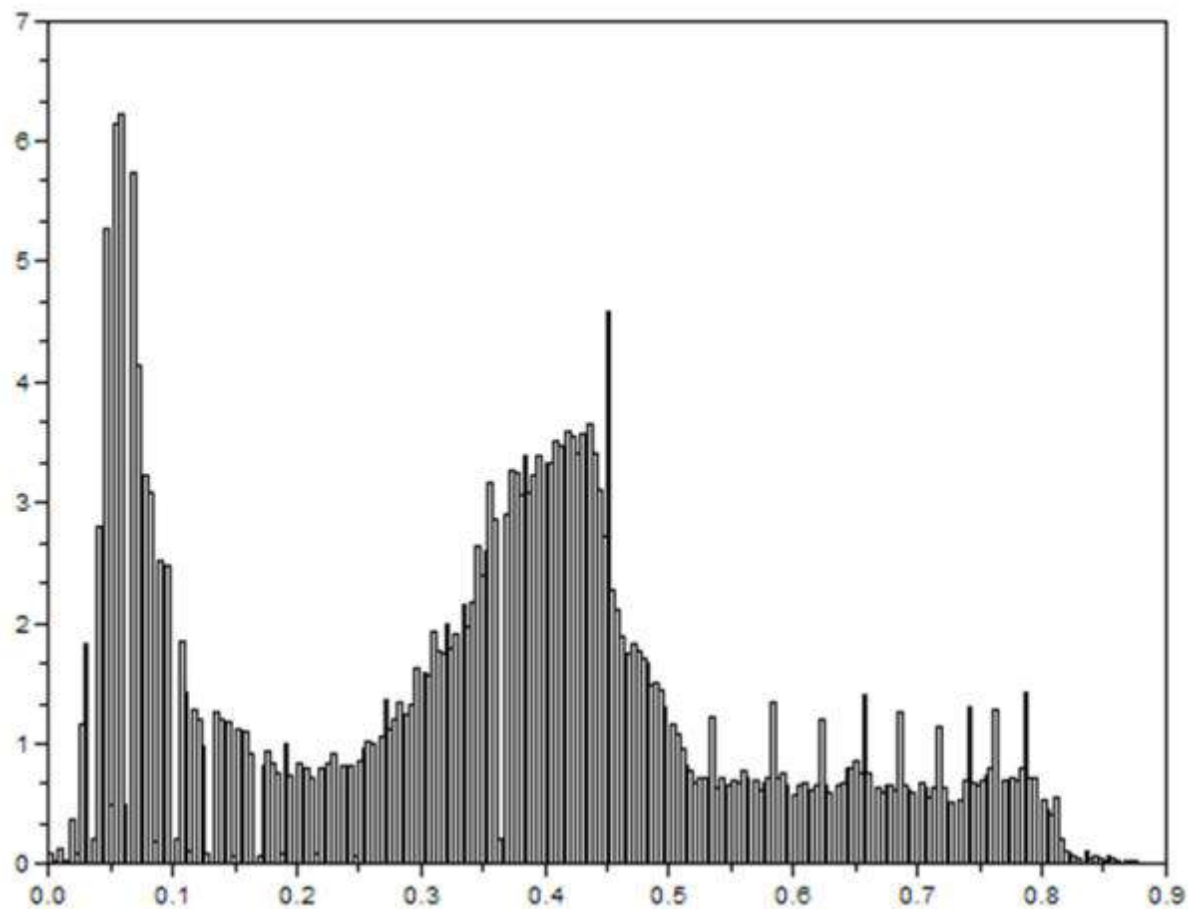
Original image



Graph of the histogram function



هستوگرام یک تصویر



هیستوگرام یک تصویر

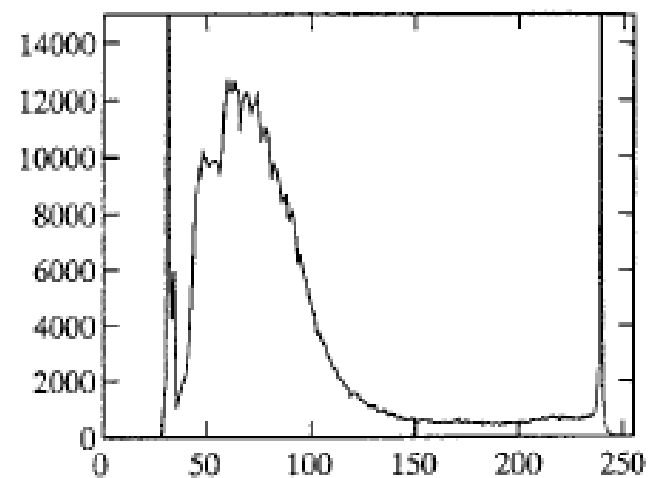
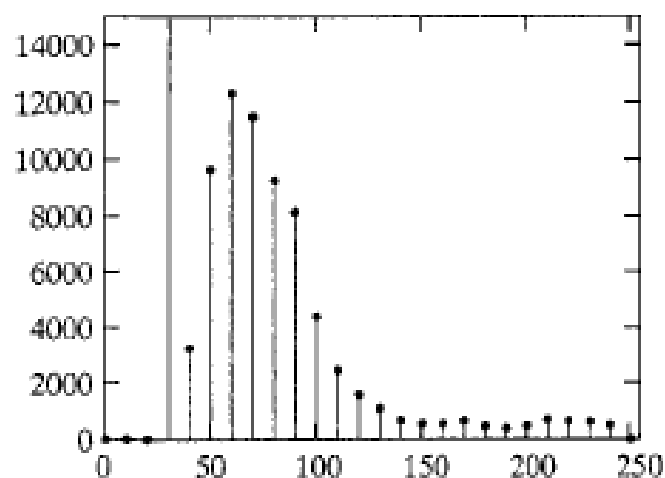
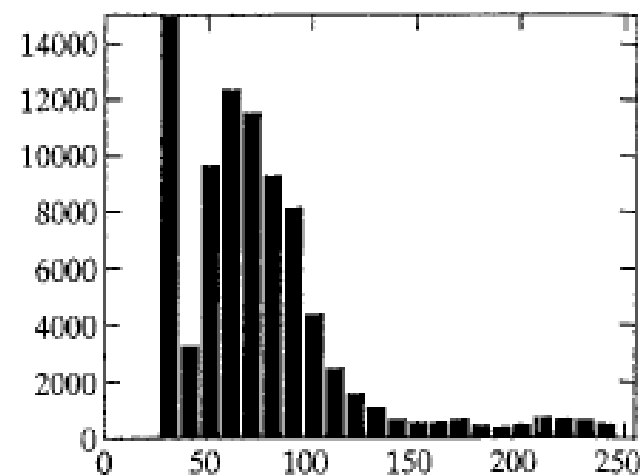
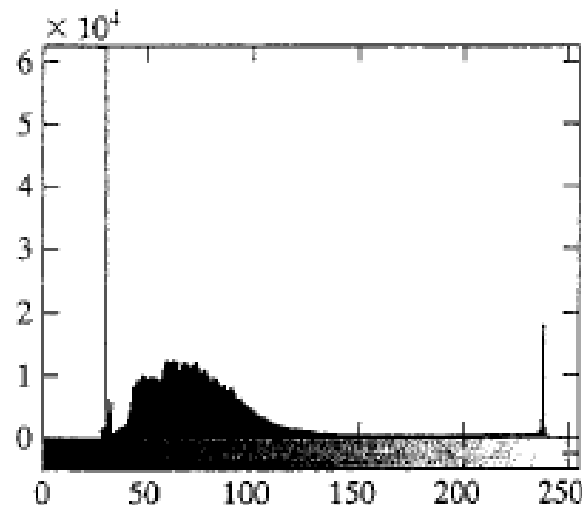
هیستوگرام به روش‌های متفاوتی ممکن است نمایش داده شود

a b
c d

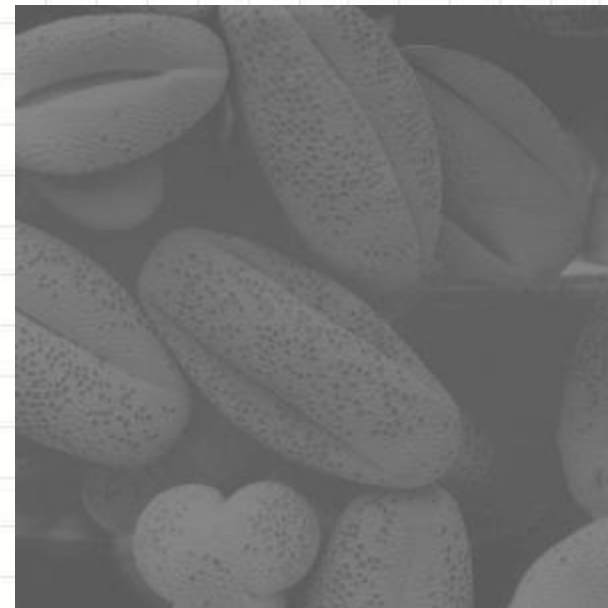
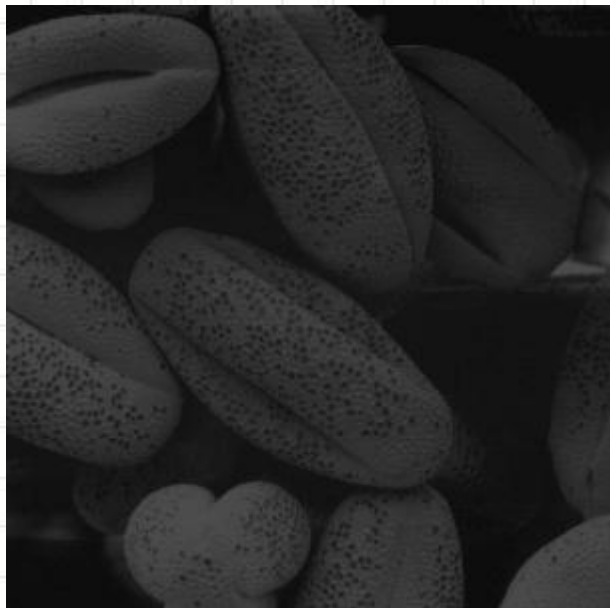
FIGURE 3.7

Various ways to plot an image histogram.

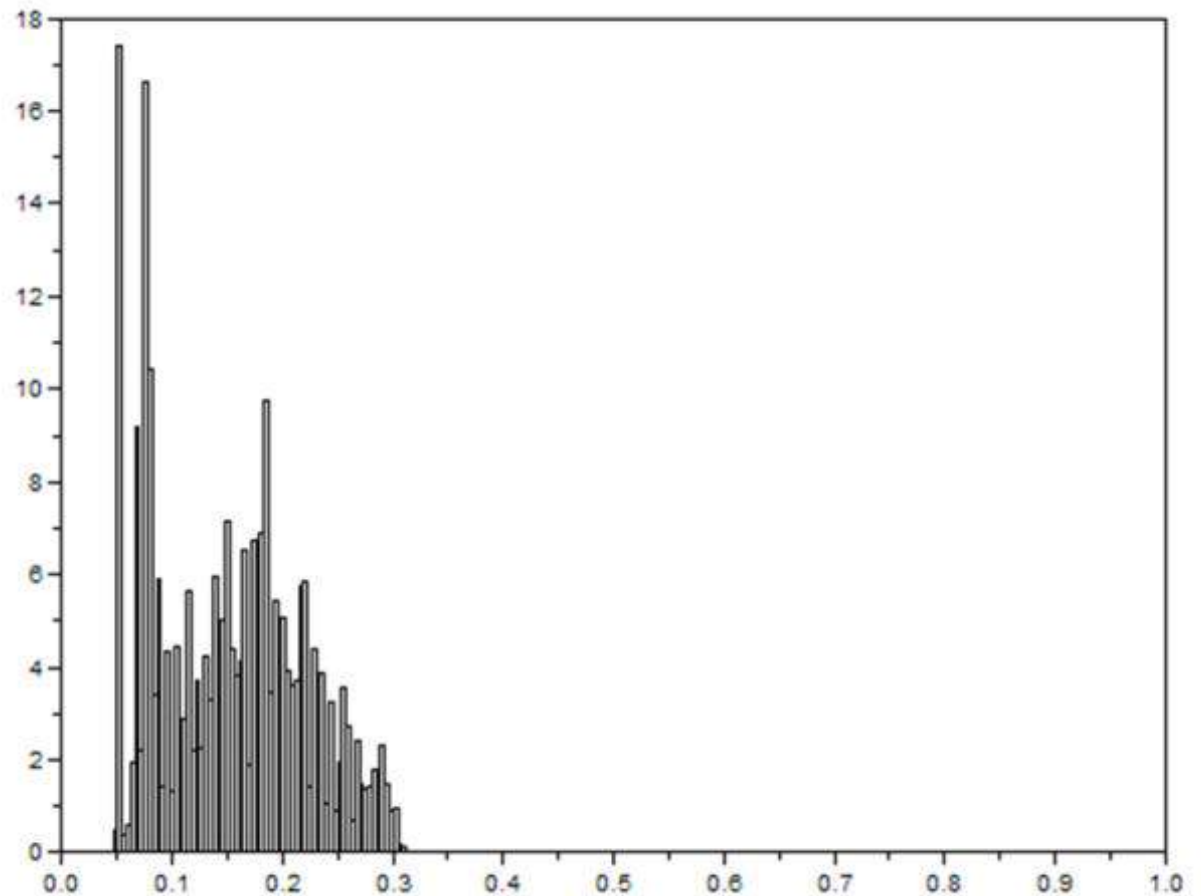
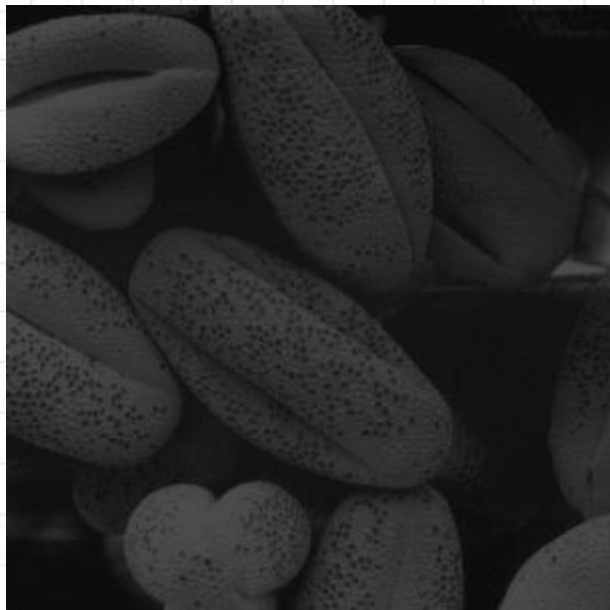
(a) imhist,
(b) bar,
(c) stem,
(d) plot.



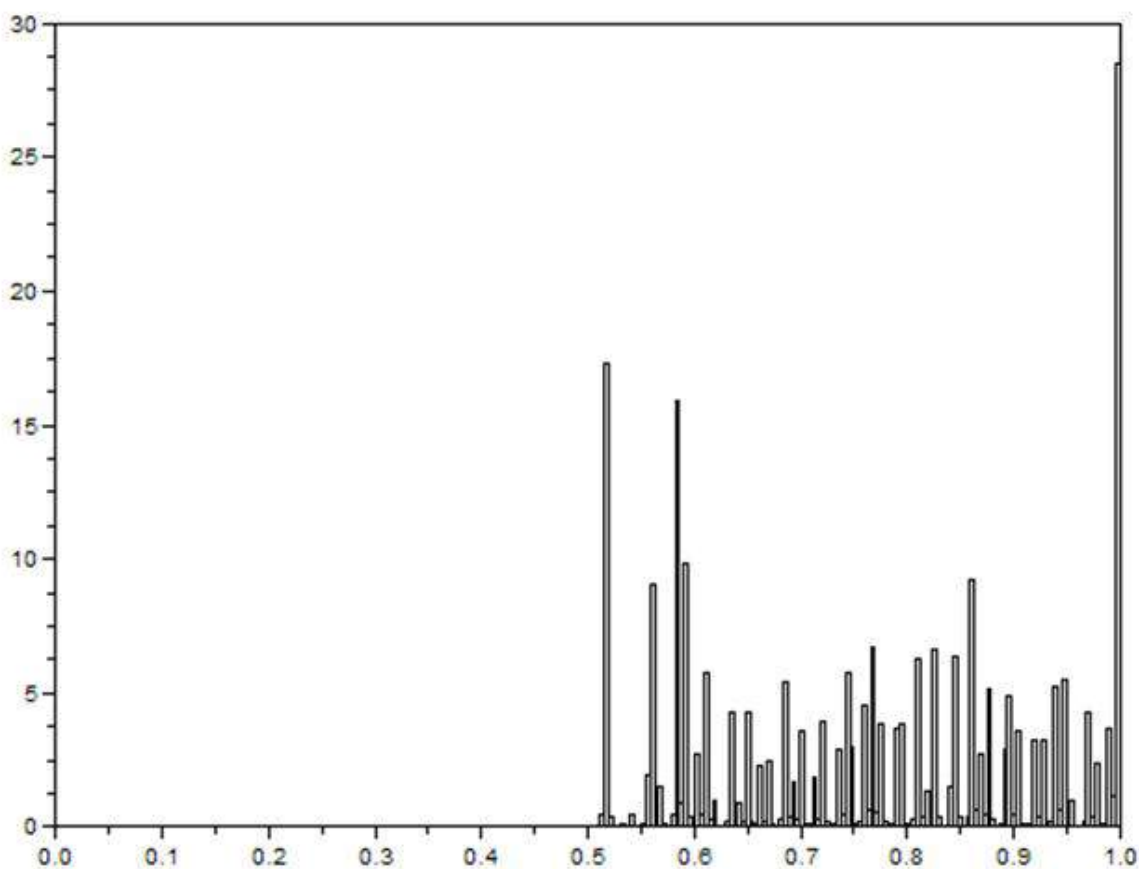
هستوگرام یک تصویر



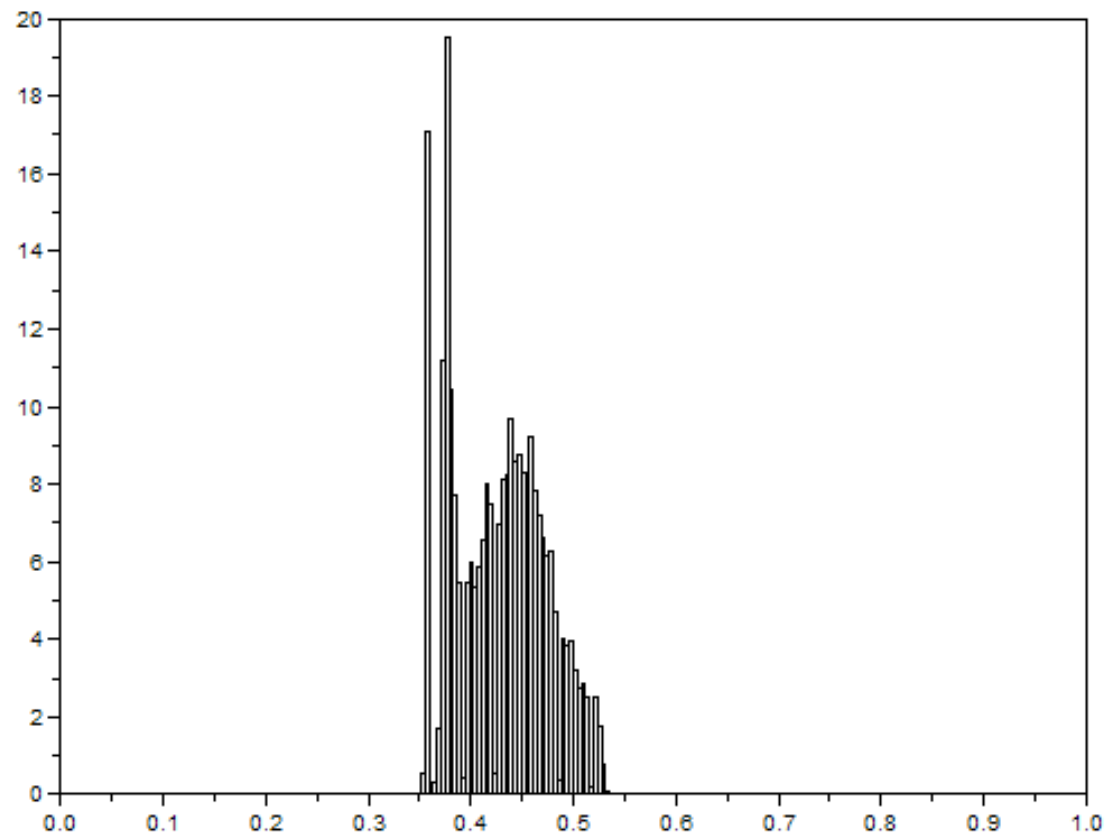
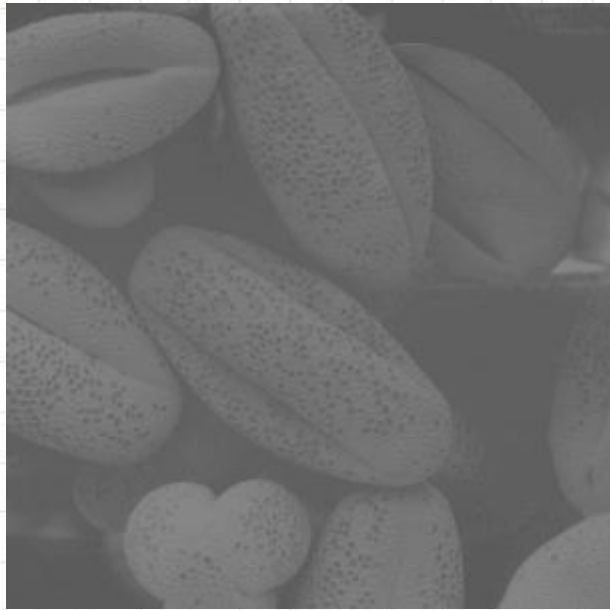
هستوگرام یک تصویر



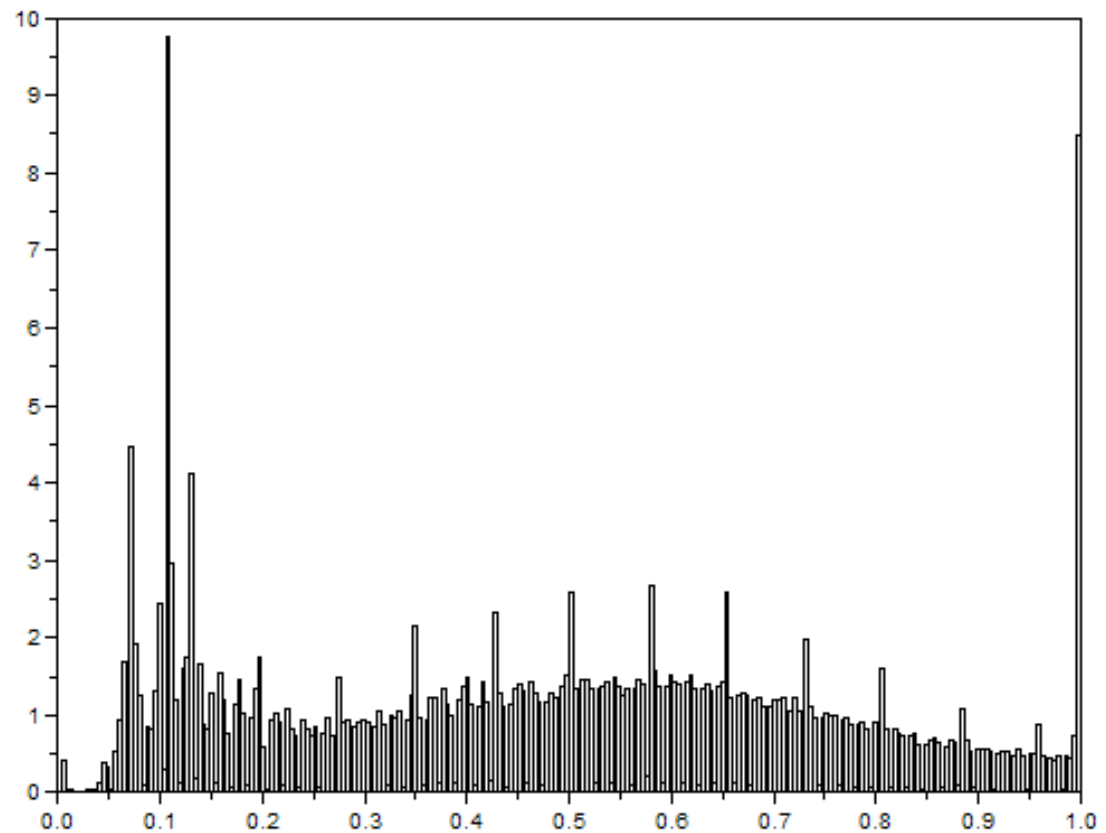
هستوگرام یک تصویر



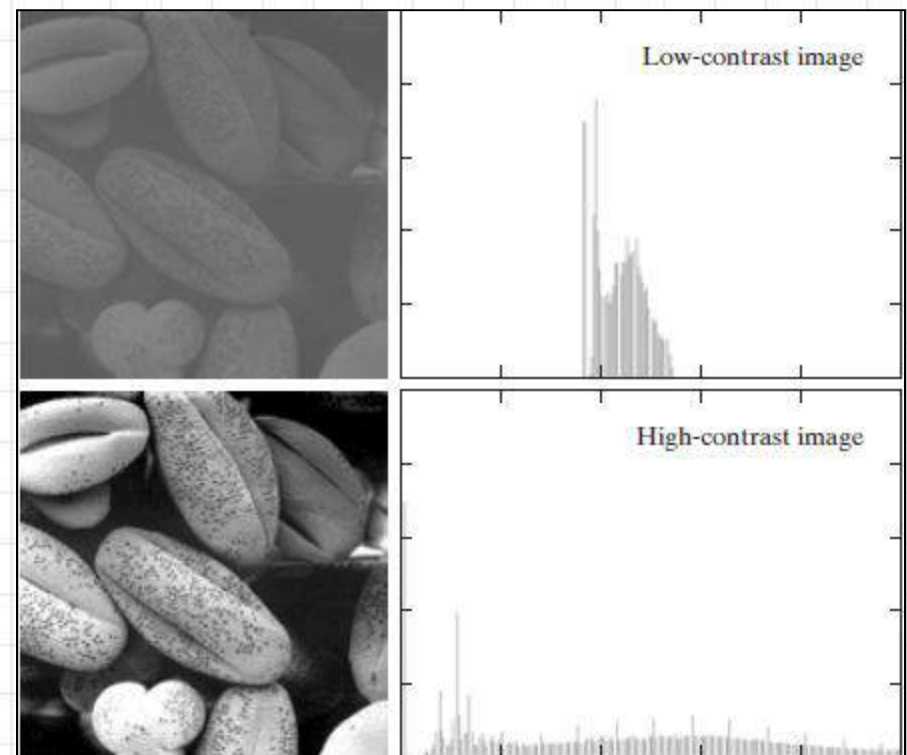
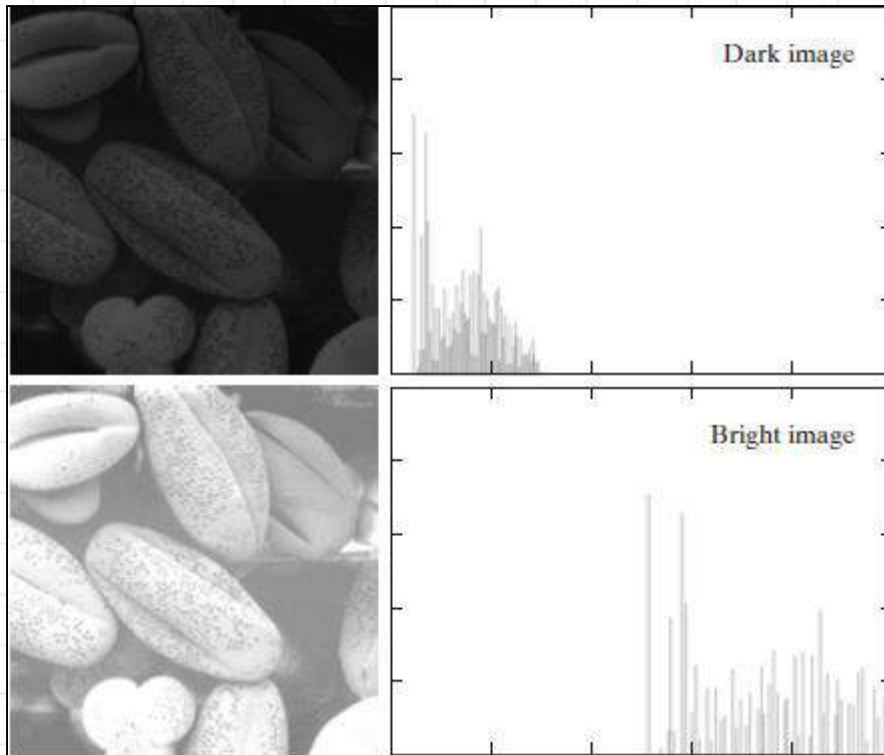
هستوگرام یک تصویر



هستوگرام یک تصویر



هستوگرام یک تصویر



هیستوگرام یک تصویر

بهبود کنتراست تصویر با استفاده از هیستوگرام به این معنی است که با اطلاعات موجود در هیستوگرام تصویر به تغییر شدت سطح خاکستری پیکسل‌ها پردازیم.

یک راه ساده برای بهبود کنتراست یک تصویر آن است که شدت سطوح خاکستری طوری تغییر کند که هیستوگرام تصویر اولیه کشیده شود و کل محدوده شدت رنگ را بپوشاند.

مقدمه

هیستوگرام اطلاعات مناسبی از یک تصویر در اختیار می‌گذارد که به وسیله آن می‌توان ویژگی‌های تصویر را مشخص کرد.

همان‌گونه که گفته شد، هر تبدیل شدت با یک تابع متناظر است. حال یافتن یک تابع مناسب از اهداف پردازش تصویر می‌باشد. در اکثر تصاویر می‌توان این تابع را با کمک هیستوگرام ساخت. به عبارت دیگر با توجه به هیستوگرام یک تابع تبدیل شدت ساخته می‌شود و سپس با استفاده از این تابع به بهبود کنتراست تصویر پرداخته می‌شود.

ساخت تابع تبدیل شدت با استفاده از هیستوگرام با دو رویکرد زیر انجام می‌شود

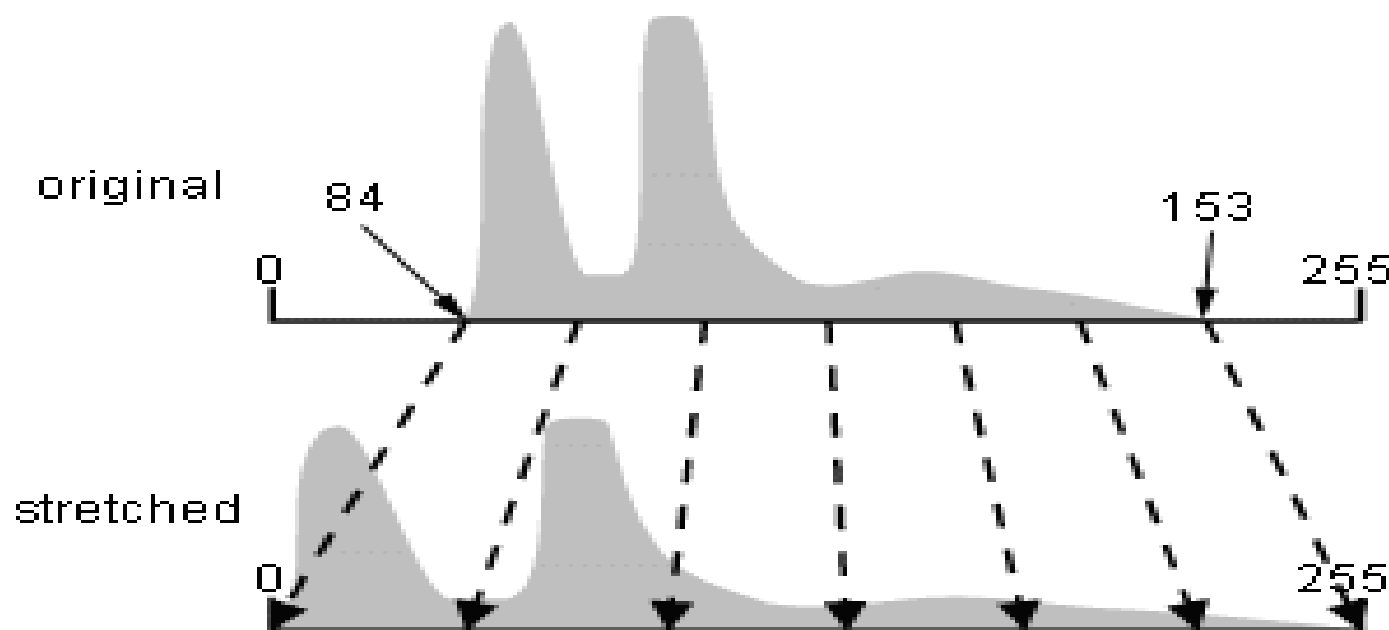
- 1- کشش هیستوگرام (Histogram Stretching)
- 2- مسطح‌سازی هیستوگرام (Histogram Equalization)

کشش هیستوگرام

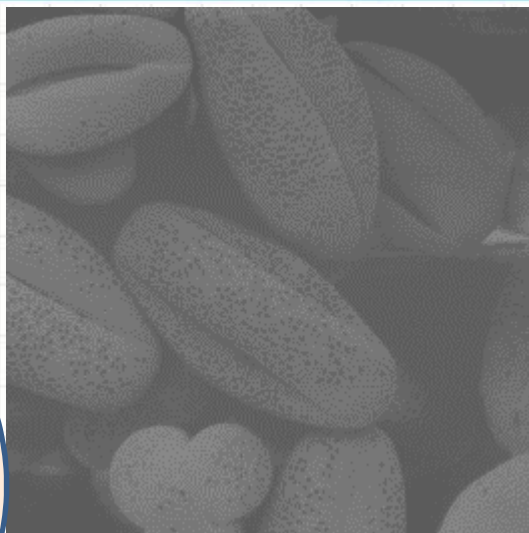
Histogram Stretching

مقدمه

ایده: در روش کشش هیستوگرام، کنتراست به صورت فاصله بین کمترین و بیشترین شدت در نظر گرفته می‌شود. لذا برای افزایش کنتراست باید این فاصله ماکزیمم گردد. به عبارت دقیق‌تر: فرض کنید که شدت رنگ پیکسل‌های تصویر I در بازه $[a, b]$ واقع شده باشند و عمق تصویر 8 باشد. در روش کشش هیستوگرام، شدت پیکسل‌ها به گونه‌ای تغییر می‌کند که شدت رنگ پیکسل‌ها در بازه $[0, 255]$ قرار گیرد.

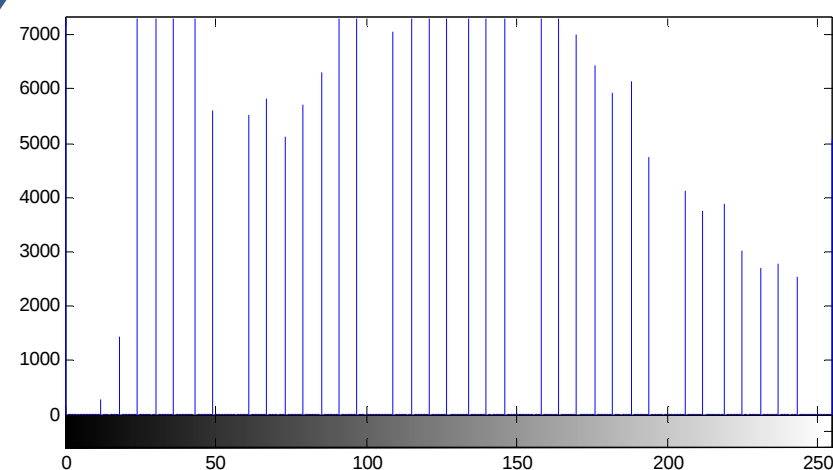
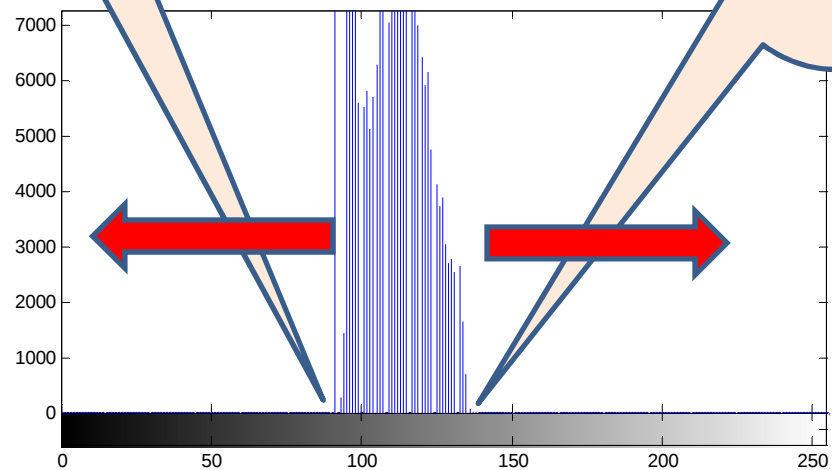


گشش هیستوگرام-نمونه



کمترین
شدت
91

بیشترین
شدت
138



کشش هیستوگرام از دیدگاه ریاضی

برای مطالعه دقیق کشش هیستوگرام لازم است که تابع تبدیل شدت متناظر با آن را استخراج کرد.

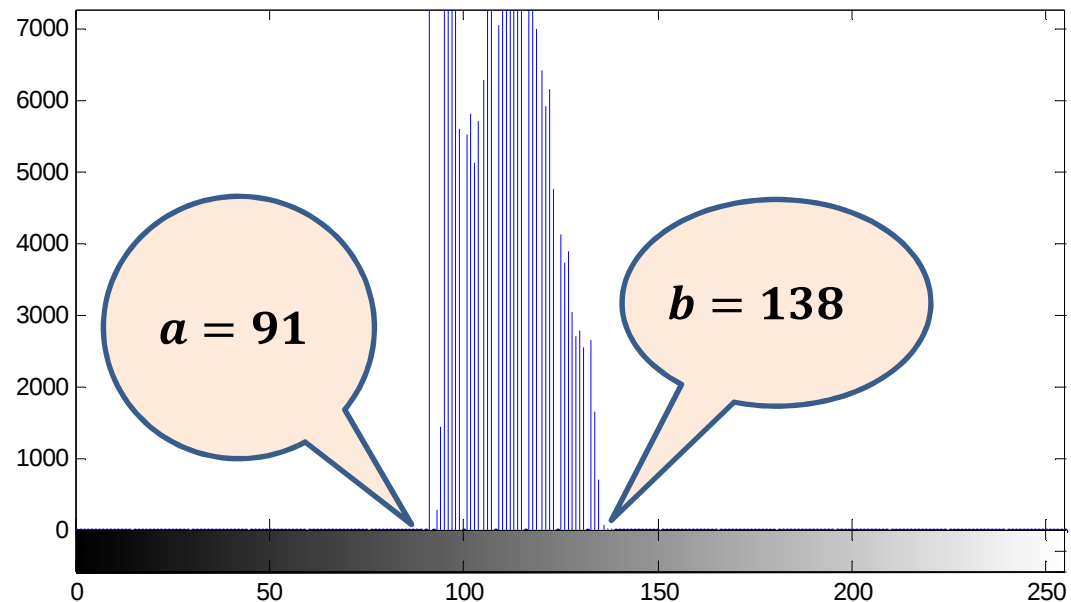
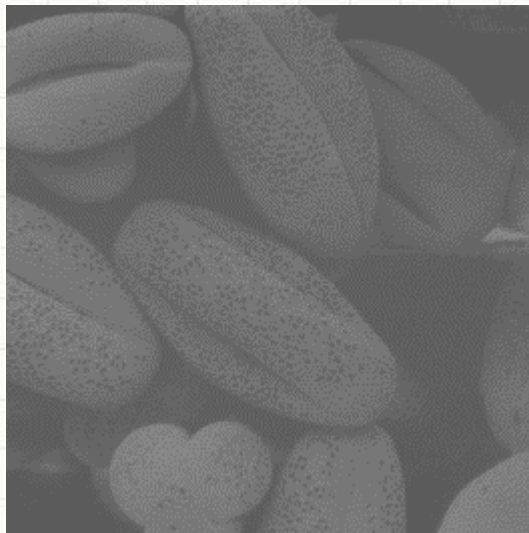
اگر شدت پیکسل‌های تصویر مبدا در بازه $[a,b]$ باشند، آن گاه باید این شدت‌ها به بازه $[0,255]$ تبدیل شوند. یک تابع خطی ساده زیر این کار را انجام می‌دهد

$$g(i) = 255 \frac{i - a}{b - a}$$

این تابع را می‌توان به عنوان تابع تبدیل شدت در نظر گرفت. به عبارت دیگر اگر $I(x,y)$ شدت رنگ پیکسل (x,y) در تصویر I باشد، آن گاه شدت این پیکسل به صورت زیر تغییر می‌کند

$$J(x,y) = 255 \frac{I(x,y) - a}{b - a}$$

کشش هیستوگرام – مثال

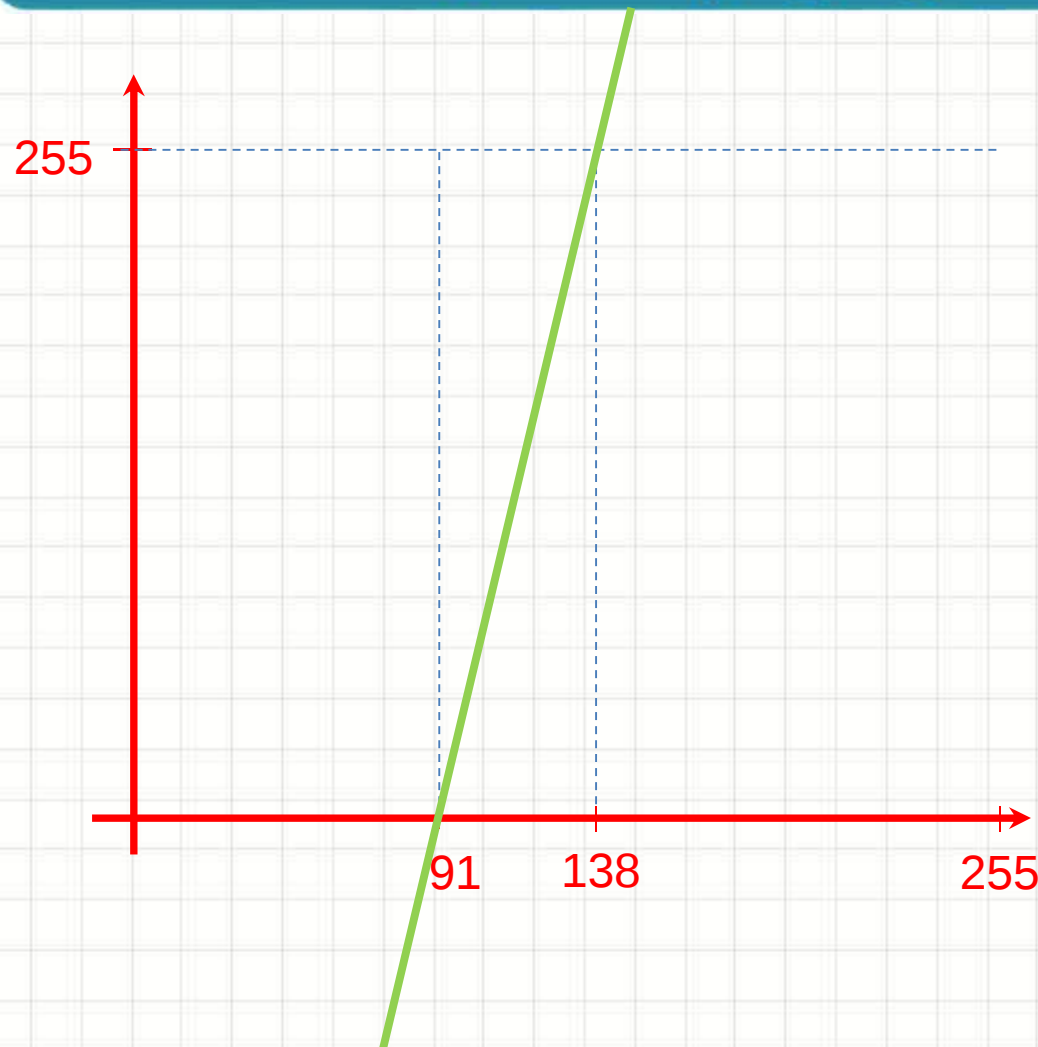


تابع تبدیل شدت در این مثال به صورت زیر در نظر گرفته می شود

$$g(i) = 255 \frac{i - 91}{47}$$

توجه شود که تابع تبدیل فوق شدت 91 را به 0 و شدت 138 را به شدت 255 تبدیل می کند. و شدت های بیت 91 الی 138 را به صورت خطی تغییر می دهد

کشش هیستوگرام-مثال



تابع تبدیل شدت

$$g(i) = 255 \frac{i-91}{47}$$

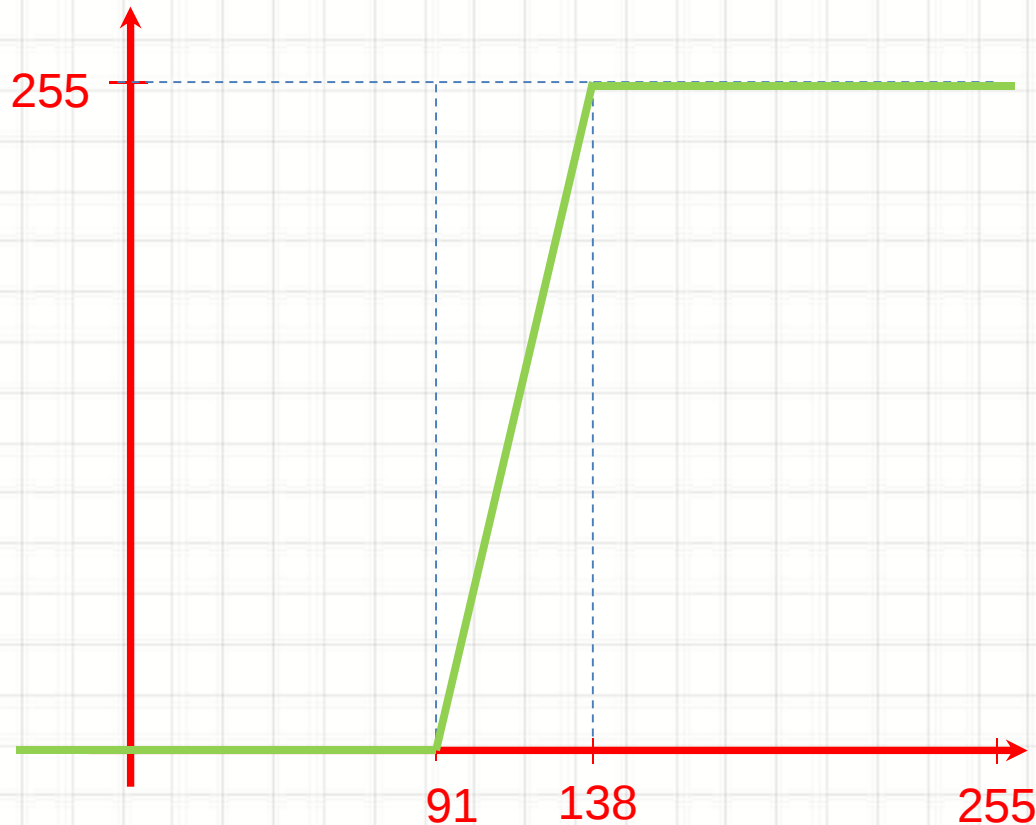
در زیر رسم شده است. این تابع

خطی است که از نقاط

$(91,0), (135,255)$

می‌گذرد

کشش هیستوگرام-مثال



برای اینکه تابع g یک تابع تبدیل شدت کامل باشد، باید حاصل آن اعداد صحیح در محدوده 0 الی 255 باشد. بنابراین لازم است که تابع g به صورت زیر تغییر داده شود.

$$g(i) = \begin{cases} 0, & i < 91 \\ \text{round}\left(255 \frac{i-91}{47}\right), & 91 \leq i \leq 138 \\ 255, & i > 138 \end{cases}$$

همان طور که ملاحظه می شود، تابع تبدیل فوق یک تابع تکه ای خطی می باشد.

کشش هیستوگرام-یک نقص

اگر در یک تصویر کمترین شدت برابر $a = 0$ و بیشترین شدت برابر $b = 255$ باشد، آن گاه کشش هیستوگرام منجر به هیچ تغییری در تصویر نمی‌شود. چرا که در این حالت تابع تبدیل g به صورت زیر درمی‌آید

$$g(i) = 255 \frac{i - a}{b - a} = 255 \frac{i - 0}{255 - 0} = i$$

که همان تابع همانی است و هیچ تغییری روی تصویر اعمال نمی‌کند.

کشش هیستوگرام برای تصاویری با $a = 0, b = 255$ انجام نمی‌شود و باید از تکنیک‌های دیگر برای بهبود کنتراست استفاده شود. در واقع این تکنیک برای تصاویری مناسب است که هیستوگرام آن‌ها در یک محدوده متمرکز شده است و روی کل بازه پخش نشده است.

کشش هیستوگرام – یک نقص

حال فرض کنید که تمامی پیکسل‌های آن دارای شدتی بین 100 و 120 می‌باشد و فقط یک پیکسل آن دارای شدت 0 و یک پیکسل دیگر دارای شدت 255 است. به خاطر این دو پیکسل داریم

$$a = 0, \quad b = 255$$

همان‌گونه که ذکر شد اعمال روش کشش هیستوگرام روی این تصویر منجر به هیچ تغییر و بهبودی نمی‌شود. حال آن که در صورت عدم وجود این دو پیکسل، کشش هیستوگرام منجر به بهبود قابل توجه‌ای در کنتراست تصویر می‌شود.

اما نادیده گرفتن این دو پیکسل به کلیت تصویر لطمه‌ای نمی‌زند. در واقع اگر این دو پیکسل را در روش کشش هیستوگرام حذف کنیم، آن‌گاه نتیجه خوبی خواهیم گرفت و ندیده گرفته شدن دو پیکسل تاثیر بسیار ناچیزی روی ظاهر تصویر می‌گذارد.

از سوی دیگر به تجربه ثابت شده است که در نظر نگرفتن و یا تغییر دادن یک درصد از پیکسل‌های تصویر منجر به تغییر محسوسی در تصویر نمی‌شود.

بهبود روش کشش هیستوگرام

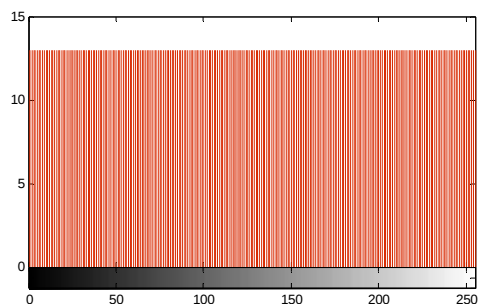
با توجه به نقص مطرح شده در اسلاید قبل و توضیحاتی در خصوص نادیده گرفتن یک درصد پیکسل‌ها روش کشش هیستوگرام قابل بهبود می‌باشد.

سطح سازی هیستوگرام Histogram Equalization

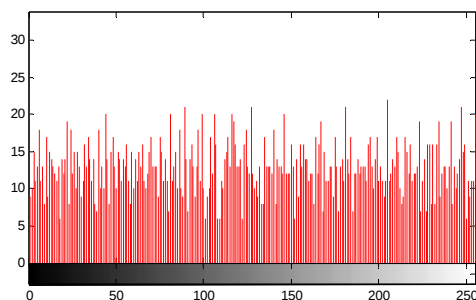
ایده

در دیدگاه مسطح‌سازی هیستوگرام، تصویر مطلوب عبارت است از: تصویری که هیستوگرام آن مسطح باشد و در تمام ناحیه $[0, 255]$ پخش شده باشد. به عبارت دیگر تصویری که میله‌های هیستوگرام آن هم‌اندازه باشند و به طور یکنواخت در ناحیه $[0, 255]$ پخش شده باشد، یک تصویر ایده‌آل و مناسب است.

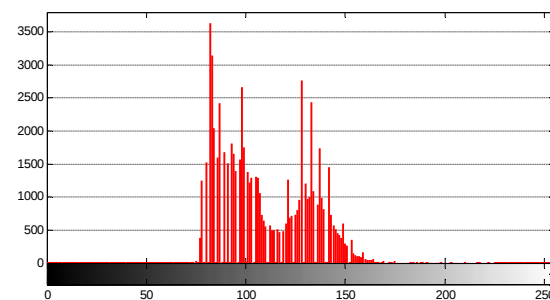
در این دیدگاه، هرچه هیستوگرام صاف‌تر (Flat) باشد، آن‌گاه تصویر مطلوب‌تر می‌باشد.



هیستوگرام ایده‌آل



هیستوگرام نسبتاً خوب



هیستوگرام نامطلوب

حال در رویکرد مسطح‌سازی هیستوگرام، تابع تبدیلی یافت می‌شود که هیستوگرام تصویر را تا آن‌جا که ممکن است به حالت مسطح ایده‌آل نزدیک می‌کند.

الگوریتم

یک تصویر خاکستری با ماتریس I را در نظر می‌گیریم. فرض می‌کنیم که دقت شدت این تصویر برابر L باشد. یعنی در این تصویر هر پیکسل می‌تواند L سطح رنگ خاکستری متفاوت را اختیار کند. همچنین فرض کنید که تعداد کل پیکسل‌ها N باشد.

تابع p را به صورت زیر تعریف می‌کنیم:

$$p: \{0, 1, \dots, L - 1\} \rightarrow [0, 1]$$

$$p(i) = \frac{n_i}{N} = \frac{\text{تعداد پیکسل‌هایی که شدت رنگ آن‌ها برابر } i \text{ است}}{\text{تعداد کل پیکسل‌ها}}$$

با توجه به تابع p تابع S را به صورت زیر تعریف می‌کنیم

$$S: \{0, 1, \dots, L - 1\} \rightarrow [0, 1]$$

$$S(i) = \sum_{j=0}^i p(j) = \sum_{j=0}^i \frac{n_j}{N}$$

الگوریتم

توجه شود که:

تابع p همان تابع چگالی احتمال (Probability Density Function) و
تابع S همان تابع توزیع تجمعی احتمال (Cumulative Distribution Function)
برای شدت رنگ پیکسل‌های تصویر می‌باشد.

برای شهود بیشتر، در زیر این توابع رسم شده‌اند.

الگوریتم

در تئوری تبدیلات شدت دیدیم که تقریباً هر تابع را می‌توان به عنوان **تابع تبدیل شدت** در نظر گرفت. از قضا تابع $S(i)$ را می‌توان به عنوان تابع تبدیل شدت در نظر گرفت. نقطه قابل توجه این است که:

انتخاب تابع $S(i)$ به عنوان تابع تبدیل شدت باعث مسطح‌سازی شدن هیستوگرام تصویر و به تبع آن بهبود کنتراست تصویر می‌شود.

البته لازم به ذکر است که لازم است که در تابع p تغییراتی اعمال شود تا به صورت یک تابع تبدیل شدت مناسب تبدیل شود. این تابع اعدادی اعشاری بین 0 و 1 تولید می‌کند. برای آن که از این تابع اعدادی به صورت $\{0, 1, \dots, L - 1\}$ استخراج کنیم، کافی است که تابع تبدیل شدت را به صورت زیر تعریف کنیم

$$g(i) = \text{Round}((L - 1) \times S(i))$$

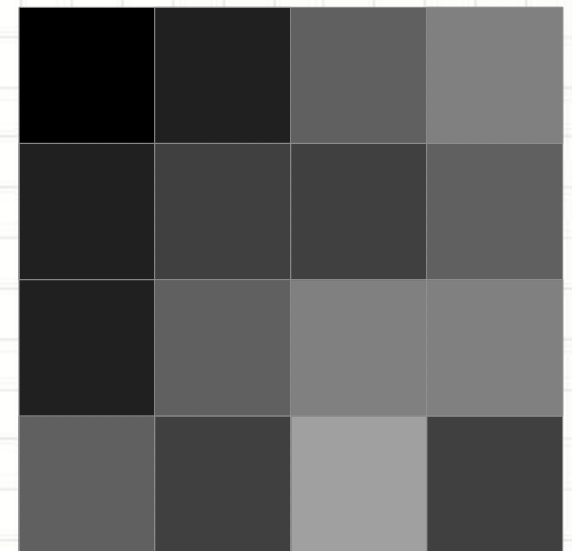
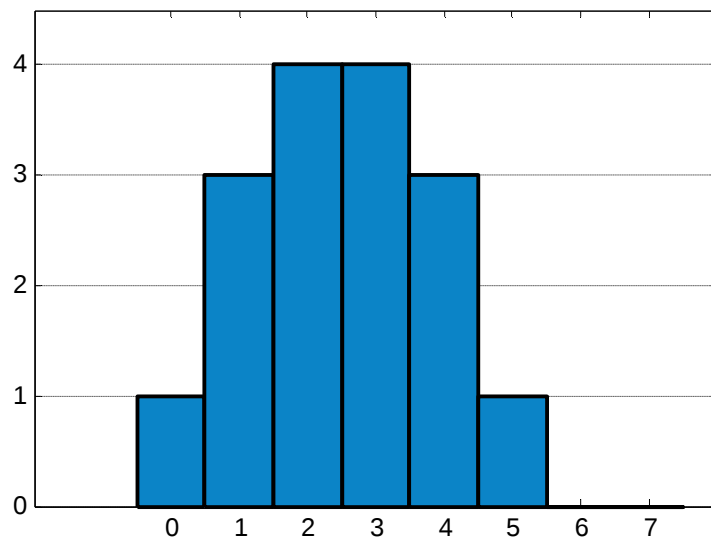
البته در برخی منابع پیشنهاد می‌شود که تابع g به صورت زیر در نظر گرفته شود

$$g(i) = \text{Round} \left(\frac{p(i) - p(i_{\min})}{1 - p(i_{\min})} \times L \right)$$

سطح سازی هیستوگرام – مثال ۱

0	1	3	4
1	2	2	3
1	3	4	4
3	2	5	2

یک تصویر با عمق بیت 3 ($L = 8$) با ماتریس تصویر روبرو را در نظر بگیرید:
در زیر این تصویر (در مقیاس بزرگ) و هیستوگرام آن رسم شده است.



همانطور که مشاهده می شود این تصویر تیره است و در ضمن دارای هیستوگرامی است که با هیستوگرام یکنواخت تفاوت نسبتاً زیادی دارد.

سطح سازی هیستوگرام-مثال ۱

فراوانی شدت‌های رنگ خاکستری به صورت زیر است:

$$\begin{bmatrix} 0 & 1 & 3 & 4 \\ 1 & 2 & 2 & 3 \\ 1 & 3 & 4 & 4 \\ 3 & 2 & 5 & 2 \end{bmatrix}$$

در ماتریس تصویر

شدت i	0	1	2	3	4	5	6	7
فراوانی	1	3	4	4	3	2	0	0

توابع $p(i)$ و $S(i)$ را محاسبه می‌کنیم

شدت i	0	1	2	3	4	5	6	7
فراوانی	1	3	4	4	3	2	0	0
$p(i)$	1/16	3/16	4/16	4/16	3/16	2/16	0/16	0/16
$S(i)$	1/16	4/16	8/16	12/16	15/16	16/16	16/16	16/16

مسطح سازی هیستوگرام – مثال ۱

با توجه به تابع S تابع تبدیل شدت g را به صورت زیر در نظر گرفته

$$g(i) = \text{Round}(s(i) \times 7)$$

و نتایج به صورت زیر خواهد بود:

شدت i	0	1	2	3	4	5	6	7
فراوانی	1	3	4	4	3	2	0	0
$p(i)$	1/16	3/16	4/16	4/16	3/16	2/16	0/16	0/16
$S(i)$	1/16	4/16	8/16	12/16	15/16	16/16	16/16	16/16
$g(i)$	0	2	4	5	7	7	7	7

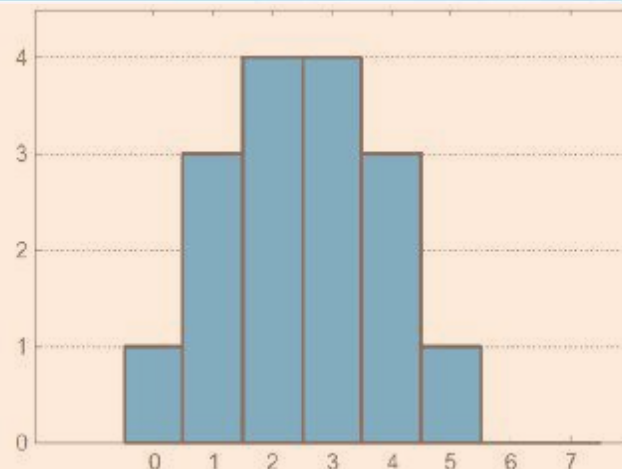
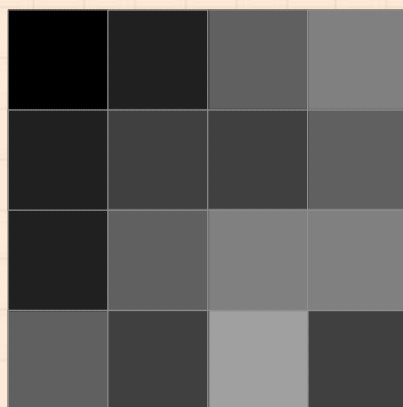
سطر اول و آخر این جدول نحوه تبدیل شدت رنگ را مشخص می کند.

شدت i	0	1	2	3	4	5	6	7
$g(i)$	0	2	4	5	7	7	7	7

مطابق با جدول فوق شدت رنگ 1 به 2 تبدیل می شود و شدت رنگ 2 به 4 تبدیل می شود و ...

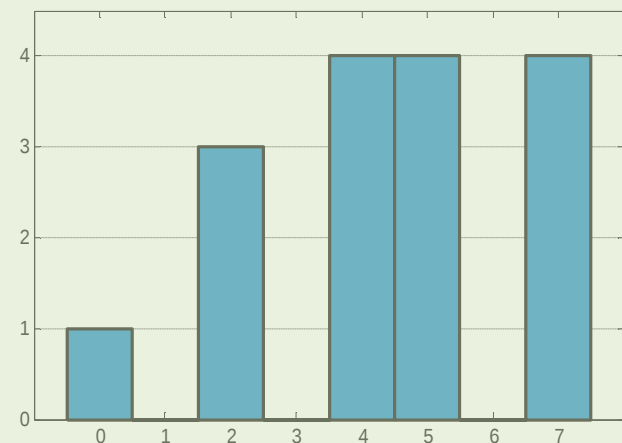
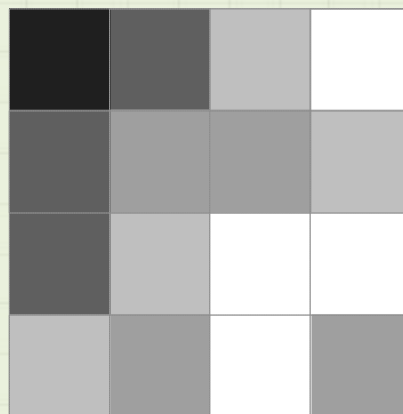
سطح سازی هیستوگرام-مثال ۱

0	1	3	4
1	2	2	3
1	3	4	4
3	2	5	2

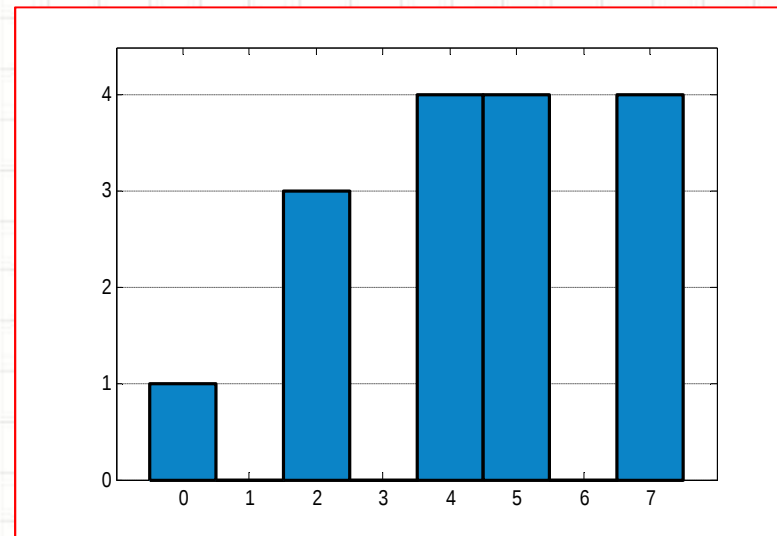
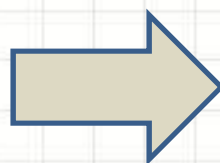
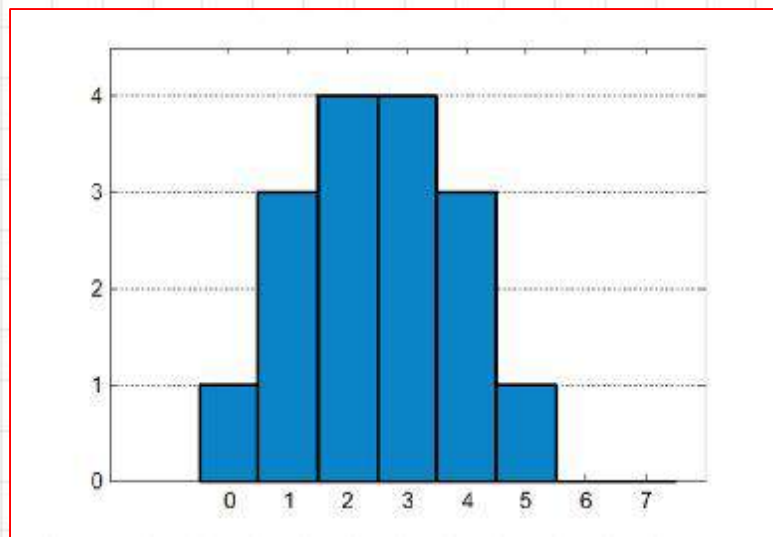


شدت i	0	1	2	3	4	5	6	7
$g(i)$	0	2	4	5	7	7	7	7

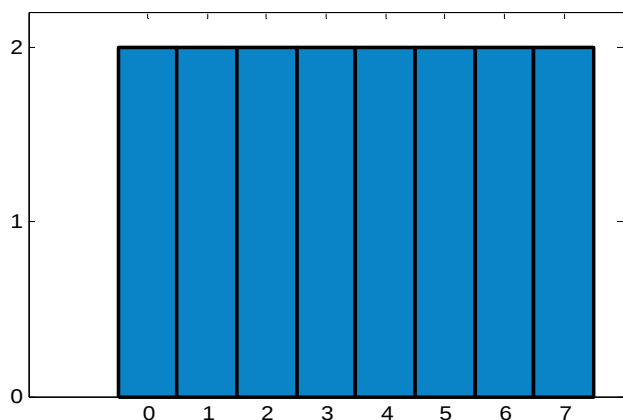
0	2	5	7
2	4	4	5
2	5	7	7
5	4	7	4



سطح سازی هیستوگرام - مثال ۱



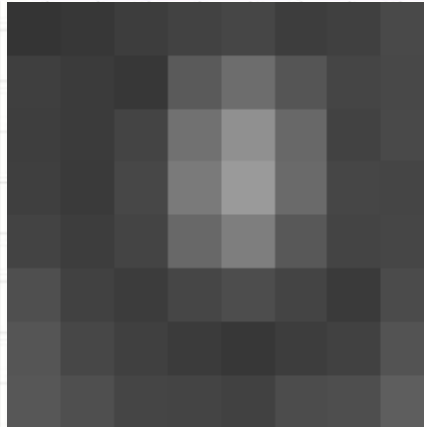
همان گونه که ملاحظه می شود، هیستوگرام در تصویر دوم به هیستوگرام یکنواخت نزدیک تر است. به عبارت دیگر در هیستوگرام اول تفاوت ارتفاع ها کم تر است.



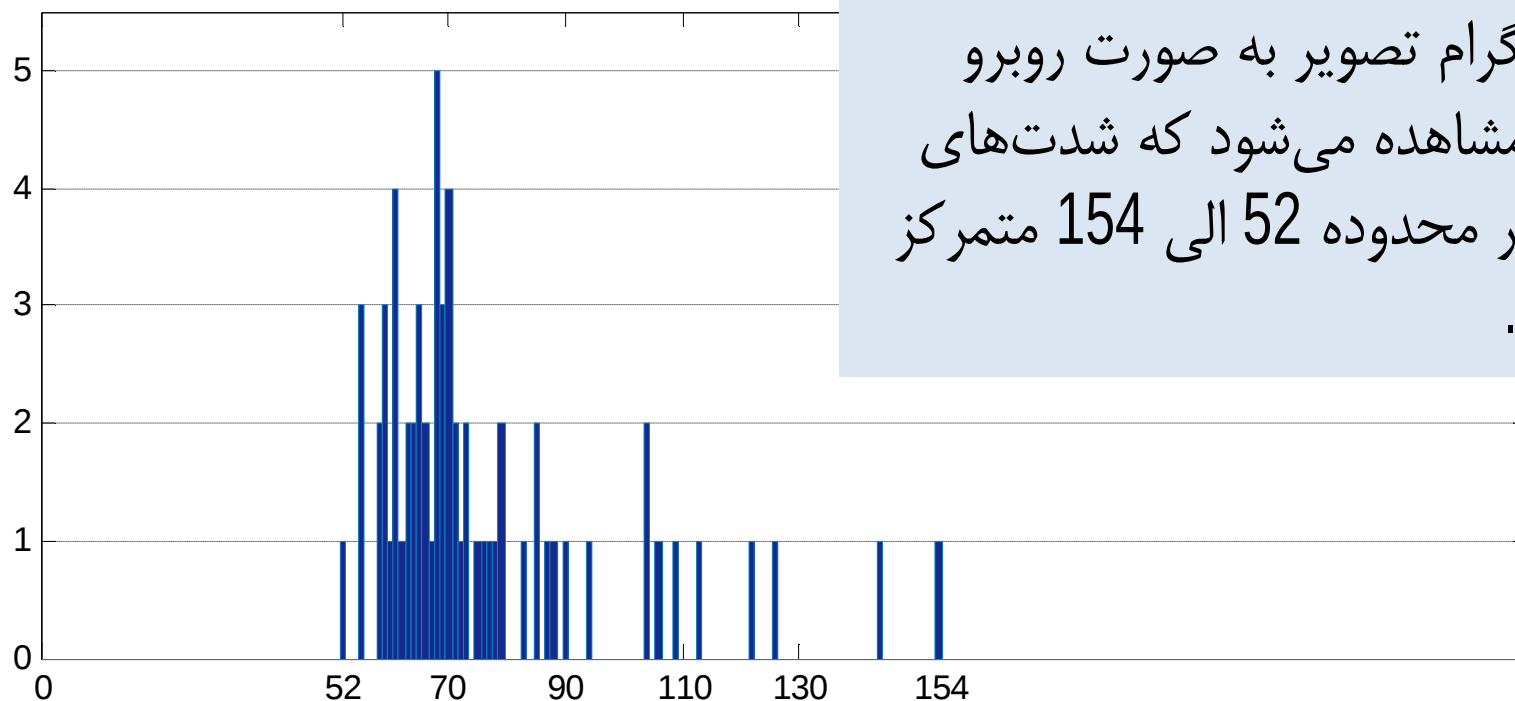
ایده آل این است که هیستوگرام تصویر دوم به صورت روبرو باشد. اما این تبدیل غیرممکن است و در روش فوق سعی شده که هیستوگرام تصویر تا آن جا که ممکن است به هیستوگرام ایده آل روبرو نزدیک باشد

سطح سازی هیستوگرام – مثال ۲

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94



تصویر روبرو را در نظر می گیریم
عمق بیت تصویر $8(L = 256)$
است.



هیستوگرام تصویر به صورت روبرو
است. مشاهده می شود که شدت های
رنگ در محدوده 52 الی 154 متمرکز
شده اند.

سطح سازی هیستوگرام – مثال ۲

در زیر نتایج اعمال الگوریتم سطح سازی هیستوگرام آورده شده است. البته برای خلاصه بودن جدول فقط شدت‌های موجود در تصویر را در جدول در نظر گرفته‌ایم. تابع تبدیل شدت به صورت زیر در نظر گرفته شده است:

$$g(i) = \text{Round}(S(i) \times 255)$$

شدت i	52	55	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	75	76	77	78	79	83	85	87	88	90	94	104	106	109	113	122	126	144	154
فراوانی	1	3	2	3	1	4	1	2	2	3	2	1	5	3	4	2	1	2	1	1	1	1	2	1	2	1	1	1	2	1	1	1	1	1	1	1	
$p(i)$	0.015625	0.046875	0.03125	0.046875	0.015625	0.0625	0.015625	0.03125	0.03125	0.046875	0.03125	0.015625	0.078125	0.046875	0.0625	0.03125	0.015625	0.03125	0.015625	0.015625	0.015625	0.015625	0.03125	0.015625	0.03125	0.015625	0.015625	0.015625	0.03125	0.015625	0.015625	0.015625	0.015625	0.015625	0.015625	0.015625	
$S(i)$	0.015625	0.0625	0.09375	0.14063	0.15625	0.21875	0.23438	0.26563	0.29688	0.34375	0.375	0.39063	0.46875	0.51563	0.57813	0.60938	0.625	0.65625	0.67188	0.6875	0.70313	0.71875	0.75	0.76563	0.79688	0.8125	0.82813	0.84375	0.85938	0.89063	0.90625	0.92188	0.9375	0.95313	0.96875	0.98438	1
$g(i)$	0	12	20	32	36	53	57	65	73	85	93	97	117	130	146	154	158	166	170	174	178	182	190	194	202	206	210	215	219	227	231	235	239	243	247	251	255

سطح سازی هیستوگرام – مثال ۲

به طور خلاصه شدت های تصویر اولیه به صورت زیر تغییر می کند:

شدت i	52	55	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	75	76	77	78	79	83	85	87	88	90	94	104	106	109	113	122	126	144	154
$g(i)$	0	12	20	32	36	53	57	65	73	85	93	97	117	130	146	154	158	166	170	174	178	182	190	194	202	206	210	215	219	227	231	235	239	243	247	251	255

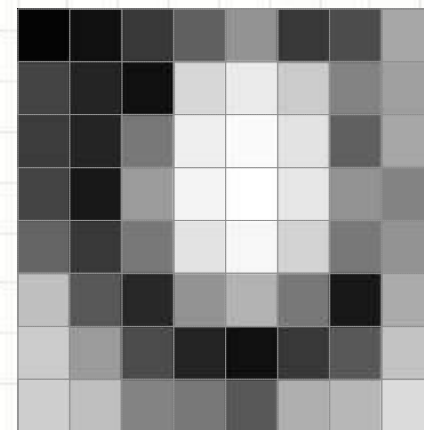
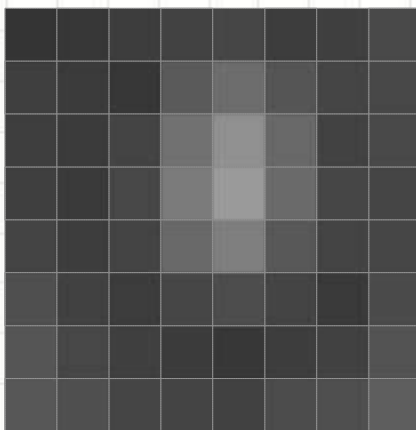
با این تبدیل ماتریس شدت تصویر و تصویر به صورت زیر تغییر می کند

52	55	61	66	70	61	64	73
63	59	55	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	68	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94



4	16	56	96	147	56	76	167
68	36	16	215	235	203	131	159
60	36	120	239	251	227	96	167
68	24	155	243	255	231	147	131
100	56	120	227	247	211	120	147
191	88	40	147	179	120	24	171
203	155	76	36	16	56	88	195
207	191	131	120	88	175	183	219

-4

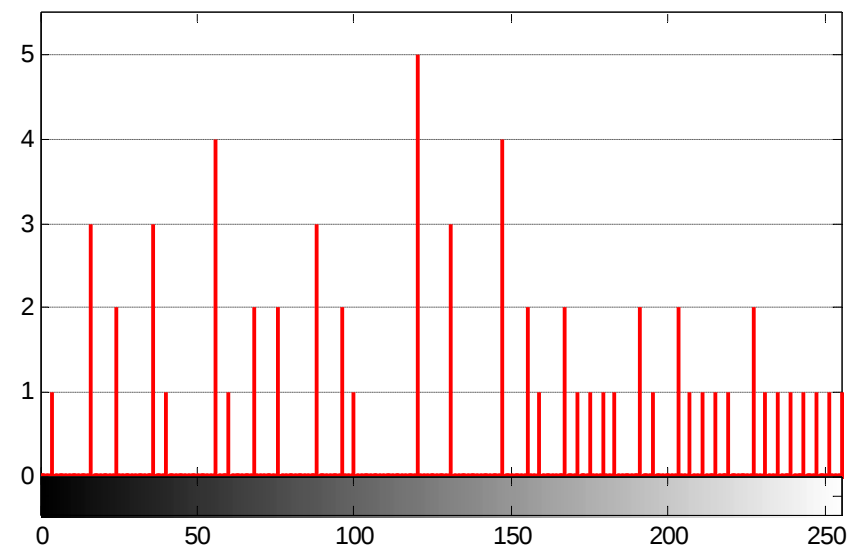
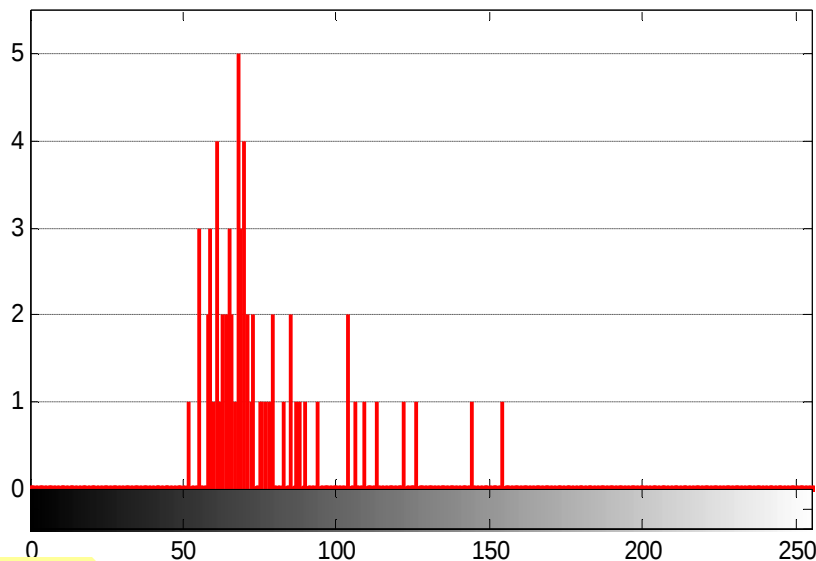
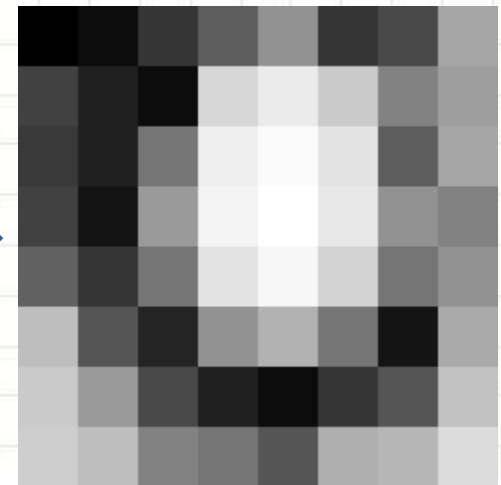


سطح سازی هیستوگرام – مثال ۲

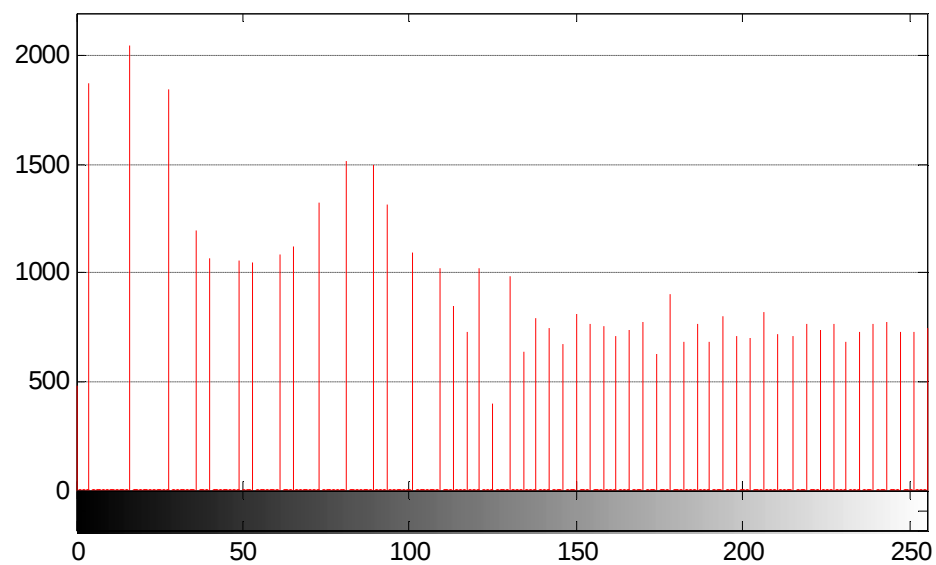
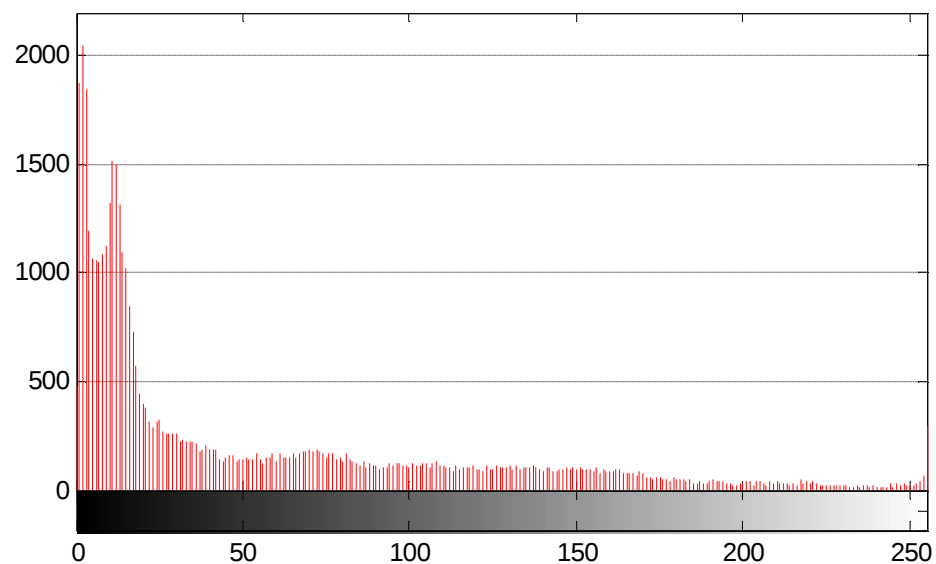
برای مشاهده بهتر تصویر اولیه و تصویر تبدیل شده در کنار هم رسم شده اند.



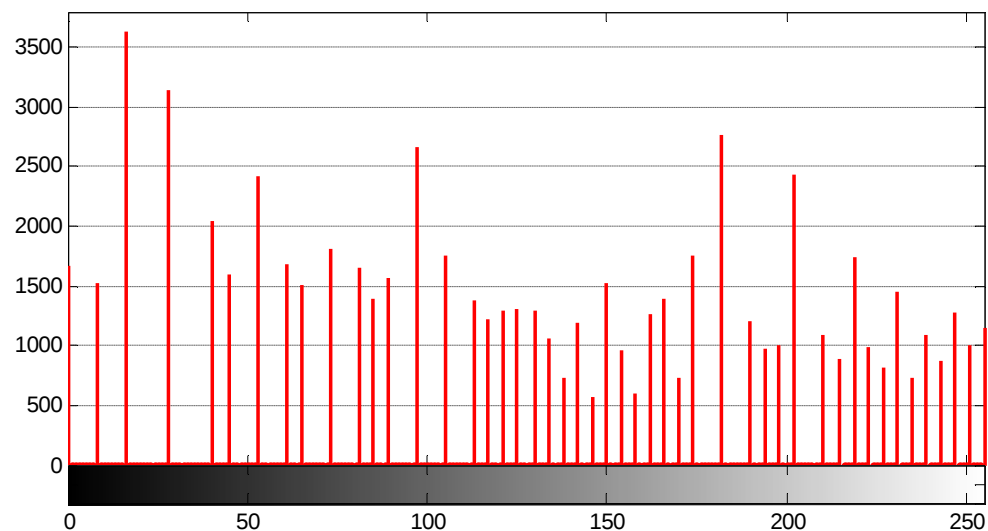
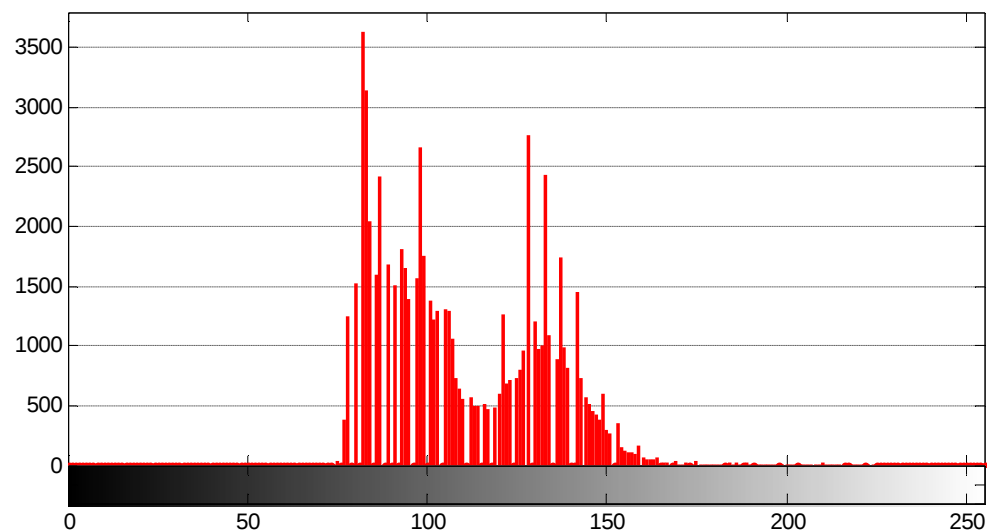
$$g(i) = \text{Round}(S(i) \times 255)$$



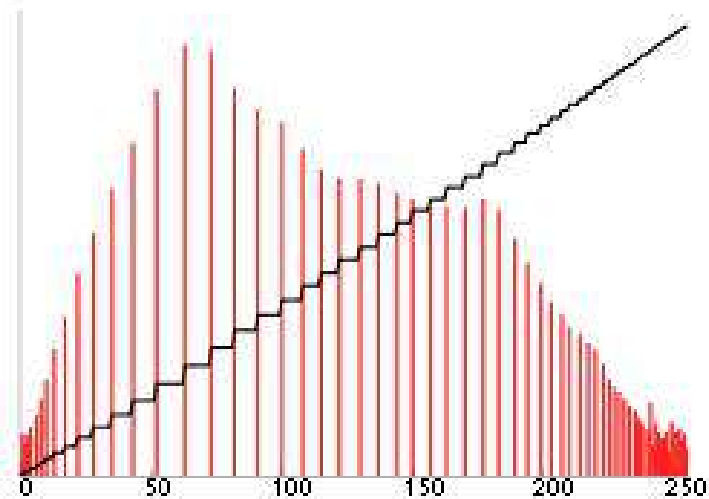
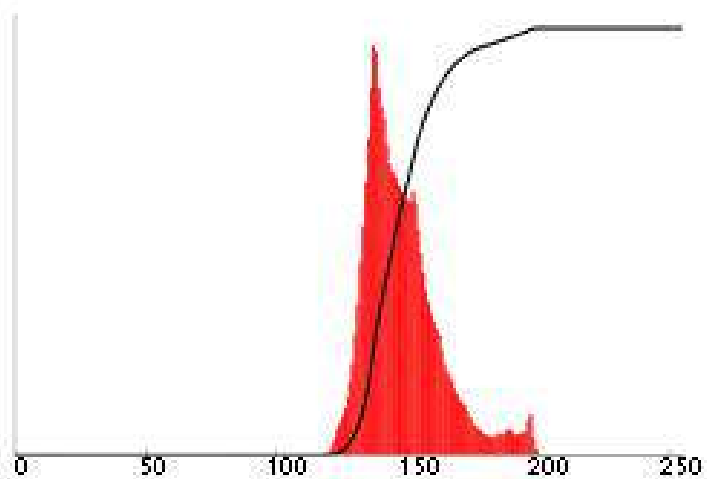
مسطح سازی هیستوگرام — نمونه



سطح سازی هیستوگرام - نمونه



مسطح سازی هیستوگرام — نمونه



مقایسه کشش و مسطح سازی هیستوگرام

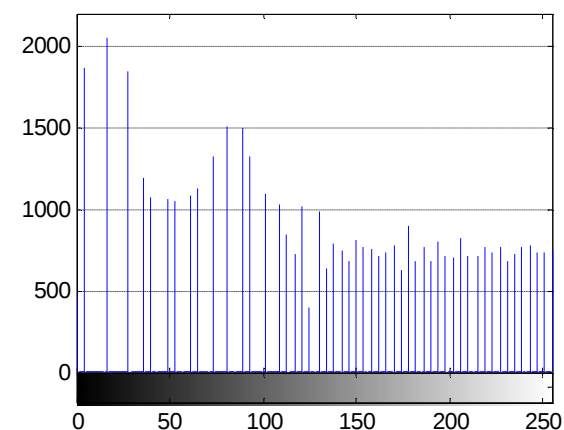
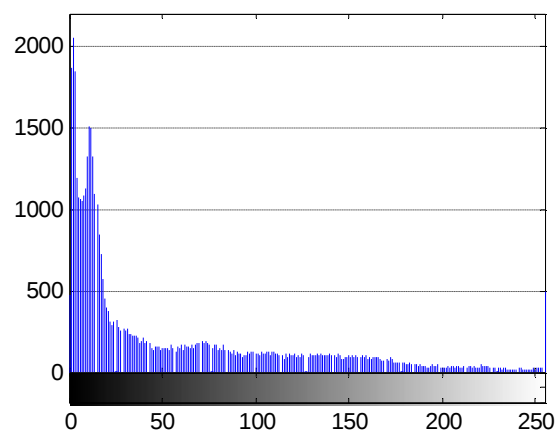
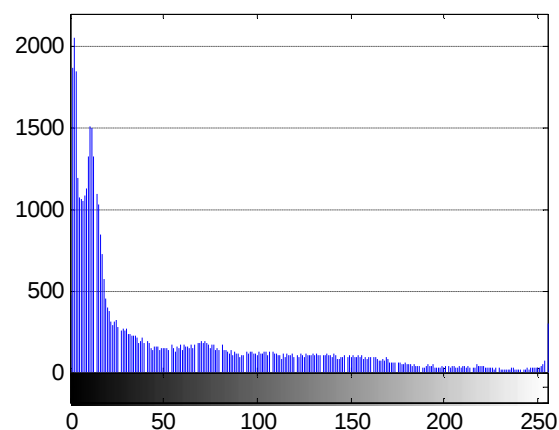
تصویر اصلی



تصویر پس از کشش هیستوگرام



تصویر پس از مسطح سازی هیستوگرام



مقایسه کشش و مسطح‌سازی هیستوگرام

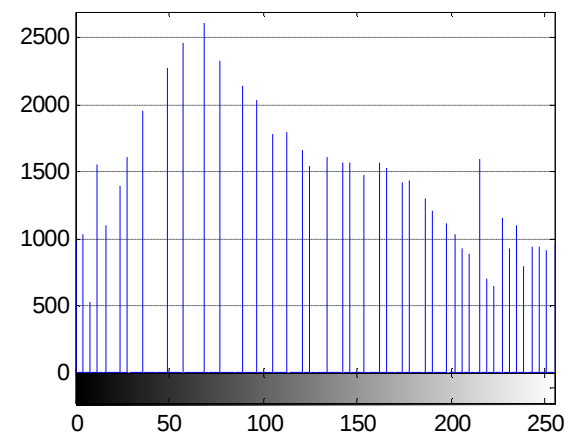
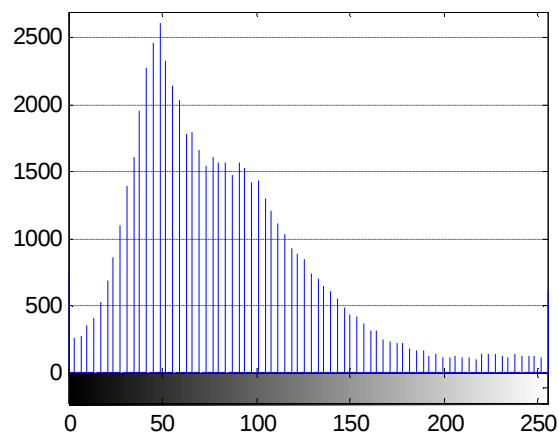
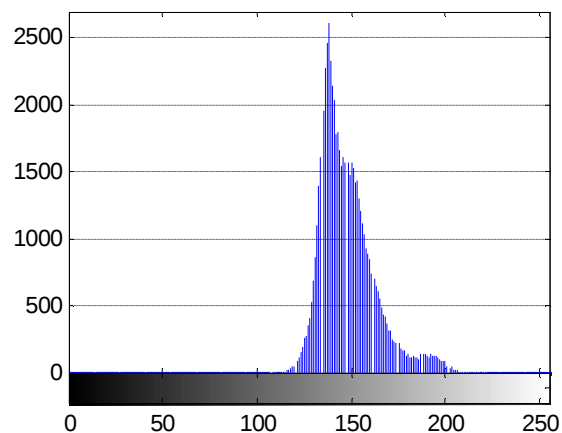
تصویر اصلی



تصویر پس از کشش هیستوگرام



تصویر پس از مسطح‌سازی هیستوگرام



مقایسه کشش و مسطح سازی هیستوگرام

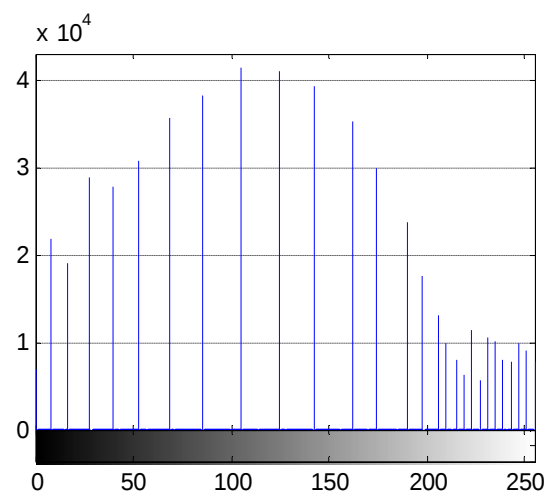
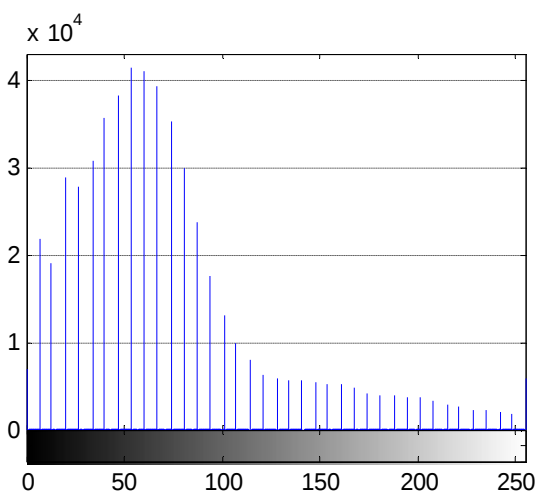
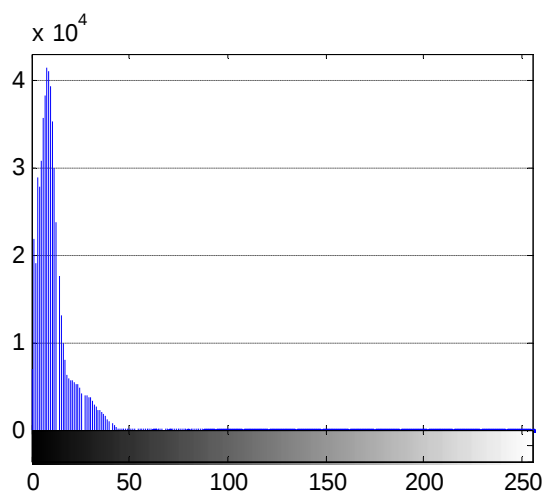
تصویر اصلی



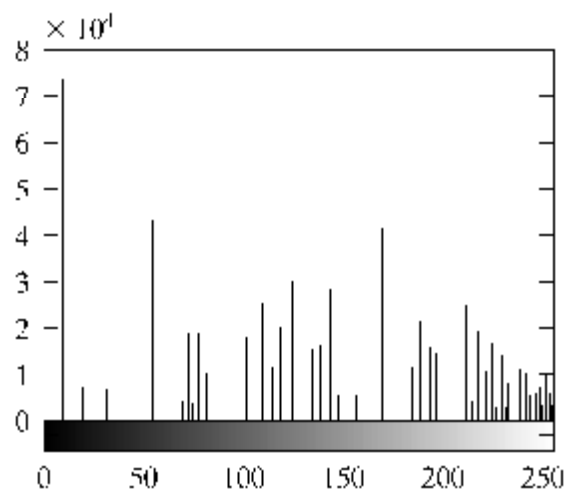
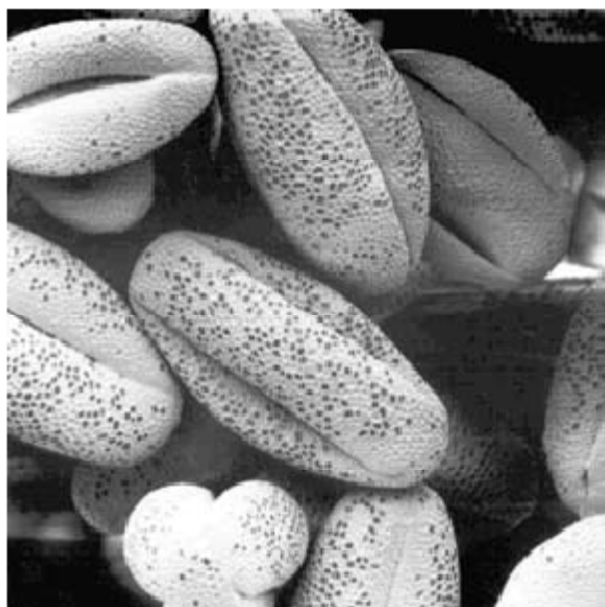
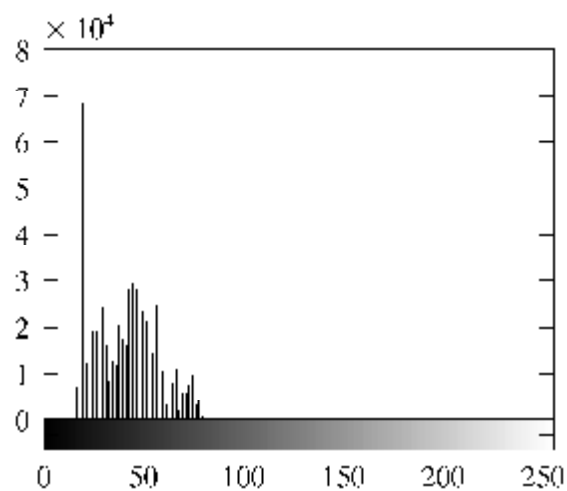
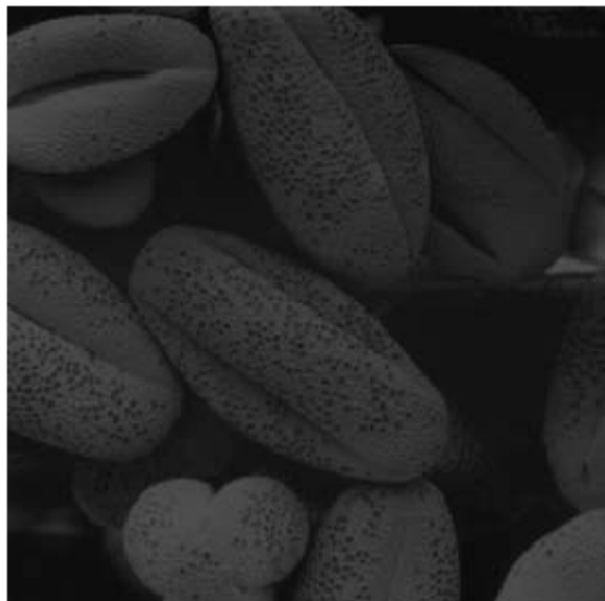
تصویر پس از کشش هیستوگرام



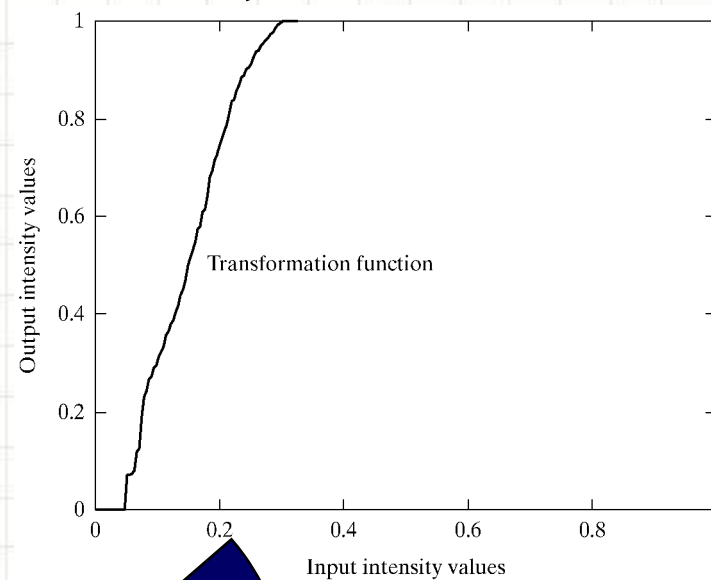
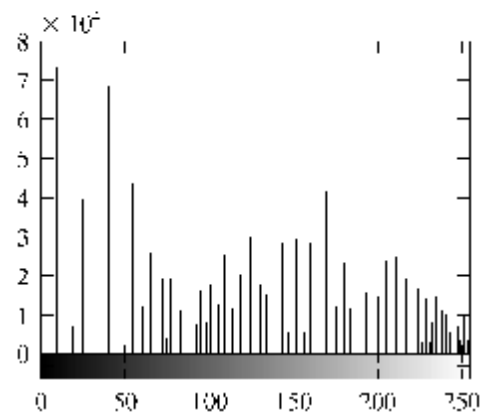
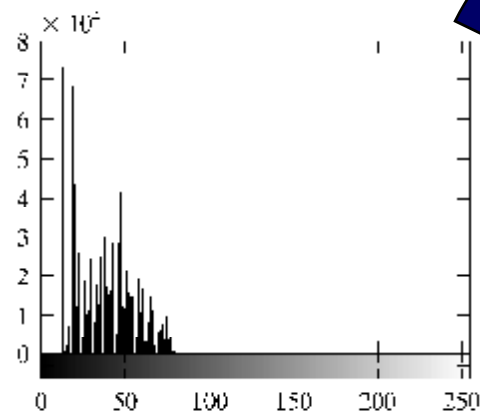
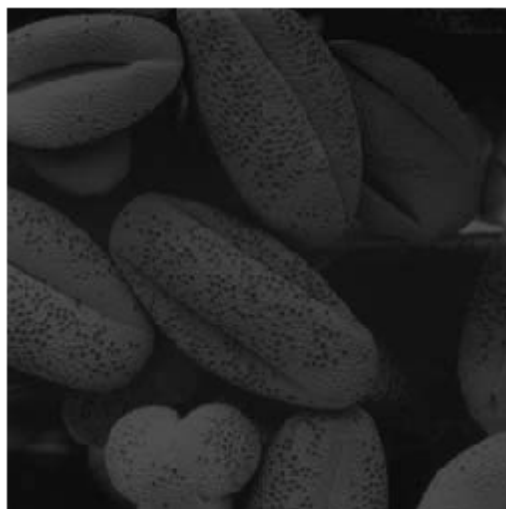
تصویر پس از مسطح سازی هیستوگرام



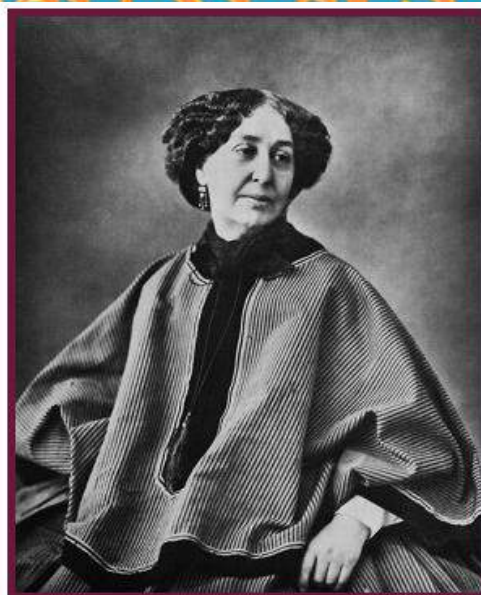
هیستوگرام یک تصویر



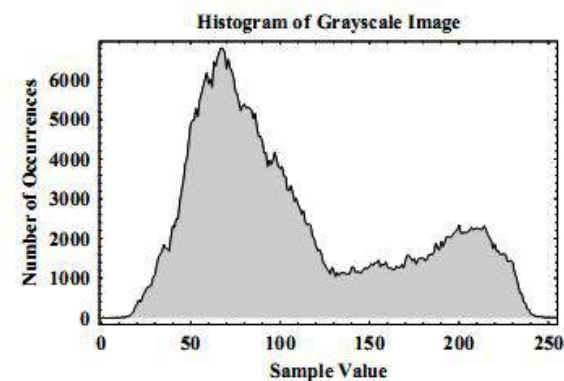
هیسٹوگرام یک تصویر



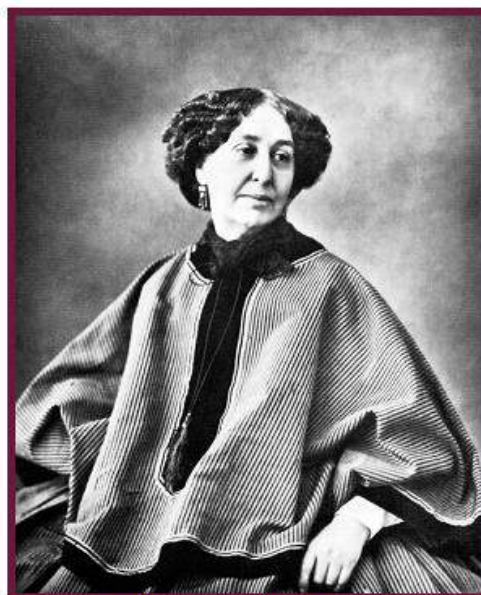
Example



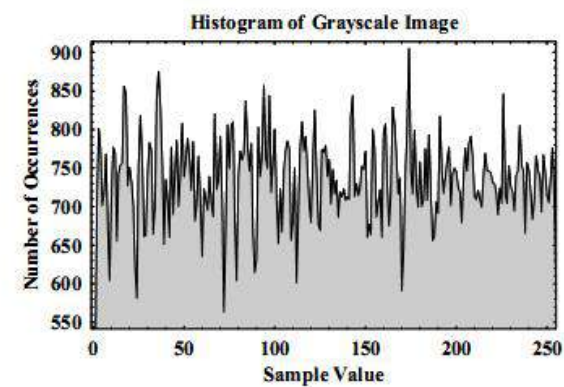
(a) Underexposed image.



(b) Histogram of the underexposed image.



(c) Histogram-equalized image.

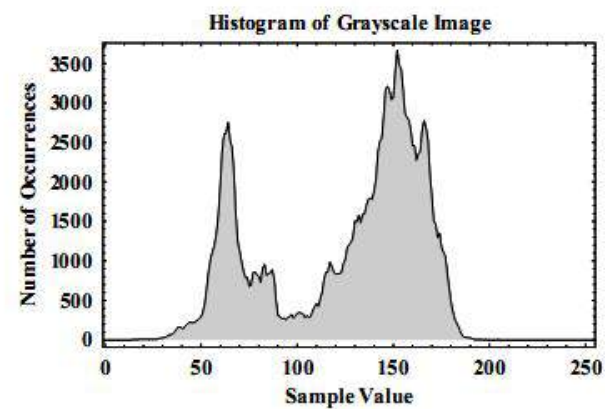


(d) Histogram of the equalized image.

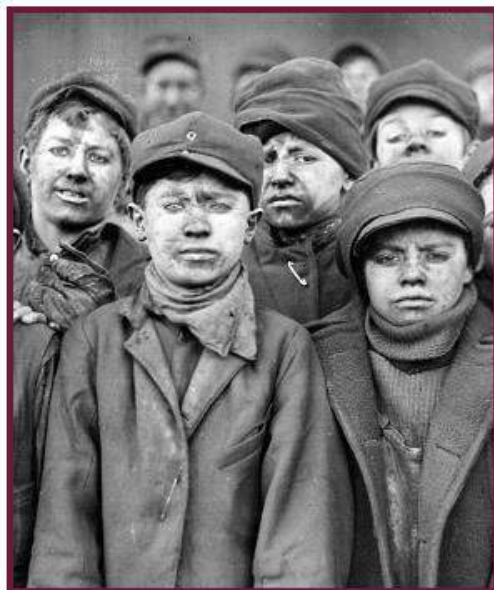
Figure 5.10. Histogram equalization of an underexposed image.



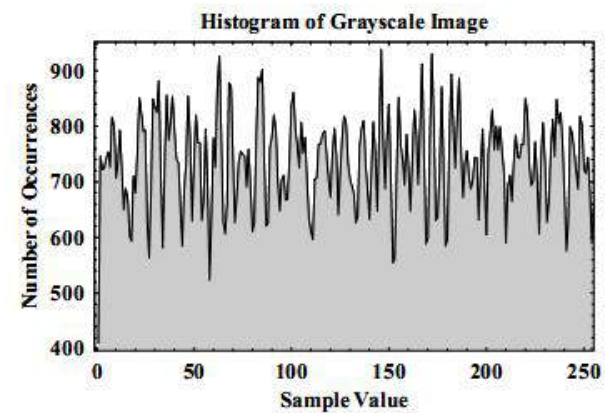
(a) Low-contrast image.



(b) Histogram of the low-contrast image.



(c) Histogram equalized image.



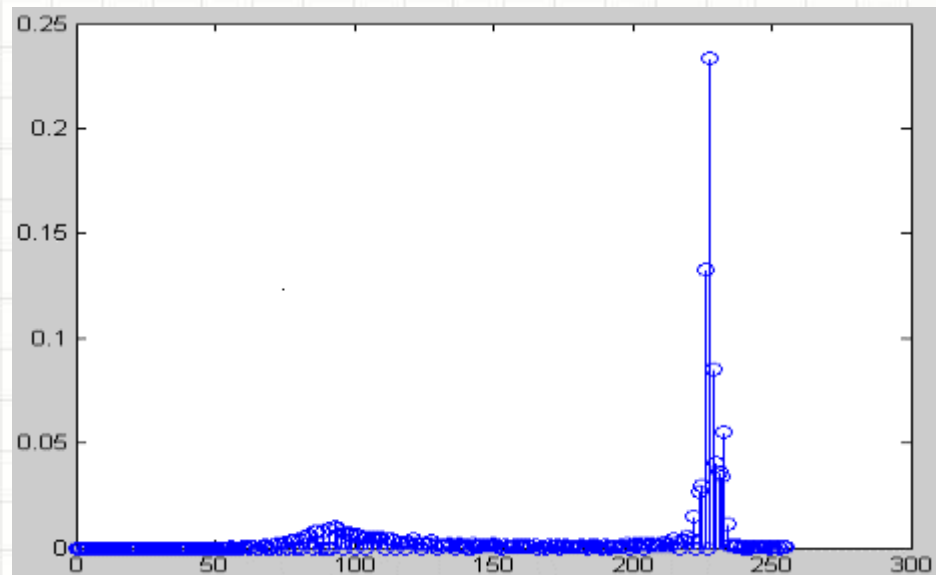
(d) Histogram of the equalized image.

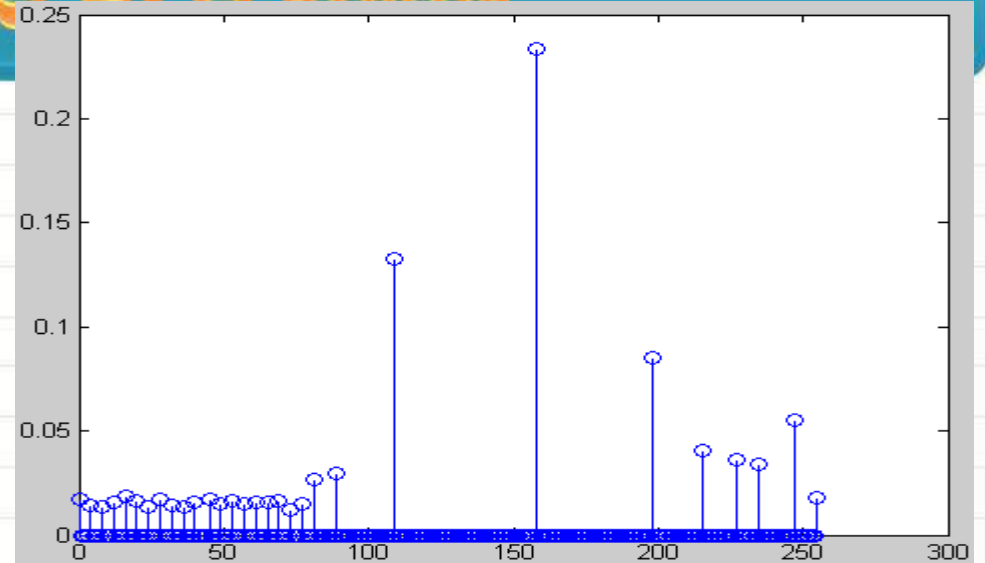
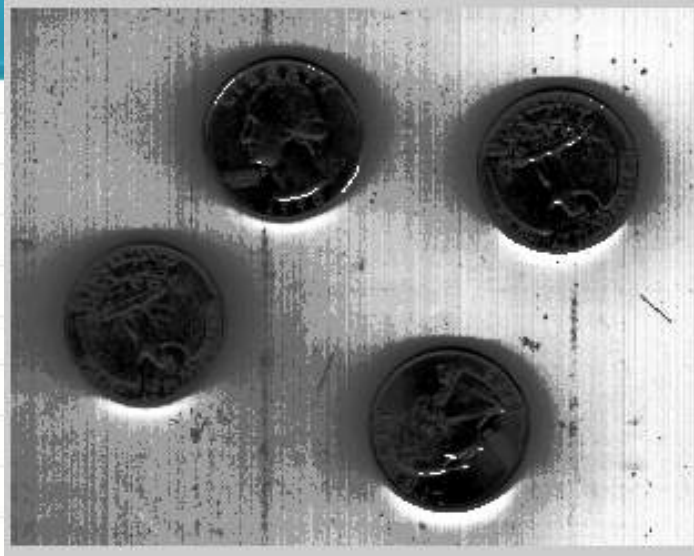
Figure 5.11. Histogram equalization of a low-contrast image.

هستوگرام یک تصویر

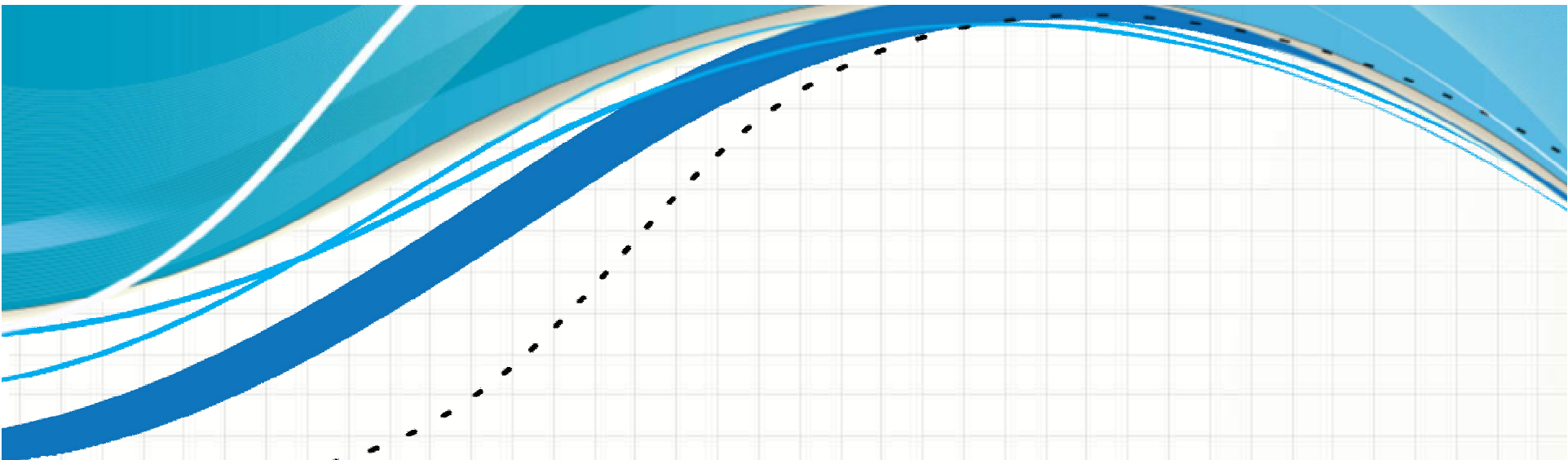
- Comments:

Histogram equalization may not always produce desirable results, particularly if the given histogram is very narrow. It can produce false edges and regions. It can also increase image “graininess” and “patchiness.”





```
>> clear
>> ix=imread('eight.tif');
>> imshow(ix);
>> iy=histeq(ix);
>> figure
>> imshow(iy);
>> ix=double(ix);
>> figure
>> hx=histogram(ix);
>> stem(0:255,hx);
>> iy=double(iy);
>> figure
>> hy=histogram(iy);
>> stem(0:255,hy);
..
```



فیلتر (تبدیلات همسایگی) در

پردازش تصویر

Filtering in Image

Processing

فیلتر (تبدیل همسایگی) – معرفی

تبدیل همسایگی عملگری است که روی تصویر اعمال می‌شود و نتیجه آن تصویر دیگری است که شدت رنگ پیکسل‌ها تغییر کرده است.

بنابراین تبدیل همسایگی نوعی **تبدیل شدت تصویر** می‌باشد. همان‌گونه که دیدیم، تبدیلات لگاریتمی، گاما، تکه‌ای خطی و تبدیلات مبتنی بر هیستوگرام نیز جزو روش‌های **تبدیل شدت تصویر** بودند. اما در تبدیلات همسایگی با رویکرد متفاوتی شدت رنگ پیکسل‌ها را تغییر می‌دهد.

به عبارت دیگر فرض کنیم I یک تصویر باشد و شدت رنگ خاکستری پیکسل (x, y) برابر $I(x, y)$ باشد. در تبدیلات لگاریتمی، گاما و ... تصویر I به تصویر J تبدیل می‌شود که شدت رنگ پیکسل (x, y) در تصویر J به صورت زیر محاسبه می‌شود:

$$J(x, y) = g(I(x, y))$$

یعنی شدت پیکسل (x, y) در تصویر J فقط بر حسب شدت رنگ پیکسل (x, y) در تصویر I محاسبه می‌شود.

اما در تبدیلات همسایگی شدت رنگ پیکسل (x, y) در تصویر J نه تنها شدت پیکسل (x, y) در تصویر I بلکه **برخی پیکسل‌های اطراف** (x, y) نیز در محاسبه شدت استفاده می‌شوند.

فیلتر (تبدیل همسایگی) – معرفی

در مقایسه با تبدیلات همسایگی، نوع اول تبدیلات، یعنی تبدیلات لگاریتمی، گاما و ... به تبدیلات نقطه‌ای و یا **تبدیلات شدت نقطه‌ای** موسوم می‌باشند. بنابراین تبدیلات شدت را می‌توان به دو دسته زیر تقسیم کرد

- تبدیلات شدت نقطه‌ای (Point Processing Intensity Transform)
 - تبدیلات شدت همسایگی (Neighborhood Processing Intensity Transform)
- غالباً تبدیلات شدت همسایگی به **فیلتر (Filter)** موسوم می‌باشند.

در تبدیلات نقطه‌ای مانند تبدیلات لگاریتمی، گاما و ... **تابع تبدیل شدت** نحوه تغییر شدت پیکسل‌ها را مشخص می‌کند. این تابع یک عدد را می‌گیرد و یک عدد دیگر را پس می‌دهد. در واقع این تابع یک شدت رنگ را می‌گیرد و آن را به یک شدت رنگ دیگر تبدیل می‌کند.

با تبدیلات همسایگی نیز یک تابع متناظر است که این تابع یک **تابع چند متغیره** است. یعنی چند عدد را می‌گیرد و یک عدد را پس می‌دهد. در واقع این تابع شدت رنگ پیکسل و شدت رنگ برخی پیکسل‌های اطراف (**همسایگی**) آن را می‌گیرد و یک شدت رنگ پس می‌دهد.

معمولاً این تابع خطی می‌باشد.

فیلتر (تبدیل همسایگی) - معرفی

		$(x-1, y-1)$	$(x-1, y)$	$(x-1, y+1)$	
		$(x, y-1)$	(x, y)	$(x, y+1)$	
		$(x+1, y-1)$	$(x+1, y)$	$(x+1, y+1)$	

فرض کنیم I یک تصویر باشد و شدت رنگ خاکستری پیکسل (x, y) برابر $I(x, y)$ باشد.

حال پیکسل‌های اطراف پیکسل (x, y) عبارتند از

$(x-1, y-1), (x, y-1), (x+1, y-1), (x+1, y), (x+1, y+1), (x, y+1), (x-1, y+1), (x, y-1)$

در واقع پیکسل‌های فوق تشکیل

یک همسایگی 9 تایی حول

پیکسل (x, y) را می‌دهند.

حال یک تبدیل همسایگی مبتنی بر همسایگی 9 تایی فوق، یک تابع 9 متغیره به صورت زیر است:

$$J(x, y) =$$

$$F(I(x-1, y-1), I(x-1, y), I(x-1, y+1), I(x, y-1), I(x, y), I(x, y+1), I(x+1, y+1), I(x+1, y), I(x+1, y+1))$$

فیلتر (تبدیل همسایگی) – معرفی

در واقع شدت رنگ هر پیکسل (x, y) در تصویر J باید از رابطه زیر محاسبه شود

$$J(x, y) =$$

$$F(I(x-1, y-1), I(x-1, y), I(x-1, y+1), I(x, y-1), I(x, y), I(x, y+1), I(x+1, y+1), I(x+1, y), I(x+1, y+1))$$

اما اگر تابع تبدیل همسایگی یک تابع خطی باشد، آن گاه می‌توان آن را به فرم زیر نوشت:

$$J(x, y) =$$

$$F(I(x-1, y-1), I(x-1, y), I(x-1, y+1), I(x, y-1), I(x, y), I(x, y+1), I(x+1, y+1), I(x+1, y), I(x+1, y+1)) =$$

$$\begin{aligned} &w_{-1,-1}I(x-1, y-1) + w_{-1,0}I(x-1, y) + w_{-1,1}I(x-1, y+1) \\ &+ w_{0,-1}I(x, y-1) + w_{0,0}I(x, y) + w_{0,1}I(x, y+1) \\ &+ w_{1,-1}I(x+1, y-1) + w_{1,0}I(x+1, y) + w_{1,1}I(x+1, y+1) \end{aligned}$$

تابع F در فوق را می‌توان با نماد مناسب به صورت خلاصه زیر نیز نوشت:

$$J(x, y) = \sum_{i=-1}^1 \sum_{j=-1}^1 w_{i,j} I(x+i, y+j)$$

فیلتر (تبدیل همسایگی) – معرفی

ضرایب W_{ij} در تابع فوق نقش مهمی در تعریف تابع فوق دارد. معمولاً این ضرایب در یک ماتریس 3 در 3 به صورت زیر قرار داده می‌شود.

$$\begin{bmatrix} W_{-1,-1} & W_{-1,0} & W_{-1,1} \\ W_{0,-1} & W_{00} & W_{0,1} \\ W_{1,-1} & W_{1,0} & W_{1,1} \end{bmatrix}$$

این ماتریس به **ماتریس ماسک (Mask)** موسوم است. البته این ماتریس با نام‌های دیگری زیر نیز شناخته شده است:

ماتریس هسته (Kernel)، ماتریس فیلتر، ماتریس پنجره (Window)

تبدیل همسایگی

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	2	7	8	5
0	3	2	4	6
3	4	8	9	3
1	1	0	0	2

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	2	7	8	5
0	3	2	4	6
3	4	8	9	3
1	1	0	0	2

	0			

$$1 * 1 + (-1) * 2 + 0 * 7 + 3 * 0 + 1 * 3 + 5 * 2 + 0 * 3 + 1 * 4 + (-2) * 8 = 0$$

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	2	7	8	5
0	3	2	4	6
3	4	8	9	3
1	1	0	0	2

	0	12		

$$1 * 2 + (-1) * 7 + 0 * 8 + 3 * 3 + 1 * 2 + 5 * 4 + 0 * 4 + 1 * 8 + (-2) * 9 = 12$$

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	2	7	8	5
0	3	2	4	6
3	4	8	9	3
1	1	0	0	2

	0	12	42	

$$1 * 7 + (-1) * 8 + 0 * 5 + 3 * 2 + 1 * 4 + 5 * 6 + 0 * 8 + 1 * 9 + (-2) * 3 = 42$$

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	2	7	8	5
0	3	2	4	6
3	4	8	9	3
1	1	0	0	2

	0	12	42	
	51			

$$1 * 0 + (-1) * 3 + 0 * 2 + 3 * 3 + 1 * 4 + 5 * 8 + 0 * 1 + 1 * 1 + (-2) * 0 = 51$$

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	2	7	8	5
0	3	2	4	6
3	4	8	9	3
1	1	0	0	2

	0	12	42	
	51	66		

$$1 * 3 + (-1) * 2 + 0 * 4 + 3 * 4 + 1 * 8 + 5 * 9 + 0 * 1 + 1 * 0 + (-2) * 0 = 66$$

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	2	7	8	5
0	3	2	4	6
3	4	8	9	3
1	1	0	0	2

	0	12	42	
	51	66	42	

$$1 * 2 + (-1) * 4 + 0 * 6 + 3 * 8 + 1 * 9 + 5 * 3 + 0 * 0 + 1 * 0 + (-2) * 2 = 42$$

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

	1	2	7	8	5
	0	3	2	4	6
	3	4	8	9	3
	1	1	0	0	2

	0	12	42	
	51	66	42	

$$1 * 2 + (-1) * 4 + 0 * 6 + 3 * 8 + 1 * 9 + 5 * 3 + 0 * 0 + 1 * 0 + (-2) * 2 = 42$$

تبدیلات همسایگی-فیلتر

1	-1	0
3	1	5
0	1	-2

1	1	2	7	8	5	5
1	1	2	7	8	5	5
0	0	3	2	4	6	6
3	3	4	8	9	3	3
1	1	1	0	0	2	2
1	1	1	0	0	2	2

	0	12	42	
	51	66	42	

فیلتر - لایه گذاری تکراری

1	1	2	7	8	5	5
1	1	2	7	8	5	5
0	0	3	2	4	6	6
3	3	4	8	9	3	3
1	1	1	0	0	2	2
1	1	1	0	0	2	2

1	-1	0		
3	1	5		
0	1	-2		

8				
	0	12	42	
	51	66	42	

$$1 * 1 + (-1) * 1 + 0 * 2 + 3 * 1 + 1 * 1 + 5 * 2 + 0 * 0 + 1 * 0 + (-2) * 3 = 8$$

فیلتر - لایه گذاری صفر

1	-1	0
3	1	5
0	1	-2

0	0	0	0	0	0	0
0	1	2	7	8	5	0
0	0	3	2	4	6	0
0	3	4	8	9	3	0
0	1	1	0	0	2	0
0	0	0	0	0	0	0

	0	12	42	
	51	66	42	

فیلتر - لایه گذاری صفر

1	-1	0
3	1	5
0	1	-2

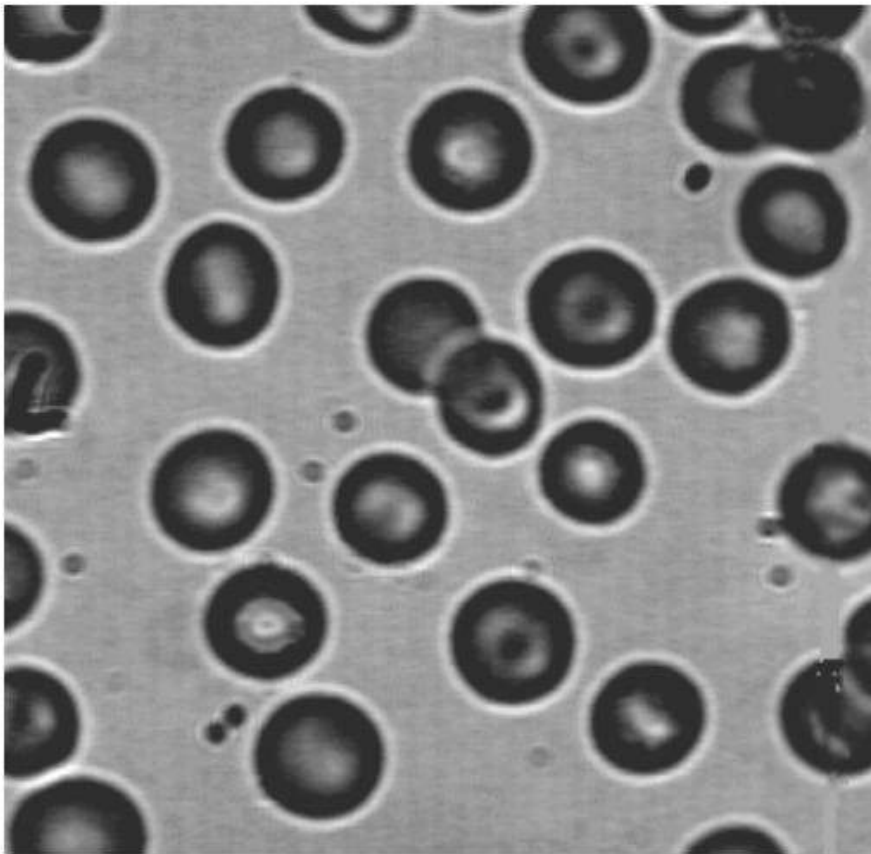
0	0	0	0	0	0	0
0	1	2	7	8	5	0
0	0	3	2	4	6	0
0	3	4	8	9	3	0
0	1	1	0	0	2	0
0	0	0	0	0	0	0

2				
	0	12	42	
	51	66	42	

$$1 * 0 + (-1) * 0 + 0 * 0 + 3 * 0 + 1 * 1 + 5 * 2 + 0 * 0 + 1 * 0 + (-2) * 3 = 2$$

مثال

فیلتر زیر را برای تصویر در نظر می گیریم

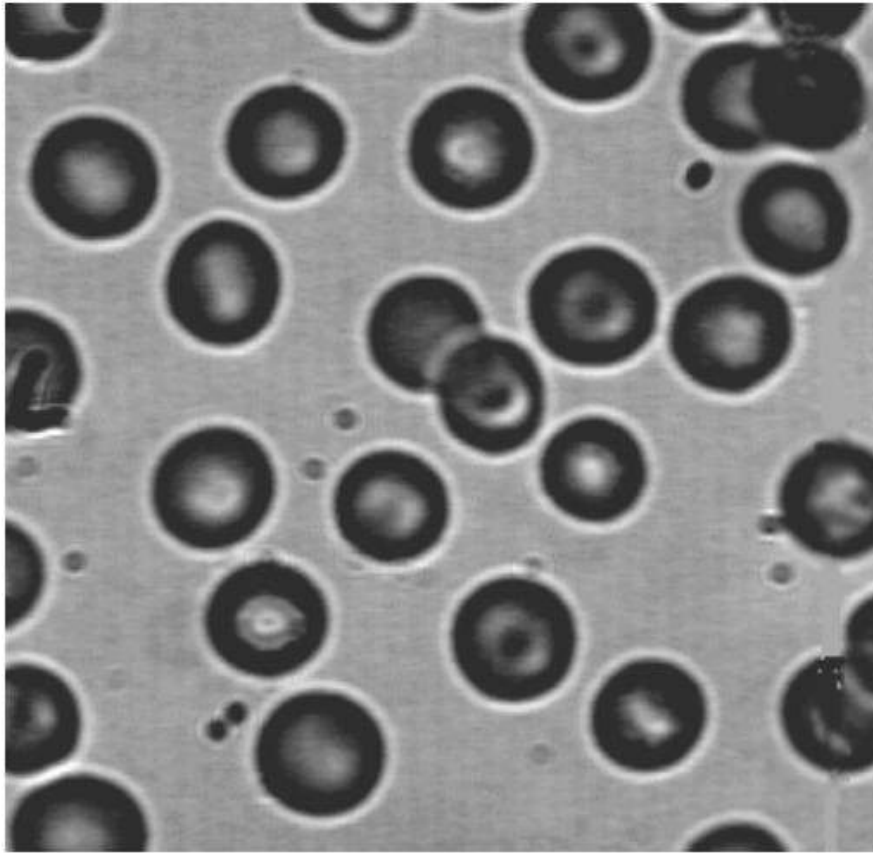


Original Image

$$\frac{1}{25} \times$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

مثال - لایه گذاری صفر



Original Image

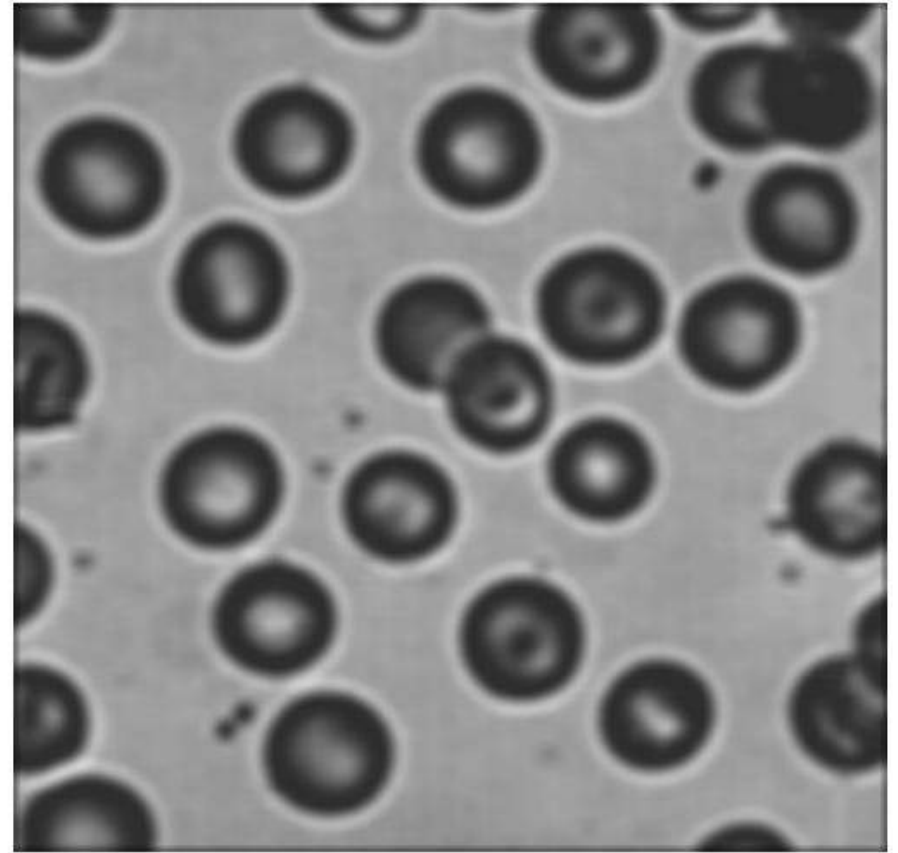
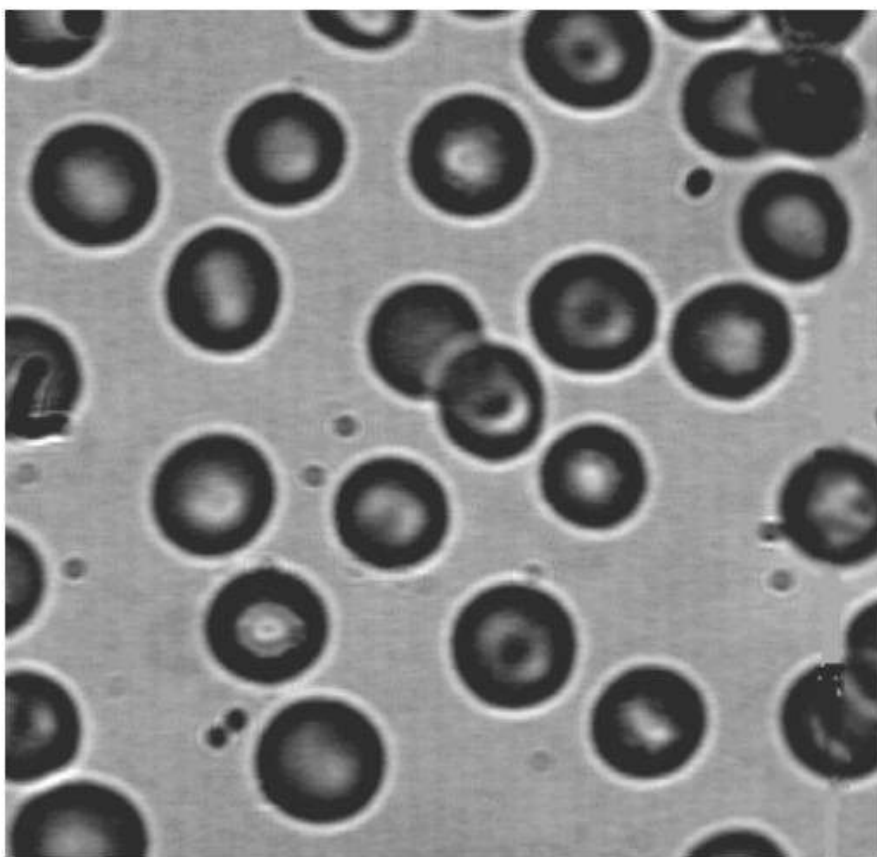


Image with Black Border

لایه گذاری تکراری



Original Image

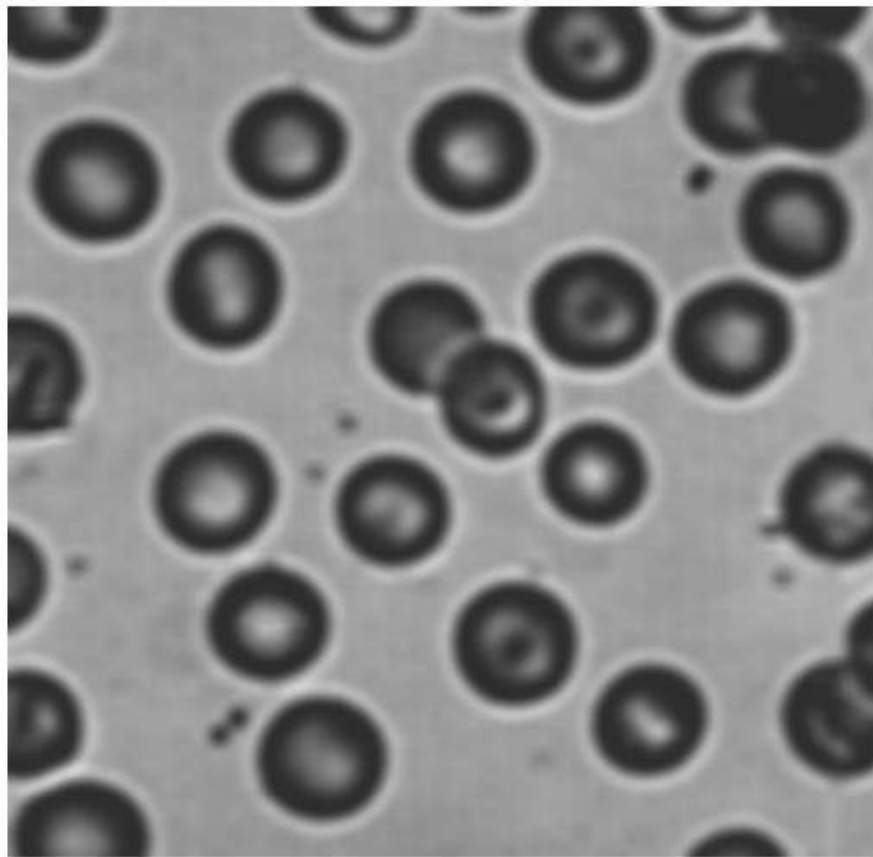


Image Border with Replication

کاربرد فیلتر در پردازش تصاویر

از جمله کاربرد فیلترها در پردازش تصاویر می‌توان به موارد زیر اشاره کرد

- بهبود تصاویر (مانند بهبود کنتراست)
- ملایم‌سازی تصاویر (حذف نویز)
- تشخیص الگوها در تصاویر (مانند قطعه‌بندی تصویر)

برخی فیلترهای مشهور

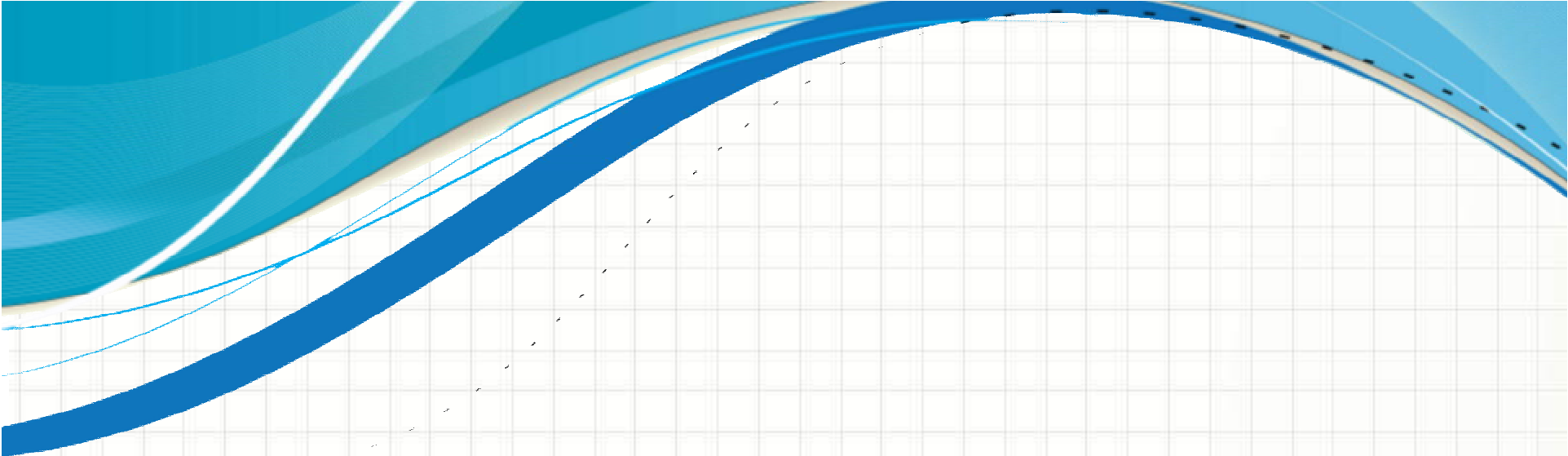
فیلترهای همسایگی
Neighborhood Filters

فیلترهای تیزکننده
Sharpening Filters

فیلتر لاپلاس (Laplace)
فیلتر سوبل (Sobel)

فیلترهای ملایم کننده
Smoothing Filters

فیلتر میانگین (Mean)
فیلتر میانه (Median)
فیلتر گوسی (Gaussian)
فیلتر



فیلترهای هموار کننده

Smoothing Filters

فیلترهای ملایم کننده – معرفی

فیلترهای ملایم کننده، یک دسته مهم از فیلترها می باشند که برخی از انواع آن به قرار زیر است:

1. فیلترهای میانگین
2. فیلترهای گوسی
3. فیلتر میانه

در ادامه این فیلترها تعریف می گردد و تاثیر آنها روی تصاویر نشان داده می شود و در انتها خاصیت این فیلترها گزارش می شود.

فیلتر میانگین

(Average or Mean Filter)

فیلتر میانگین (Average or Mean Filter)

فیلتر میانگین، ابتدا یک همسایگی حول پیکسل در نظر گرفته می‌شود و سپس میانگین شدت پیکسل‌های موجود در آن همسایگی، به عنوان مقدار جدید آن پیکسل در نظر گرفته می‌شود. معمولاً همسایگی حول پیکسل به صورت مربعی در نظر گرفته می‌شود که این مربع در هر وجه خود $2k + 1$ پیکسل دارد.

اگر I تصویر اولیه باشد که شدت پیکسل (x, y) آن برابر $I(x, y)$ باشد، آن‌گاه یک فیلتر میانگین با همسایگی $(2k + 1) \times (2k + 1)$ شدت پیکسل (x, y) را از $I(x, y)$ به $J(x, y)$ تغییر می‌دهد که

$$J(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k \frac{1}{(2k + 1)^2} I(x + i, y + j)$$

همان‌گونه که ملاحظه می‌شود، فیلتر میانگین یک **فیلتر خطی** می‌باشد و ماتریس ماسک آن یک ماتریس $(2k + 1) \times (2k + 1)$ است که تمامی مولفه‌های به صورت

$$w_{i,j} = \frac{1}{(2k+1)^2} \text{ است.}$$

فیلتر میانگین (Average or Mean Filter)

فیلتر میانگین یک فیلتر خطی است که ماتریس ماسک آن در ابعاد 3، 4 و 5 به صورت زیر است.

$$\frac{1}{9}$$

1	1	1
1	1	1
1	1	1

$$\frac{1}{16}$$

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

$$\frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

در فیلتر میانگین با همسایگی $(2k + 1) \times (2k + 1)$ تایی نیاز به لایه گذاری k تایی داریم.

فیلتر میانگین (Average or Mean Filter)

ماتریس ماسک در فیلتر میانگین می تواند به صورت **غیرمربعی** و یا **وزن دار** (غیریکنواخت) نیز در نظر گرفته شود. در زیر مثال هایی آورده شده است.

$$\frac{1}{15}$$

3	2	1
1	4	2
1	1	0

ماتریس ماسک مربوط به یک
فیلتر میانگین وزن دار

$$\frac{1}{12}$$

1	1	1	1
1	1	1	1
1	1	1	1

ماتریس ماسک غیرمربعی

البته در عمل از فیلترهای غیرمربعی و وزن دار کمتر استفاده می شود.

اعمال فیلترهای میانگین روی یک تصویر



$\frac{1}{9}$

1	1	1
1	1	1
1	1	1



اعمال فیلترهای میانگین روی یک تصویر



$$\frac{1}{25}$$

1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1



اعمال فیلترهای میانگین روی یک تصویر

Filtered Image with Av(3)



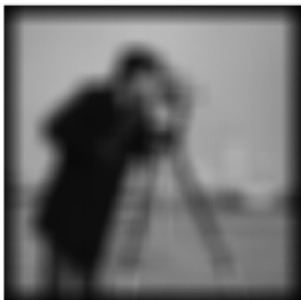
Filtered Image with Av(4)



Filtered Image with Av(5)



Filtered Image with Av(20)



Original Image



Filtered Image with Av(6)



Filtered Image with Av(155)



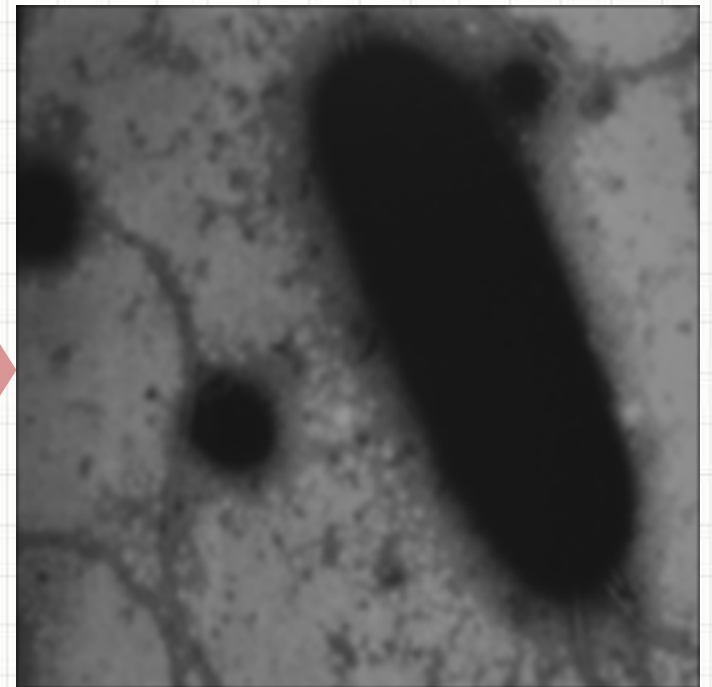
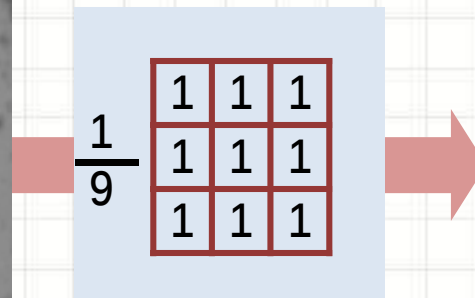
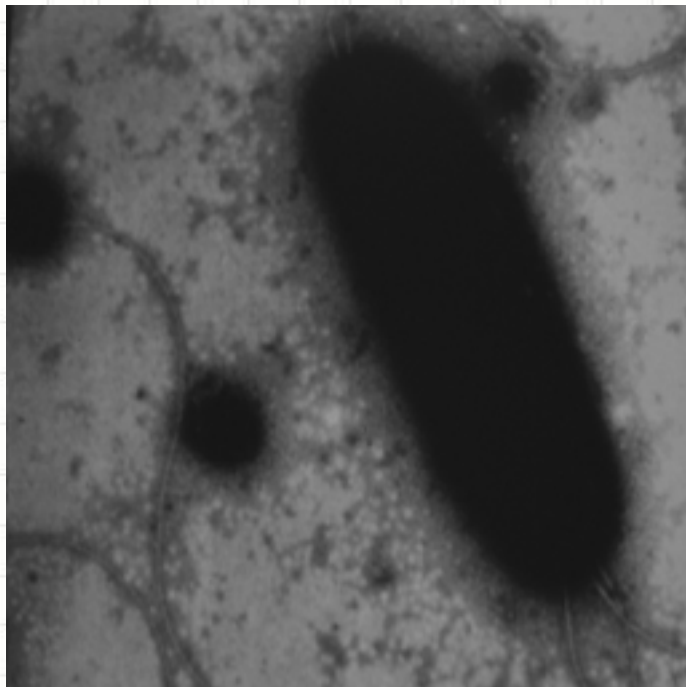
Filtered Image with Av(10)



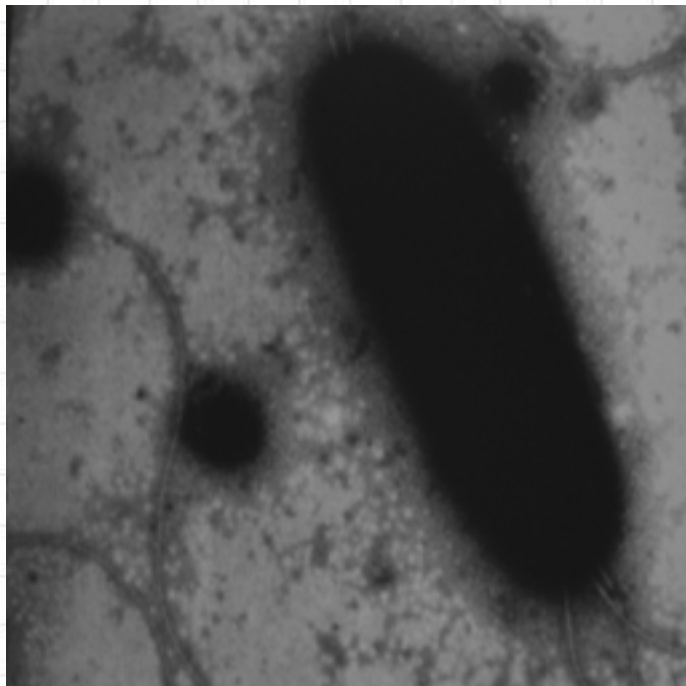
Filtered Image with Av(7)



اعمال فیلترهای میانگین روی یک تصویر

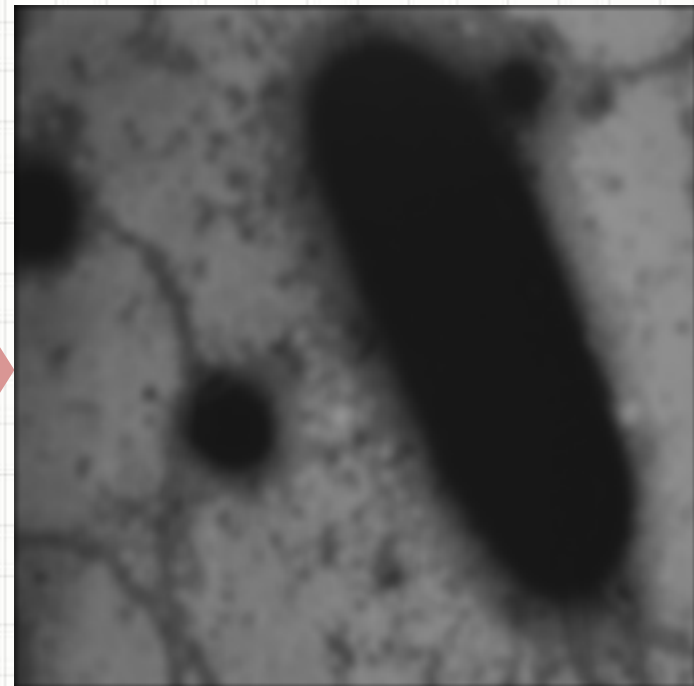


اعمال فیلترهای میانگین روی یک تصویر

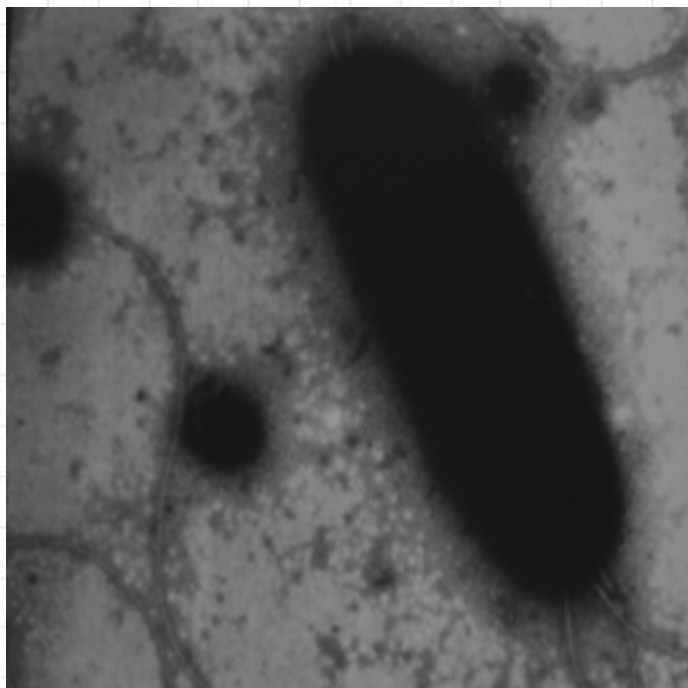


$$\frac{1}{25}$$

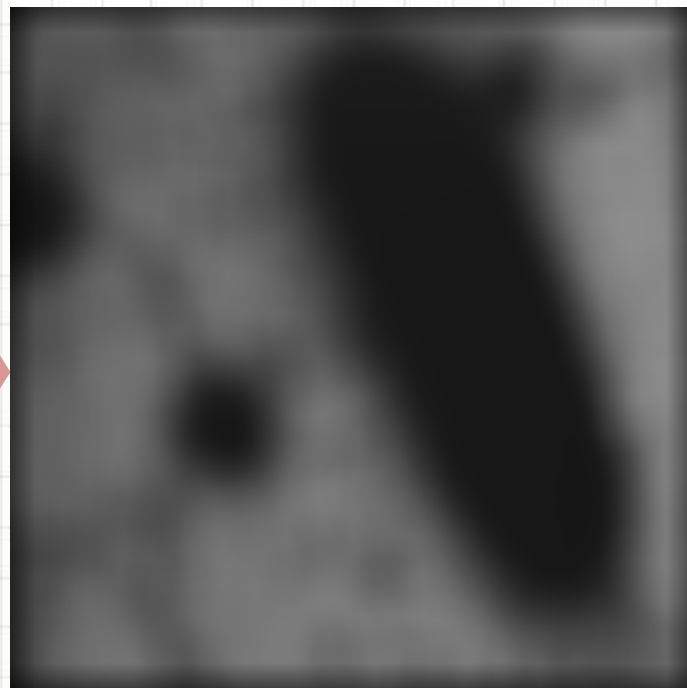
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1



اعمال فیلترهای میانگین روی یک تصویر

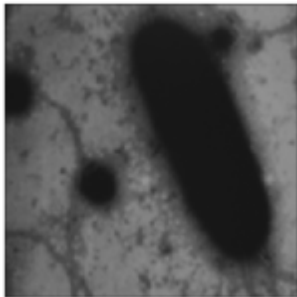


فیلتر میانگین با
ماتریس ماسک
20x20

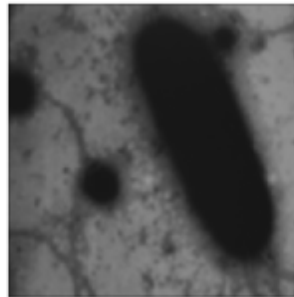


اعمال فیلترهای میانگین روی یک تصویر

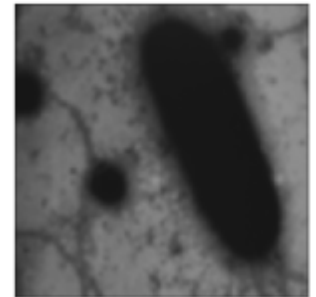
Filtered Image with Av(3)



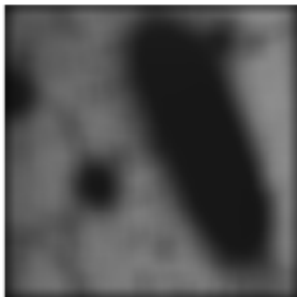
Filtered Image with Av(4)



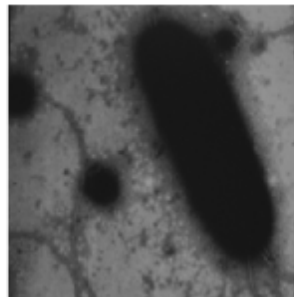
Filtered Image with Av(5)



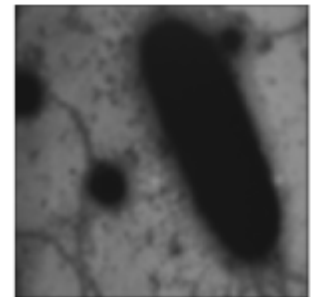
Filtered Image with Av(20)



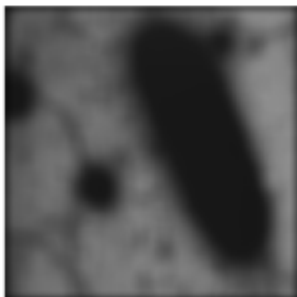
Original Image



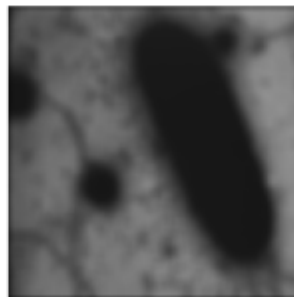
Filtered Image with Av(6)



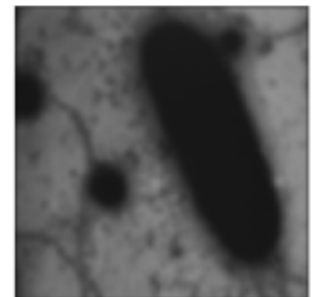
Filtered Image with Av(155)



Filtered Image with Av(10)

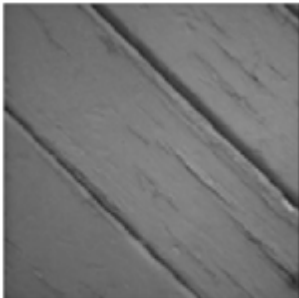


Filtered Image with Av(7)

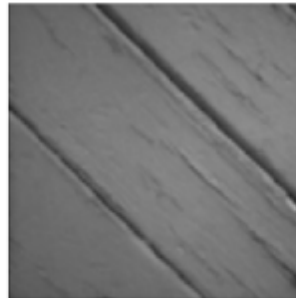


اعمال فیلترهای میانگین روی یک تصویر

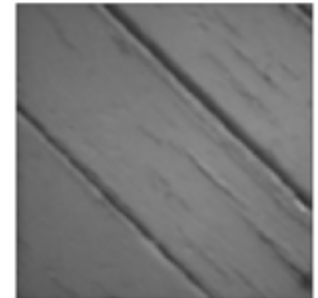
Filtered Image with $A_v(3)$



Filtered Image with $A_v(4)$



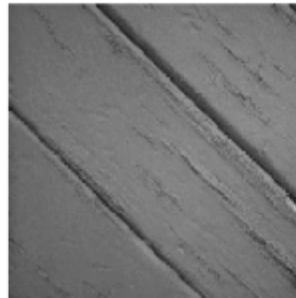
Filtered Image with $A_v(5)$



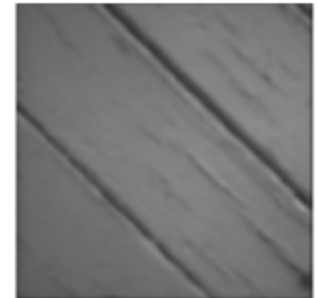
Filtered Image with $A_v(20)$



Original Image



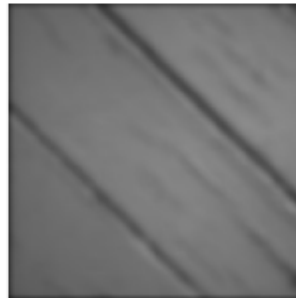
Filtered Image with $A_v(6)$



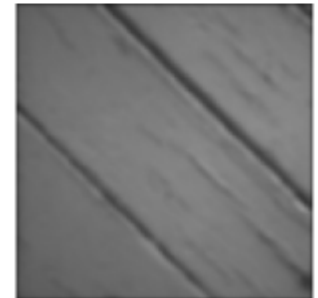
Filtered Image with $A_v(155)$



Filtered Image with $A_v(10)$



Filtered Image with $A_v(7)$



اعمال فیلترهای میانگین روی یک تصویر

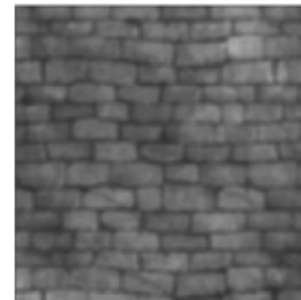
Filtered Image with Av(3)



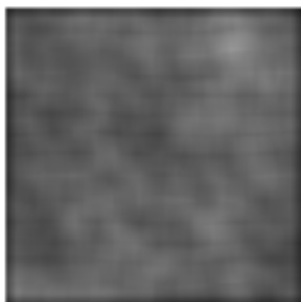
Filtered Image with Av(4)



Filtered Image with Av(5)



Filtered Image with Av(20)



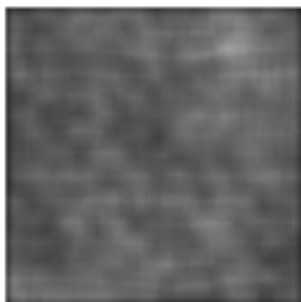
Original Image



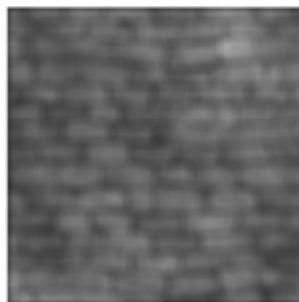
Filtered Image with Av(6)



Filtered Image with Av(155)



Filtered Image with Av(10)

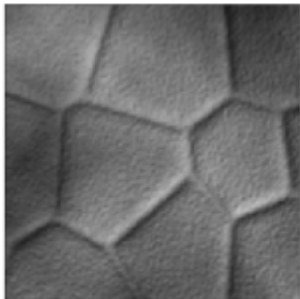


Filtered Image with Av(7)

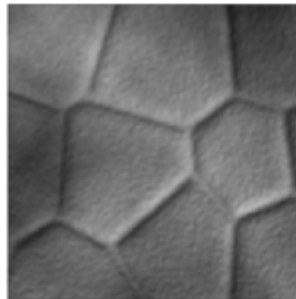


اعمال فیلترهای میانگین روی یک تصویر

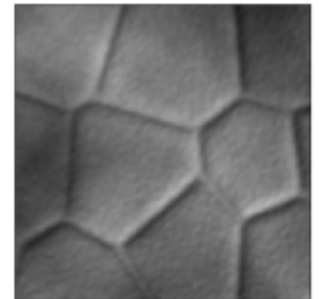
Filtered Image with $A_v(3)$



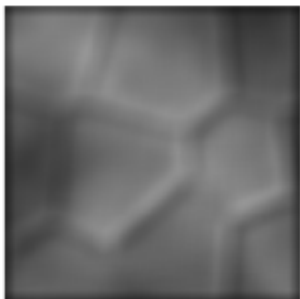
Filtered Image with $A_v(4)$



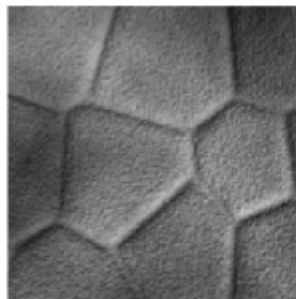
Filtered Image with $A_v(5)$



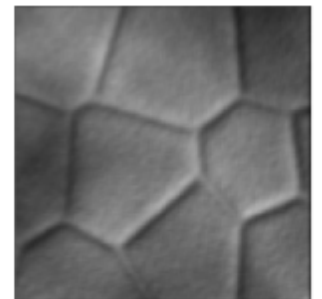
Filtered Image with $A_v(20)$



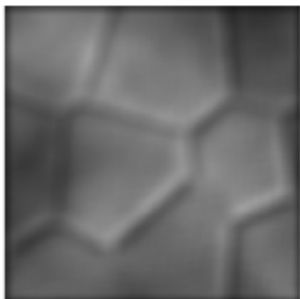
Original Image



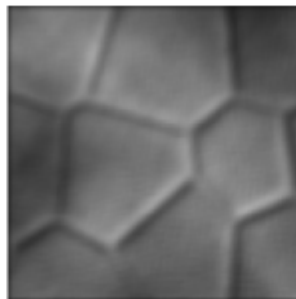
Filtered Image with $A_v(6)$



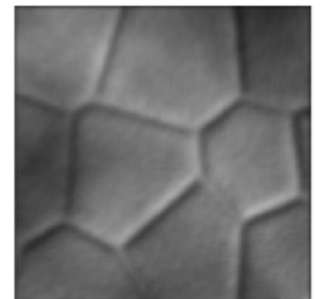
Filtered Image with $A_v(155)$



Filtered Image with $A_v(10)$



Filtered Image with $A_v(7)$



خاصیت فیلترهای میانگین

با اعمال فیلترهای میانگین تفاوت بین شدت پیکسل‌ها کاهش می‌یابد. لذا

- فیلتر میانگین باعث **مات شدن (Blurred)** تصویر می‌شود.
- فیلتر میانگین باعث **حذف جزئیات** می‌شود.
- هرچه ماتریس ماسک بزرگ‌تر باشد، آن‌گاه درجه مات شدن تصویر و حذف جزئیات تصویر بیشتر خواهد بود.

فیلتر گوسین (Gaussian Filter)

فیلتر گوسی (Gaussian Filter) – مقدمه

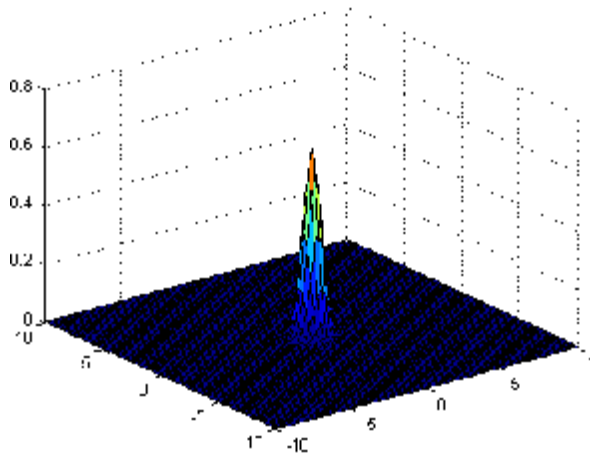
- فیلتر گوسی یک نوع فیلتر **میانگین وزندار** می باشد که مولفه های ماتریس ماسک آن از **تابع گوس** اقتباس می گردد.
- در ماسک فیلتر گوسی، بیشترین ارزش به پیکسل اصلی (مرکزی) داده می شود و پیکسل های همسایه وزنی متناسب با فاصله شان تا پیکسل مرکزی به خود اختصاص می دهند. هرچه فاصله دورتر می شود مقدار وزن نیز کوچک تر می شود. این نکته ای است که باعث می شود لبه ها و مرزها بهتر حفظ گردد.

تابع گاوسی (Gaussian Function)

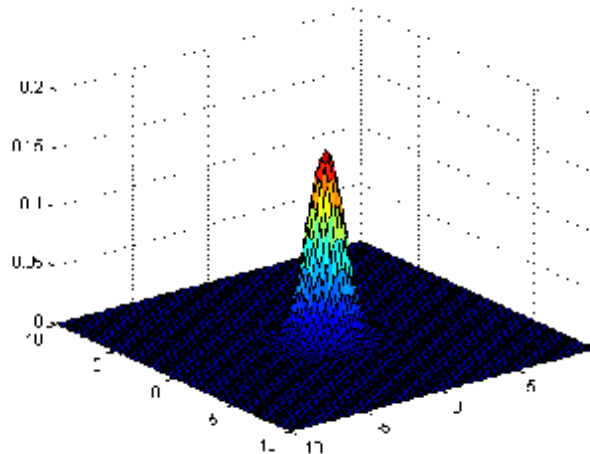
تابع گاوسی دومتغیره به صورت زیر می باشد

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

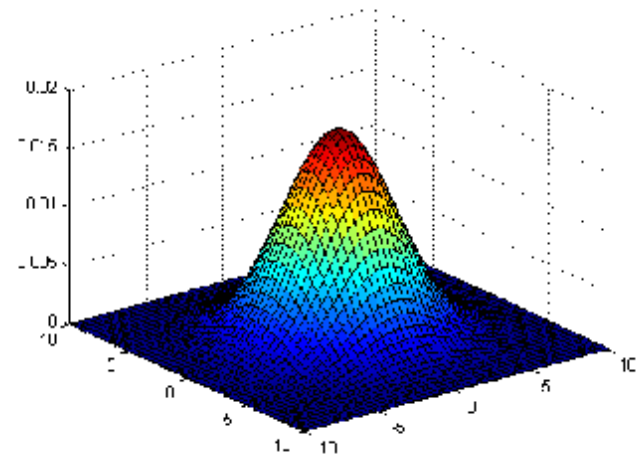
که σ یک عدد حقیقی مثبت است. این تابع یک تابع پیوسته می باشد و در همه جا مثبت است. تابع گاوسی ماکزیمم خود را در $(0,0)$ اختیار می کند و در بی نهایت به صفر میل می کند. هرچه مقدار σ بزرگ تر باشد، آن گاه تابع با سرعت کمتری به صفر میل می کند.



$$\sigma = 0.5$$



$$\sigma = 1$$



$$\sigma = 3$$

ماتریس ماسک در فیلتر گوسی

فیلتر گوسی از تابع گوسی ساخته می‌شود. تابع گوسی یک تابع پیوسته است، اما فیلتر ماهیتی گسسته دارد. لذا یک **تقریب گسسته** از تابع گوسی مورد استفاده قرار می‌گیرد. **رویکردهای** متفاوتی برای ساخت این تقریب گسسته در نظر گرفته می‌شود که ماهیتا با یکدیگر معادلند.

در فیلتر گوسی ماتریس ماسک $(2k + 1) \times (2k + 1)$ تایی به صورت زیر است.

$$\frac{1}{S} \begin{bmatrix} G(-k, -k) & \cdots & G(-k, 0) & \cdots & G(-k, +k) \\ \vdots & & \vdots & & \vdots \\ G(0, -k) & \cdots & G(0, 0) & \cdots & G(0, +k) \\ \vdots & & \vdots & & \vdots \\ G(+k, -k) & \cdots & G(+k, 0) & \cdots & G(+k, +k) \end{bmatrix}$$

به طوری که S عددی است که باعث می‌شود که مجموع مولفه‌های ماتریس ماسک H برابر یک گردد. یعنی $S = \sum_{i=-k}^k \sum_{j=1}^k G(i, j)$.

ماتریس ماسک در فیلتر گوسی

در این رویکرد که در **matlab** نیز استفاده می‌شود، اگر H ماتریس ماسک با بعد دلخواه n باشد، آن‌گاه مولفه سطر i ام و ستون j ام ماتریس ماسک به صورت زیر می‌باشد

$$h(i, j) = \frac{1}{S} G \left(i - \frac{n+1}{2}, j - \frac{n+1}{2} \right)$$

به طوری که S عددی است که باعث می‌شود که مجموع مولفه‌های ماتریس ماسک H برابر یک گردد. یعنی

$$S = \sum_{i=1}^n \sum_{j=1}^n G \left(i - \frac{n+1}{2}, j - \frac{n+1}{2} \right)$$

توجه شود که امکان تعریف ماتریس ماسک با بعد زوج نیز وجود دارد.

فیلتر گوسی (Gaussian Filter)

ماتریس ماسک 5 در 5 فیلتر گوسی با $\sigma=1$

0.0030	0.0133	0.0219	0.0133	0.0030
0.0133	0.0596	0.0983	0.0596	0.0133
0.0219	0.0983	0.1621	0.0983	0.0219
0.0133	0.0596	0.0983	0.0596	0.0133
0.0030	0.0133	0.0219	0.0133	0.0030

تقارن در ماتریس ماسک به خوبی دیده می شود.

اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
3x3
با
 $\sigma=1.0$



اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
5x5
با
 $\sigma=1.0$



اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
20x20
با
 $\sigma=1.0$



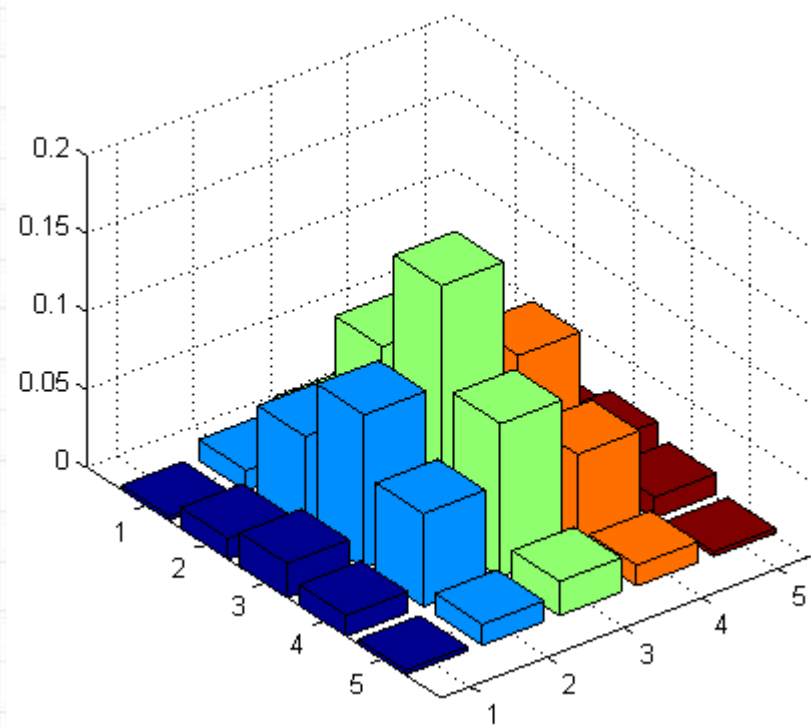
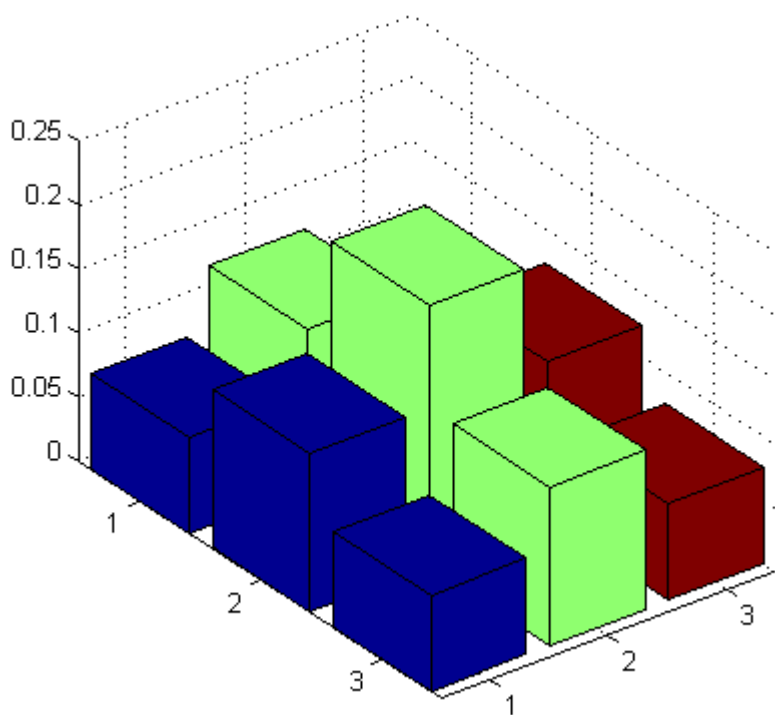
اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
30x30
با
 $\sigma=1.0$

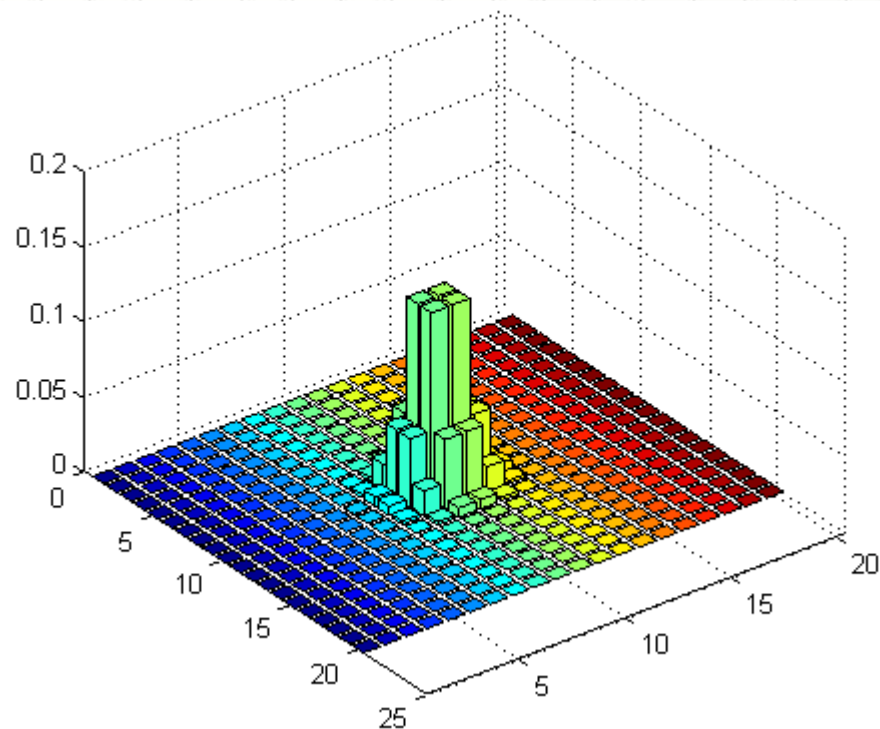
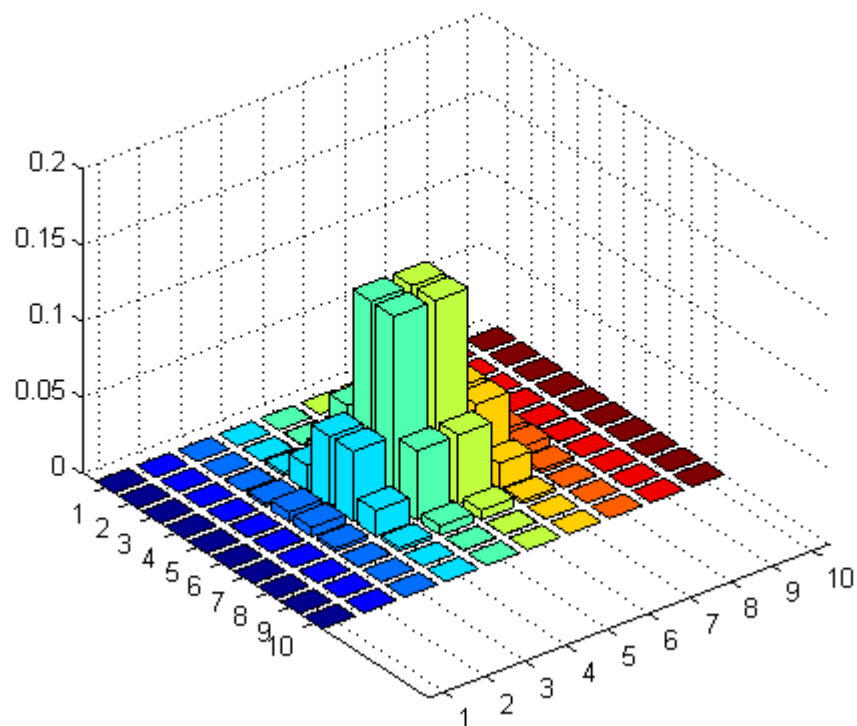


اعمال فیلترهای گوسی روی یک تصویر



اندازه مولفه‌های ماتریس ماسک گوسی با $\sigma=1$

اعمال فیلترهای گوسی روی یک تصویر



اندازه مولفه‌های ماتریس ماسک گوسی با $\sigma=1$

اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
 3×3
با
 $\sigma = 10$



اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
5x5
با
 $\sigma=10$



اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
10x10
با
 $\sigma=10$



اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
20x20
با
 $\sigma=10$



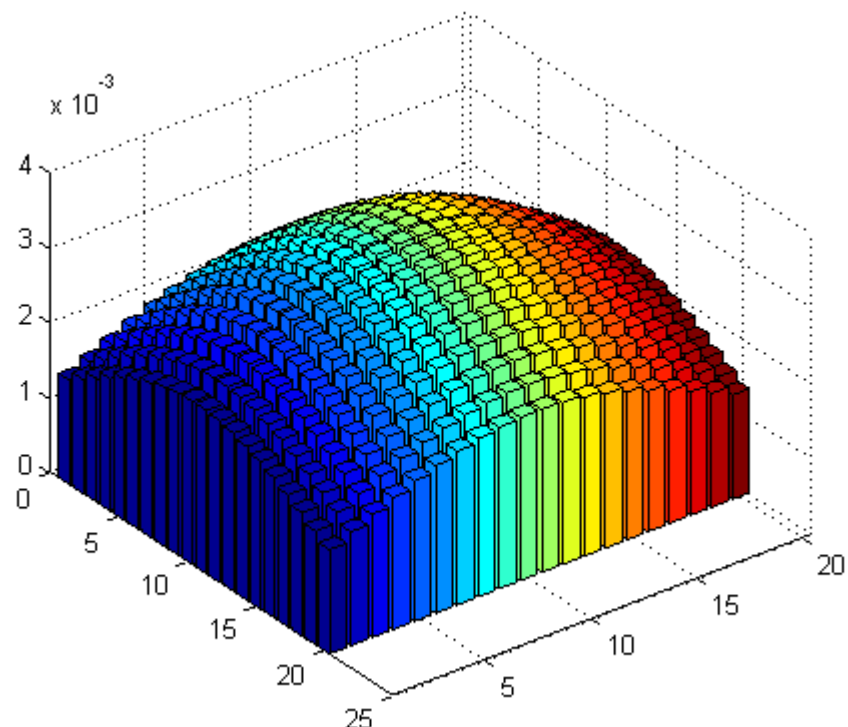
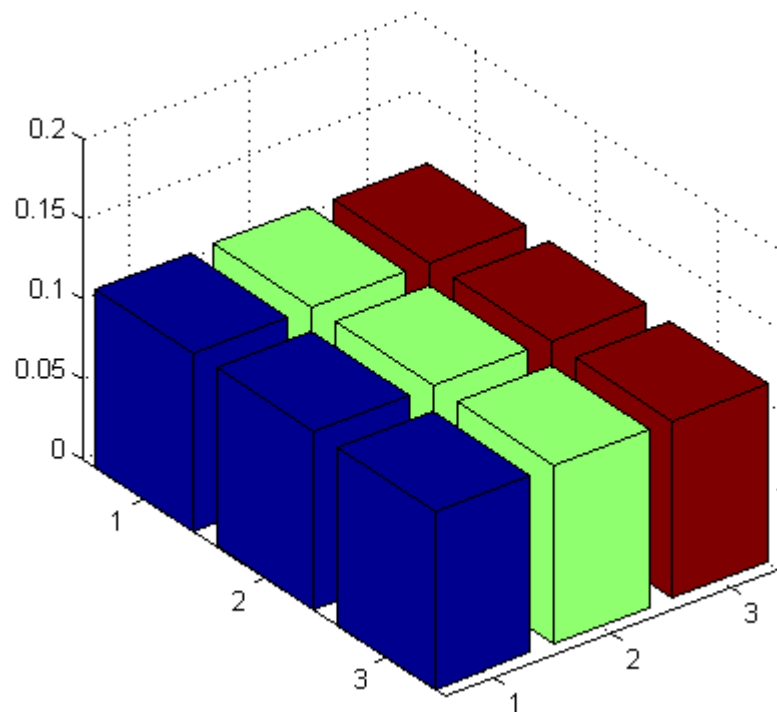
اعمال فیلترهای گوسی روی یک تصویر



فیلتر گوسی
30x30
با
sigma=10

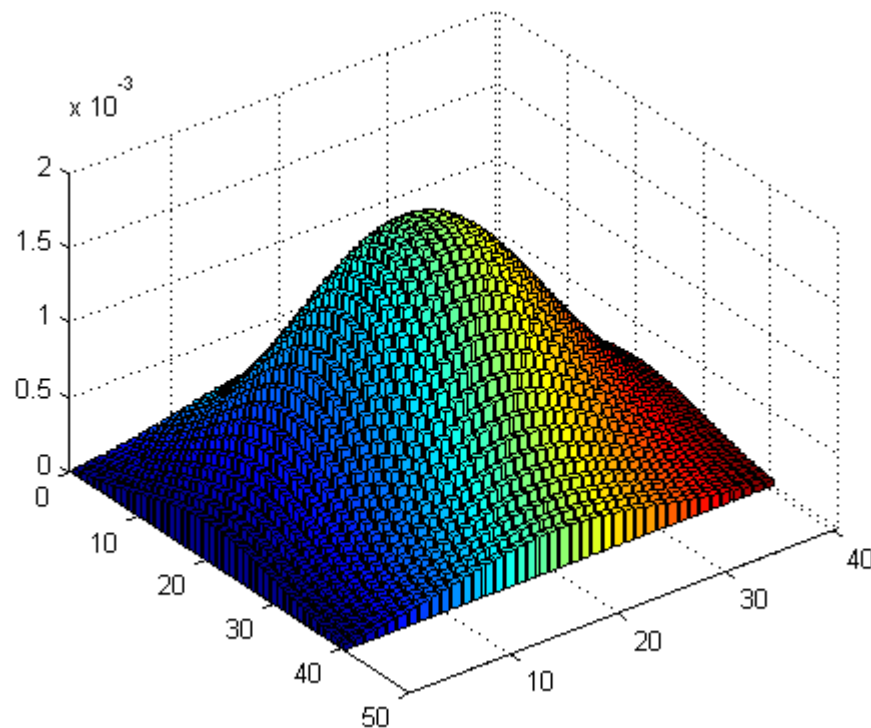
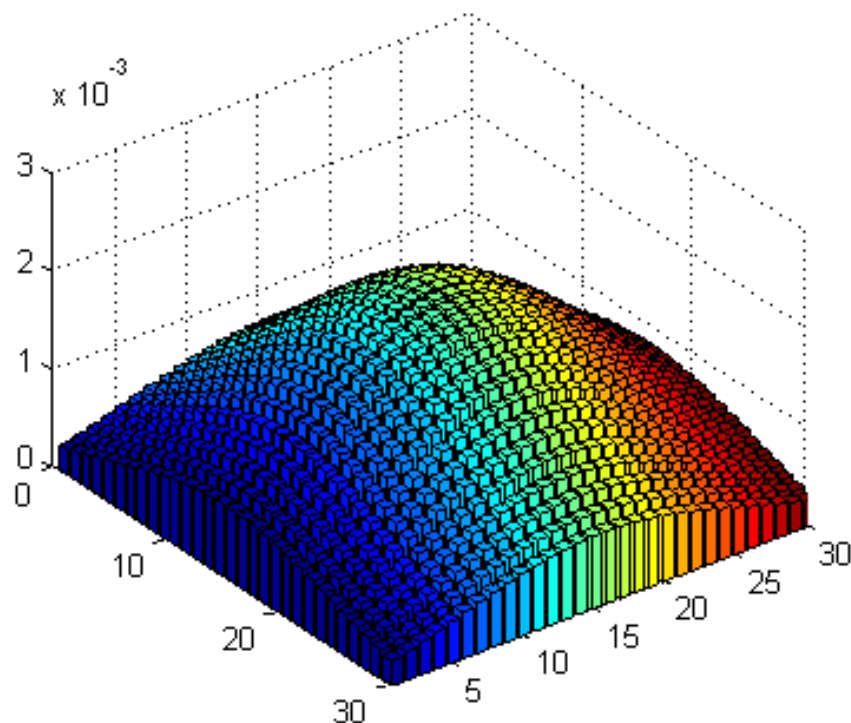


اعمال فیلترهای گوسی روی یک تصویر



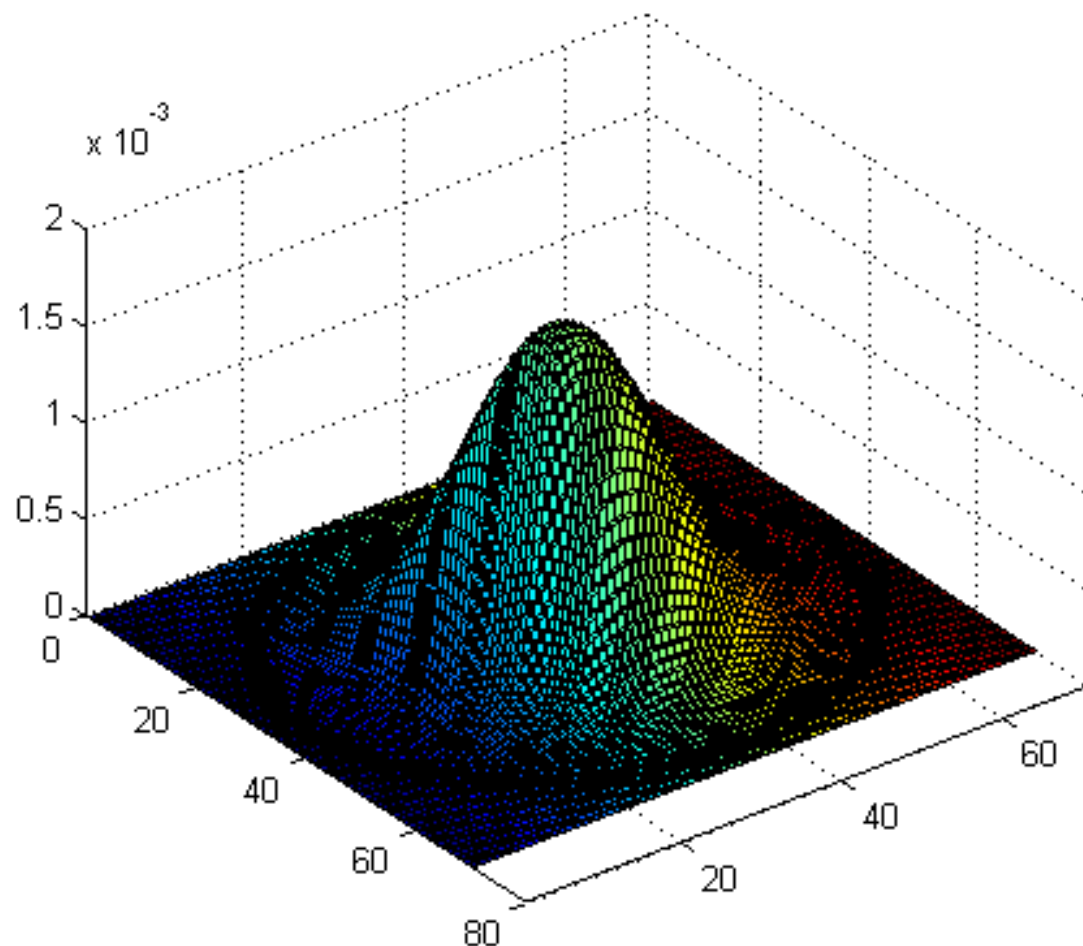
اندازه مولفه‌های ماتریس ماسک گوسی با $\sigma=10$

اعمال فیلترهای گوسی روی یک تصویر



اندازه مولفه‌های ماتریس ماسک گوسی با $\sigma=10$

اعمال فیلترهای گوسی روی یک تصویر



اندازه مولفه‌های ماتریس ماسک گوسی با $\sigma=10$

اعمال فیلترهای گوسی روی یک تصویر

در ماتریس ماسک فیلتر گوسی با پارامتر σ ، بهتر است که اندازه ماتریس ماسک برابر

$$2k + 1 = 2 \lceil 3\sigma \rceil$$

انتخاب شود. چرا که در این حالت اندازه تمامی مولفه‌ها به اندازه کافی بزرگ است و لذا فیلتر در کردن با این ماتریس تمامی مولفه‌ها تاثیرگذار خواهند بود. اگر اندازه ماتریس ماسک بزرگ‌تر از مقدار پیشنهادی فوق باشد، آن‌گاه مولفه‌های کناری ماتریس بسیار کوچک و یا هستند و در فرایند فیلتر کردن تاثیر بسیار ناچیزی دارند. در این شرایط فقط حجم محاسبات بالا می‌رود و همچنین کناره‌های تصویر به خاطر لایه‌گذاری بیشتر به هم می‌ریزد

فیلترهای آماری

فیلتر ماکزیمم (Max Filter)

فیلتر مینیمم (Min Filter)

فیلتر میانه (Median Filter)

فیلتر ماکزیمم (Max Filter)

فیلتر ماکزیمم یک فیلتر غیرخطی است، که یک همسایگی اطراف هر پیکسل را در نظر می‌گیرد و **ماکزیمم** اعداد موجود در آن همسایگی را به عنوان تبدیل آن پیکسل در نظر می‌گیرد.

ماسک ندارد
حایک دارد

12	14	18	21	34	12	5	2
13	15	17	24	45	78	90	99
34	45	48	60	65	70	90	101
33	37	40	49	60	64	71	89
41	46	48	60	34	70	83	90
50	67	90	90	70	77	94	102
55	73	92	85	81	98	121	111

		60					

15, 17, 24, 45, 48, 60, 37, 40, 49

Max

60

فیلتر مینیمم (Min Filter)

فیلتر مینیمم یک فیلتر **غیرخطی** است، که یک همسایگی اطراف هر پیکسل را در نظر می‌گیرد و **مینیمم** اعداد موجود در آن همسایگی را به عنوان تبدیل آن پیکسل در نظر می‌گیرد.

12	14	18	21	34	12	5	2
13	15	17	24	45	78	90	99
34	45	48	60	65	70	90	101
33	37	40	49	60	64	71	89
41	46	48	60	34	70	83	90
50	67	90	90	70	77	94	102
55	73	92	85	81	98	121	111

		15					

15, 17, 24, 45, 48, 60, 37, 40, 49

Min

15

فیلتر میانه (Median Filter)

فیلتر میانگین یک فیلتر **غیرخطی** است، که یک همسایگی اطراف پیکسل را در نظر می‌گیرد و میانه اعداد موجود در آن همسایگی را به عنوان تبدیل آن پیکسل در نظر می‌گیرد.

12	14	18	21	34	12	5	2
13	15	17	24	45	78	90	99
34	45	48	60	65	70	90	101
33	37	40	49	60	64	71	89
41	46	48	60	34	70	83	90
50	67	90	90	70	77	94	102
55	73	92	85	81	98	121	111

فیلتر میانه (Median Filter) ادامه...

فیلتر میانگین یک فیلتر **غیرخطی** است، که یک همسایگی اطراف پیکسل را در نظر می‌گیرد و میانه اعداد موجود در آن همسایگی را به عنوان تبدیل آن پیکسل در نظر می‌گیرد.

12	14	18	21	34	12	5	2
13	15	17	24	45	78	90	99
34	45	48	60	65	70	90	101
33	37	40	49	60	64	71	89
41	46	48	60	34	70	83	90
50	67	90	90	70	77	94	102
55	73	92	85	81	98	121	111

		40					

15, 17, 24, 45, 48, 60, 37, 40, 49

SORT

15, 17, 24, 37, 40, 45, 48, 49, 60

اعمال فیلتر میانه روی یک تصویر



فیلتر میانه با
همسایگی
3x3



اعمال فیلتر میانه روی یک تصویر



فیلتر میانه با
همسایگی
5x5



اعمال فیلتر میانه روی یک تصویر



فیلتر میانه با
همسایگی
7x7



اعمال فیلتر میانه روی یک تصویر



فیلتر میانه با
همسایگی
9x9



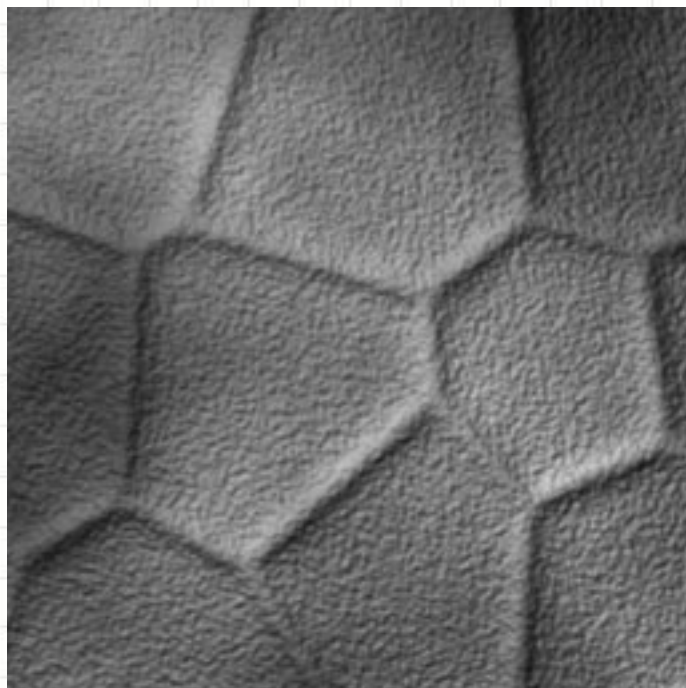
اعمال فیلتر میانه روی یک تصویر



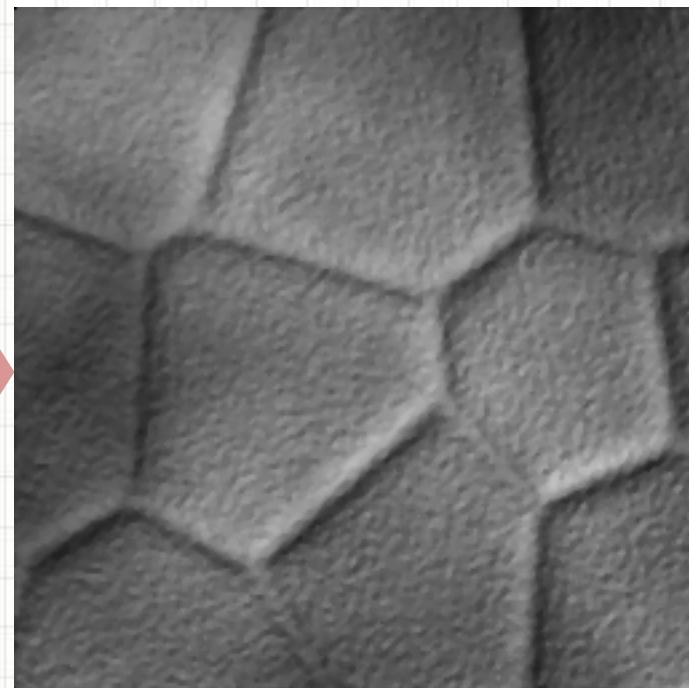
فیلتر میانه با
همسایگی
21x21



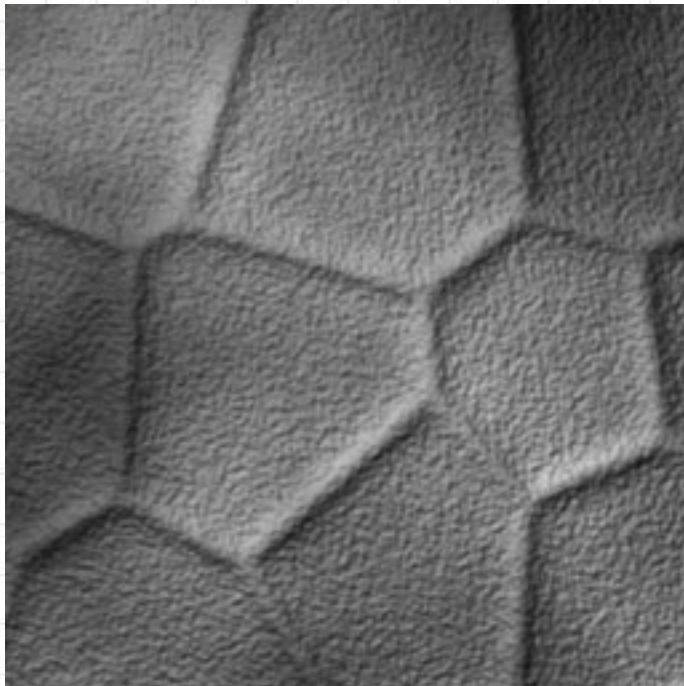
اعمال فیلتر میانه روی یک تصویر



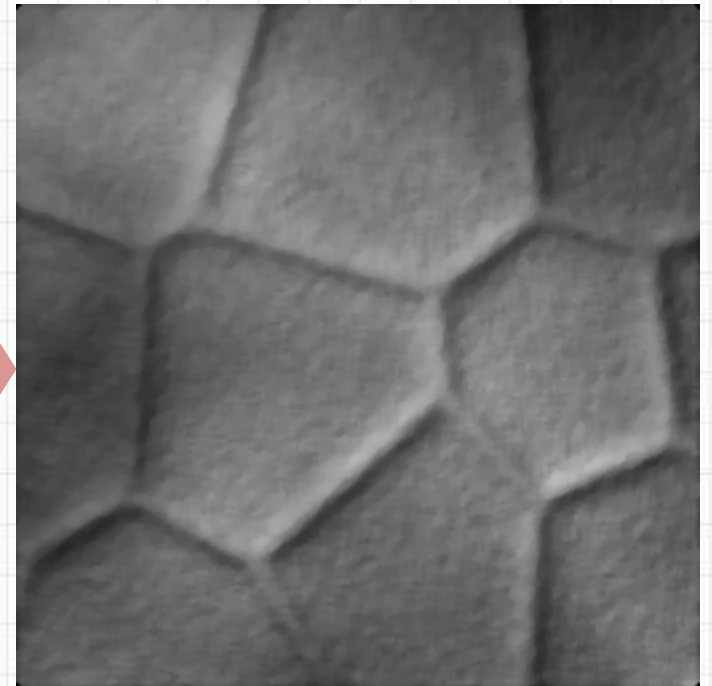
فیلتر میانه با
همسایگی
 3×3



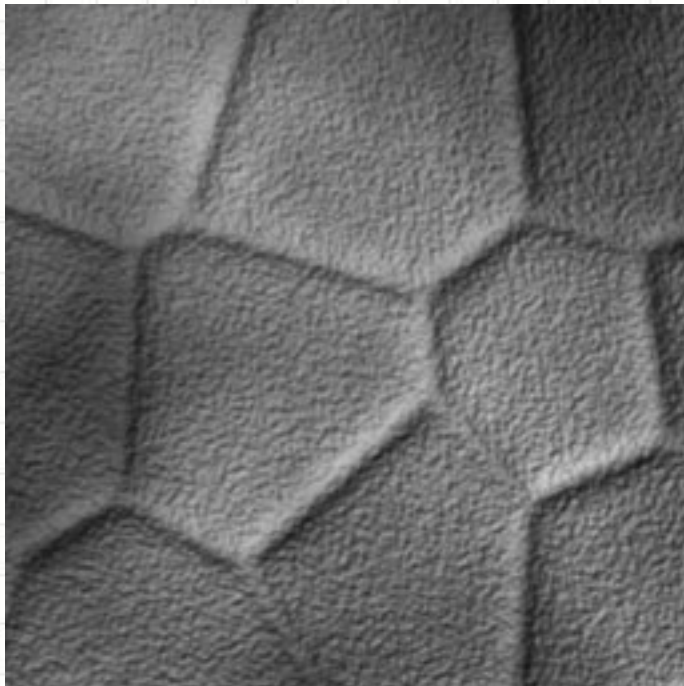
اعمال فیلتر میانه روی یک تصویر



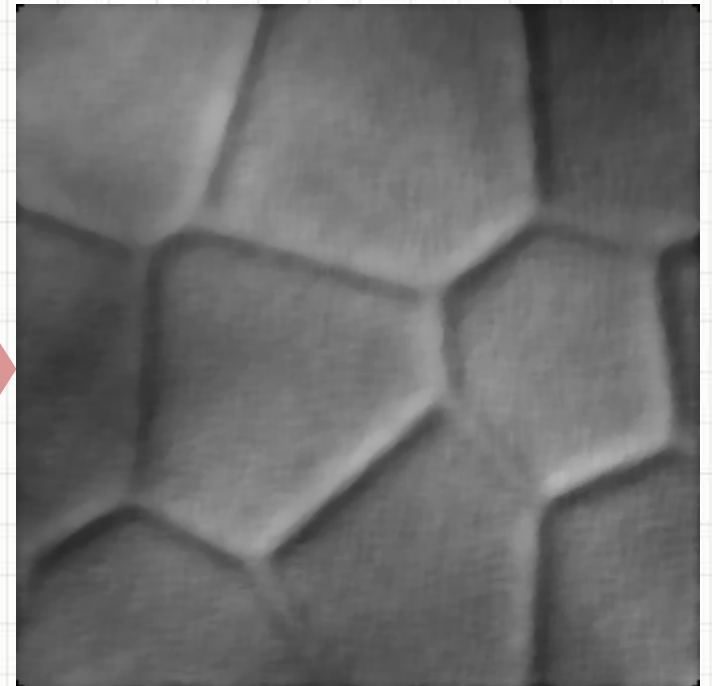
فیلتر میانه با
همسایگی
5x5



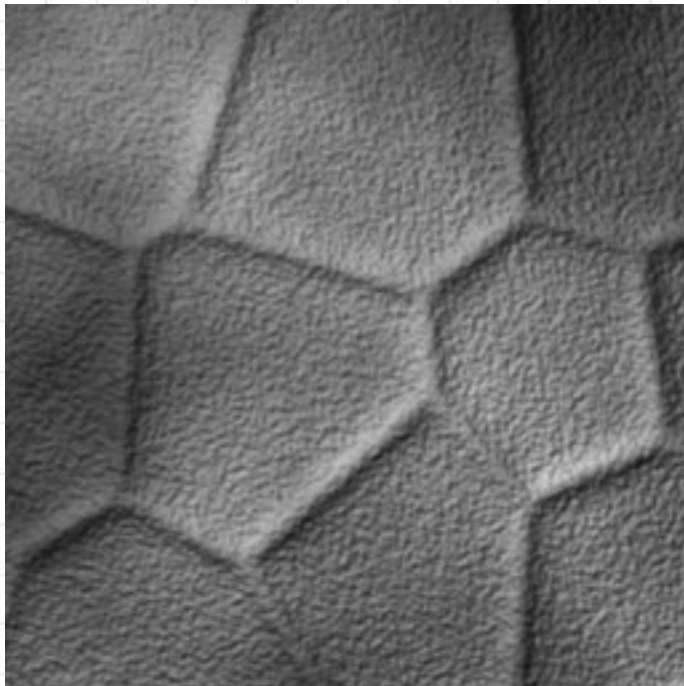
اعمال فیلتر میانه روی یک تصویر



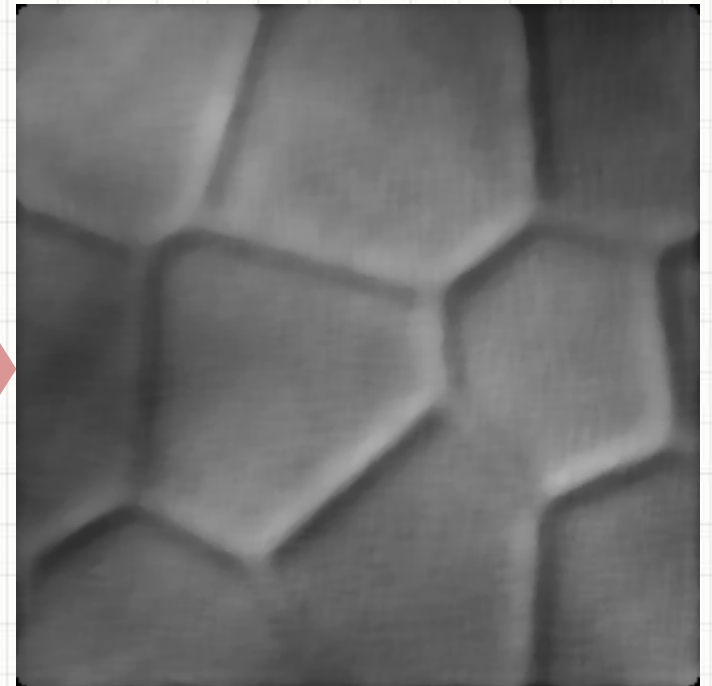
فیلتر میانه با
همسایگی
 7×7



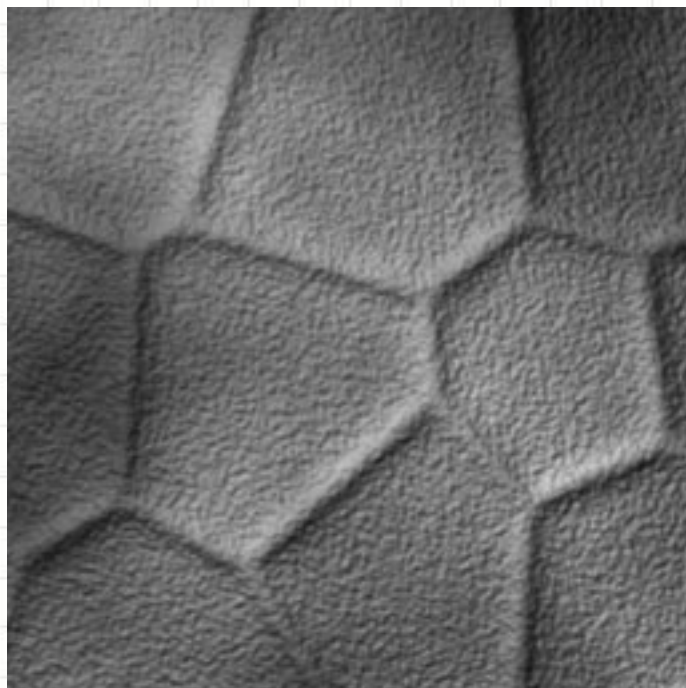
اعمال فیلتر میانه روی یک تصویر



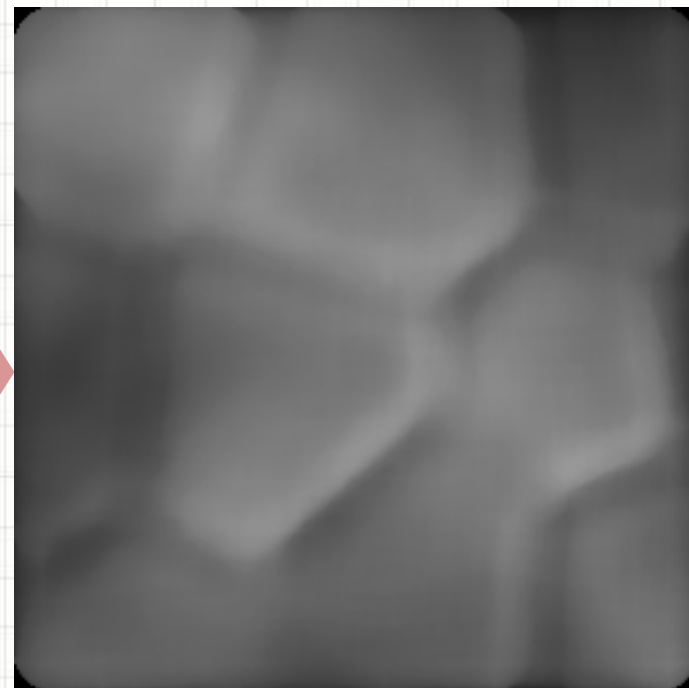
فیلتر میانه با
همسایگی
9x9



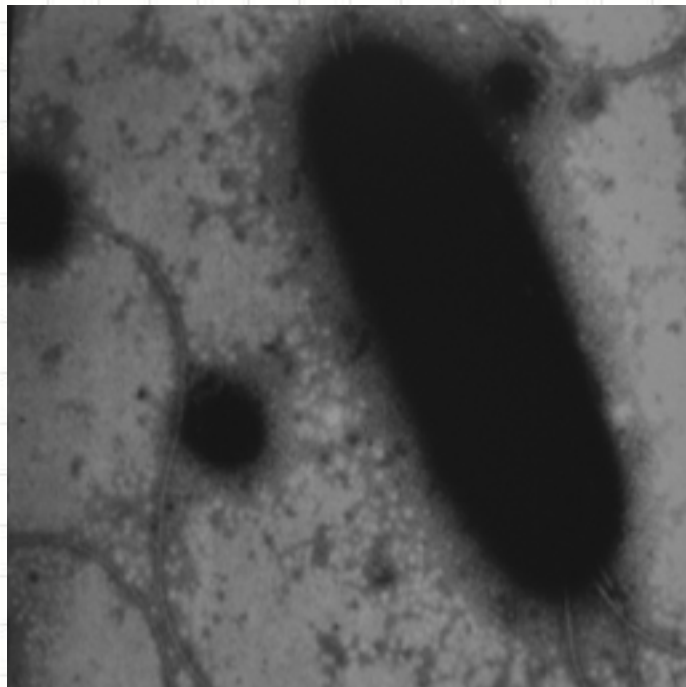
اعمال فیلتر میانه روی یک تصویر



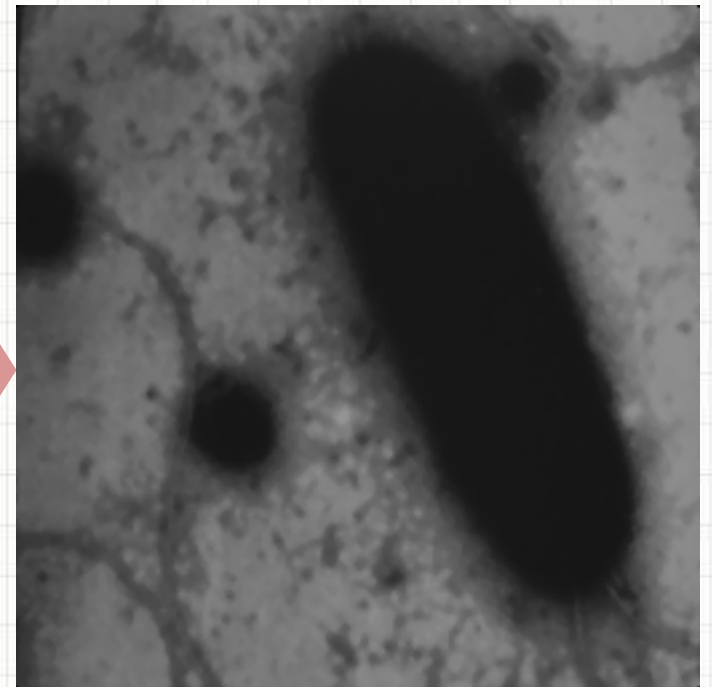
فیلتر میانه با
همسایگی
21x21



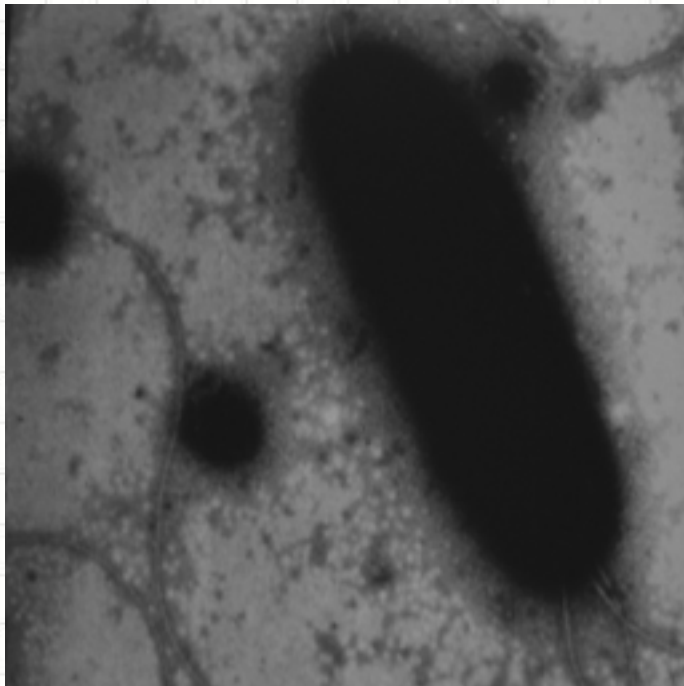
اعمال فیلتر میانه روی یک تصویر



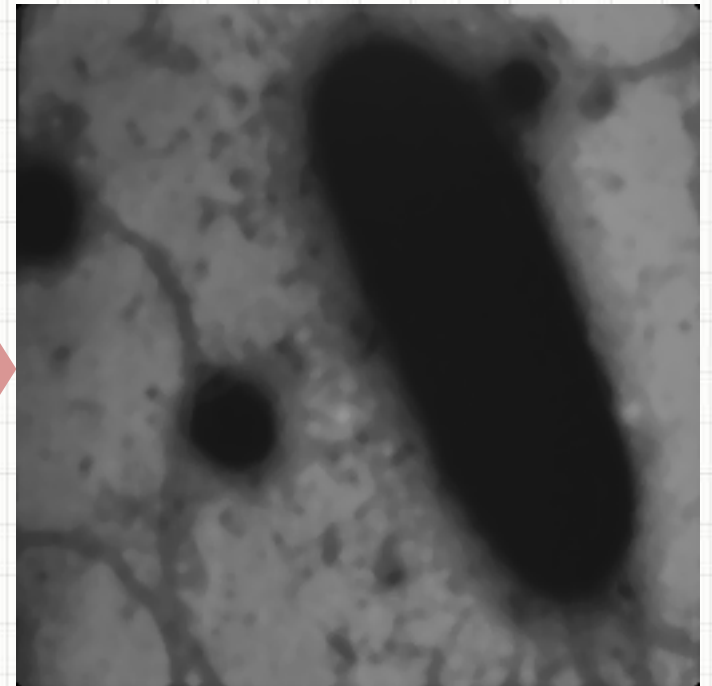
فیلتر میانه با
همسایگی
 3×3



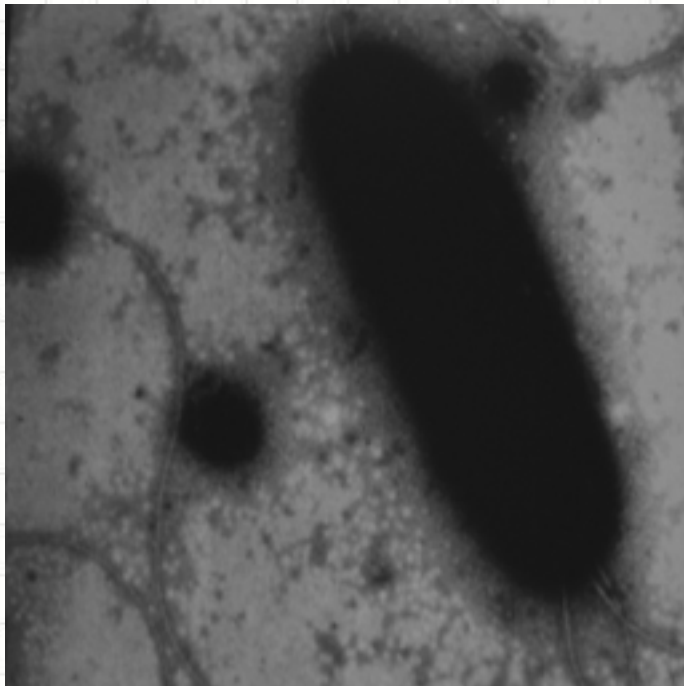
اعمال فیلتر میانه روی یک تصویر



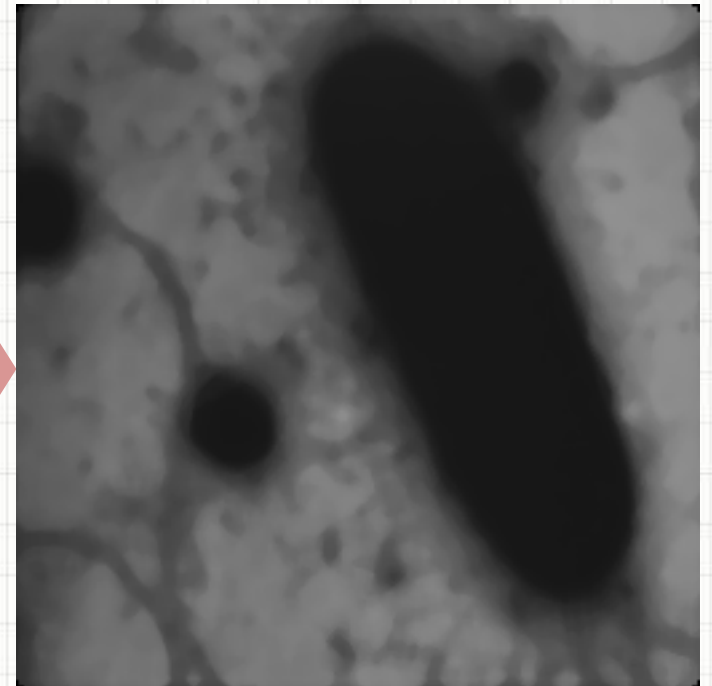
فیلتر میانه با
همسایگی
5x5



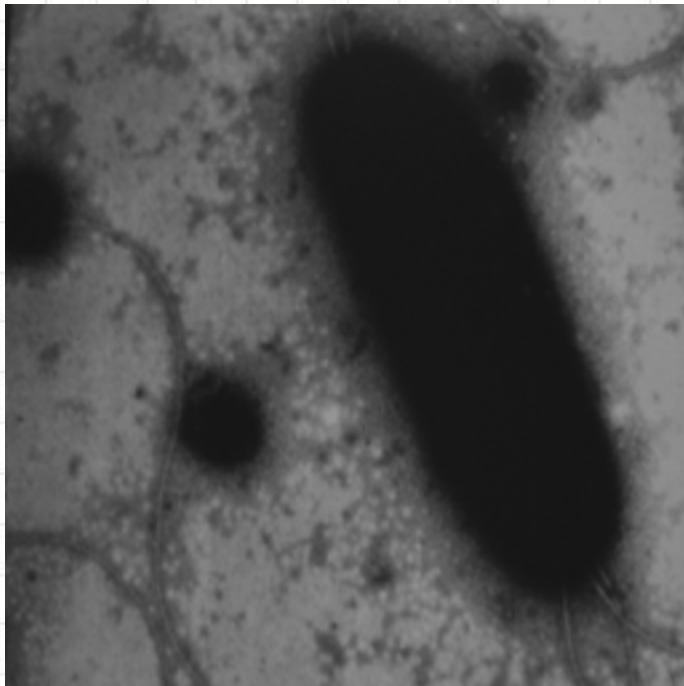
اعمال فیلتر میانه روی یک تصویر



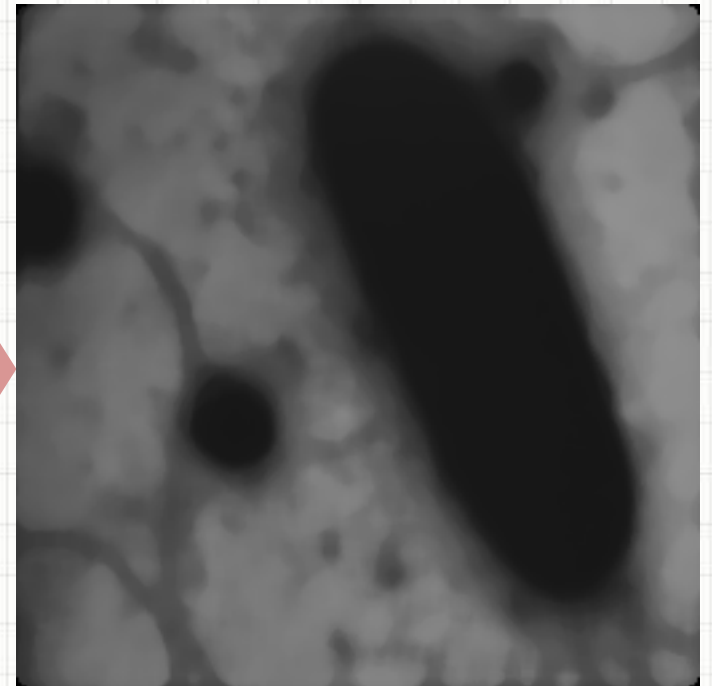
فیلتر میانه با
همسایگی
7x7



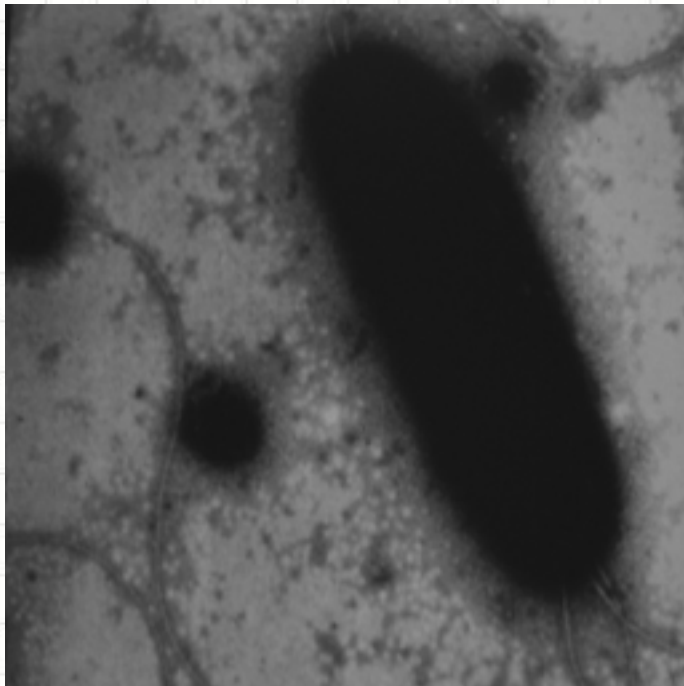
اعمال فیلتر میانه روی یک تصویر



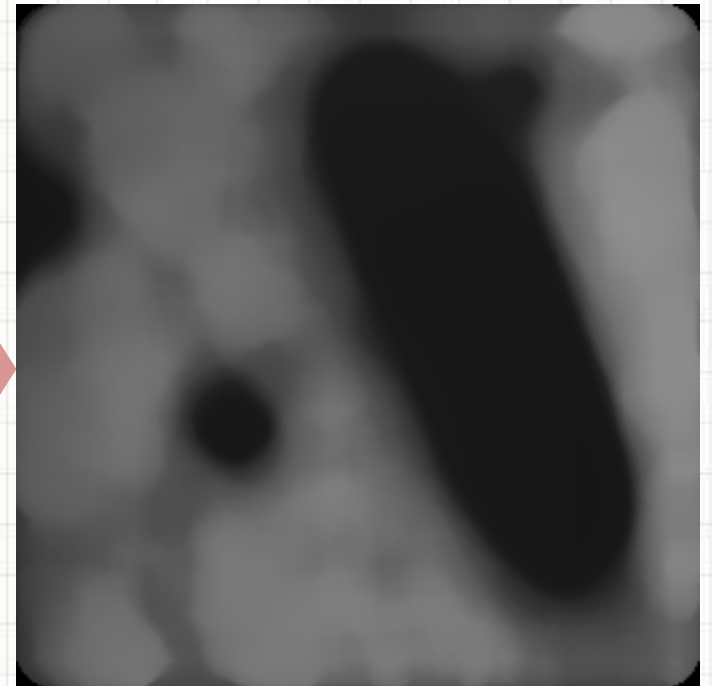
فیلتر میانه با
همسایگی
9x9



اعمال فیلتر میانه روی یک تصویر

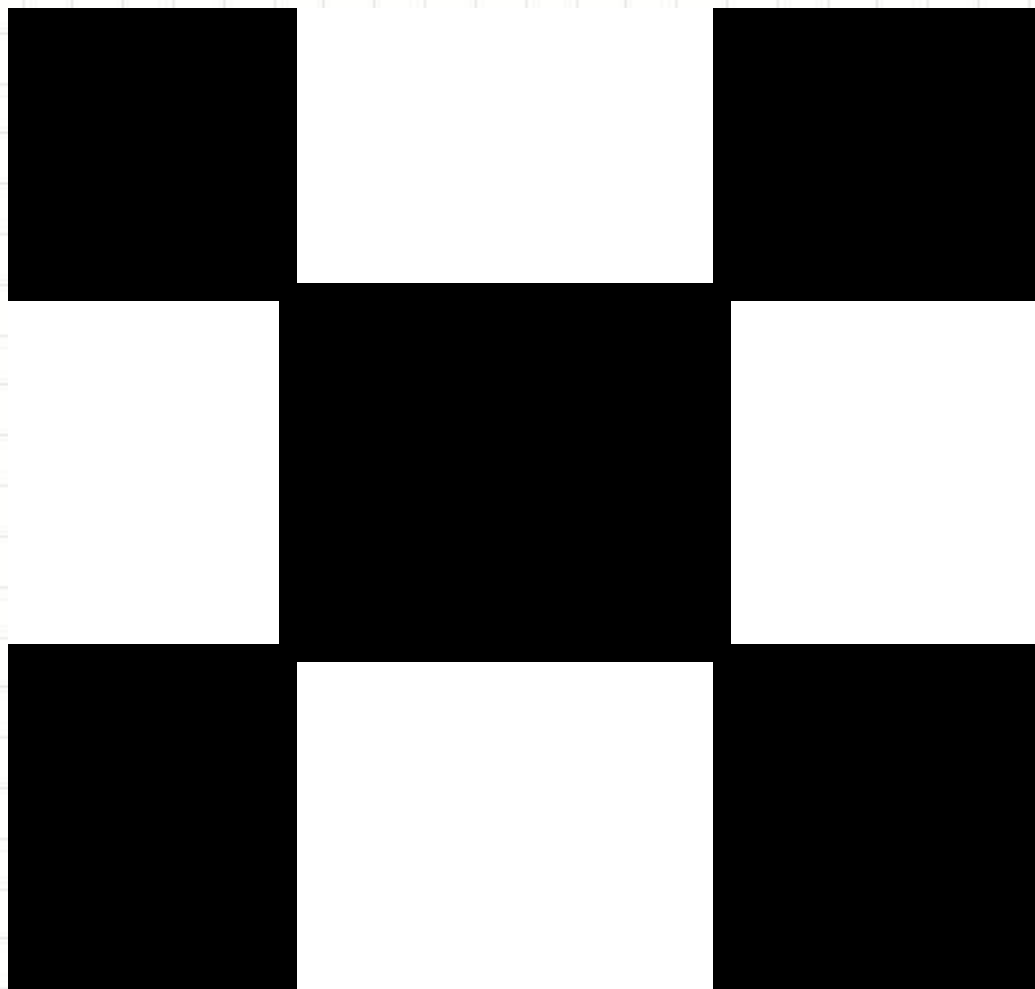


فیلتر میانه با
همسایگی
21x21



مقایسه فیلترهای ملایم کننده

مقایسه فیلترها در حفظ لبه‌ها

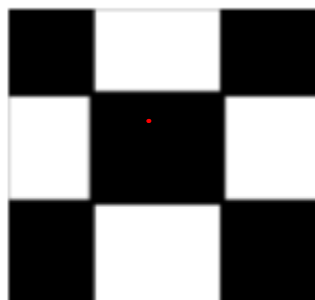


مقایسه فیلترها در حفظ لبه‌ها

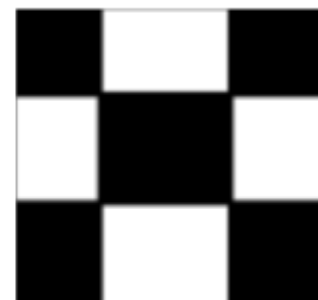
Filtered Image with $\Delta v(3)$



Filtered Image with $\Delta v(4)$



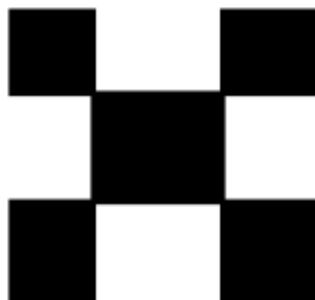
Filtered Image with $\Delta v(5)$



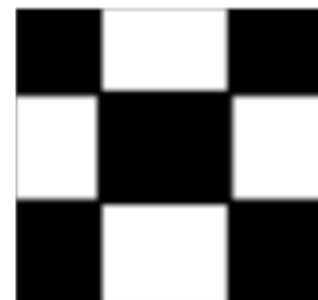
Filtered Image with $\Delta v(20)$



Original Image



Filtered Image with $\Delta v(6)$



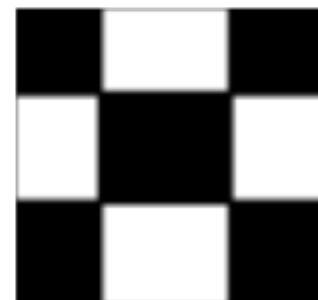
Filtered Image with $\Delta v(155)$



Filtered Image with $\Delta v(10)$

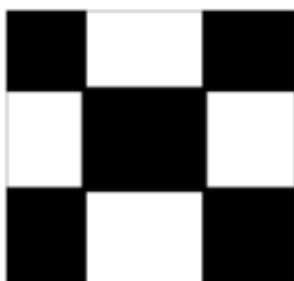


Filtered Image with $\Delta v(7)$



مقایسه فیلترها در حفظ لبه‌ها

Filtered Image with Gaussian($n=3$, $\sigma=20$)



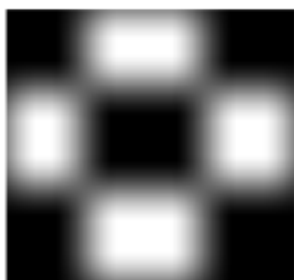
Filtered Image with Gaussian($n=5$, $\sigma=20$)



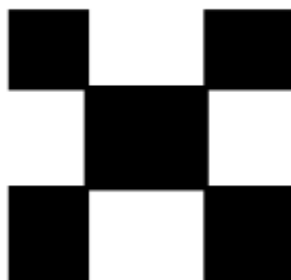
Filtered Image with Gaussian($n=15$, $\sigma=20$)



Filtered Image with Gaussian($n=60$, $\sigma=20$)



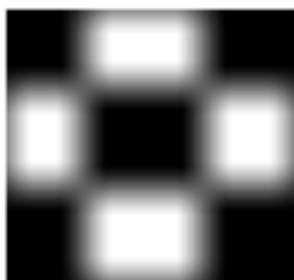
Original Image



Filtered Image with Gaussian($n=20$, $\sigma=20$)



Filtered Image with Gaussian($n=50$, $\sigma=20$)



Filtered Image with Gaussian($n=40$, $\sigma=20$)

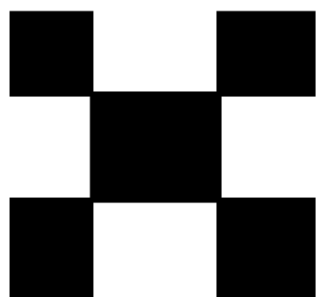


Filtered Image with Gaussian($n=30$, $\sigma=20$)

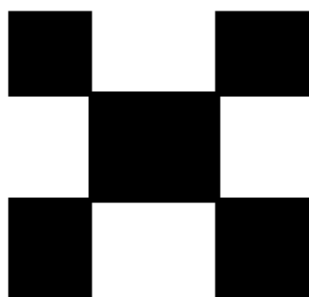


مقایسه فیلترها در حفظ لبه‌ها

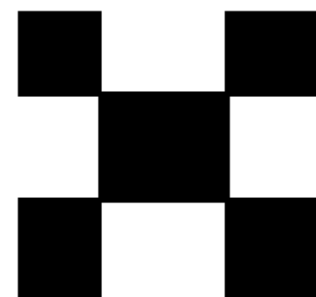
Filtered Image with Median(3x3)



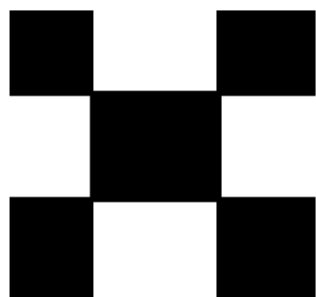
Filtered Image with Median(4x4)



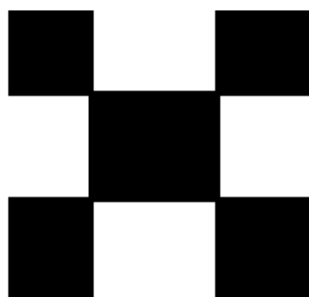
Filtered Image with Median(5x5)



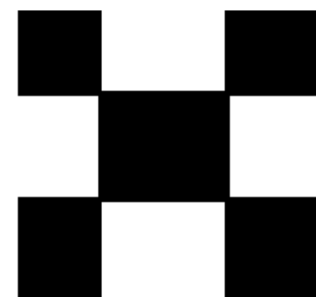
Filtered Image with Median(60x60)



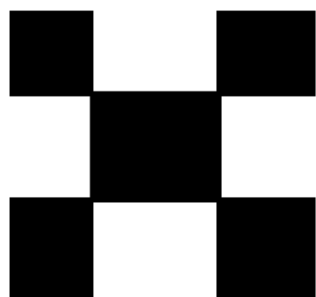
Original Image



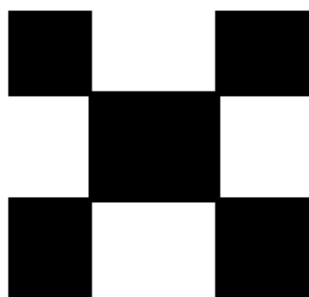
Filtered Image with Median(6x6)



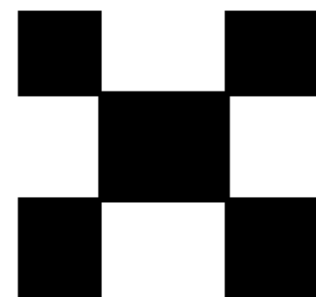
Filtered Image with Median(15x15)



Filtered Image with Median(10x10)



Filtered Image with Median(7x7)





فیلترهای لبه‌یاب و تیزکننده

Edge Detector & Sharpening Filter

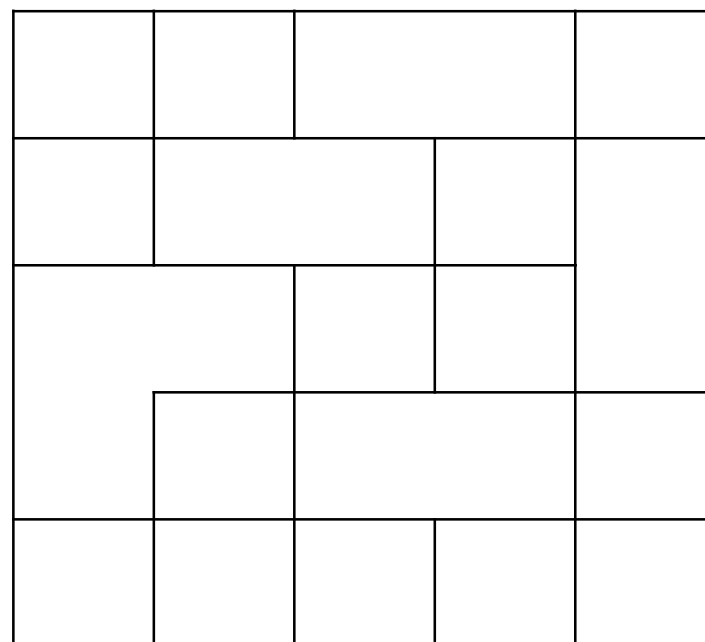
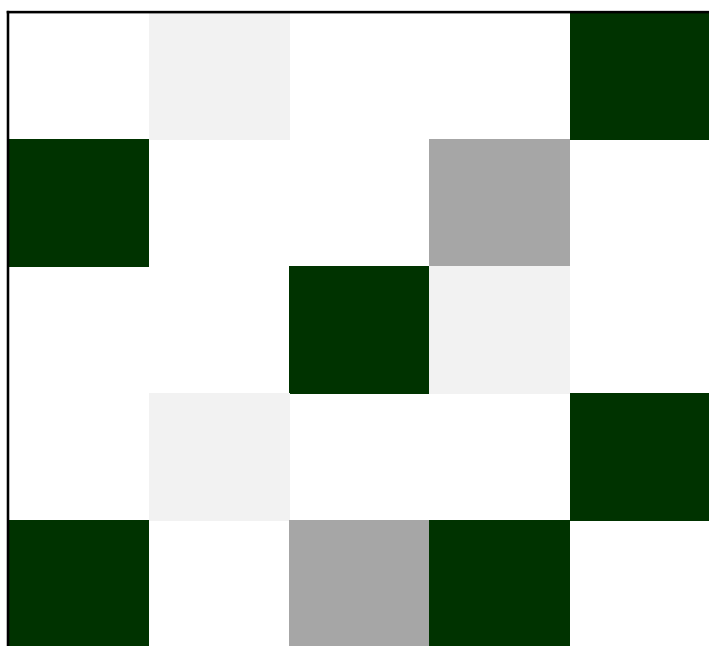
لبه، انواع و خصوصیات آن

Edge, Types and its characteristic

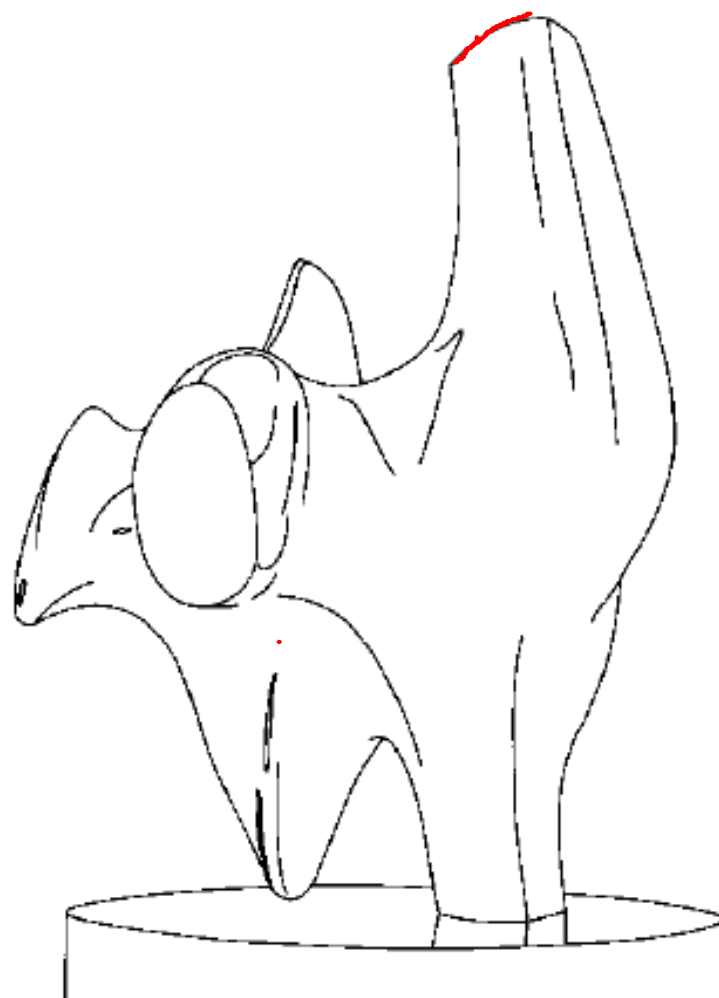
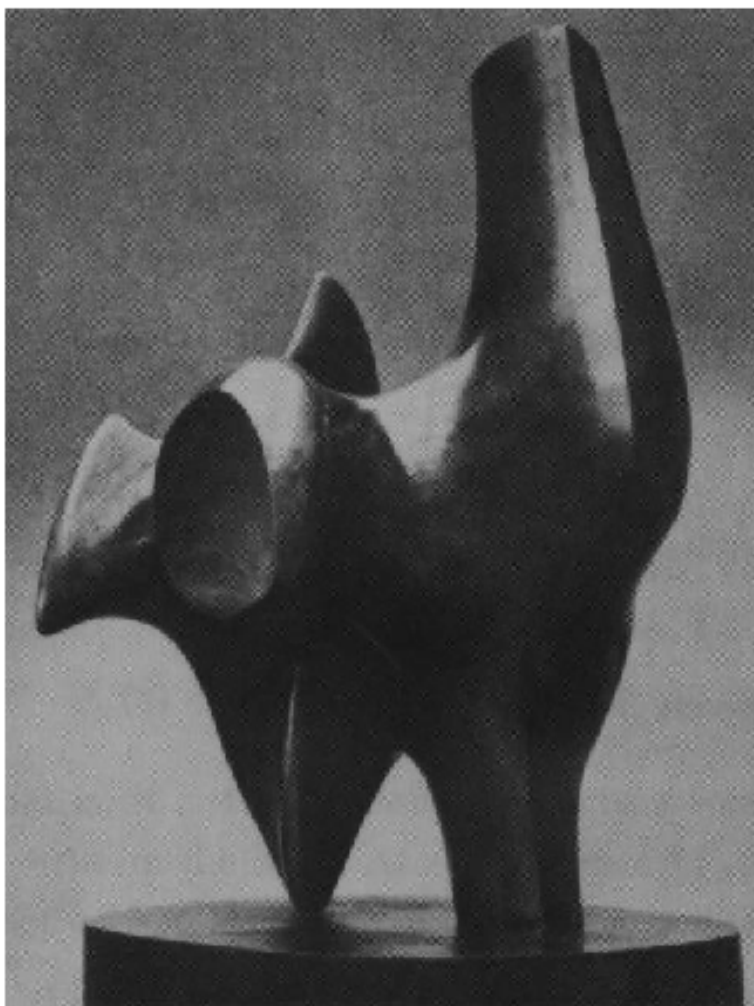
لبه

- لبه قسمتهایی از یک تصویر است که در آنجا شدت رنگ به طور ناگهانی تغییر می‌کند.
- به نوعی می‌توان گفت که لبه همان مرزهای اشیاء موجود در تصویر می‌باشد.
- اگر یک تصویر را یک تابع در نظر بگیریم (image intensity function)، آن‌گاه لبه‌ها همان نقاط ناپیوستگی یا ناهمواری این تابع می‌باشند.
- البته تعریف دقیق ریاضی برای لبه وجود ندارد.

لبه-مثال

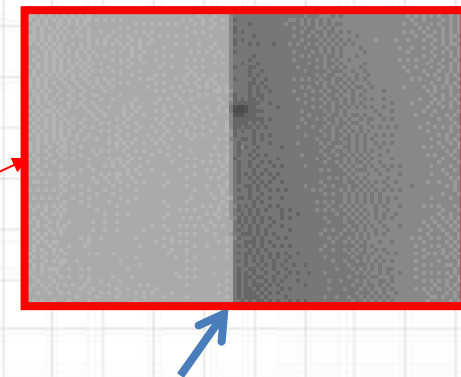
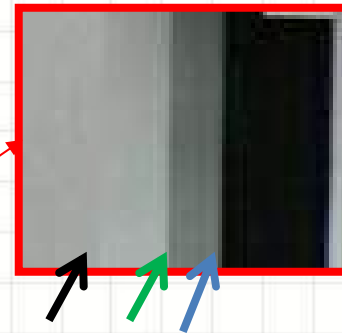
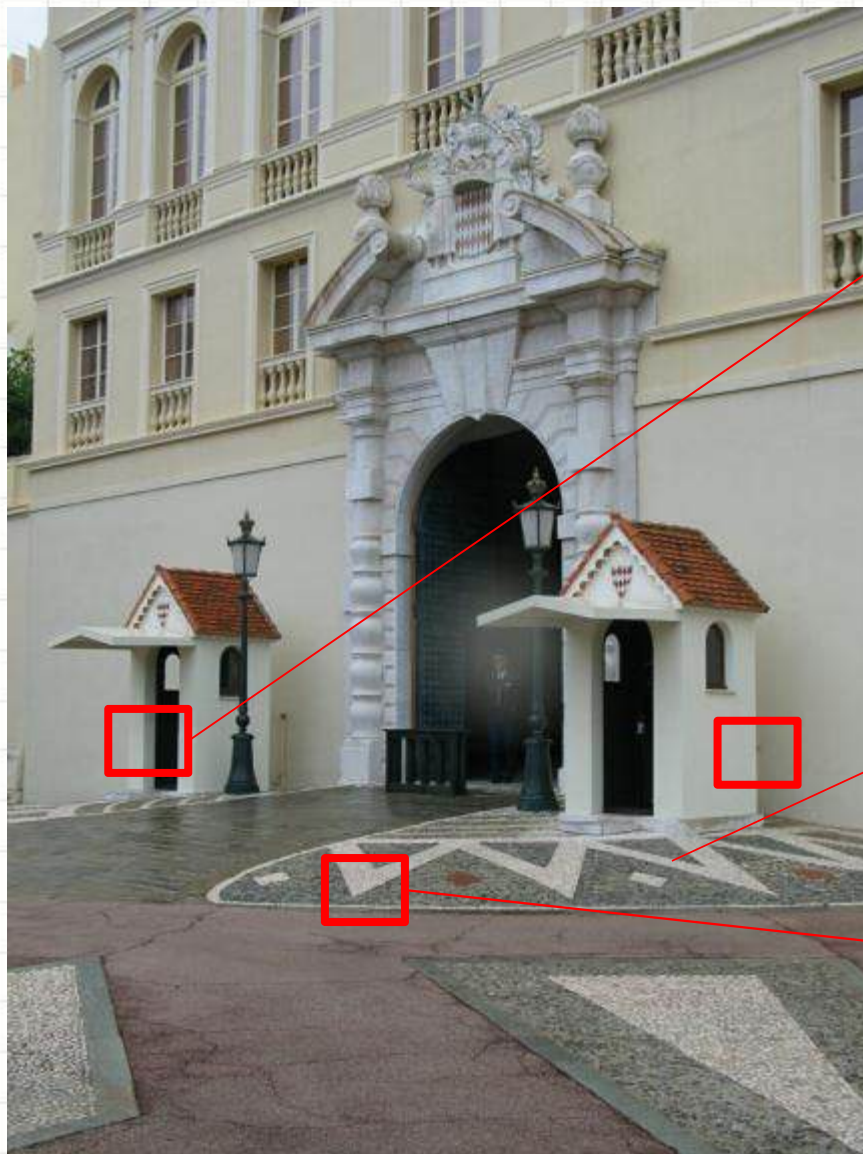


لبه-مثال



لبه-مثال

عموماً در یک تصویر لبه‌های زیادی وجود دارد



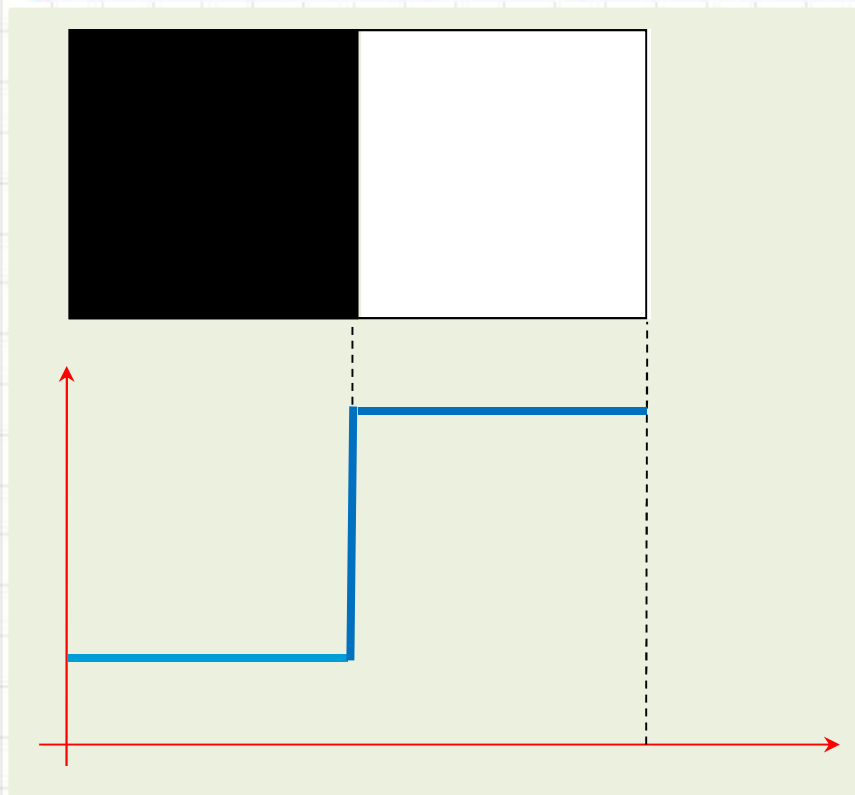
انواع لبه

در حالت کلی، تمامی لبه‌های موجود در تصویر را نمی‌توان دسته‌بندی و نامگذاری کرد. اما چهار نوع لبه بیشتر شناخته شده‌اند که به قرار زیر می‌باشند:

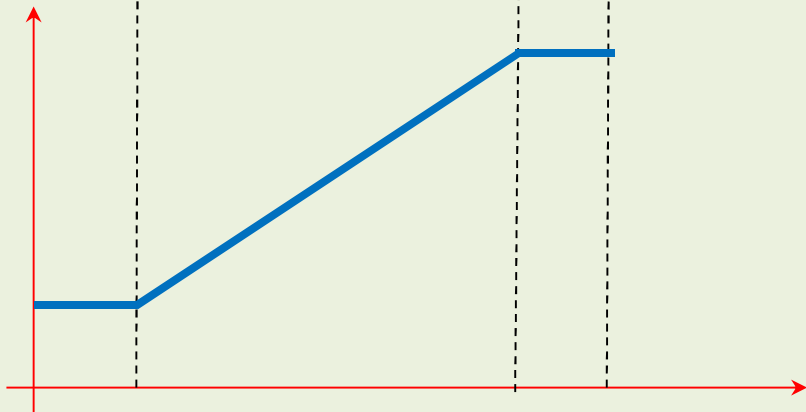
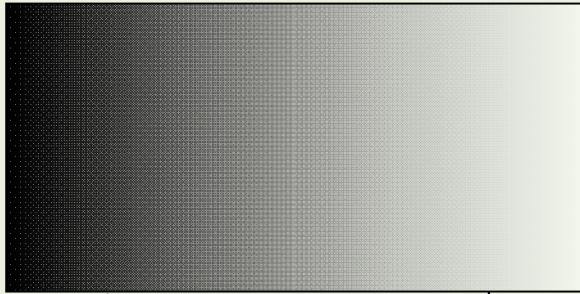
1. لبه پله‌ای (Step Edge)
2. لبه شیب‌دار (Ramp Edge)
3. لبه خطی (Line Edge)
4. لبه شیروانی (Roof Edge)

لبه‌ی پله‌ای

در لبه‌های پله‌ای ناگهان شدت رنگ در یک پیکسل تغییر می‌کند و این تغییر محسوس می‌باشد.
برای نمونه شکل روبرو را در نظر بگیرید:



لبه‌ی شیب‌دار

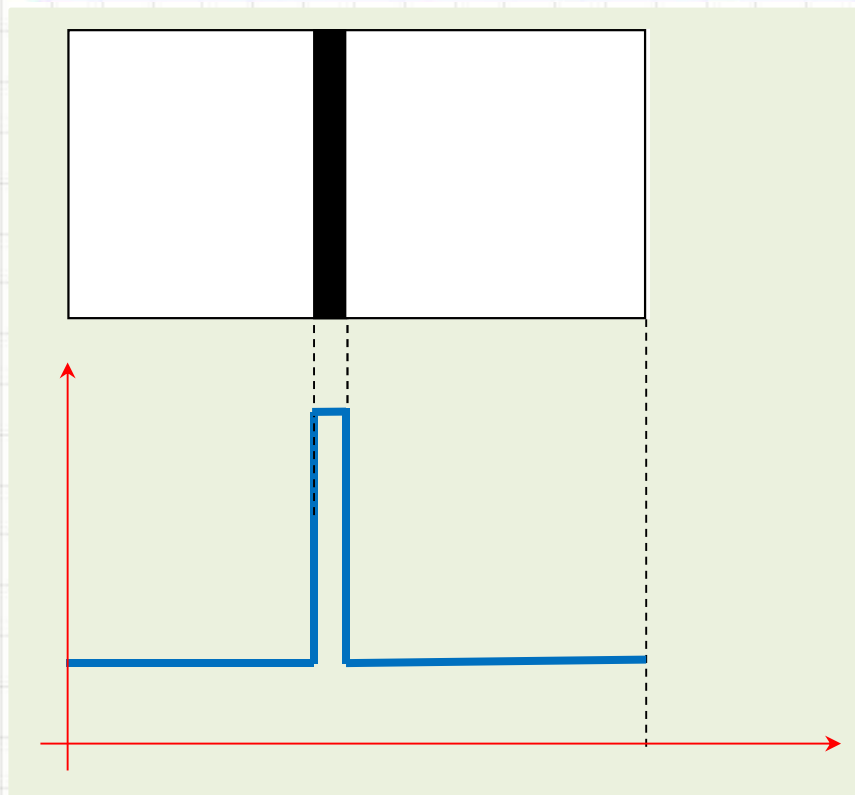


در لبه‌های شیب‌دار شدت رنگ به صورت ملایم تغییر می‌کند و این تغییر محسوس در یک پیکسل محسوس نمی‌باشد، بلکه در یک مجموعه از پیکسل‌ها این تغییر قابل تشخیص می‌باشد.

برای نمونه شکل روبرو را در نظر بگیرید:

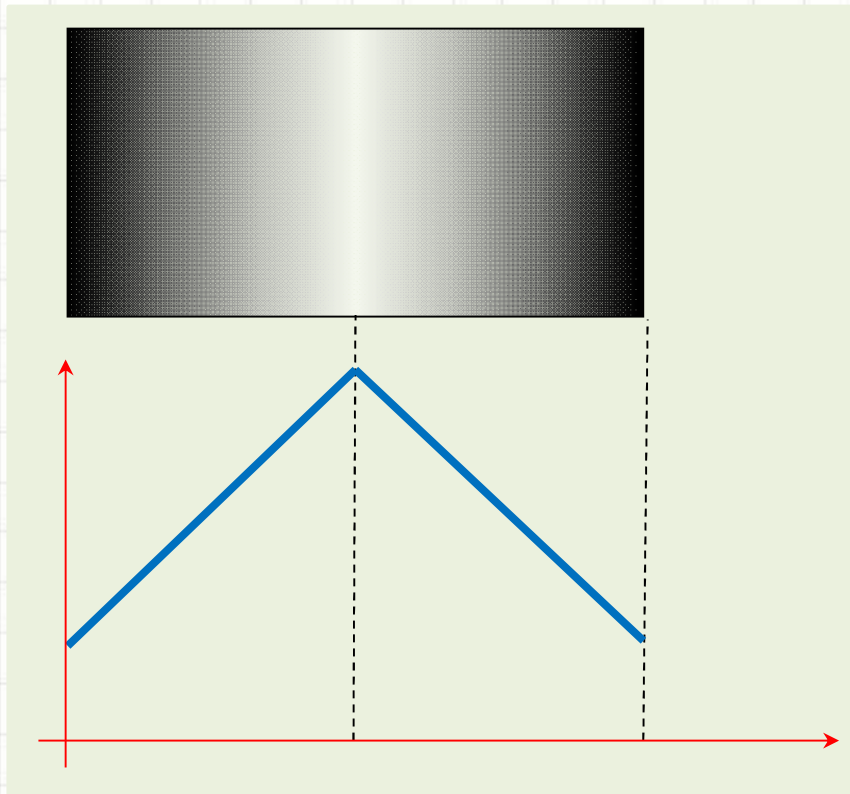
لبه‌ی خطی

در لبه‌های خطی شدت رنگ به صورت ناگهانی تغییر می‌کند و دوباره به حالت اول برمی‌گردد. لبه‌های خطی همان اثری هستند که از کشیدن قلم روی یک صفحه حاصل می‌شوند.



لبه‌ی شیروانی

در لبه‌های شیروانی شدت شدت رنگ در ابتدا به تدریج کم شده ولی دوباره به تدریج افزایش می‌یابد.

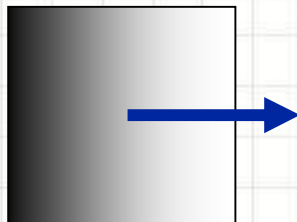


خصوصیات لبه

برای لبه دو خصوصیت زیر را می توان برشمرد:

- جهت لبه
- اندازه لبه

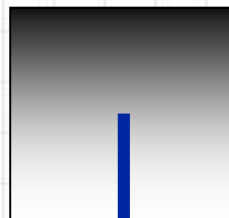
Vertical



$$\|\nabla I\| = G_x$$

$$q(x, y) = 0$$

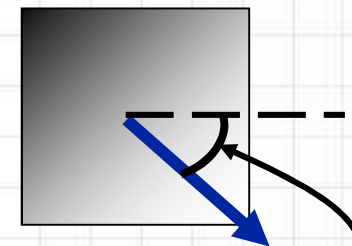
Horizontal



$$\|\nabla I\| = G_y$$

$$q(x, y) = -\frac{p}{2}$$

Generic



$$\|\nabla I\| = \sqrt{G_x^2 + G_y^2}$$

$$q(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

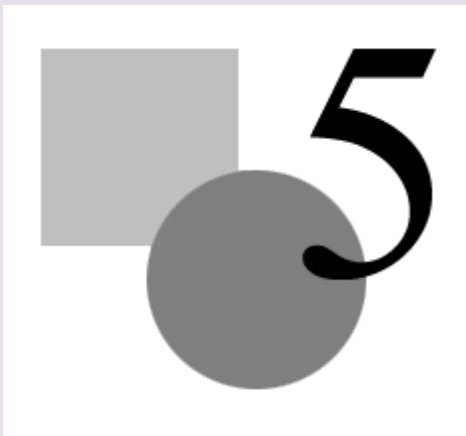
تشخیص و یافتن لبه (Edge Detection)

تشخیص لبه (Edge Detection)

تشخیص لبه به معنی مشخص کردن لبه‌ها در تصویر می‌باشد.
تشخیص لبه در تصاویر با الگوریتم‌هایی موسوم به **الگوریتم‌های لبه‌یاب** انجام می‌شود.

در الگوریتم‌های لبه‌یاب ورودی یک فایل تصویر است و خروجی عبارت است از موقعیت لبه‌های تصویر. که این خروجی می‌تواند در قالب یک تصویر دیگر باشد و یا در قالب ساختارهای ریاضی که لبه‌ها را مشخص می‌کند.

در ساده‌ترین حالت خروجی الگوریتم‌های لبه‌یاب یک باینری است که با تصویر اصلی هم‌سایز است و پیکسل‌هایی که در موقعیت لبه قرار دارند با رنگ سفید و دیگر پیکسل‌ها با رنگ سیاه مشخص می‌شوند.



فیلتر لبه‌یاب و مشتق

- الگوریتم‌های لبه‌یاب عموماً از **فیلتر** برای یافتن لبه‌های تصویر استفاده می‌کنند که آن فیلترها به **فیلترهای لبه‌یاب** موسوم می‌باشند.
- اساس فیلترهای لبه‌یاب بر مبنای **مشتق** می‌باشد.
- مشتق موقعیت تغییرات ناگهانی در تابع را مشخص می‌کند و اگر تصویر را یک تابع دوبعدی در نظر بگیریم، آن‌گاه با مشتق می‌توان مکان‌هایی که تابع تغییر ناگهانی دارد را مشخص کنیم. توجه شود که این مکان‌ها همان لبه‌های تصویر می‌باشد.
- توجه شود که تصویر یک تابع گسسته است و همان مشتق معمولی که در حسابان استفاده می‌شد را نمی‌توان برای توابع گسسته به کار برد.
- برای توابع گسسته از **مشتق گسسته** یا **مشتق عددی** استفاده می‌شود. این نوع مشتق‌ها در درس آنالیز عددی و یا محاسبات عددی بررسی می‌شوند.
- **مشتق گسسته** یا **مشتق عددی** فرمول‌هایی جبری می‌باشند که با استفاده از مقدار تابع در نقاط معلوم، به تخمین مشتق اول یا دوم تابع در همان نقاط معلوم می‌پردازند. البته فرمول‌های متفاوتی برای انجام این کار ارائه شده است.

مشتق گسسته برای توابع یک متغیره

فرض کنید f تابعی گسسته باشد که در نقاط $x = 0, 1, \dots, n$ تعریف شده باشد. حال مشتق تابع f در نقاط فوق را می‌توان به صورت زیر تخمین زد

$$\frac{d}{dx} f(x) = f(x+1) - f(x)$$

فرمول فوق یک فرمول دو نقطه‌ای می‌باشد.

همچنین مشتق دوم تابع f را می‌توان با فرمول زیر تخمین زد

$$\frac{d^2}{dx^2} f(x) = f(x+1) - 2f(x) + f(x-1)$$

فرمول فوق به فرمول سه نقطه‌ای می‌باشد.

توجه شود که علاوه بر فرمول‌های فوق، فرمول‌های دیگری نیز برای مشتق مرتبه اول و دوم وجود دارد. مثلاً فرمول فوق که برای مشتق مرتبه اول ارائه شده یک فرمول دو نقطه‌ای است که به فرمول دونقطه‌ای پیشرو موسوم است. علاوه بر این فرمول فرمول‌های دونقطه‌ای پس‌رو، فرمول‌های سه نقطه‌ای پیش‌رو، مرکزی و پس‌رو و ... وجود دارد

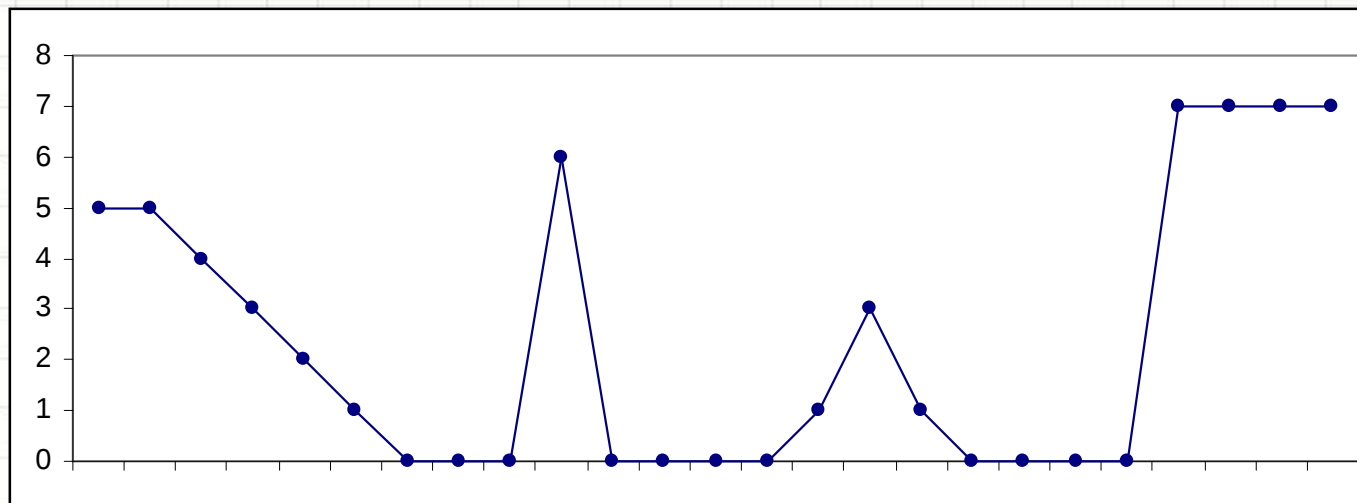
مشتق گسسته برای توابع یک متغیره

برای درک بهتر مشتق گیری گسسته یک مثال آورده می شود.

فرض کنید f تابعی گسسته باشد که در نقاط $x = 0, 1, \dots, 25$ به صورت زیر تعریف شده باشد.

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$f(x)$	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7

برای درک بهتر تصویر این تابع در زیر رسم شده است

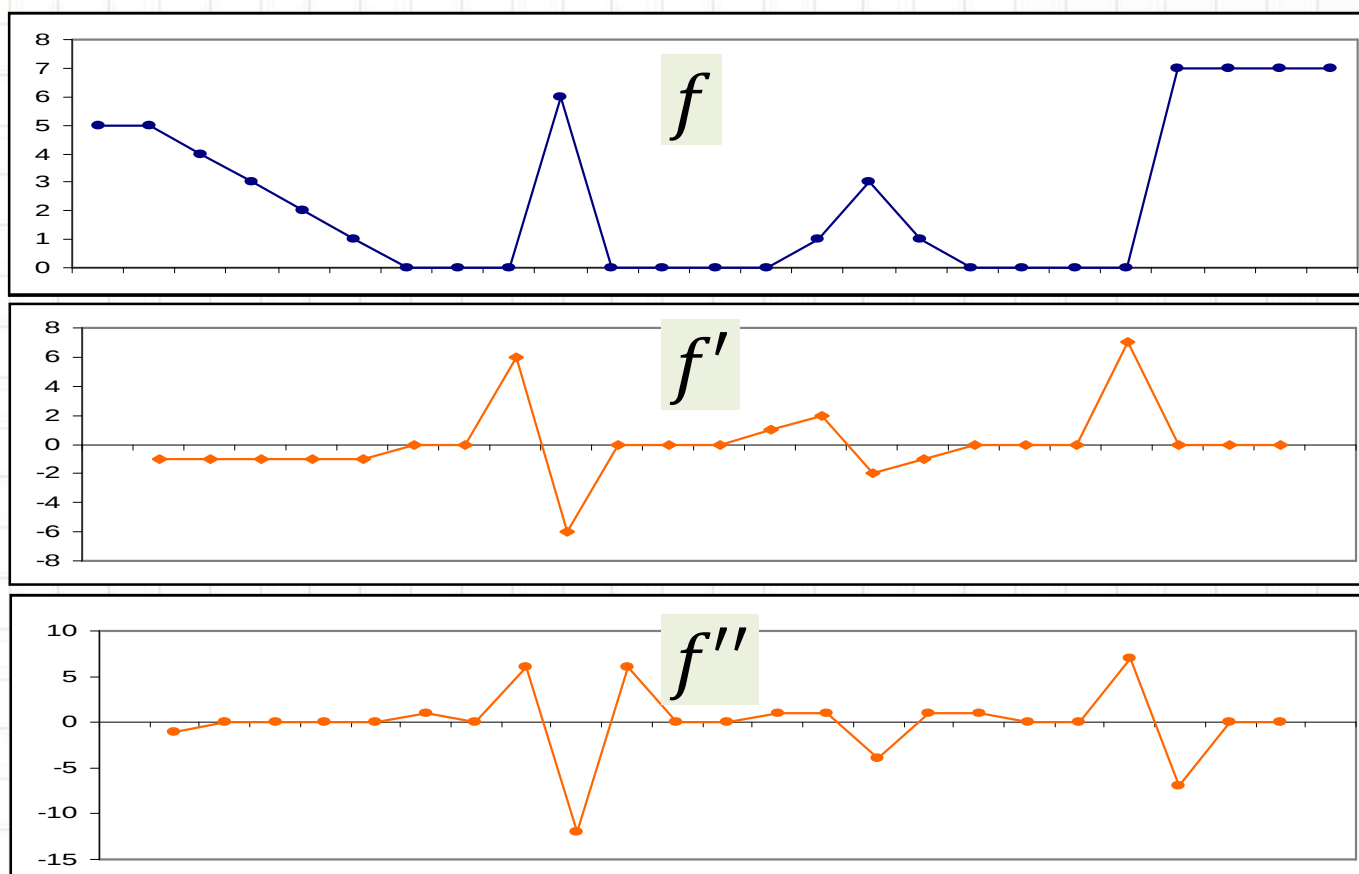


مشتق گسسته برای توابع یک متغیره

با استفاده از فرمول‌های فوق، مشتق اول و دوم تابع به صورت زیر محاسبه می‌شود

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$f(x)$	5	5	4	3	2	1	0	0	0	6	0	0	0	0	1	3	1	0	0	0	0	7	7	7	7
$f'(x)$		0	-1	-1	-1	-1	0	0	6	-6	0	0	0	1	2	-2	-1	0	0	0	7	0	0	0	
$f''(x)$		-1	0	0	0	0	1	0	6	-12	6	0	0	1	1	-4	1	1	0	0	7	-7	0	0	

ملاحظه می‌شود
که در مکان‌هایی
که تابع f تغییرات
زیادی دارد، مقدار
تابع f' و f''
بزرگ است.



مشتق گسسته برای توابع دو متغیره

تاکنون مشتق توابع گسسته یک متغیره را بررسی کردیم. اما تصویر یک تابع گسسته دومتغیره است. لذا باید مشتق گسسته را برای این توابع نیز معرفی کنیم.

در توابع دومتغیره مشتق‌های متفاوتی تعریف می‌شود. از جمله گرادیان، مشتق‌های نسبی، لاپلاسین، هسین و
در پردازش تصویر بیشتر از لاپلاسین و گرادیان استفاده می‌شود.

اگر $f(x, y)$ یک تابع دو متغیره باشد، آن‌گاه لاپلاسین f به صورت زیر تعریف می‌شود:

$$\Delta f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

تعریف فوق برای توابع پیوسته می‌باشد و اگر بخواهیم تعریف فوق را برای توابع گسسته نیز به کار ببریم، آن‌گاه لازم است که لاپلاسین را گسسته کنیم. برا گسسته سازی لاپلاسین باید مشتق‌ها را با فرمول‌های گسسته مشتق جایگزین کنیم.

مشتق گسسته برای توابع دو متغیره

گسسته کردن لاپلاسین به معنی جایگزین کردن مشتق‌ها با فرمول‌های گسسته مشتق است. از سوی دیگر فرمول‌های مشتق گسسته متفاوتی می‌توان به کار برد. لذا اپراتور لاپلاسین رامی‌توان به فرمول‌های متفاوتی گسسته نمود که در زیر به یک مورد اشاره می‌شود.

برای فرض کنیم $f(x, y)$ یک تابع گسسته دو متغیره باشد، آن‌گاه می‌توان نوشت

$$\frac{\partial^2 f}{\partial^2 y} = f(x, y + 1) + f(x, y - 1) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial^2 x} = f(x + 1, y) + f(x - 1, y) - 2f(x, y)$$

و لذا

$$\begin{aligned} \Delta f(x, y) &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \\ &= f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y) \end{aligned}$$

رابطه فوق را می‌توان یک تبدیل همسایگی و یا فیلتر در نظر گرفت. که این فیلتر خطی نیز می‌باشد و ماتریس ماسک متناظر با آن به صورت زیر است:

0	1	0
1	-4	1
0	1	0

جمع بندی

- برای یافتن لبه‌های تصویر از الگوریتم‌هایی موسوم به الگوریتم‌های لبه‌یاب استفاده می‌شود.
- الگوریتم‌های لبه‌یاب می‌تواند از **مشتق** برای یافتن لبه‌ها بهره بگیرند. چرا که مشتق قادر است که مکان‌هایی که تغییرات زیادی در مقدار تابع وجود دارد را مشخص کند.
- تصویر یک تابع دومتغیره گسسته است و لذا باید از **مشتق گسسته** در تصاویر استفاده شود.
- در توابع دومتغیره گسسته می‌توان انواع متفاوتی از مشتق‌های گسسته را تعریف و به کار برد.
- اعمال مشتق گسسته روی یک تصویر باعث می‌شود که تصویر دیگری حاصل شود که اندازه پیکسل‌هایی که در موقعیت لبه قرار دارند بزرگ‌تر از اندازه پیکسل‌های دیگر باشد.
- اعمال مشتق گسسته روی یک تصویر را از یک دیدگاه می‌توان معادل **فیلتر کردن** تصویر دانست. که این فیلتر خطی است و دارای یک ماتریس ماسک می‌باشد.

دسته‌بندی

همان‌گونه که ذکر شد، الگوریتم‌های لبه‌یاب از فیلتر خطی استفاده می‌کنند که این فیلترها به فیلترهای لبه‌یاب موسوم می‌باشند.

ماتریس ماسک در فیلترهای لبه‌یاب با توجه به نوع مشتق و نوع مشتق گسسته به کار رفته تعریف می‌شود و به خاطر تنوع در نوع مشتق و نوع فرمول مشتق گسسته، ماتریس‌های ماسک متفاوتی قابل تعریف می‌باشد. که از جمله آن‌ها می‌توان به موارد زیر اشاره کرد:

- فیلتر لاپلاس (Laplace)
- فیلتر سوبل (Sobel)
- فیلتر پریویت (Prewitt)
- فیلتر رابرت (Robert)

بر مبنای فیلترهای لبه‌یاب الگوریتم‌های لبه‌یاب متفاوتی ارائه می‌شود که در زیر به برخی از آن‌ها اشاره می‌گردد

- الگوریتم لبه‌یاب سوبل
- الگوریتم لبه‌یاب کانی (Canny)

فیلتر لبہ یاب لاپلاس

فیلتر لاپلاس (Laplace Filter)

- فیلتر لاپلاس یک فیلتر خطی است با ماتریس ماسک 3 در 3 است.
- ماتریس ماسک در این فیلتر از اپراتور لاپلاس اقتباس می شود

$$\Delta f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

- مولفه های ماتریس ماسک ضرایب تقریب عددی اپراتور لاپلاس می باشد.
- با توجه به این که تقریب عددی به صورت های متفاوتی انجام می شود، لذا ماتریس های ماسک متفاوتی نیز وجود دارد.
- یک فرم کلی برای ماتریس های ماسک لاپلاسی به صورت زیر است:

$\frac{\alpha}{1 + \alpha}$	$\frac{1 - \alpha}{1 + \alpha}$	$\frac{\alpha}{1 + \alpha}$
$\frac{1 - \alpha}{1 + \alpha}$	$\frac{-4}{1 + \alpha}$	$\frac{1 - \alpha}{1 + \alpha}$
$\frac{\alpha}{1 + \alpha}$	$\frac{1 - \alpha}{1 + \alpha}$	$\frac{\alpha}{1 + \alpha}$

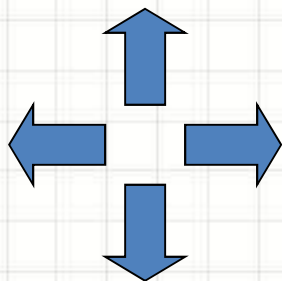
به طوری که α عددی بین 0 و 1 انتخاب می شود

فیلتر لاپلاس (Laplace Filter)

در زیر ماتریس ماسک لاپلاسن، به ازای برخی مقادیر از α ارایه شده است.

0	1	0
1	-4	1
0	1	0

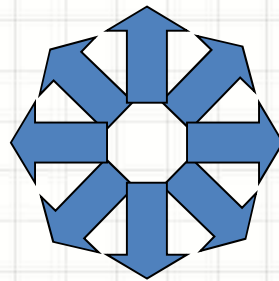
$$\alpha = 0$$



1	1	1
1	-8	1
1	1	1

$$\frac{1}{3}$$

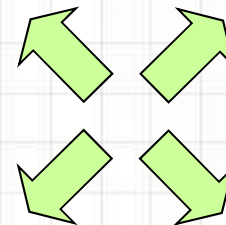
$$\alpha = 0.5$$



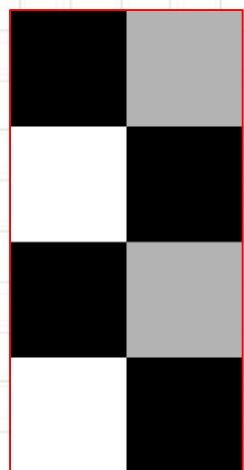
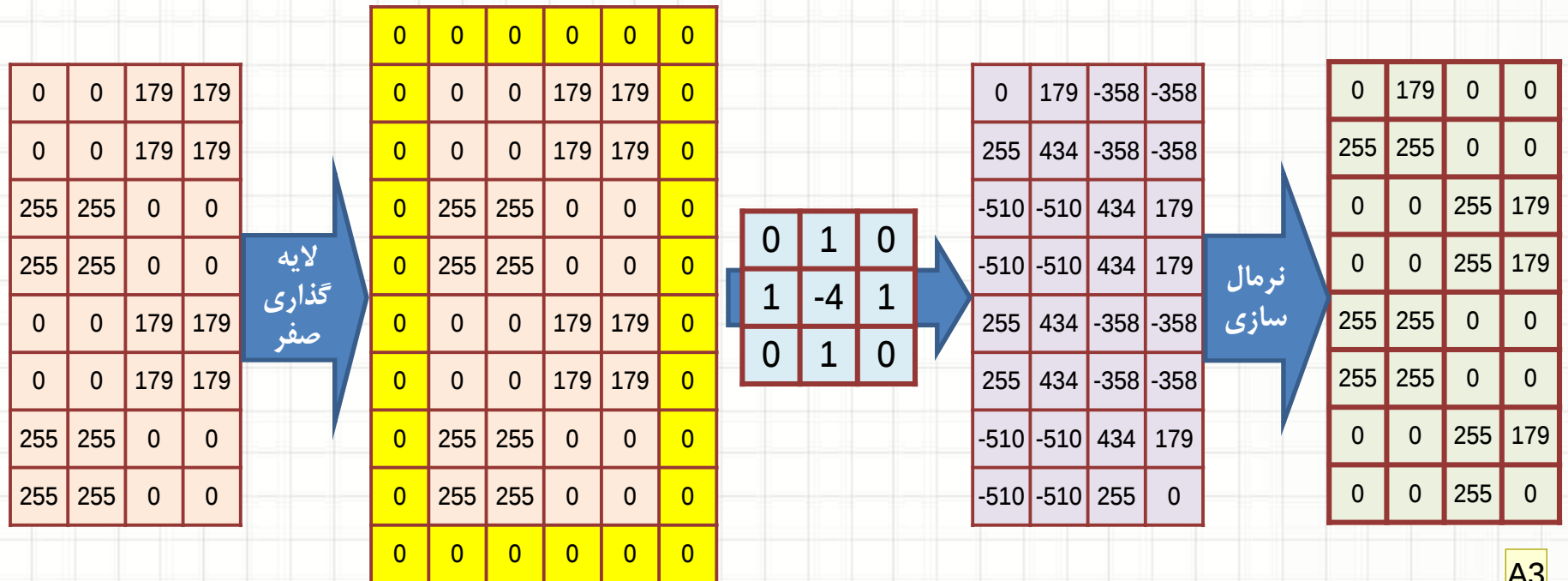
1	0	1
0	-4	0
1	0	1

$$\frac{1}{2}$$

$$\alpha = 1$$

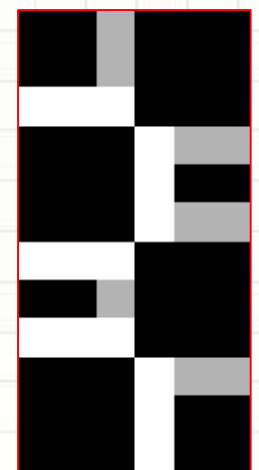


فیلتر لاپلاس - مثال ۱



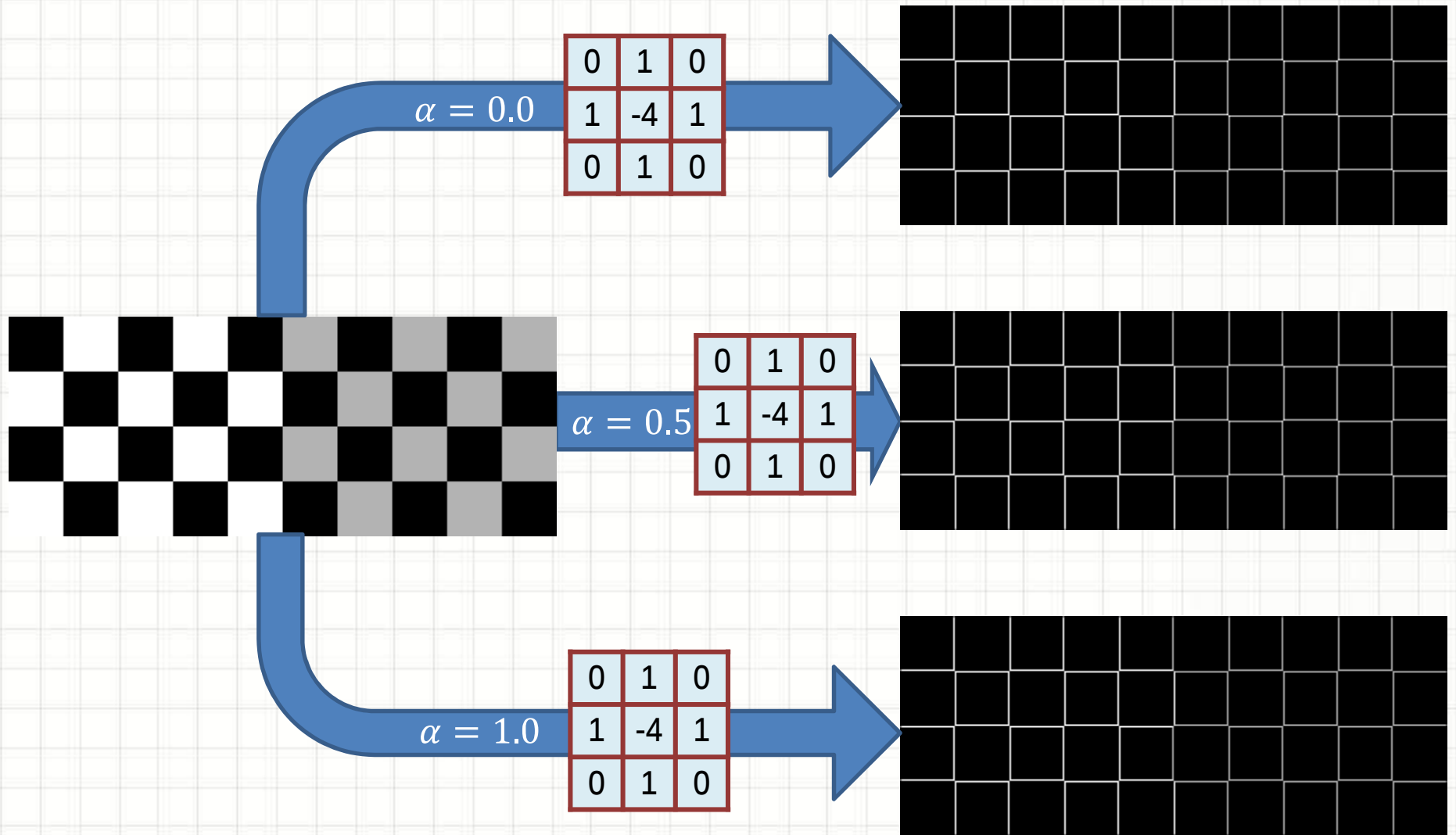
0	1	0
1	-4	1
0	1	0

فیلتر لاپلاسی با ماسک



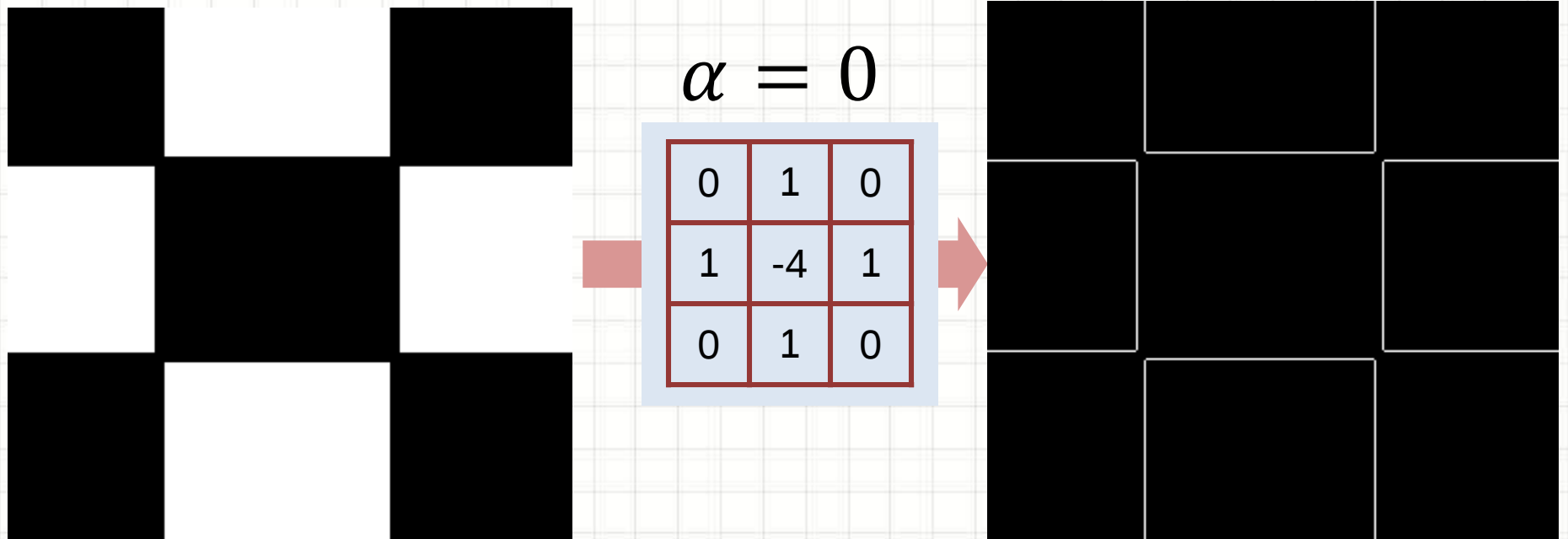
A3

اعمال فیلترهای لاپلاس روی یک تصویر

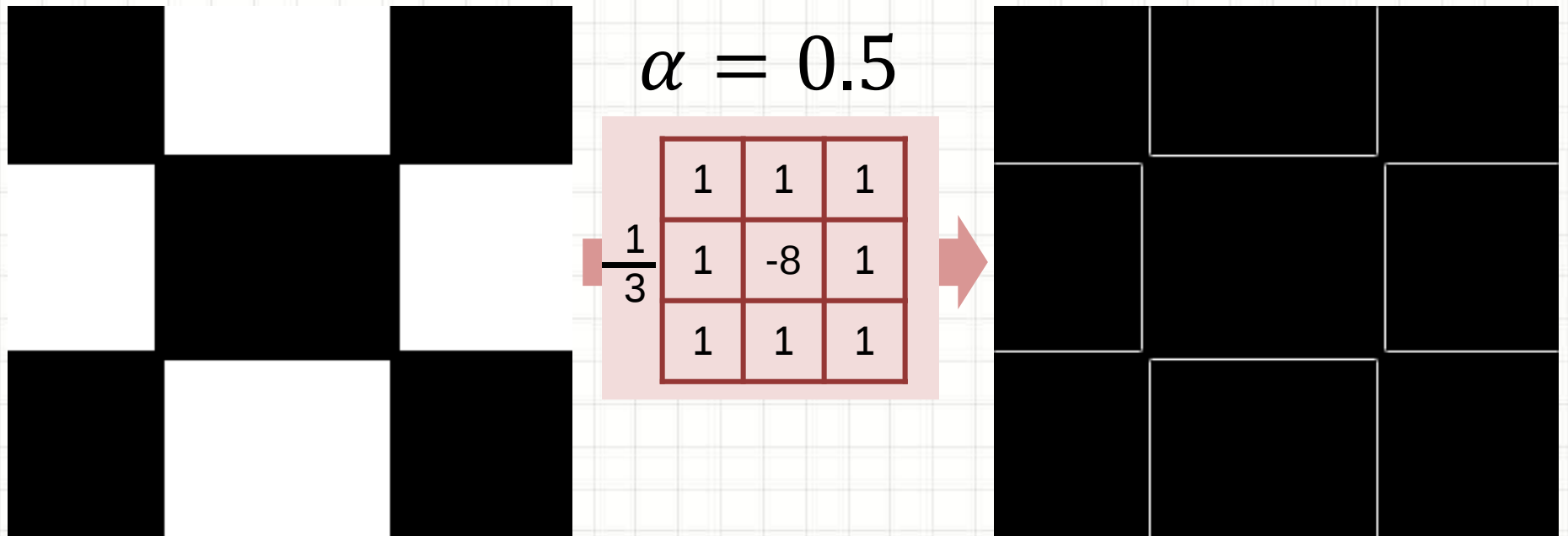


فیلتر لاپلاس به ازای α های متفاوت تقریباً یکسان عمل می کند.

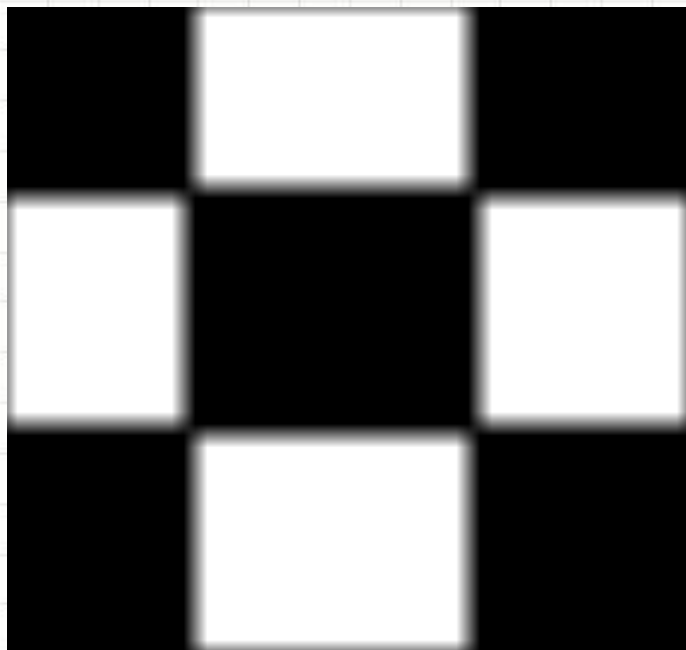
اعمال فیلترهای لاپلاس روی یک تصویر



اعمال فیلترهای لاپلاس روی یک تصویر



اعمال فیلترهای لاپلاس روی یک تصویر



$$\alpha = 0$$

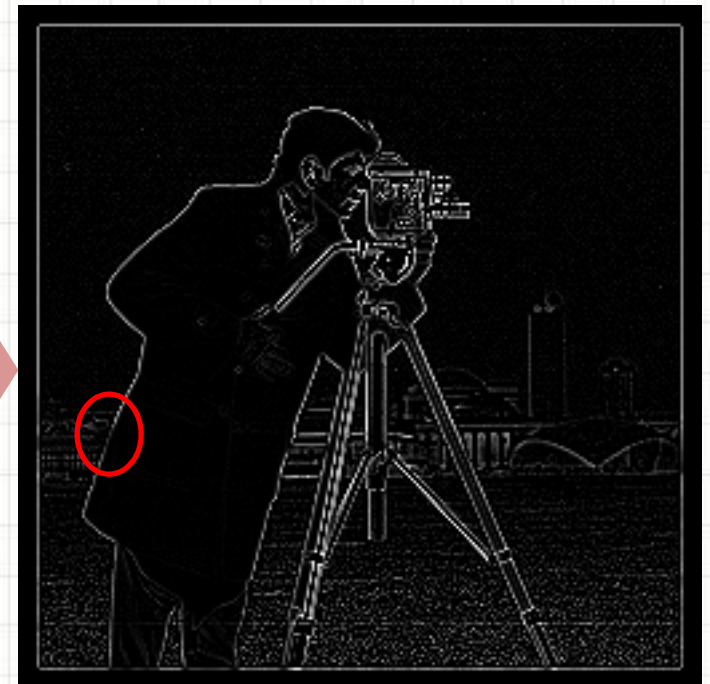
0	1	0
1	-4	1
0	1	0



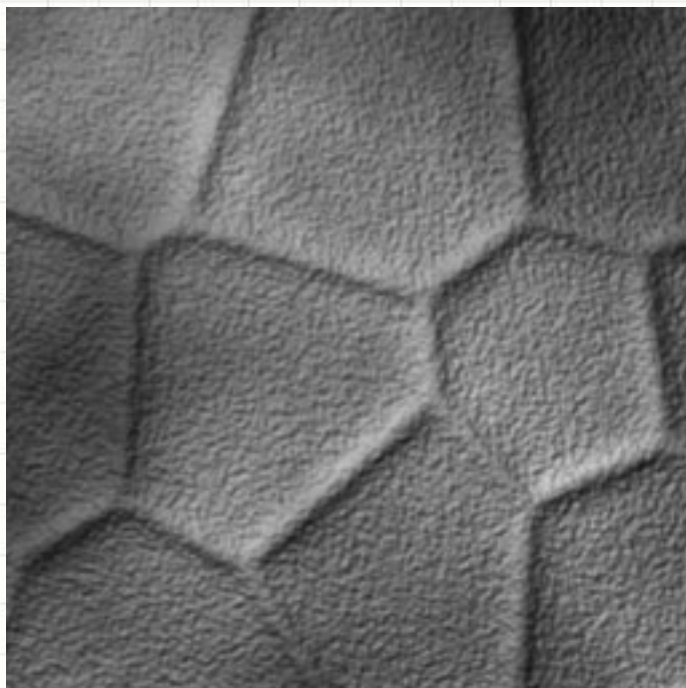
اعمال فیلترهای لاپلاس روی یک تصویر



0	1	0
1	-4	1
0	1	0



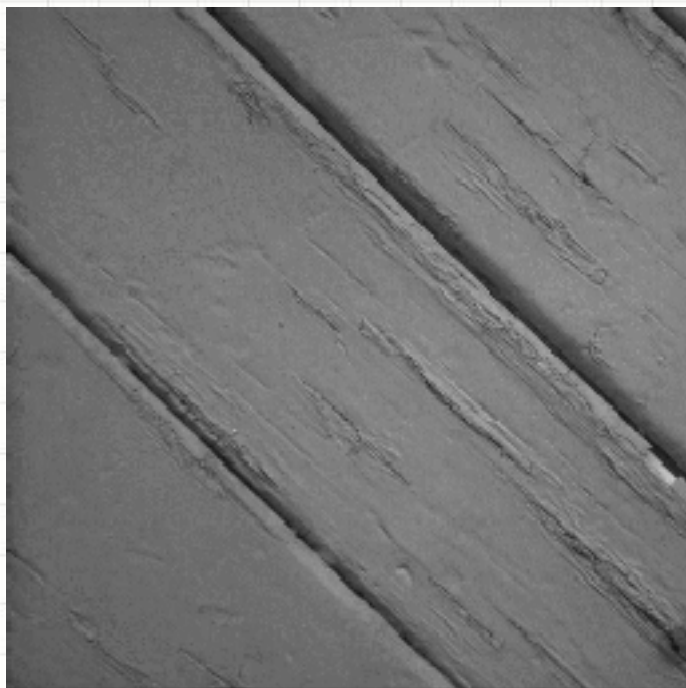
اعمال فیلترهای لاپلاس روی یک تصویر



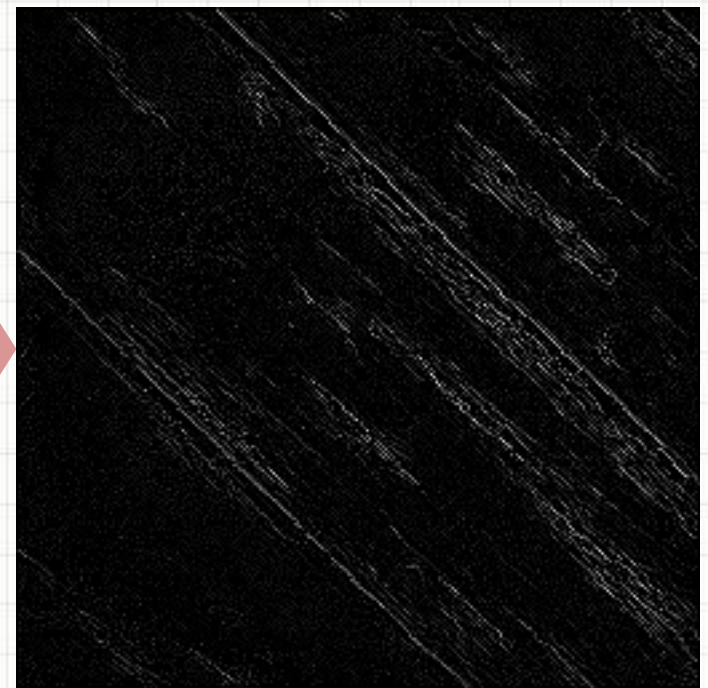
0	1	0
1	-4	1
0	1	0



اعمال فیلترهای لاپلاس روی یک تصویر



0	1	0
1	-4	1
0	1	0



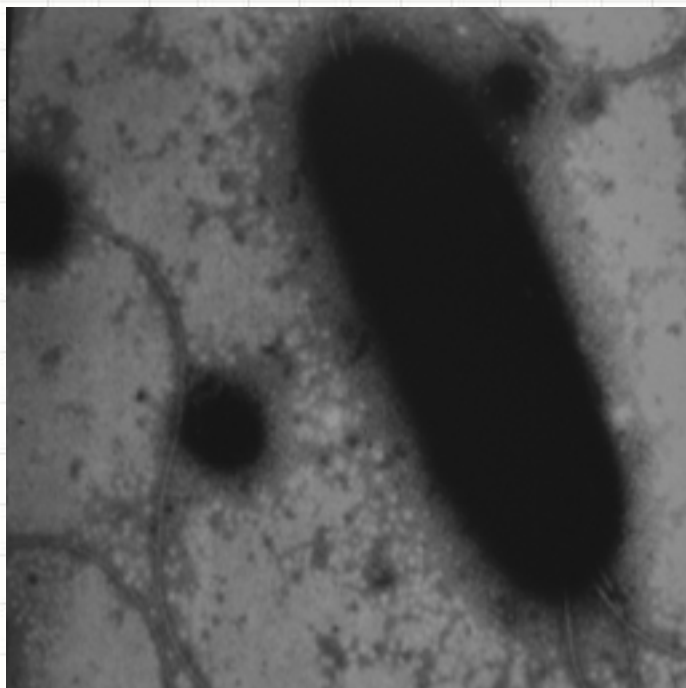
اعمال فیلترهای لاپلاس روی یک تصویر



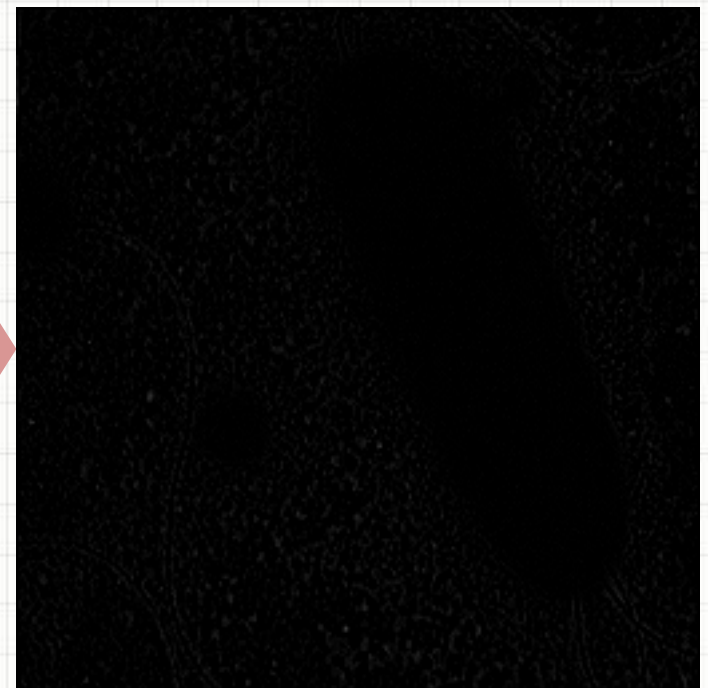
0	1	0
1	-4	1
0	1	0



اعمال فیلترهای لاپلاس روی یک تصویر



0	1	0
1	-4	1
0	1	0



فیلتر لاپلاس

- همان گونه که ملاحظه گردید، فیلترهای لاپلاسی، لبه ها را برجسته می کند و قسمت های غیر لبه را به پس زمینه ای تاریک و بدون ویژگی بارز تبدیل می کند.

فیلتر لبہ یاب سوہل

فیلترهای Sobel

فیلتر سوبل از جمله فیلترهای تشخیص لبه می باشد. در فیلتر سوبل دو ماسک به صورت زیر وجود دارد:

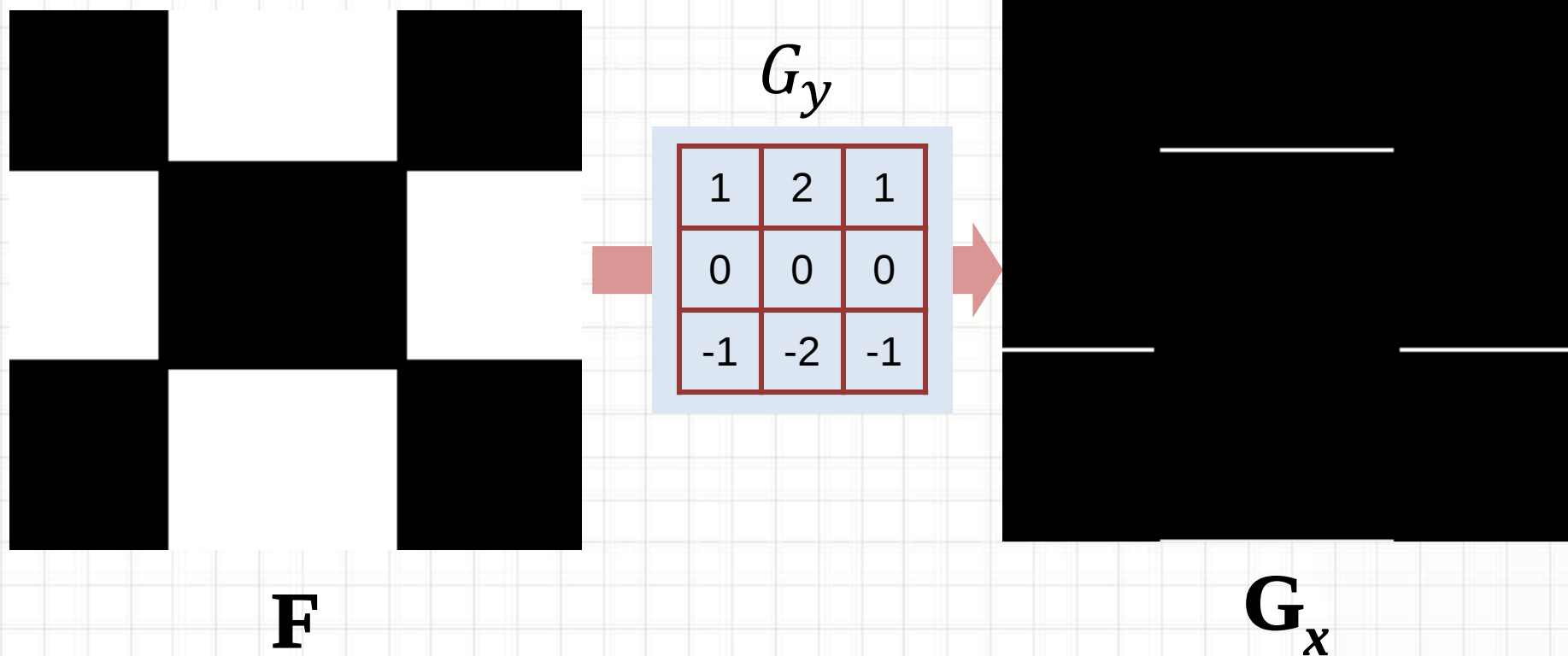
$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

ماسک سوبل افقی

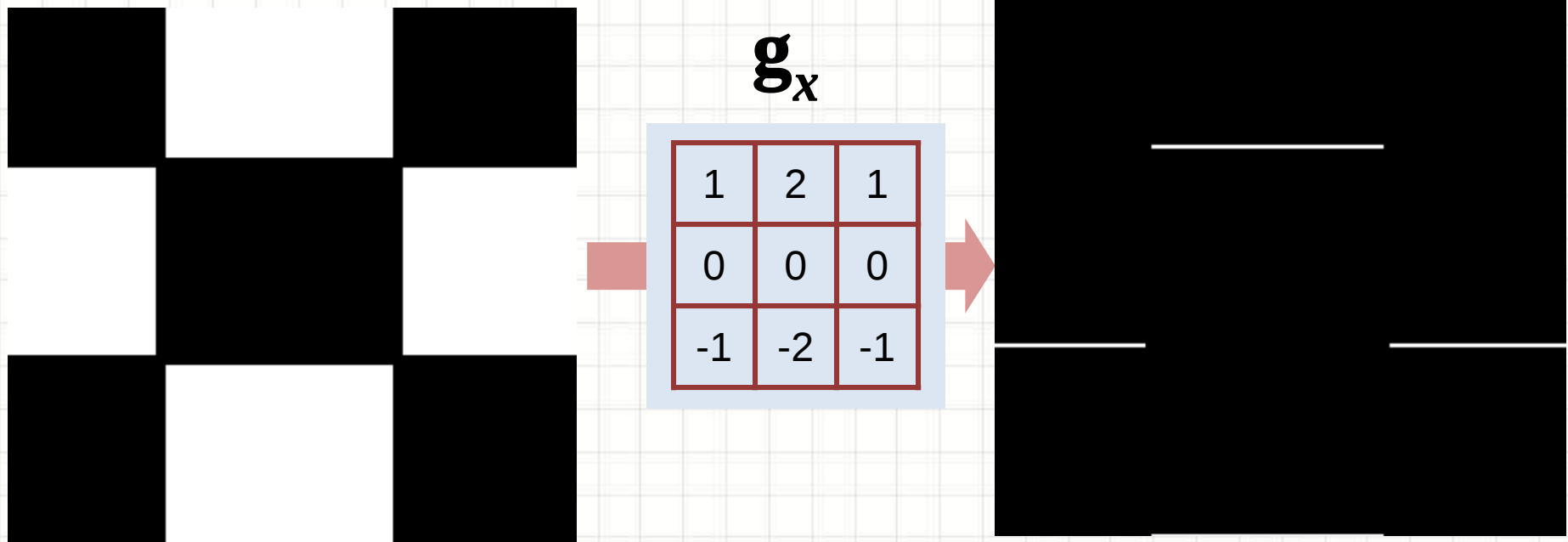
$$G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

ماسک سوبل عمودی

اعمال فیلترهای سو بل یک تصویر



اعمال فیلترهای سوبل یک تصویر



ملاحظه می شود که فیلتر سوبل با ماسک افقی، کلیه لبه های افقی را استخراج نکرده است.

اعمال فیلترهای سو بل یک تصویر

I

0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0

G_x

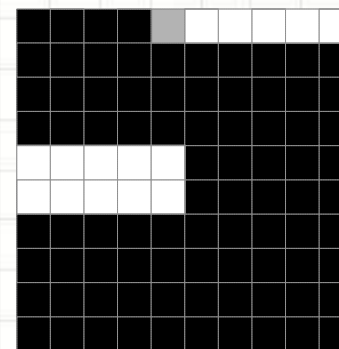
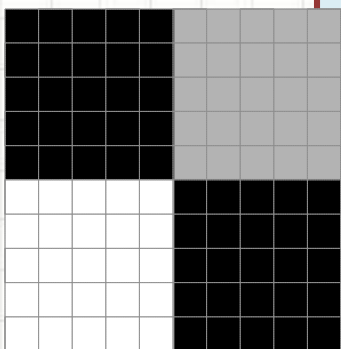
0	0	0	0	179	537	716	716	716	537
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
765	1020	1020	1020	586	-282	-716	-716	-716	-537
765	1020	1020	1020	586	-282	-716	-716	-716	-537
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
-765	-1020	-1020	-1020	-765	-255	0	0	0	0

0	0	0	0	179	255	255	255	255	255
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$g_x =$

-1	-2	-1
0	0	0
1	2	1

نرمال سازی به بازه $[0,255]$



اعمال فیلترهای سو بل یک تصویر

 I

0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0

 G_x

0	0	0	0	179	537	716	716	716	537
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
765	1020	1020	1020	586	282	-716	-716	-716	-537
765	1020	1020	1020	586	282	-716	-716	-716	-537
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
-765	-1020	-1020	-1020	-586	-282	716	716	716	537

0	0	0	0	179	255	255	255	255	255
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

این اعداد به خاطر وجود لبه ظاهر شده‌اند ولی با علامت منفی. تبعاً در نرمال سازی این مقادیر صفر می‌شوند و صفر نشانه عدم وجود لبه است. یعنی به اشتباه این لبه مشخص نمی‌شود.

برای رفع این مشکل بهتر است که قدر مطلق مولفه‌ها در نظر گرفته شود. به عبارت دیگر بهتر است که ماتریس $|G_x|$ به جای ماتریس G_x در نظر گرفته شود

اعمال فیلترهای سو بل یک تصویر

I

0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
0	0	0	0	0	179	179	179	179	179
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0
255	255	255	255	255	0	0	0	0	0

$|G_x|$

0	0	0	0	179	537	716	716	716	537
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
765	1020	1020	1020	586	282	716	716	716	537
765	1020	1020	1020	586	282	716	716	716	537
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
765	1020	1020	1020	765	255	0	0	0	0

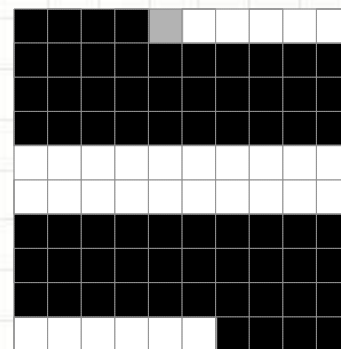
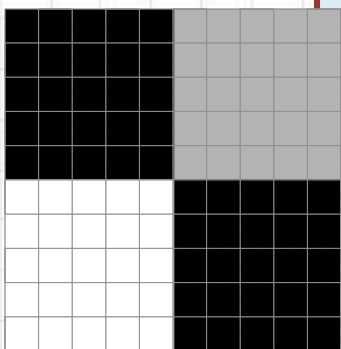
0	0	0	0	179	255	255	255	255	255
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
255	255	255	255	255	255	0	0	0	0

$g_x =$

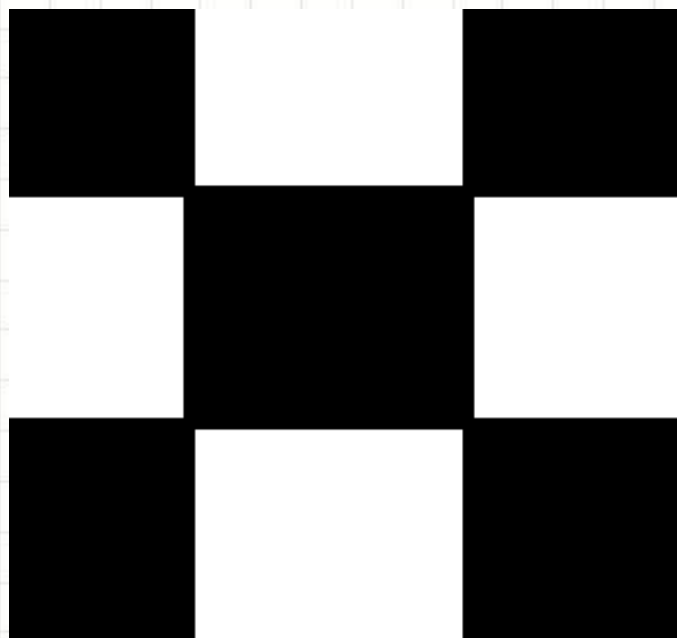
-1	-2	-1
0	0	0
1	2	1

به همراه
قدر مطلق گیری

نرمال سازی به بازه $[0,255]$

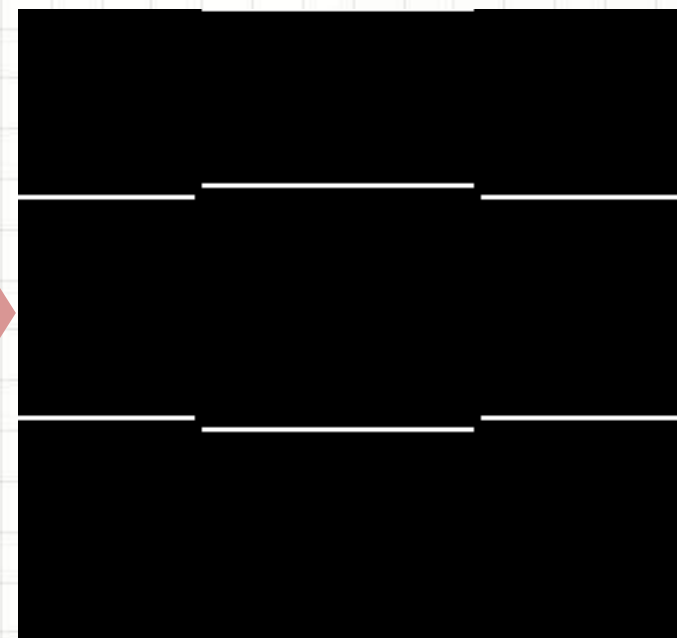


اعمال فیلترهای سوبل یک تصویر



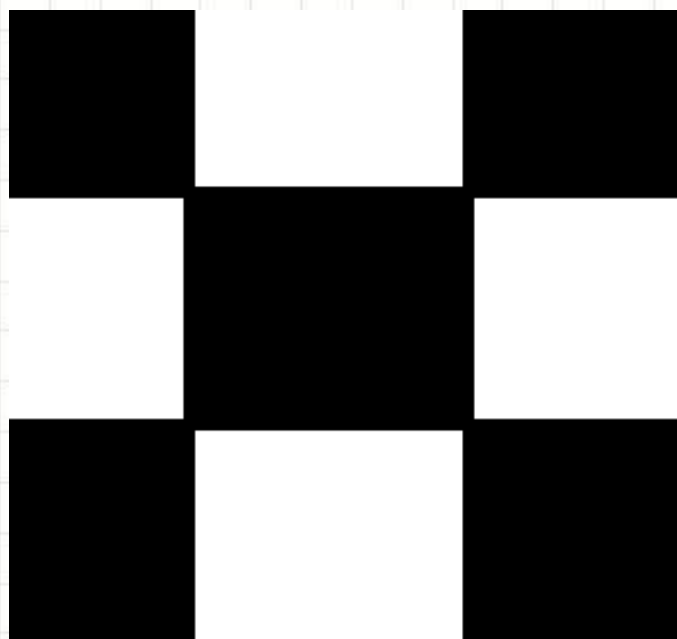
F

قدرمطلق فیلتر
سوبل با ماسک
افقی



G_x

اعمال فیلترهای سو بل روی یک تصویر



F

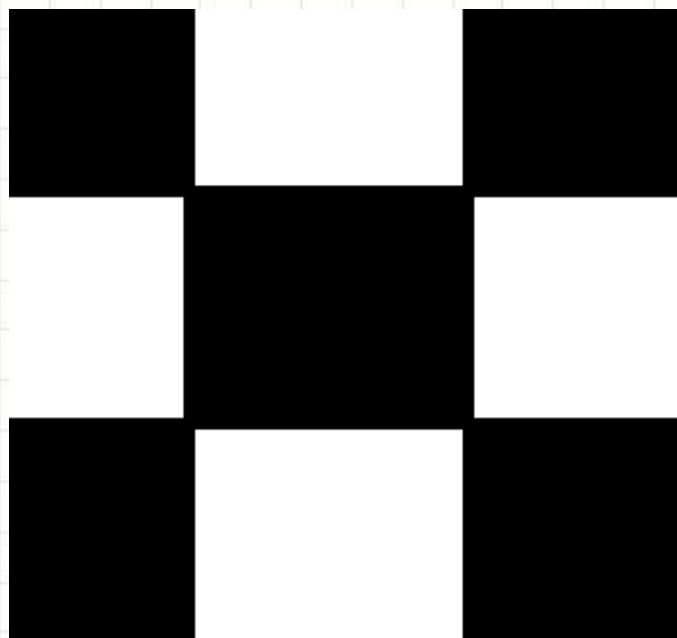


1	0	-1
2	0	-2
1	0	-1



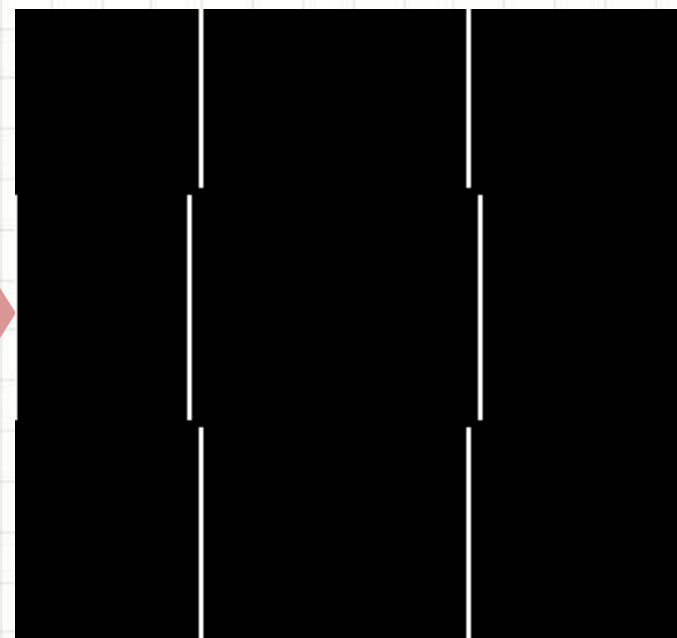
G_y

اعمال فیلترهای سو بل یک تصویر



F

قدرمطلق فیلتر
سو بل با ماسک
عمودی



G_y

فیلترهای Sobel

ماسک سوبل افقی بیشتر لبه‌های افقی را مشخص می‌کند و ماسک سوبل عمودی، لبه‌های عمودی را مشخص می‌کند.
برای مشخص شدن کلیه لبه‌ها:

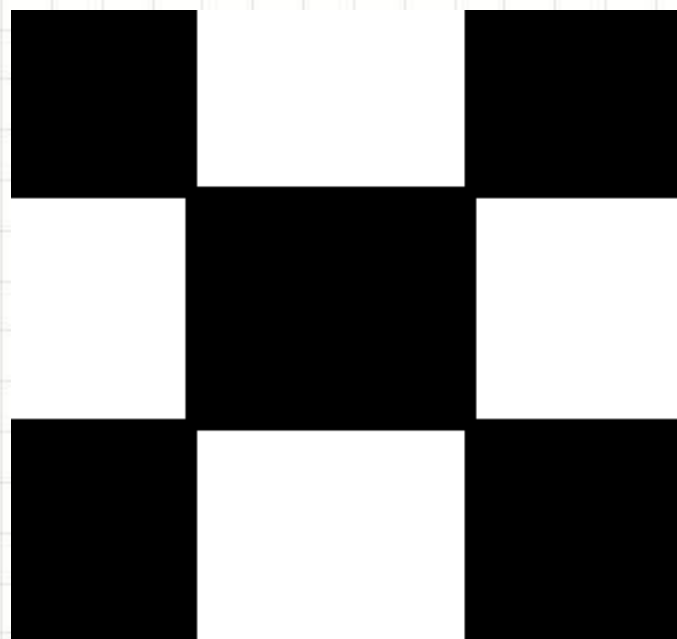
اگر G_x و G_y تصاویر فیلتر شده به وسیله ماسک افقی و عمودی باشند، آن‌گاه تصویر

$$\sqrt{[G_x]^2 + [G_y]^2}$$

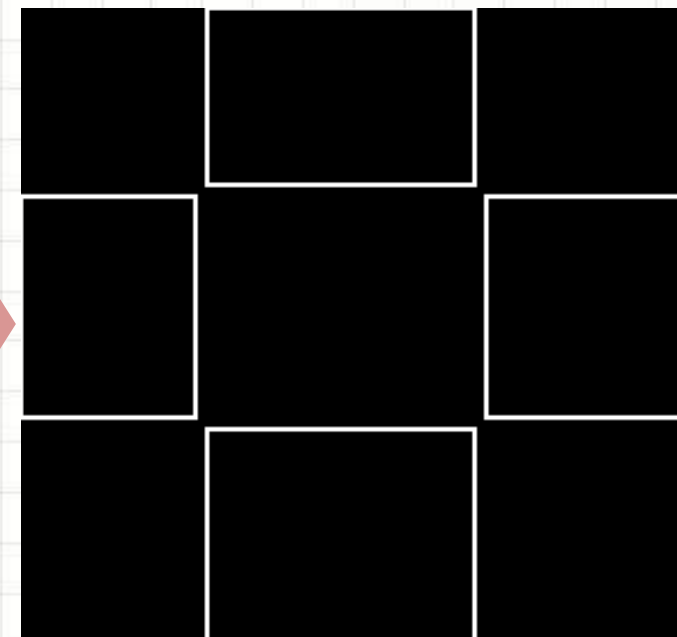
لبه‌های تصویر را بهتر نشان می‌دهد.
روال فوق به **عملگر یا الگوریتم سوبل** موسوم است.

در عمل، به منظور کاهش هزینه محاسبات، به جای $\sqrt{[G_x]^2 + [G_y]^2}$ می‌توان از تقریب $|G_x| + |G_y|$ استفاده می‌شود. توجه شود که نتیجه این دو فرمول تقریباً یکسان است ولی فرمول دوم با هزینه کمتری قابل محاسبه است.

اعمال فیلترهای سو بل یک تصویر



F



$$\sqrt{[G_x]^2 + [G_y]^2}$$

اعمال فیلترهای سو بل یک تصویر



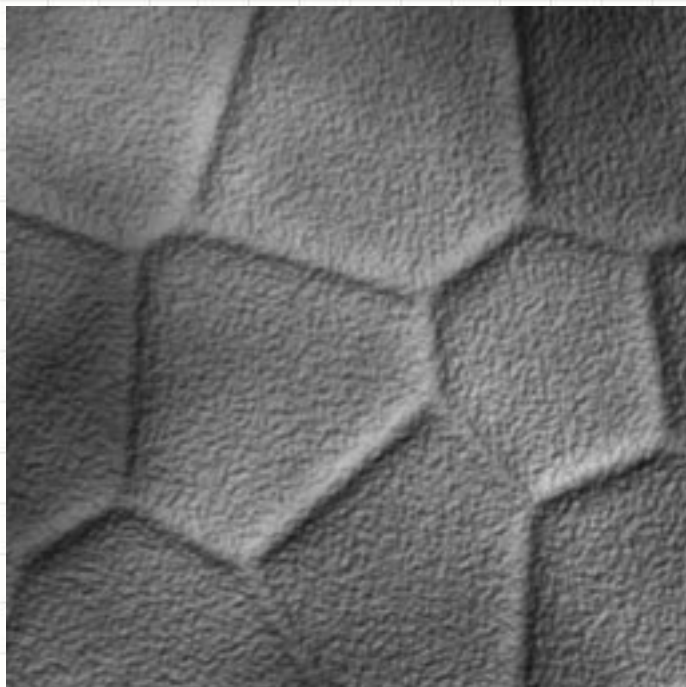
F

عملگر سو بل



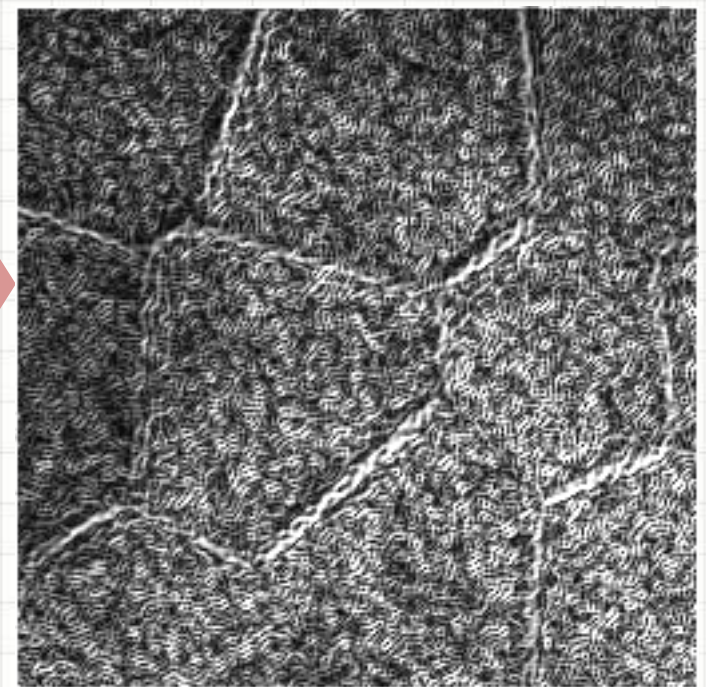
$|G_x| + |G_y|$

اعمال فیلترهای سو بل یک تصویر



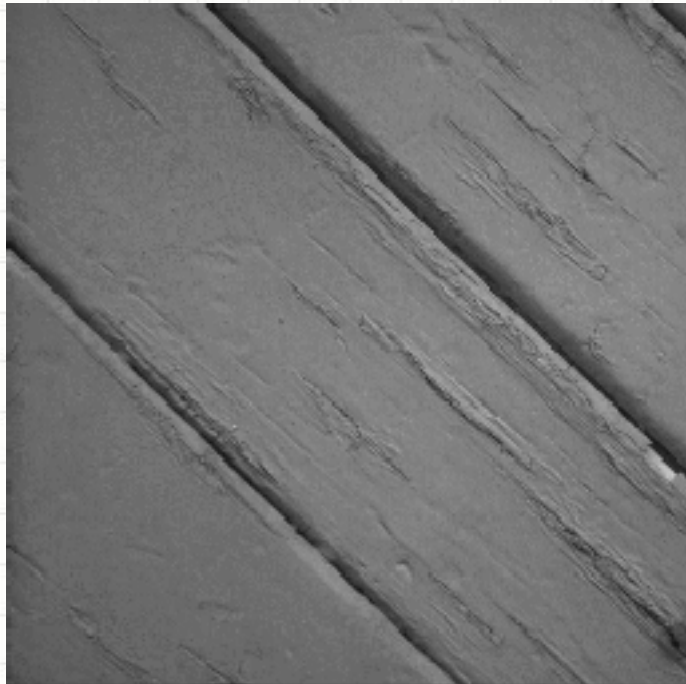
F

عملگر سو بل



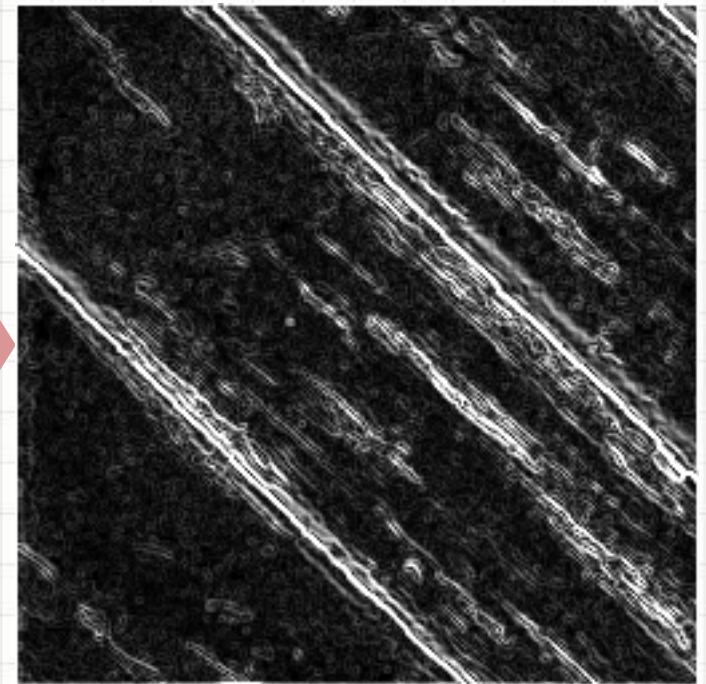
$|G_x| + |G_y|$

اعمال فیلترهای سو بل یک تصویر



F

عملگر سو بل



$|G_x| + |G_y|$

مقایسه فیلتر لاپلاسین و سوبل

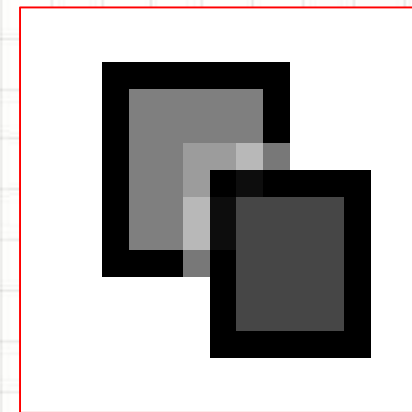
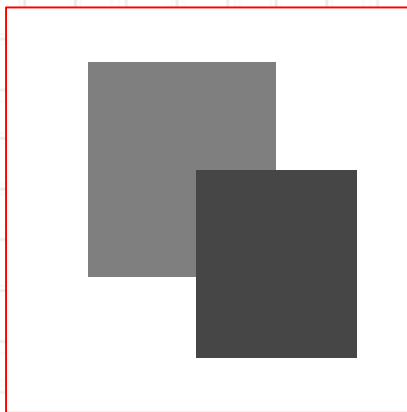
فیلتر لاپلاسین نسبت به فیلتر سوبل، لبه‌های نازک‌تری را تولید می‌کند.
فیلتر لاپلاسین حساسیت بیشتری به جزئیات کوچک مانند خط‌های نازک دارد.



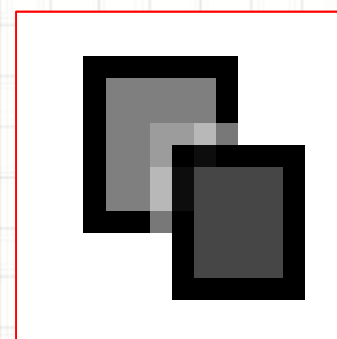
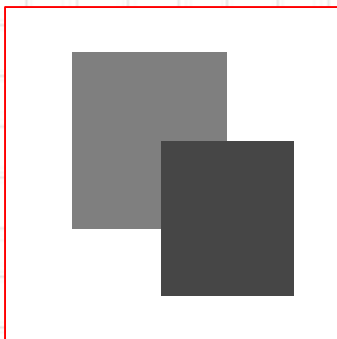
تیز کردن لبه‌های تصویر (Sharpening)

تیز کردن لبه‌های تصویر (Sharpening)

- تیزی (Sharpness) به معنی تمایز ملموس (contrast) بین رنگ‌ها می‌باشد.
- تیزی یک انتقال سریع از تاریکی به روشنی و یا روشنی به تاریکی می‌باشد.
- تیز کردن لبه‌ها به معنی افزایش تمایز سطوح خاکستری با رنگ‌ها در لبه‌ها. توجه شود که لبه‌ها محل برخورد رنگ‌ها و یا سطوح خاکستری متفاوت می‌باشد.



تیز کردن لبه‌های تصویر (Sharpening)



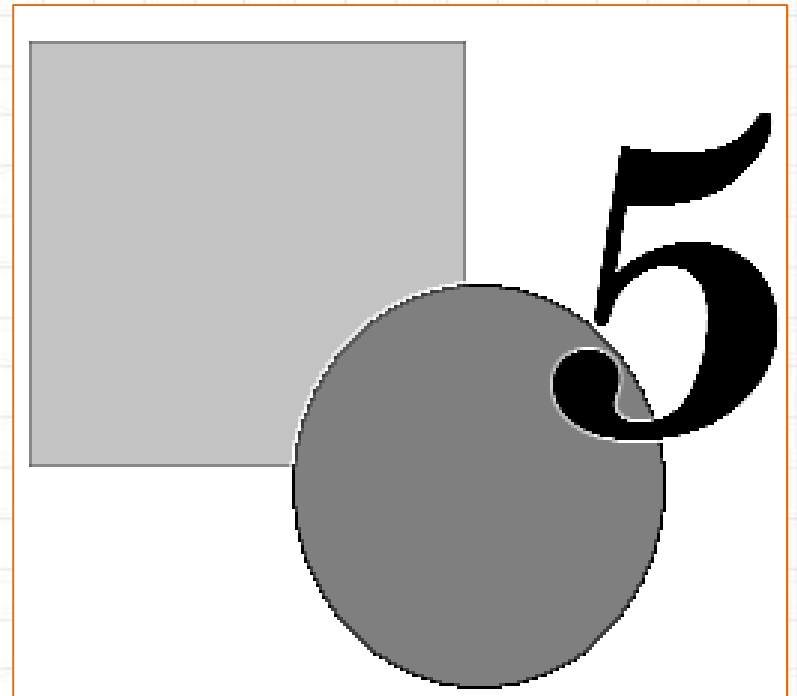
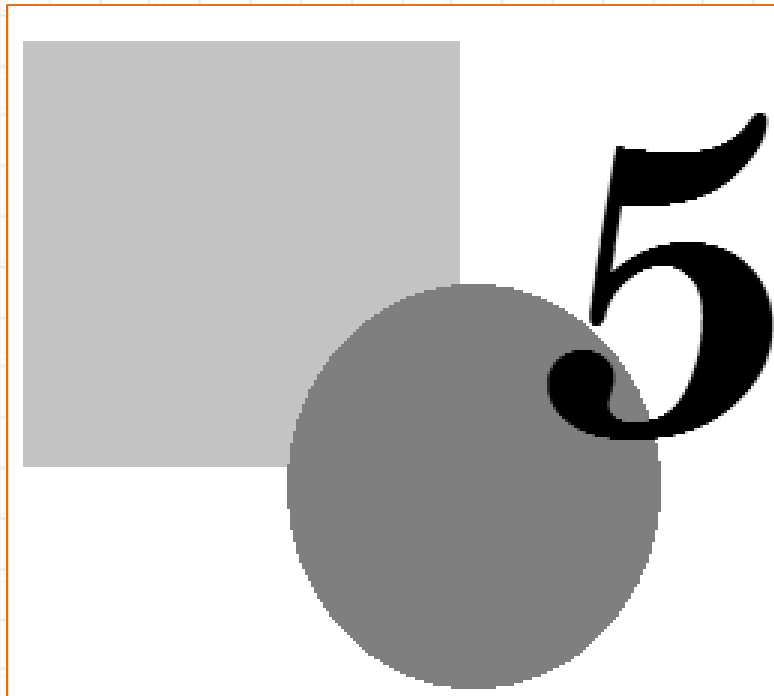
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	0	0	0	0	0	0	0	255	255	255	255	255
255	255	255	0	127	127	127	127	127	0	255	255	255	255	255
255	255	255	0	127	127	127	127	127	0	255	255	255	255	255
255	255	255	0	127	127	156	156	184	120	255	255	255	255	255
255	255	255	0	127	127	156	0	13	0	0	0	0	255	255
255	255	255	0	127	127	184	13	70	70	70	70	0	255	255
255	255	255	0	127	127	184	13	70	70	70	70	0	255	255
255	255	255	0	0	0	120	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	0	0	0	0	0	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

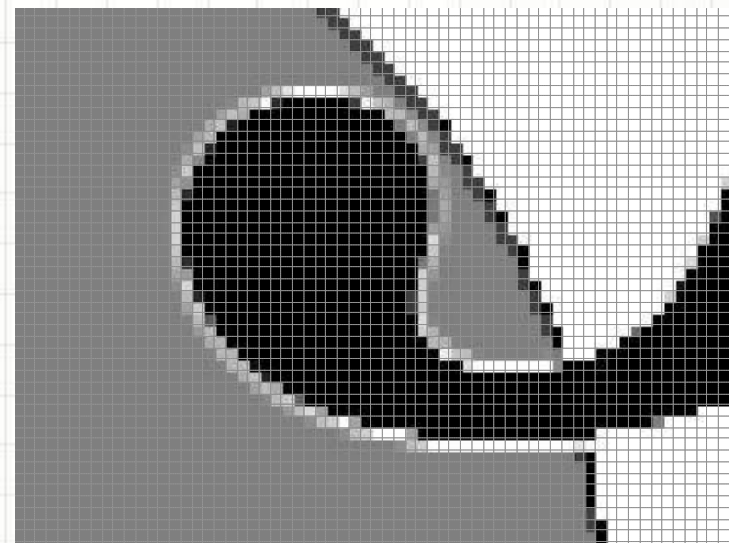
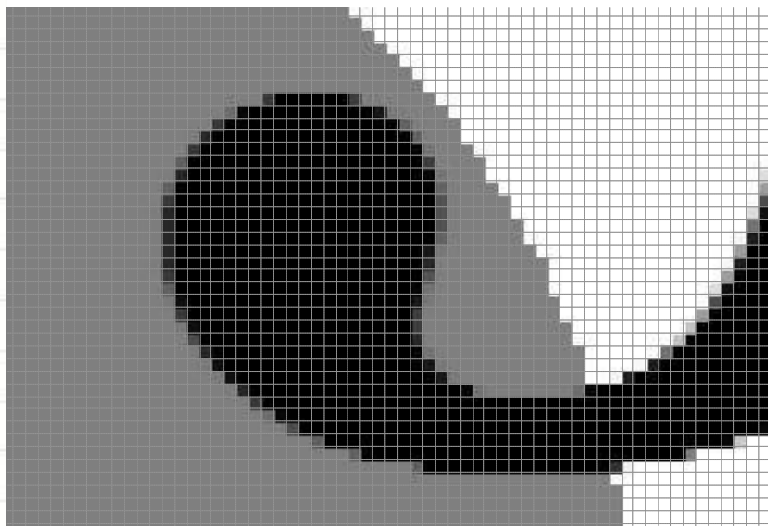
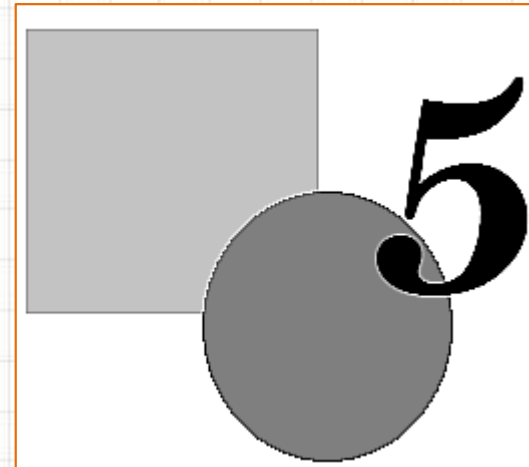
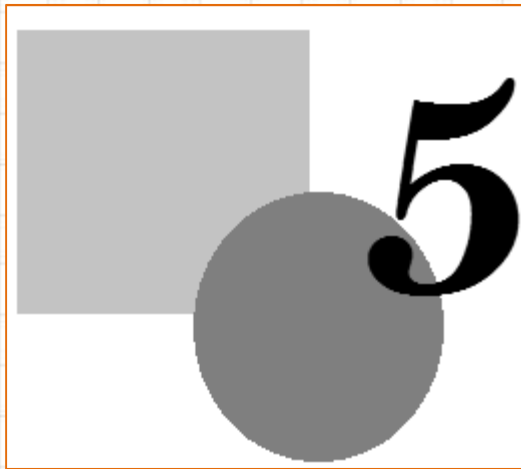
تیز کردن تصویر – مثال ۱



تیز کردن تصویر – مثال ۲



تیز کردن تصویر – مثال ۲



تیز کردن تصویر (Sharpening)

در تیز کردن تصویر هدف برجسته کردن جزئیات می باشد. در واقع با تیز کردن تصویر دو اتفاق زیر می افتد:

1- مات بودن تصویر کاهش می یابد.

2- لبه ها برجسته می شوند.

بنابراین تیز کردن تصویر متضاد هموار کردن (ملایم کردن) تصویر می باشد.

هموار کردن (ملایم کردن) تصویر با کمک فیلتر قابل انجام می باشد. برای تیز کردن تصویر نیز می توان از فیلتر استفاده کرد.

فیلترهایی که برای تیز کردن تصویر به کار می روند به

فیلترهای تیز کننده (Sharpening Filter)

موسوم می باشند.

ایده فیلترهای تیز کننده

فیلترهای تیز کننده بر مبنای فیلترهای لبه یاب ساخته می باشند. همان گونه که ملاحظه گردید، فیلترهای لبه یاب، لبه ها را برجسته می کند و قسمت های غیر لبه را به پس زمینه ای تاریک و بدون ویژگی بارز تبدیل می کند.

اما می توان اطلاعات پس زمینه را در تصویر فیلتر شده بازیابی کرد. برای انجام این کار تصویر اصلی را از تصویر فیلتر شده تفریق می کنیم.



بر مبنای هر کدام از فیلترهای لبه یاب می توان یک فیلتر تیز کننده ساخت. اما در عمل بیشتر از فیلتر لاپلاس برای تیز کردن لبه لبه ها استفاده می شود که در این صورت آن را فیلتر تیز کننده لاپلاس می نامیم.

فیلتر لاپلاس برای تیز کردن تصویر - مثال



تصویر اصلی: I

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

تصویر اصلی: L

-382.5	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-382.5
-255	0	-64	-64	-128	-128	-128	-128	-128	-64	-64	0	0	0	-255
-255	0	-64	192	128	128	128	128	128	192	-64	0	0	0	-255
-255	0	-128	128	0	0	0	0	0	128	-128	0	0	0	-255
-255	0	-128	128	0	0	0	0	0	128	-128	0	0	0	-255
-255	0	-128	128	0	0	-28.5	-28.5	-57	7	-249	-185	-92.5	-92.5	-255
-255	0	-128	128	0	0	-28.5	85.5	57	121	121	185	277.5	-92.5	-255
-255	0	-128	128	0	0	-57	57	0	0	0	0	185	-185	-255
-255	0	-128	128	0	0	-57	57	0	0	0	0	185	-185	-255
-255	0	-64	192	128	128	7	121	0	0	0	0	185	-185	-255
-255	0	-64	-64	-128	-128	-249	121	0	0	0	0	185	-185	-255
-255	0	0	0	0	0	-185	185	0	0	0	0	185	-185	-255
-255	0	0	0	0	0	-92.5	277.5	185	185	185	185	277.5	-92.5	-255
-255	0	0	0	0	0	-92.5	-92.5	-185	-185	-185	-185	-92.5	-92.5	-255
-382.5	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-255	-382.5

فیلتر لاپلاسین بدون نرمال سازی

فیلتر لاپلاس برای تیز کردن تصویر – مثال

تفریق تصویر اصلی از لاپلاسین تصویر: $I - L$

637.5	510	510	510	510	510	510	510	510	510	510	510	510	510	637.5
510	255	319	319	383	383	383	383	319	319	255	255	255	510	
510	255	319	-65	-1	-1	-1	-1	-1	-65	319	255	255	510	
510	255	383	-1	127	127	127	127	127	-1	383	255	255	510	
510	255	383	-1	127	127	127	127	127	-1	383	255	255	510	
510	255	383	-1	127	127	155.5	155.5	184	120	504	440	347.5	347.5	510
510	255	383	-1	127	127	155.5	15.50	13	-51	-51	-115	-207.5	347.5	510
510	255	383	-1	127	127	184	13	70	70	70	70	-115	440	510
510	255	383	-1	127	127	184	13	70	70	70	70	-115	440	510
510	255	319	-65	-1	-1	120	-51	70	70	70	70	-115	440	510
510	255	319	319	383	383	504	-51	70	70	70	70	-115	440	510
510	255	255	255	255	255	440	-115	70	70	70	70	-115	440	510
510	255	255	255	255	255	347.5	207.5	-115	-115	-115	-115	207.5	347.5	510
510	255	255	255	255	255	347.5	347.5	440	440	440	440	347.5	347.5	510
637.5	510	510	510	510	510	510	510	510	510	510	510	510	510	637.5

تصویر تیز شده

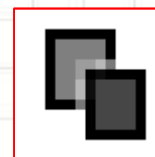
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	0	0	0	0	0	0	0	255	255	255	255	255
255	255	255	0	127	127	127	127	127	0	255	255	255	255	255
255	255	255	0	127	127	127	127	127	0	255	255	255	255	255
255	255	255	0	127	127	156	156	184	120	255	255	255	255	255
255	255	255	0	127	127	156	0	13	0	0	0	0	255	255
255	255	255	0	127	127	184	13	70	70	70	70	0	255	255
255	255	255	0	127	127	184	13	70	70	70	70	0	255	255
255	255	255	0	0	0	120	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	0	0	0	0	0	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

نرمال سازی به بازه $[0, 255]$

فیلتر لاپلاس برای تیز کردن تصویر - مثال

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	127	127	127	255	255	255	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	127	127	127	127	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	70	70	70	70	70	70	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	0	0	0	0	0	0	0	255	255	255	255	255
255	255	255	0	127	127	127	127	127	0	255	255	255	255	255
255	255	255	0	127	127	127	127	127	0	255	255	255	255	255
255	255	255	0	127	127	156	156	184	120	255	255	255	255	255
255	255	255	0	127	127	156	0	13	0	0	0	0	255	255
255	255	255	0	127	127	184	13	70	70	70	70	0	255	255
255	255	255	0	127	127	184	13	70	70	70	70	0	255	255
255	255	255	0	0	0	120	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	70	70	70	70	0	255	255
255	255	255	255	255	255	255	0	0	0	0	0	0	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255



ماتریس ماسک در فیلتر تیز کننده لاپلاس

فیلتر تیز کننده لاپلاس از اعمال فیلتر لاپلاس و تفاضل آن از تصویر اصلی حاصل می شود. اما می توان برای فیلتر تیز کننده لاپلاس، مستقیماً یک ماتریس ماسک ساخت. ماتریس ماسک در فیلتر تیز کننده لاپلاس در زیر ارائه شده است.

0	0	0
0	1	0
0	0	0

-

$$\frac{1}{1+\alpha}$$

α	$1-\alpha$	α
$1-\alpha$	-4	$1-\alpha$
α	$1-\alpha$	α

=

$$\frac{1}{1+\alpha}$$

$-\alpha$	$\alpha-1$	$-\alpha$
$\alpha-1$	$\alpha+5$	$\alpha-1$
$-\alpha$	$\alpha-1$	$-\alpha$

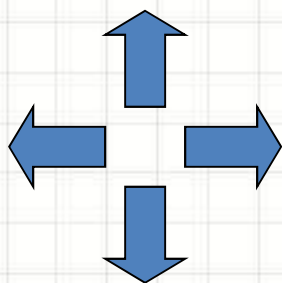
ماتریس ماسک در فیلتر تیز کننده لاپلاس به **ماسک unsharp** نیز موسوم است. توجه شود که از واژه unsharp معنی متضاد تیز کردن برداشت می شود. اما این واژه در پردازش تصویر به معنی تیز کردن معنی می شود.

فیلتر با ماسک unsharp برای تیز کردن

در زیر ماتریس ماسک unsharpen، به ازای برخی مقادیر از α ارایه شده است.

0	-1	0
-1	5	-1
0	-1	0

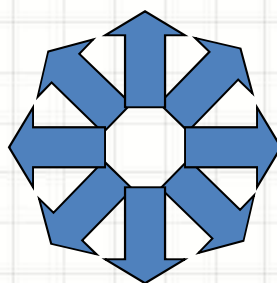
$$\alpha = 0$$



-1	-1	-1
-1	11	-1
-1	-1	-1

$$\frac{1}{3}$$

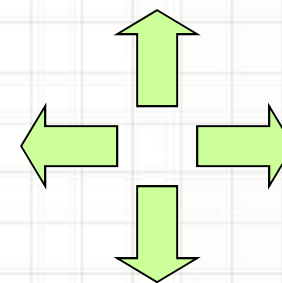
$$\alpha = 0.5$$



-1	0	-1
0	6	0
-1	0	-1

$$\frac{1}{2}$$

$$\alpha = 1$$



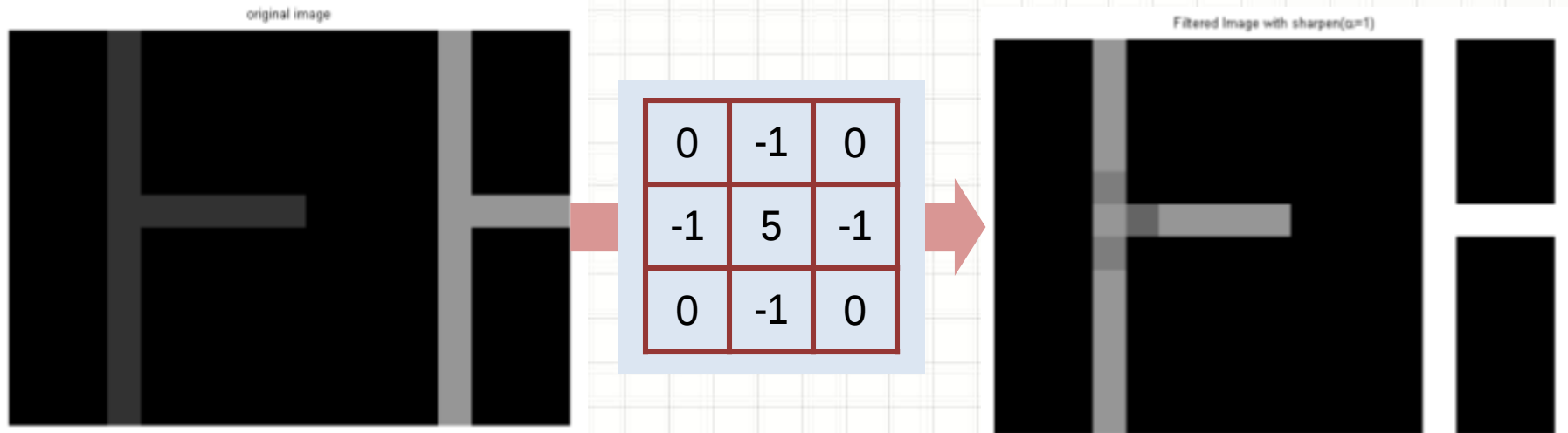
اعمال فیلتر unsharpen روی یک تصویر



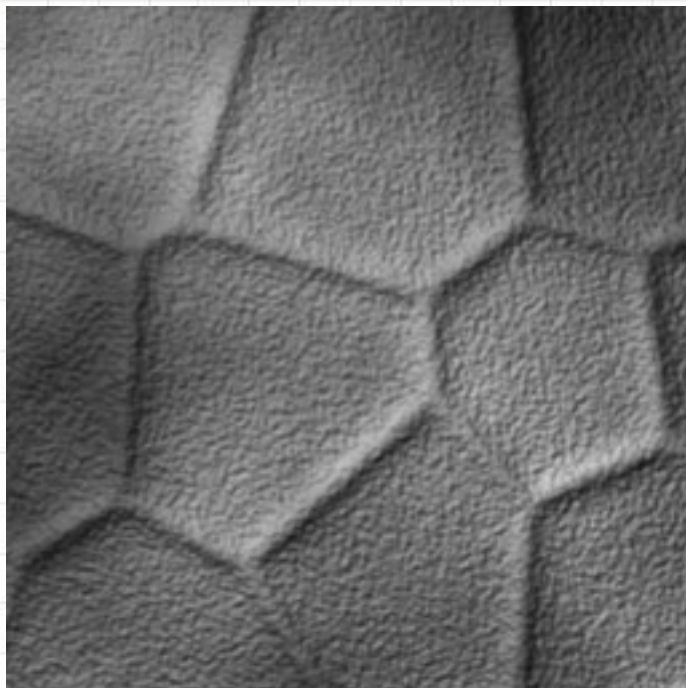
0	-1	0
-1	5	-1
0	-1	0



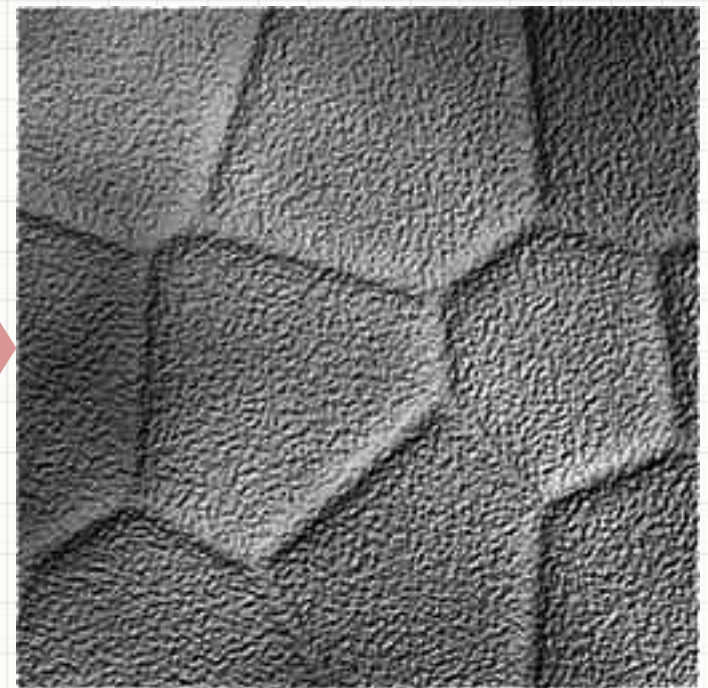
اعمال فیلتر unsharpen روی یک تصویر



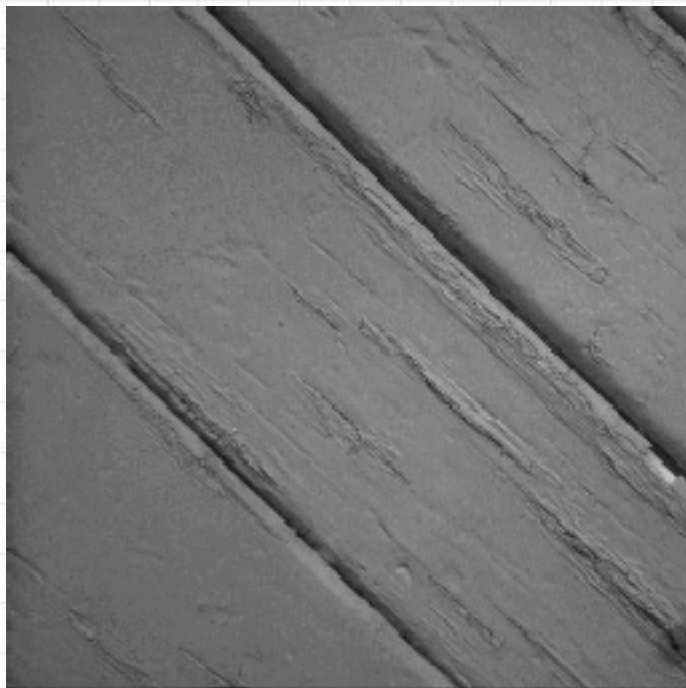
اعمال فیلتر unsharpen روی یک تصویر



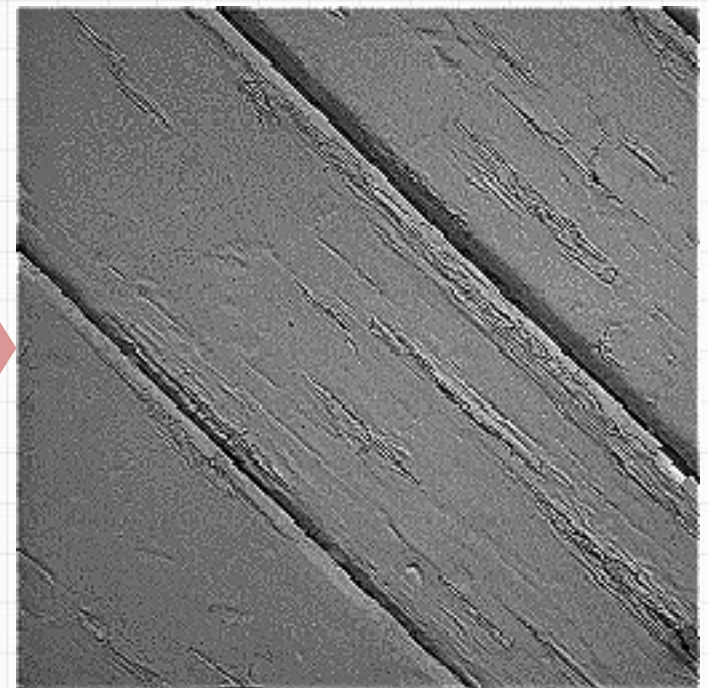
0	-1	0
-1	5	-1
0	-1	0



اعمال فیلتر unsharpen روی یک تصویر



0	-1	0
-1	5	-1
0	-1	0



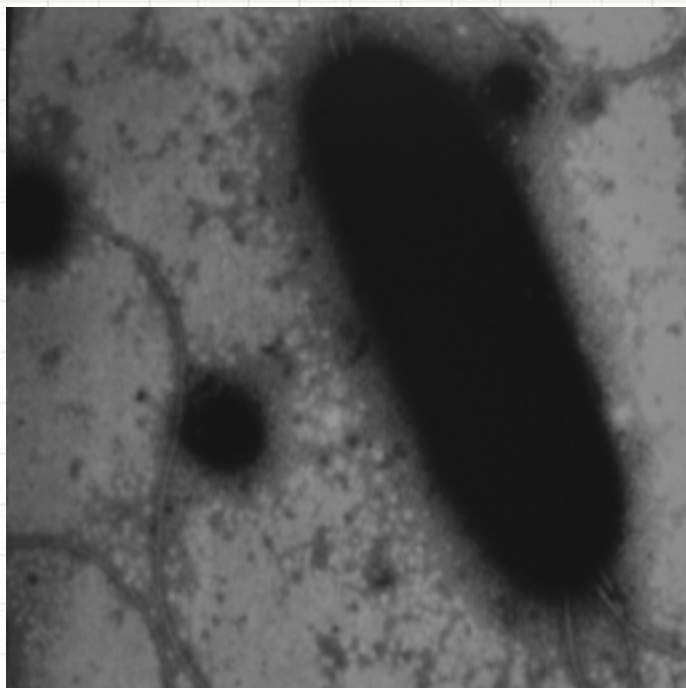
اعمال فیلتر unsharpen روی یک تصویر



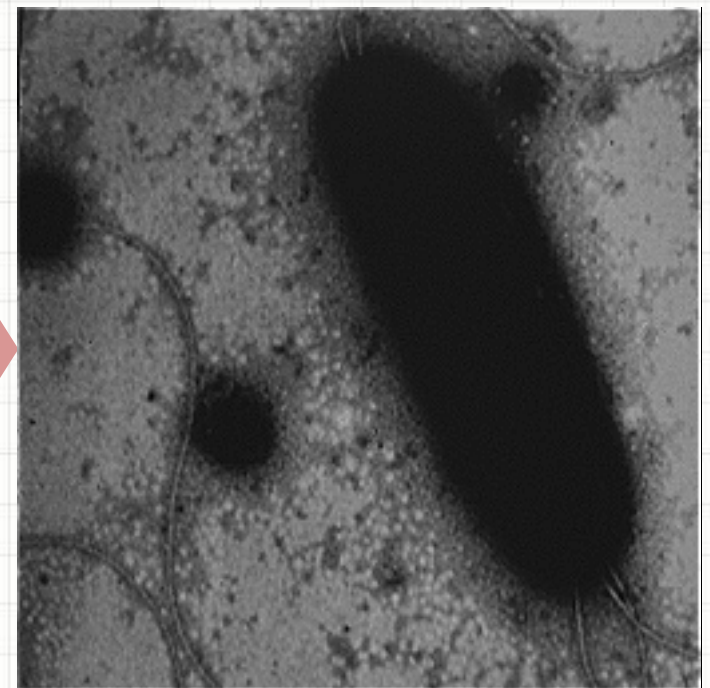
0	-1	0
-1	5	-1
0	-1	0



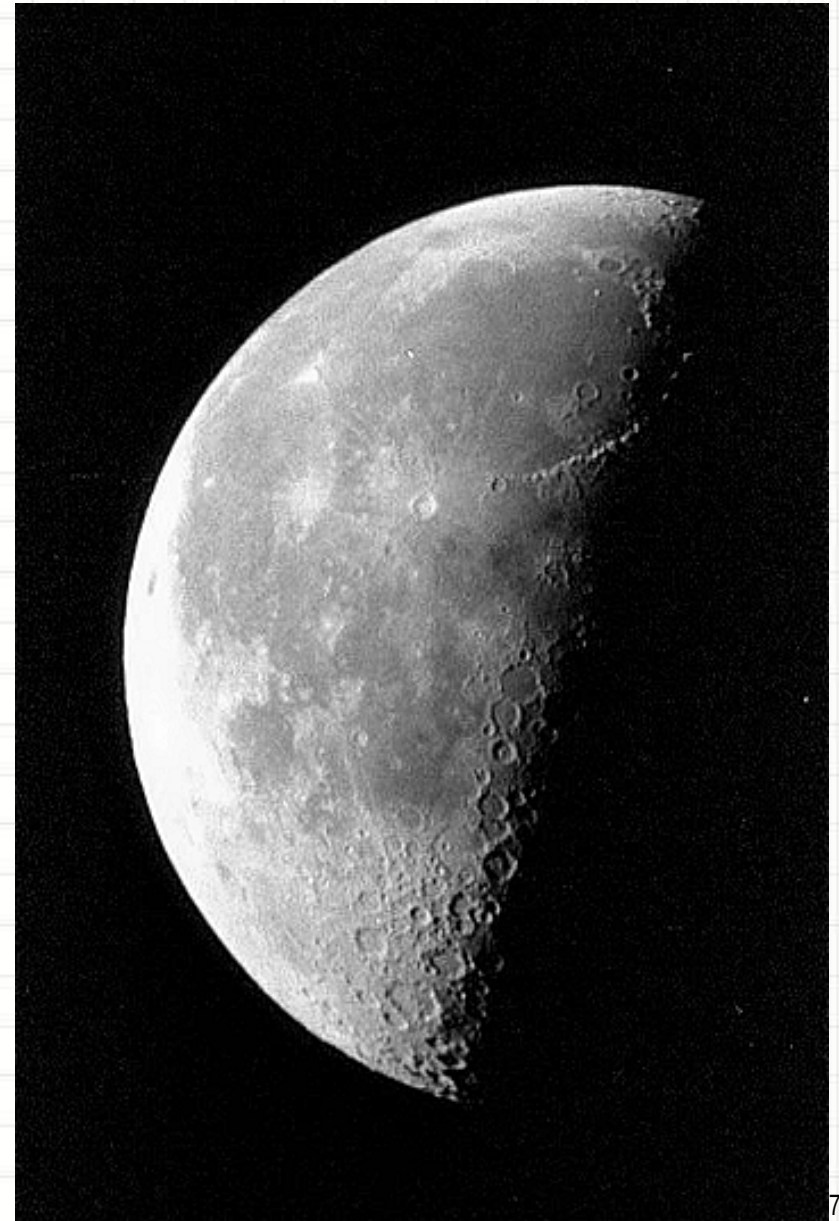
اعمال فیلتر unsharpen روی یک تصویر



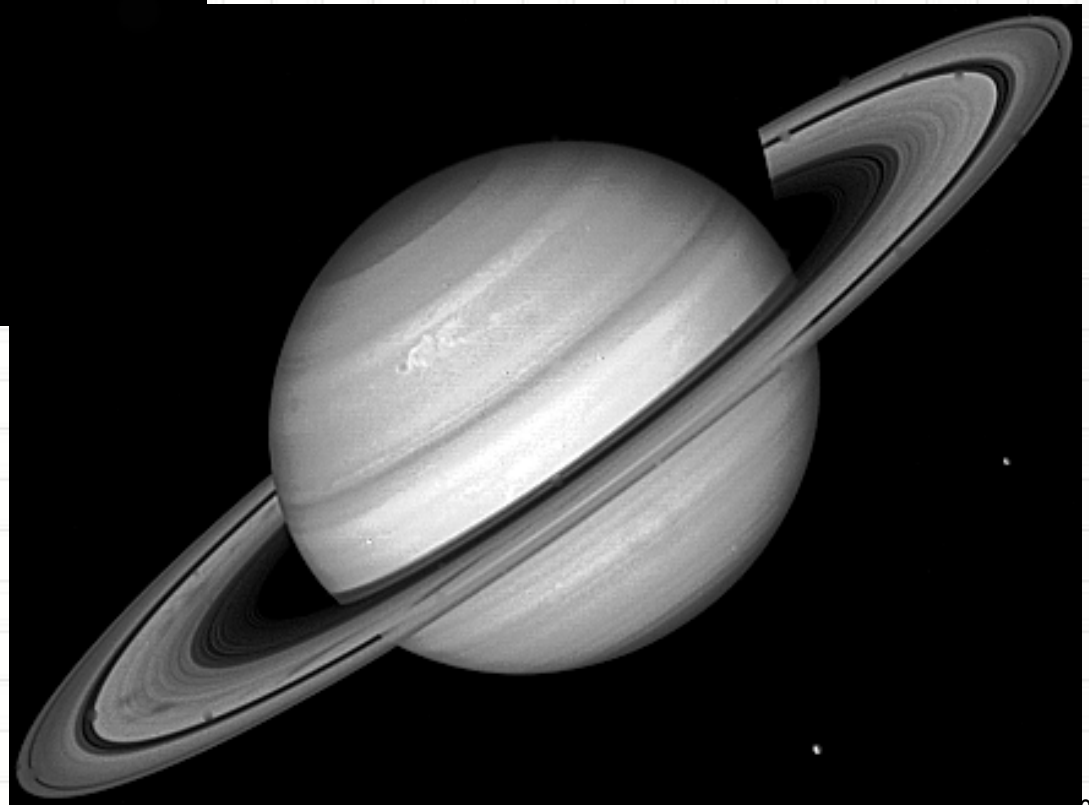
0	-1	0
-1	5	-1
0	-1	0

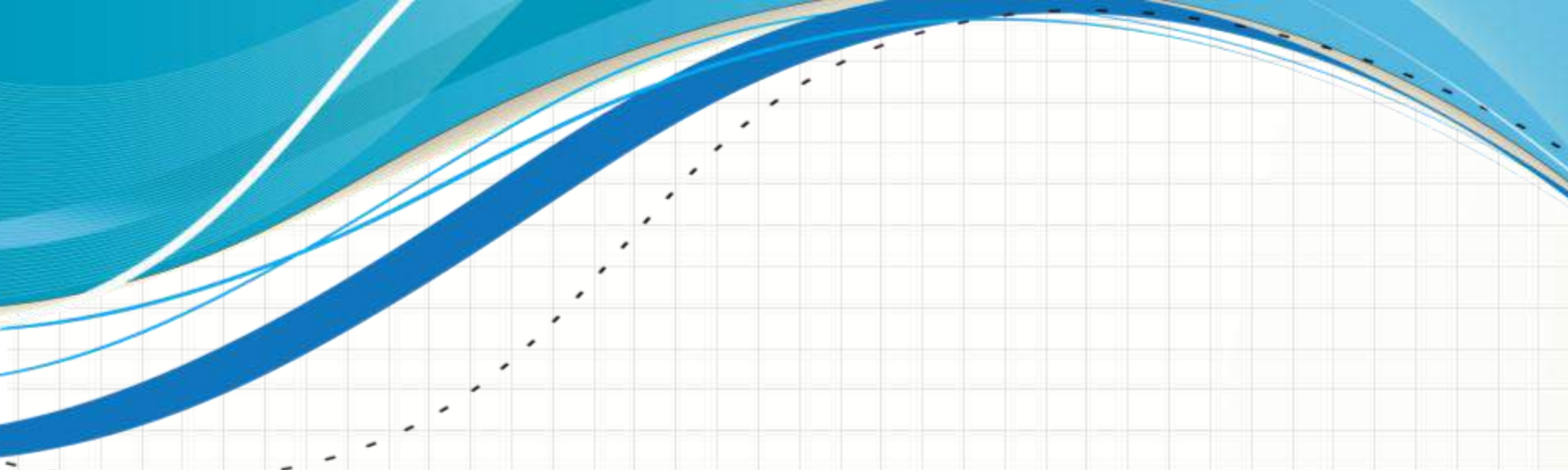


نمونه‌هایی از تیز کردن لبه‌های تصویر



نمونه‌هایی از تیز کردن لبه‌های تصویر





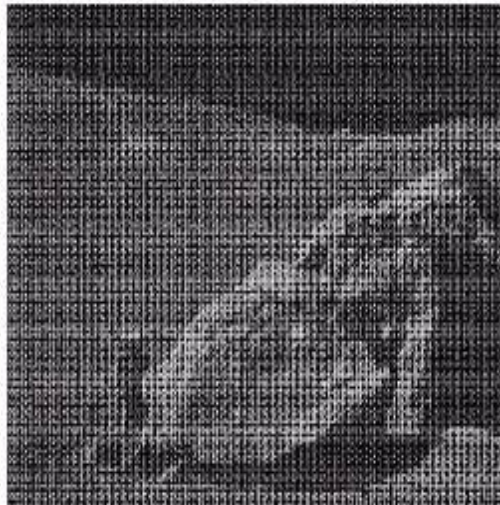
نویز در تصاویر

Noise

تعریف نویز

نویز Noise (نوفه) عبارت است از تغییرات تصادفی در شدت رنگ پیکسل‌های یک تصویر.

- نویز یکی از مشکلات کلاسیک و اولیه در پردازش تصویر می‌باشد.
- نویز ماهیت تصافی دارد و لذا از ابزار آماری برای توصیف و مطالعه آن استفاده می‌شود.
- همواره نویز وجود دارد ولی ممکن است که آنقدر کوچک باشد که قابل مشاهده و تشخیص نباشد.



تعریف نویز

از دیدگاه ماتریس تصویر، نویز یک ماتریس دیگر است که با ماتریس تصویر اصلی جمع می‌شود و باعث تغییر شدت رنگ پیکسل‌های تصویر اصلی می‌گردد.



+



=



تصویر اصلی

نویز

تصویر نویزی

انواع نویز در تصاویر

از نمونه‌های نویز در تصاویر می‌توان به موارد زیر اشاره کرد

1. نویز گوسی (Gaussian Noise)
2. نویز پواسونی (Poisson Noise)
3. نویز نمک و فلفل (salt & pepper Noise)
4. نویز خال خالی (Speckle Noise)
5. نویز متناوب (Periodic Noise)

نویز گوسی

در نویز گوسی شدت رنگ پیکسل (x,y) به اندازه $n(x,y)$ تغییر می کند که $n(x,y)$ یک متغیر تصادفی است که از توزیع نرمال با میانگین $\mu = 0$ و واریانس σ تبعیت می کند.

به عبارت دیگر مولفه های ماتریس نویز اعدادی تصادفی با توزیع نرمال $N(0,\sigma^2)$ می باشند.

معمولا در تصاویر مقدار واریانس σ کوچک می باشد و لذا میزان تغییر شدت پیکسل ها نیز کوچک است. همین تغییرات کوچک باعث افت کیفیت تصویر می شود.

نویز گوسی - مثال

تصویر اولیہ

[illegible]

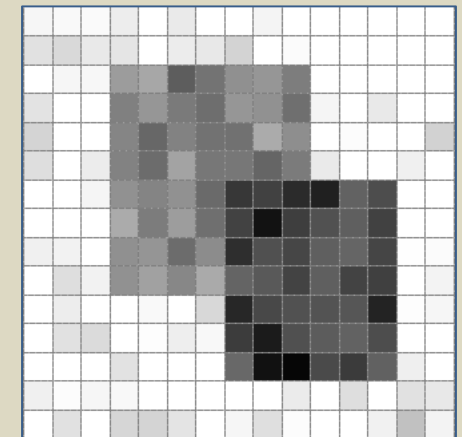
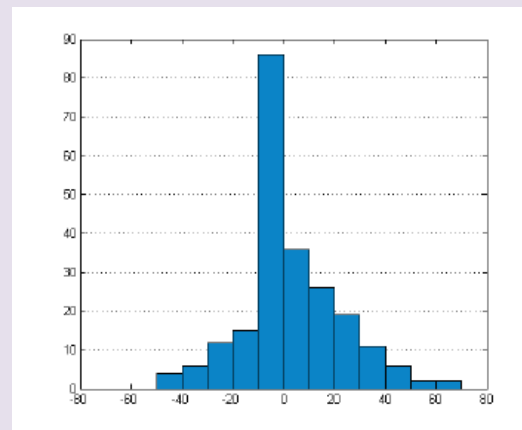
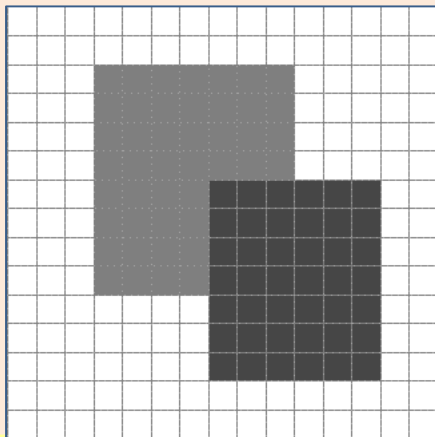
نویز نرمال ($\sigma = 0.1$ و $\mu = 0$)

9	6	5	18	0	21	0	0	11	0	0	0	0	0	0
30	38	22	26	0	19	23	44	0	4	0	0	0	0	0
0	9	8	-29	-40	34	12	-17	-23	2	0	0	0	0	0
28	0	0	-1	-23	5	17	-23	-18	16	10	0	22	0	0
43	0	0	-6	24	-3	13	14	-44	-15	0	3	0	0	45
33	0	19	-3	19	-36	8	8	26	4	21	0	0	15	0
0	0	10	-18	-5	-18	21	15	5	27	38	-28	-7	0	0
0	0	0	-44	3	-30	16	4	52	8	-11	-25	5	0	0
15	13	0	-17	-22	19	-13	24	-10	0	-25	-31	1	1	4
0	33	13	-18	-34	-8	-43	-30	-19	3	-25	3	6	0	12
0	18	0	0	6	0	39	31	-2	-11	-13	-16	35	2	9
0	30	36	0	2	17	7	10	43	-10	-22	-28	-7	0	0
0	0	0	29	0	0	0	-34	53	64	-4	11	-27	16	0
15	3	9	7	0	0	0	0	0	20	0	33	0	30	23
0	30	0	41	43	27	0	9	32	3	0	0	14	62	12



تصویر نویزی با نویز گوسی ($\sigma = 0.1$)

246	249	250	237	255	234	255	255	244	255	255	255	255	255	255	255
225	217	233	229	255	236	232	211	255	251	255	255	255	255	255	255
255	246	247	156	167	93	115	144	150	125	255	255	255	255	255	255
227	255	255	128	150	122	110	150	145	111	245	255	233	255	255	255
212	255	255	133	103	130	114	113	171	142	255	252	255	255	255	210
222	255	236	130	108	163	119	119	101	123	234	255	255	240	255	255
255	255	245	145	132	145	106	55	65	43	32	98	77	255	255	255
255	255	255	171	124	157	111	66	18	62	81	95	65	255	255	255
240	242	255	144	149	108	140	46	80	70	95	101	69	254	251	255
255	222	242	145	161	135	170	100	89	67	95	67	64	255	243	255
255	237	255	255	249	255	216	39	72	81	83	86	35	253	246	255
255	225	219	255	253	238	248	60	27	80	92	98	77	255	255	255
255	255	255	226	255	255	255	104	17	6	74	59	97	239	255	255
240	252	246	248	255	255	255	255	255	235	255	222	255	225	232	255
255	225	255	214	212	228	255	246	223	252	255	255	241	193	243	255



نویز گوسی-نمونه



نویز گوسی با $\sigma = 0.001$



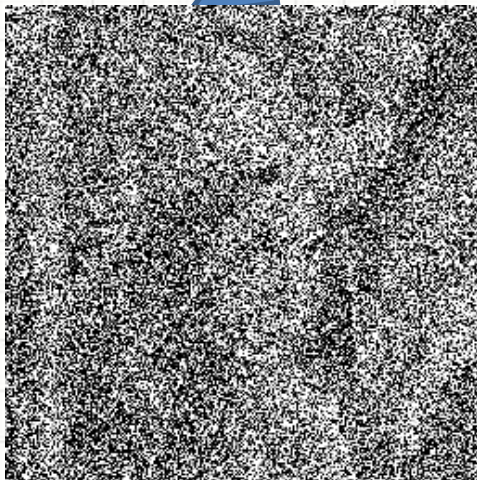
نویز گوسی با $\sigma = 0.01$



نویز گوسی با $\sigma = 0.1$



نویز گوسی با $\sigma = 1$



نویز گوسی-نمونه



نویز گوسی با $\sigma = 0.001$



نویز گوسی با $\sigma = 0.01$



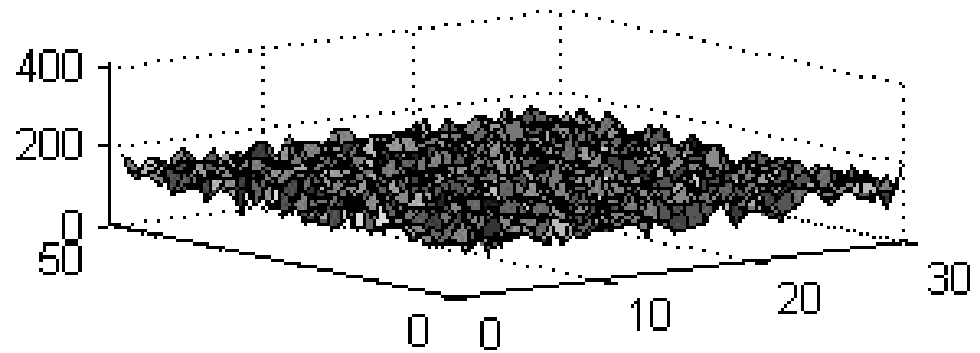
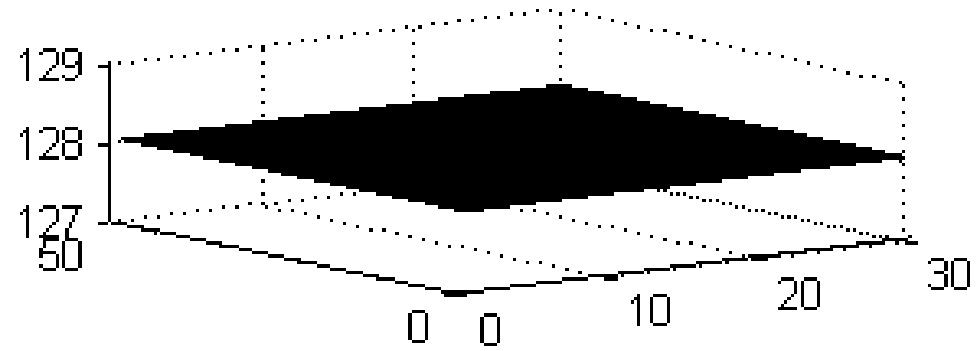
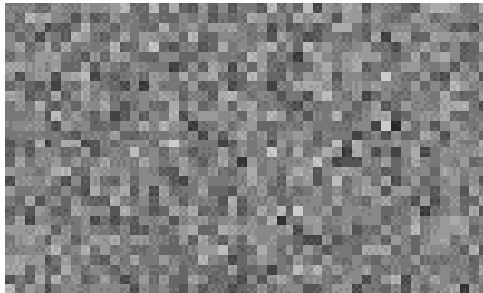
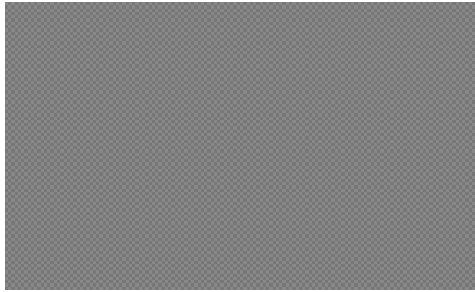
نویز گوسی با $\sigma = 0.1$



نویز گوسی با $\sigma = 1$



نویز گوسی-نمونه



نویز نمک و فلفل

در نویز نمک و فلفل، شدت بعضی از پیکسل‌ها را به شدت ماکزیمم ۲۵۵ و شدت بعضی دیگر از پیکسل‌ها به مقدار شدت می‌نیمم صفر تبدیل می‌کند. به عبارت تعدادی پیکسل به طور تصادفی انتخاب می‌شوند و مقدار آن‌ها به ۲۵۵ تبدیل می‌شود. همچنین تعدادی دیگر از پیکسل‌ها به طور تصادفی انتخاب می‌شوند و مقدار آن‌ها به صفر تبدیل می‌شود.

نویز نمک و فلفل را می‌توان یک ماتریس در نظر گرفت که مولفه‌های آن از 0، 255 و 255- تشکیل شده است.

- نویز نمک و فلفل در موارد زیر رخ می‌دهد
- تبدیل تصاویر آنالوگ به دیجیتال
- خطای بیت در انتقال اطلاعات

نویز نمک و فلفل-مثال

[illegible]

نويز نمک
و فلفل

[illegible]

نویز نمک و فلفل - نمونه



نویز نمک
و فلفل



نویز نمک و فلفل - نمونه



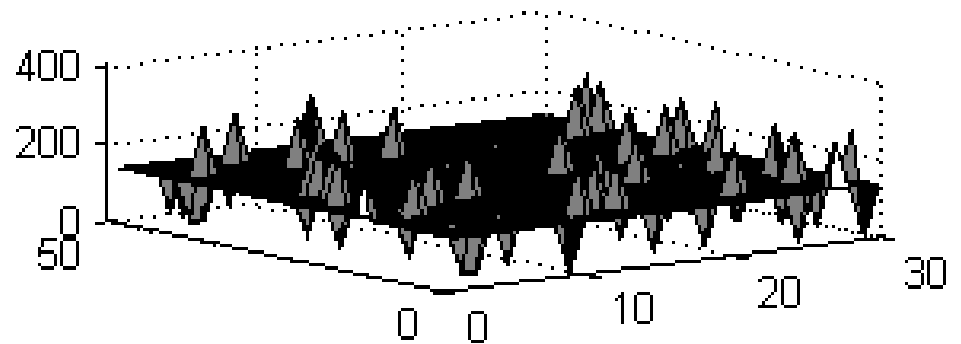
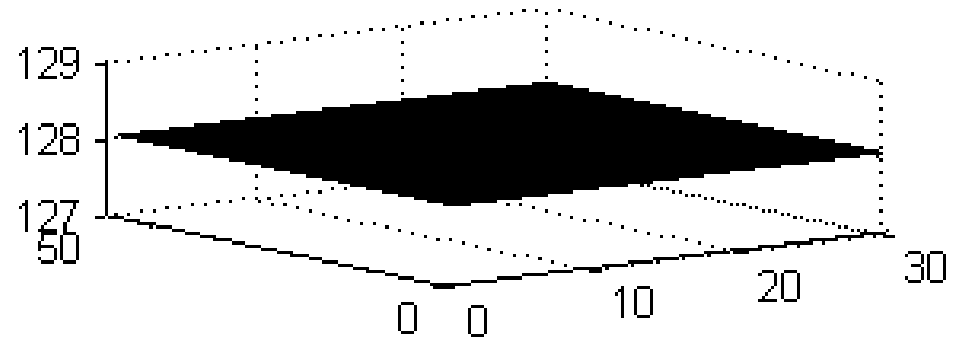
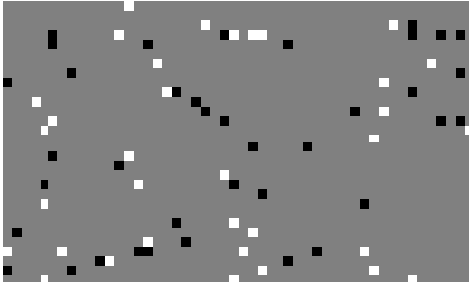
نویز نمک و فلفل-نمونه



نویز نمک
و فلفل



نویز نمک و فلفل-نمونه



حذف نویز با استفاده از فیلتر

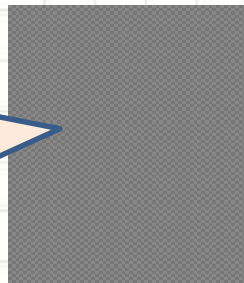
حذف نویزهای گوسی

نویز گوسی باعث ایجاد تغییراتی در مقدار شدت پیکسل‌ها می‌شود که این تغییرات به صورت تصادفی با توزیع نرمال ایجاد شده‌اند. نکته این است که اکثر این تغییرات در شدت پیکسل‌ها صفر است و یا به صفر نزدیک است. بنابراین در یک همسایگی از پیکسل‌ها، تعدادی کمی از پیکسل‌ها دارای تغییر شدت زیاد می‌باشند.

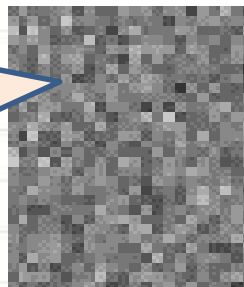
بنابراین به نظر می‌رسد که میانگین شدت‌ها دارای اطلاعات صحیح باشد. در این راستا فیلترهای میانگین برای برطرف کردن نویز گوسی مفید واقع می‌شوند.

حذف نویزهای گوسی

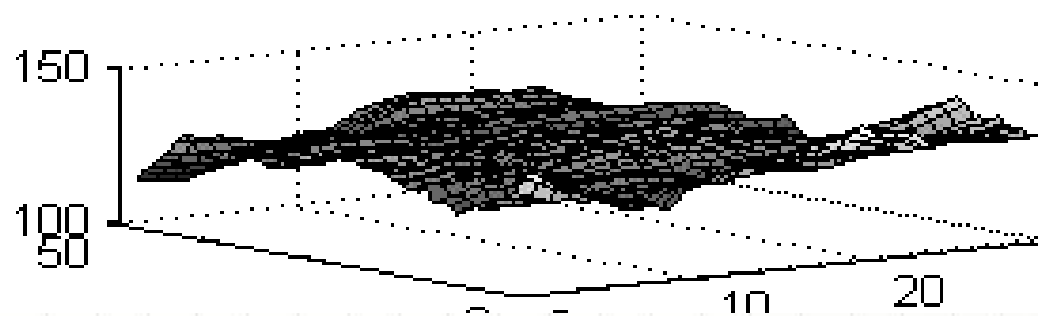
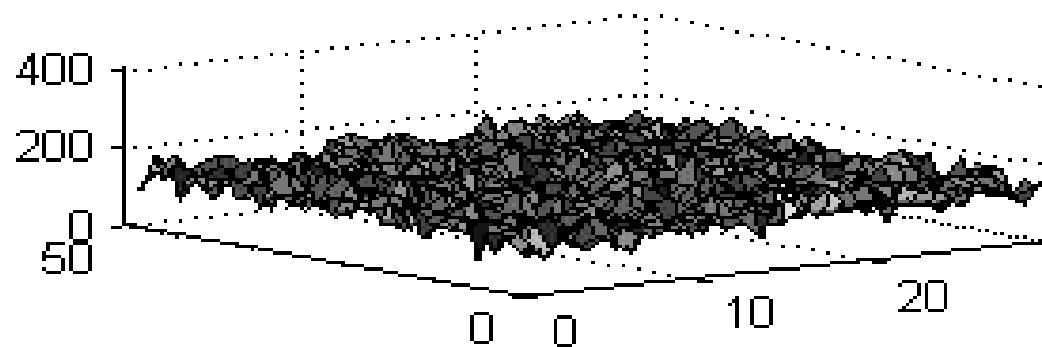
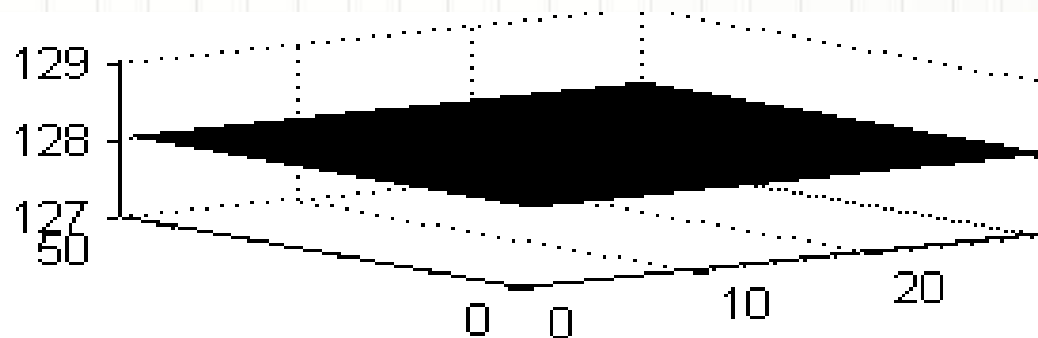
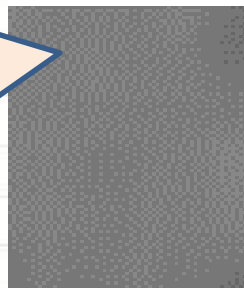
تصویر
اصلی



تصویر نویزی
نویز گوسی
 $\sigma = 0.01$

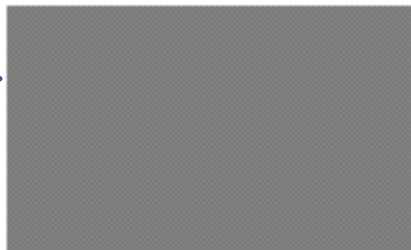


تصویر فیلتر
شده با فیلتر
میانگین
۱۱ تایی

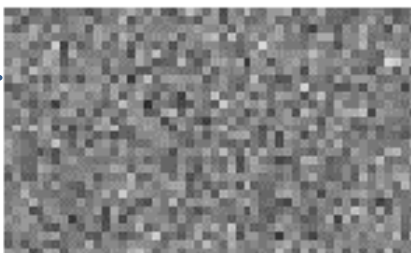


حذف نویزهای گوسی

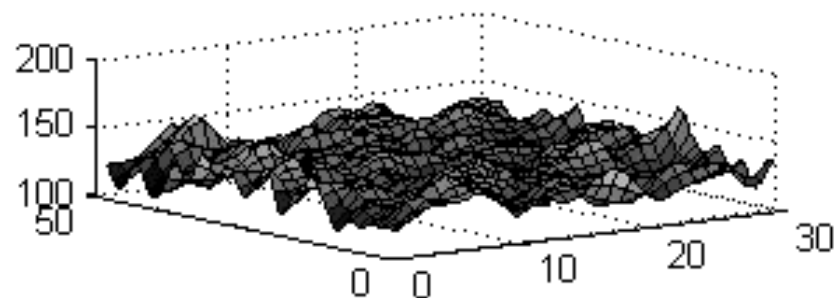
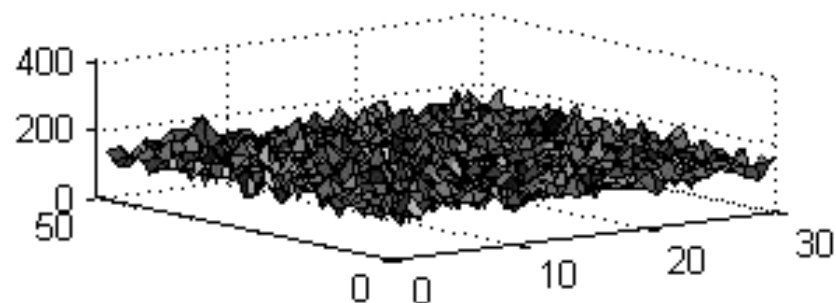
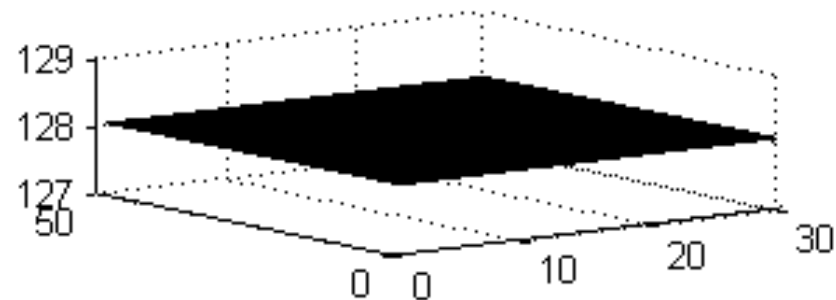
تصویر
اصلی



تصویر نویزی
نویز گوسی
 $\sigma = 0.01$



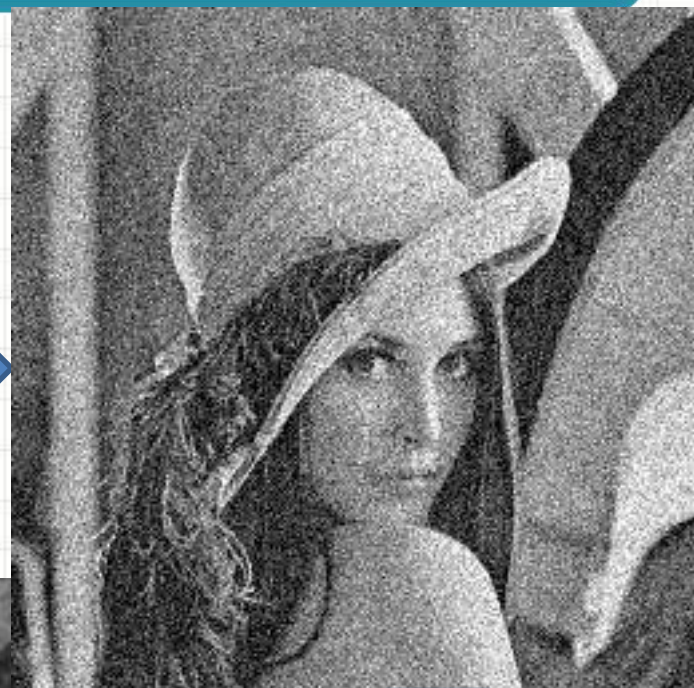
تصویر فیلتر
شده با فیلتر
گوسی ۵ تایی
 $\sigma = 1.0$



حذف نویزهای گوسی



نویز گوسی ($\sigma = 0.01$)

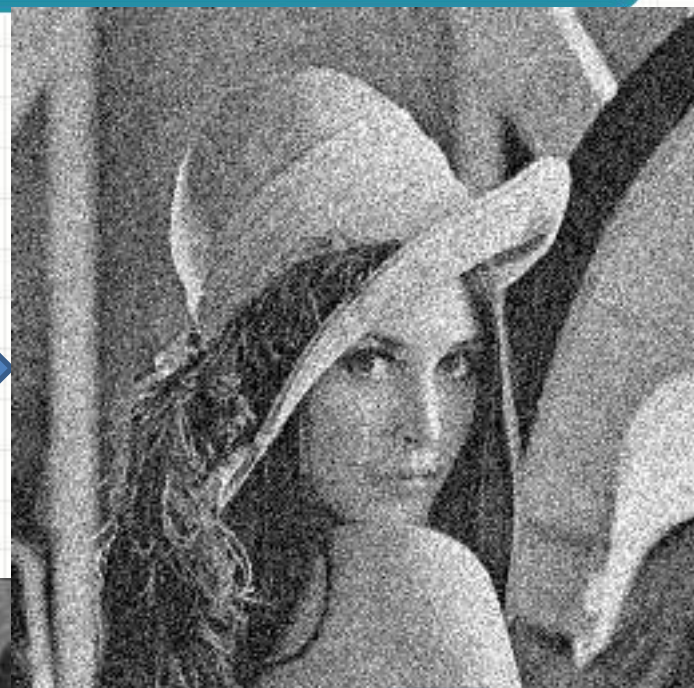


$G(3, \sigma = 1)$

حذف نویزهای گوسی



نویز گوسی ($\sigma = 0.01$)

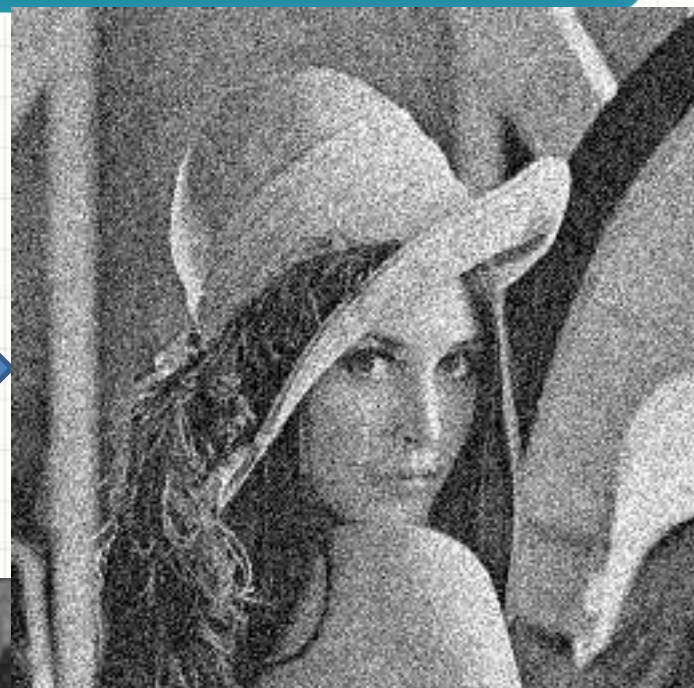


$G(3, \sigma = 5)$

حذف نویزهای گوسی



نویز گوسی ($\sigma = 0.01$)

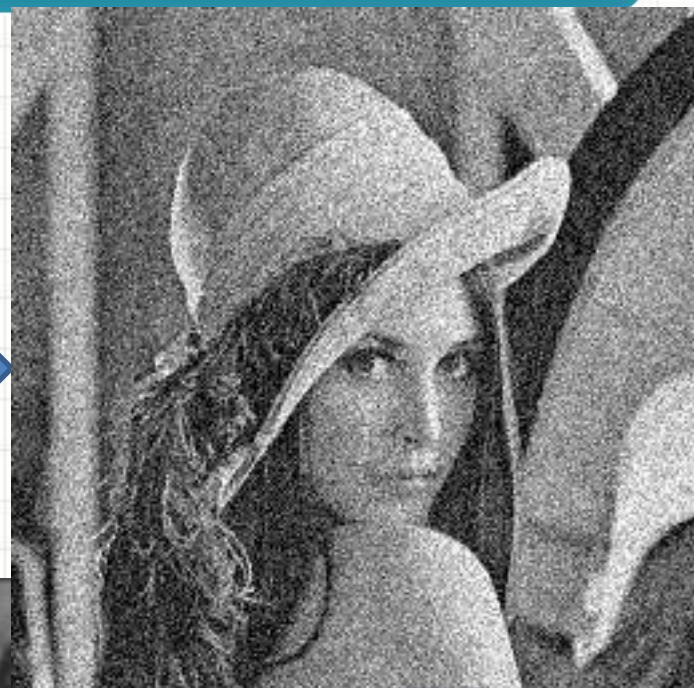


$G(5, \sigma = 1)$

حذف نویزهای گوسی



نویز گوسی ($\sigma = 0.01$)



$G(5, \sigma = 1)$

حذف نویزهای گوسی

فیلترهای میانگین گوسی باعث حذف تقریبی نویز می‌شود، اما

۱- نویز کاملاً برطرف نمی‌شود

۲- تصویر مات می‌شود.

البته انتخاب پارامتر σ و بعد ماتریس ماسک در کارایی نویززدایی فیلتر گوسی تاثیر دارد.

سوال: پارامترهای فیلتر گوسی چگونه انتخاب شوند که بهترین نویززدایی را داشته باشیم؟

حذف نویزهای نمک و فلفل

چنین به نظر می‌رسد که نویزهای نمک و فلفل هم با فیلتر میانگین از بین می‌روند.

حذف نویزهای نمک و فلفل

اما برای حذف نویزهای نمک و فلفل، فیلتر میانه بسیار بهتر از فیلترهای میانگین عمل می‌کند.

128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128

128	128	255	0	128	128	128	128	128	128
128	128	128	128	0	128	128	128	128	0
128	128	128	128	128	128	128	128	128	128
128	128	0	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
128	128	128	128	128	128	128	128	128	128
0	128	128	128	128	255	128	128	128	128
128	128	128	128	128	128	128	128	128	255
128	128	128	128	128	128	128	255	128	128

حذف نویزهای نمک و فلفل

اما برای حذف نویزهای نمک و فلفل، فیلتر میانه بسیار بهتر از فیلترهای میانگین عمل می‌کند.

225	225	225	226	226	226	226	226
225	225	255	226	226	226	225	226
226	226	225	226	0	226	226	255
255	226	225	0	226	226	226	226
225	255	0	225	226	226	226	255
255	225	224	226	226	0	225	226
226	225	225	226	255	226	226	228
226	226	225	226	226	226	226	226

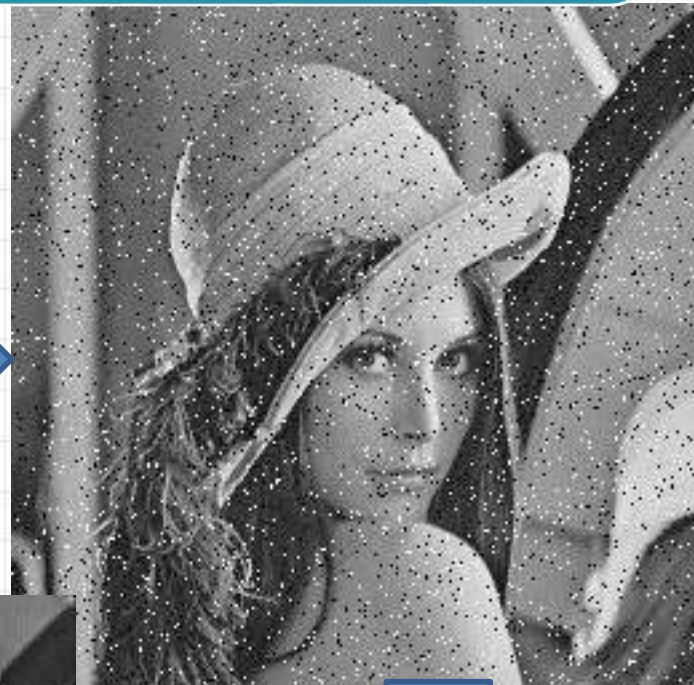
0	225	225	226	226	226	226	226
225	225	226	226	226	226	226	226
225	226	226	226	226	226	226	226
226	226	225	225	226	226	226	226
225	225	225	225	226	226	226	226
225	225	225	226	226	226	226	226
226	226	226	226	226	226	226	226

Sorted: [0, 0, 0, 225, 225, 225, 226, 226, 226]

حذف نویزهای گوسی



نویز نمک و فلفل



فیلتر میانه ۳

حذف نویزهای گوسی



نویز نمک و فلفل



?

