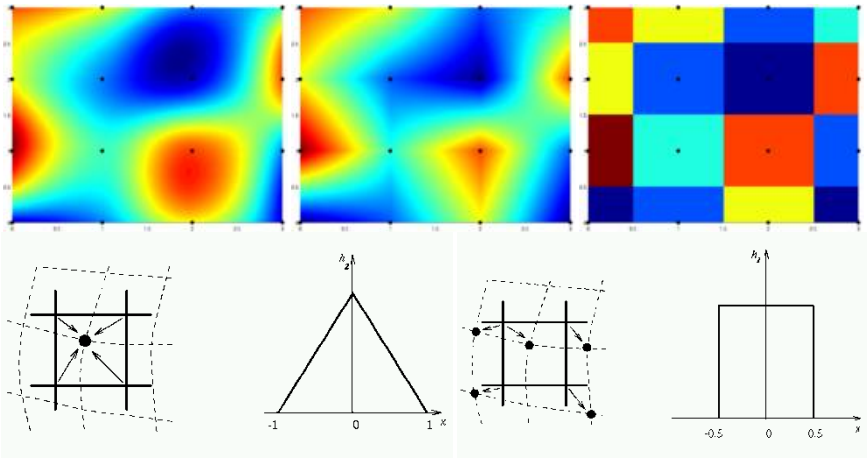
تبدیل پروژکتیو ۲ بعدی یک تصویر و بازیابی تصویر تبدیل یافته با هر یک از روشهای درون یابی

Nearest Neighbor

Bilinear Interpolation

Bicubic Interpolation



نوشته شده توسط فرید اسماعیلی (دوران دانشجویی - کارشناسی)

- **تبدیل پروژکتیو ۲ بعدی یک تصویر به فضای مورد نظر با ۴ نقطه**
- **تشکیل تصویر در فضای ثانویه با استفاده از هر یک از روشهای درون یابی زیر**

- Nearest Neighbour
- Bilinear Interpolation
- Bicubic Interpolation

- **توضیح تبدیلها و روشهای درون یابی مورد استفاده در این پروژه**
- **توضیح مراحل کار و نحوه انجام پروژه**
- **ارائه برنامه پروژه در زبان برنامه نویسی مطلب و توضیح برنامه**
- **ارائه ورودی ها، نتایج و تصاویر خروجی برنامه نوشته شده**

توضیح تصاویر روی جلد :

تصاویر رنگی روی جلد، نمایه هایی هستند که به ترتیب از راست به چپ نشان دهنده نتایج حاصل از روشهای نزدیکترین همسایگی، Bilinear و Bicubic می باشند. در زیر این تصاویر، اشکال به ترتیب از راست به چپ مفهوم نحوه اختصاص مقادیر درجات خاکستری را در ۲ روش نزدیکترین همسایگی و Bilinear نشان می دهند.

پیشگفتار

گزارشی که هم اکنون پیشرو دارید، گزارش کار پروژه انجام شده می باشد که شامل بخش های اصلی زیر است:

در *فصل اول* ابتدا مقدمه ای بر مفاهیم پایه ای پردازش رقومی تصاویر که در این پروژه مورد نیاز می باشند را توضیح خواهم داد. در بخش دوم این فصل تبدیل پروژکتیو ۲ بعدی و جزئیات آن توضیح داده می شود. سپس در بخش سوم روشهای درون یابی Nearest Neighbour ، Bilinear Interpolation ، Bicubic Interpolation شرح داده می شود.

در *فصل دوم* ابتدا روش و مراحل کار در این پروژه توضیح داده شده و سپس بخش های مختلف برنامه نوشته شده شرح داده می شود.

فصل سوم نیز شامل متن برنامه ، تصویر ورودی و تصاویر خروجی برای هر سه روش می باشد. در این فصل مقایسه ای نیز بین خروجی های حاصله از سه روش مختلف درون یابی مورد استفاده، انجام می گیرد.

نهایتاً در بخش *ضمایم* ، CD شامل فایل های برنامه نوشته شده، تصویر اولیه و تصاویر خروجی برنامه به انضمام متن گزارش کار حاضر به پیوست تقدیم می گردد.

فهرست مطالب

عنوان **صفحه**

مقدمه

پردازش تصاویر رقومی با استفاده از کامپیوتر برای داده های رقومی ذخیره شده با فرمت رقومی به کار می رود. هدف از پردازش تصاویر، واضح کردن داده های جغرافیایی و اطلاعات و عوارض موجود در تصاویر است به نحوی که در استخراج اطلاعات کیفی به کاربر کمک نماید. بنابر این به مجموعه عملیاتی که بر روی تصاویر رقومی انجام می گیرد، پردازش رقومی تصویر (Digital Image Processing) می گویند. اگر بخواهم کمی علمی تر بیان کنم، پردازش رقومی تصاویر نگاشتی از فضای تصویر به فضای تصویر است. در این گزارش من نوع خاصی از نگاشت یعنی تبدیل پروژکتیو ۲ بعدی به همراه ۳ نوع از انواع مختلف روشهای درون یابی را در فصل های آینده بیان خواهم کرد.

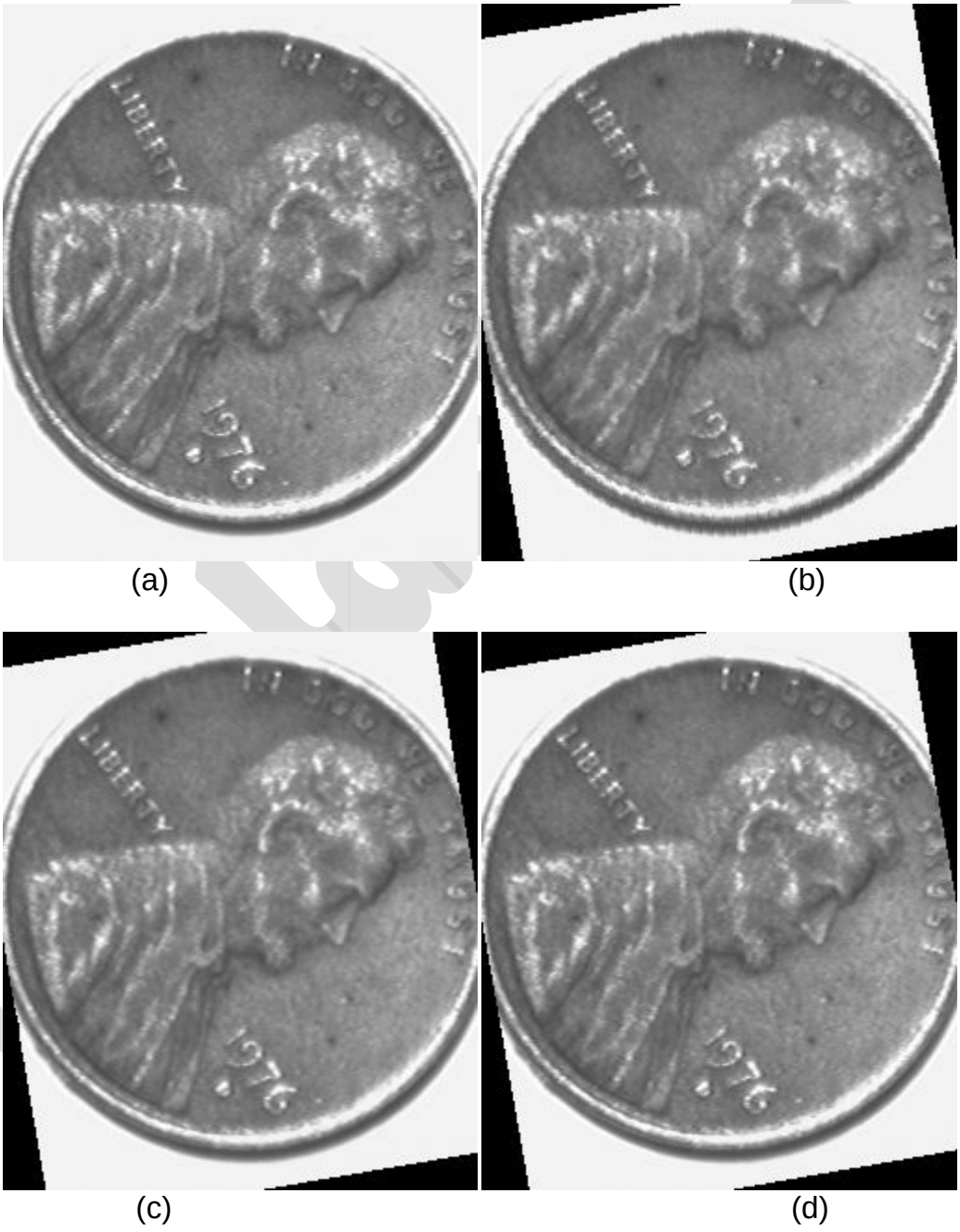
کاربرد های پردازش رقومی تصاوی در بسیاری از علوم وصنایع نمود پیدا کرده است. کاربردهای عکاسی، پزشکی، امنیتی، نظامی، سنجش از دور، صنعتی، فشرده سازی تصاویر و ... را می توان نمونه هایی از این کاربرد ها بیان نمود.

اما در ادامه و در مقدمهٔ فصل اول، به بیان مفاهیم اولیه تصاویر و پردازش رقومی که برای درک مطالب فصل های این گزارش کار به آنها نیاز داریم می پردازم.

فصل اول

مقدمه ای بر مفاهیم مورد نیاز پردازش رقومی تصاویر در این پروژه ؛

پروژکتیو ۲ بعدی و روشهای درون یابی



(a) An image of a penny. (b) - (d) Images resampled by the nearest-neighbor, bilinear, and cubic
convolution, respectively, after rotating image (a) by 10 degrees counter-clockwise about the image center.

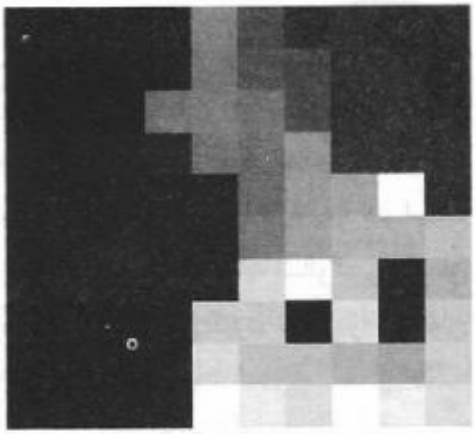
تصاویر تزئینی هستند

۱ - ۱ - مقدمه

یک تصویر رقومی در یک ردیف ۲ بعدی (شبکه) توسط جزء های تصویر کوچکی که پیکسل (Pixel) نامیده می شوند، ذخیره شده است. ترتیب یا ساختمان این شبکه ها فرمت رستری (Raster) نامیده می شود. هر پیکسل با یک شماره رقومی یا DN (Digital Number) که همان درجه خاکستری آن پیکسل می باشد، نشان داده می شود. عبارت "تصویر تکرنگ" یا به طور ساده *"تصویر"*، در حقیقت به یک تابع شدت روشنایی ۲ بعدی $f(x,y)$ اشاره می کند که x و y نشان دهنده مختصات مکانی است و مقدار f در هر نقطهٔ (x,y) متناسب با روشنایی (یا سطح خاکستری (Gray level)) تصویر در آن نقطه است. تصویر رقمی ، یک تابع تصویر $f(x,y)$ است که هم در مختصات مکانی و هم در شدت روشنایی ، گسسته شده باشد. تصویر رقمی را می توان ماتریس ۲ بعدی در نظر گرفت که هر زوج از اندیسهای سطری و ستونی آن یک نقطه از تصویر را مشخص می کند. عناصر چنین آرایه رقمی را *عناصر تصویر*، *عناصر عکس*، *پیکسلها یا پلها* می نامند. بنابر این تصویر را به سادگی می توان شبکه ای منظم از اعداد در نظر گرفت که این اعداد نشان دهنده شدت روشنایی می باشند.

در هنگام نمایش تصاویر و یا چاپ آنها ، دستگاه خروجی مقادیر هر پیکسل را تبدیل به شدت و ضعف رنگهای نمایش داده شده می کند و بنابر این یک تصویر با درجات مختلف شدت روشنایی در بخشهای آن نمایش داده می شود و می توان عوارض را در کنار یکدیگر تشخیص داد. شکل زیر نشان دهنده مفهوم این موضوع است.

12	14	16	15	164	120	70	80	69	70
15	14	10	15	160	130	120	84	84	72
13	18	19	160	163	155	120	79	82	75
55	52	60	68	154	150	180	90	83	80
42	46	45	31	30	150	182	190	250	60
33	44	46	33	27	160	188	200	200	210
30	42	42	25	24	230	250	213	30	200
12	5	8	9	220	220	265	233	33	216
13	18	4	11	231	210	210	200	190	220
10	5	3	2	250	240	230	255	240	230



یک باند تصویر به صورت خام (ماتریسی) و نمایش آن

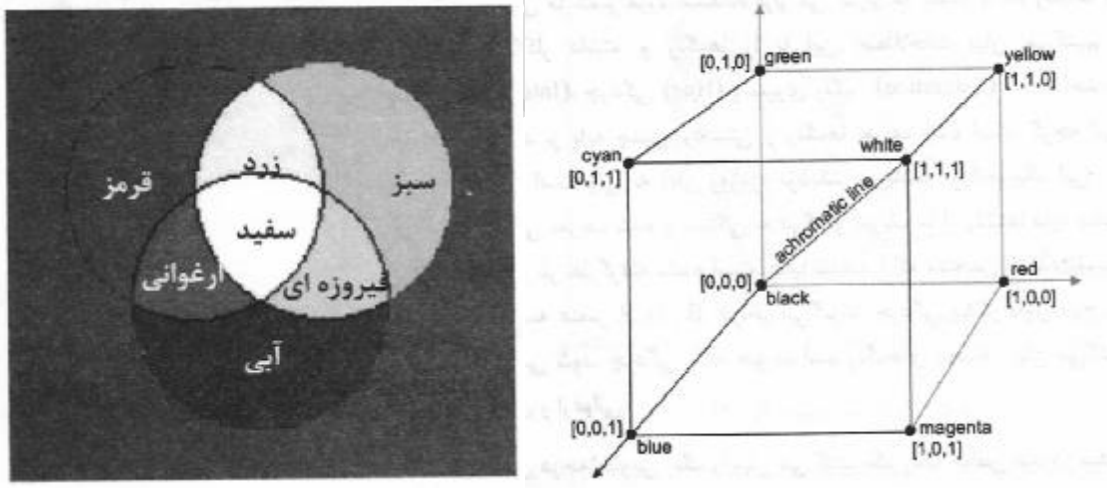
تصاویر رقومی نسبت به عکس های غیر رقومی یا آنالوگ، طبیعتی گسسته دارند و همین مسئله از کیفیت نمایشی آنها به خصوص در مواقعی که بزرگ نمایی نیز شوند به حد زیادی می کاهد. ولی همین طبیعت گسسته اما منظم ، به ما اجازه می دهد تا بتوانیم بسیاری از الگوریتم ها را به طور قاعده مند بر روی آنها اعمال نماییم. یک قاعده که برای یک پیکسل تعریف می شود می تواند با تغییر سطر یا ستون و یا هر دو، بر روی پیکسل های دیگر نیز به طور مشابه اعمال

شود. قابلیتی که پایه اصلی الگوریتم های مختلف پردازشی و یا آنالیز تصاویر است. همچنین این موضوع پایه اصلی پردازشی است که در این پروژه به آن خواهم پرداخت.
ابعاد تصویر رقومی بسته به کاربرد متغییر است. به عنوان مثال اندازه یک تصویر رقومی که از نظر کیفی با تصویر تلویزیونی قابل مقایسه باشد، آرایه ۵۱۲ * ۵۱۲ با ۱۲۸ درجه خاکستری خواهد بود.

از آنجا که پردازش من در این پروژه در سه باند RGB است، مفهوم سیستم رنگی RGB را نیز در اینجا شرح می دهم.

۱-۱-۱- سیستم رنگی RGB

سیستم رنگی RGB سه رنگ اصلی قرمز، سبز و آبی را برای تولید تمامی رنگها به کار می برد. هر رنگ با مشخص شدن این سه مؤلفه برای آن تعیین می شود و قابل تولید است. اگر یک سیستم مختصات ۳ بعدی در نظر بگیریم (به صورت یک مکعب)، در مبدأ سیستم مختصات رنگ سیاه قرار دارد(۰و۰و۰)؛ به تدریج با اضافه شدن مقادیر در سه محور، رنگهای دیگر تولید می شوند. در گوشه مقابل مبدأ در جایی که حداکثر اعداد ممکن برای سه رنگ اصلی قابل تولید است، رنگ سفید خواهد بود. (مثلاً در یک تصویر ۸ بیتی، رنگ با مقدار G۲۵۵، R۲۵۵، و B۲۵۵ ، سفید در نظر گرفته خواهد شد. شکل زیر به صورت گرافیکی مکعب رنگی مذکور را نشان می دهد.



مکعب فضای رنگی سه بعدی و سیستم رنگی تجمعی

قطری از این مکعب که سیاه را به نقطه سفید وصل می کند، خط درجات خاکستری نامیده می شود و در روی آن مقادیر هر سه مؤلفه قرمز، سبز و آبی برای هر نقطه مساوی است که در نتیجه سطوح مختلف خاکستری از سیاه تا سفید را تولید می کند. از آنجایی که با اضافه شدن مقدار در سطح سه محور اصلی رنگهای مختلف تولید می شوند، به سیستم رنگی RGB یک سیستم تجمعی (Additive) گویند. در این پروژه نیز من تصاویر را در هر سه باند RGB در قالب ماتریس های ۳ بعدی در نرم افزار مطلب ذخیره سازی، فراخوانی و پردازش خواهم کرد.
چند خط بالاتر اشاره کردم به تصویر ۸ بیتی. جا دارد همین جا این مفهوم را نیز توضیح دهم.

۱-۱-۲- قدرت تفکیک

مفهومی به نام قدرت تفکیک را در ۴ حیطه بررسی می کنم:

قدرت تفکیک مکانی، قدرت تفکیک رادیو متریک، قدرت تفکیک طیفی، قدرت تفکیک زمانی

۱-۱-۲-۱- قدرت تفکیک مکانی

قدرت تفکیک مکانی را می توان به سادگی توانایی ثبت اشیا کوچک مجاور یکدیگر تعریف کرد. دیگر پارامتری که گاه به عنوان بر آوردی نه چندان دقیق از قدرت تفکیک مکانی عنوان می شود، اندازه پیکسل (Pixel size) است. هرچه قدرت تفکیک مکانی بیشتر باشد، اطلاعات مکانی بیشتری قابل استخراج است. به عنوان مثال برای درک قدرت تفکیک مکانی، در مورد تصاویر ماهواره ای فرض کنید که تصویری با قدرت تفکیک ۱ متری در اختیار دارید. چنین تصویری برای بررسی خیابان های یک شهر و یا پوشش گیاهی داخل شهر کاملاً مناسب است. ولی اگر بخواهیم جاده های بین شهری را از تصویر استخراج کنیم، آنگاه تصویری با قدرت تفکیک ۱۰ یا ۱۵ متری مناسب تر خواهد بود. به همین ترتیب برای بررسی اثرات خشکسالی در سطح کشور ، قدرت تفکیکی کمتر از ۱۰۰ متر نیز کفایت می کند.

۱-۱-۲-۲- قدرت تفکیک طیفی

دامنه طیفی ای که یک سنجنده پوشش می دهد در توانایی آن در تشخیص عوارض کاملاً تأثیر گذار است. هرچه باند تصویر برداری (باند طیفی) بیشتر و باریکتر باشد قدرت تفکیک طیفی بیشتر خواهد بود. به عبارت دیگر هرچه تعداد باند تصویر برداری بیشتر باشد قدرت تفکیک طیفی بیشتر است.

۱-۱-۲-۳- قدرت تفکیک رادیو متریکی

قدرت تفکیک رادیو متریکی و یا حساسیت رادیو متریکی مربوط به حد درجه جزئیاتی است که داده های جمع آوری شده در آن بیان می شوند. هرچه تعداد این درجات بالاتر باشند، جزئیات بیشتری بازگو می شوند. فرض کنید که تصویر رقومی به صورت باینری در اختیار دارید که در آن پیکسل ها یکی از دو مقدار صفر و یا ۲۵۵ دارند. این تصویر با قدرت تفکیک رادیومتریکی بسیار پایین به حساب می آید چراکه داده ها فقط می توانند دو مقدار صفر و یا ۲۵۵ را بپذیرند. حال اگر تعداد درجات خاکستری بالا رود (مثلاً به ۱۶)، آنگاه جزئیات بیشتری از تصویر مشخص خواهد شد. معمولاً قدرت تفکیک رادیومتریکی به صورت توانی از ۲ عنوان می شود. مثلاً اگر داده های سنجنده ای بین مقادیر ۰ تا ۲۵۵ قرار گیرند، آنگاه قدرت تفکیک رادیومتریکی آن ۸ بیتی خواهد بود. در حقیقت این مقدار بیانگر خانه های حافظه ای است که برای ذخیره مقدار یک پیکسل در یک باند لازم است. پس با مفهوم چند بیتی بودن تصاویر نیز آشنا شدیم.

۱-۱-۲-۴- قدرت تفکیک زمانی

حداقل مدت زمان مورد نیاز یک سنجنده برای تصویر برداری مجدد از یک ناحیه مشخص را قدرت تفکیک زمانی گویند.

یک مفهوم دیگر نیز بیان کنم و بعد وارد بخش های اصلی این فصل شویم.

۱-۱-۳- همسایگی پیکسل

پیکسل هایی که در مجاورت یک پیکسل خاص قرار دارند را گویند. ۲ نوع همسایگی معمول، همسایگی ۴ تایی و همسایگی ۸ تایی هستند. (در راستای سطر و یا ستون)

۱-۲- پروژکتیو ۲ بعدی (2D projective)

با بیان مفاهیم پایه در بخش اول (مقدمه) این فصل، حال آمادگی این را داریم که وارد بخش های اصلی گزارش کار شده و کارمان را شروع کنیم.

کاری که ما قصد داریم در این پروژه انجام دهیم به طور کلی به این صورت است. یک تصویر داریم که می خواهیم با ۴ نقطه از آن، با تبدیل پروژکتیو ۲ بعدی، تصویر را به فضای جدیدی ببریم که مختصات ۴ نقطه ما در آن فضا مشخص و متفاوت با فضای اولیه است. به بیان ساده تر می خواهیم یک تبدیل پروژکتیو ۲ بعدی روی عکسمان اعمال کنیم. اما اینکه این کار را چگونه انجام می دهیم، پارامتر های تبدیل مستقیم و معکوس را چگونه حساب می کنیم، قابمان را چگونه تشکیل می دهیم و روشهای درون یابی را چگونه اعمال می کنیم، مباحثی هستند که در فصل دوم بیان خواهم کرد. در ادامه و در این بخش از این فصل به بیان مفاهیم علمی تبدیل ها و به طور خاص تبدیل پروژکتیو ۲ بعدی می پردازم و ۳ نوع از انواع روش های درون یابی مورد استفاده در این پروژه را نیز در بخش دوم این فصل شرح می دهم تا در فصل آینده از آنها استفاده کنیم.

۱-۲-۱- تبدیلات بین سیستم های مختصات

اغلب لازم است ارتباط بین دو سیستم مختصات به منظور انتقال مختصات یک رشته نقاط از سیستمی به سیستم دیگر معلوم شود. برای این منظور تبدیلات زیادی وجود دارد که در فضای سه بعدی به کار می رود و در حالت ساده تر، قابل استفاده در فضای دو بعدی است. در مورد تبدیل می توان گفت در اثر انتقال یک جسم تحت یک تبدیل ممکن است موقعیت یا وضعیت آن جسم به طور ساده (بدون تغییر در شکل و اندازه) یا پیچیده (با تغییر در شکل و اندازه) تغییر یابد و یا بین این دو حالت (تغییر در مقیاس بدون تغییر شکل) باشد. بنابراین می توان از معادلات ریاضی برای انتقال مختصات تصویر به سیستم مختصات معلوم استفاده نمود. اگر بخواهم به صورت یک تعریف بیان کنم میتوان گفت که *نگاشت یا تبدیل* معادله ای ریاضی است که رابطه بین فضای تصویر اولیه و تبدیل یافته را ارائه می دهد. این معادلات ریاضی متنوعند و هر کدام تا حدی توانایی حل و تصحیح بخشی از خطاها[1] را دارند، ولی همگی باعث انتقال از سیستم مختصات تصویر به سیستم مختصات معلوم می گردند. از بین این تبدیلات، تبدیل پروژکتیو ۲ بعدی مد نظر من در این پروژه می باشد. در ادامه این تبدیل را توضیح داده و بقیه را نیز تیر وار بیان می کنم.

[1] مانند خطاهای مختلف موجود در تصاویر هوایی (فتوگرامتری) و یا تصاویر ماهواره ای (سنجش از دور)

۱-۲-۲- تبدیلات سه بعدی

تبدیلات سه بعدی برای ارتباط دو سیستم سه بعدی با یکدیگر و انتقال نقاط از یک سیستم به سیستم دیگر مورد استفاده قرار می گیرند. تبدیلات متعددی برای این منظور وجود دارد که از مهمترین آنها می توان به موارد زیر اشاره کرد.

تبدیل ساده سه بعدی، تبدیل افاین (Affine) سه بعدی، تبدیل سه بعدی با استفاده از چند جمله ای ها، تبدیل پروژکتیو سه بعدی و ...

۱-۲-۳- تبدیلات دو بعدی

تبدیلات دو بعدی در حقیقت حالت ساده شده تبدیلات سه بعدی (که بعد سوم در آنها ثابت است) هستند. از مهمترین تبدیلات ۲ بعدی که اغلب در فتوگرامتری نیز به کار می روند به موارد زیر می توان اشاره کرد.

تبدیل ساده دو بعدی (تبدیل ۴ پارامتری) ، تبدیل افاین دو بعدی(تبدیل شش پارامتری)، تبدیل دو بعدی با استفاده از چند جمله ای ها و تبدیل پروژکتیو ۲ بعدی (تبدیل ۸ پارامتری) که این مورد آخر مد نظر من می باشد و آنرا توضیح می دهم.

۱-۲-۳-۱- تبدیل پروژکتیو (Projective) دو بعدی (تبدیل ۸ پارامتری)

در فتوگرامتری، در مواردی که یک عکس مورد استفاده قرار می گیرد، برای ارتباط بین صفحه نقشه (X,Y) و صفحه عکس (x,y) می توان از معادله پروژکتیو ۲ بعدی زیر استفاده کرد.

$$X = \frac{a_1 x + a_2 y + a_3}{c_1 x + c_2 y + 1}$$

$$Y = \frac{b_1 x + b_2 y + b_3}{c_1 x + c_2 y + 1}$$

در معادله بالا برای تعیین ضرایب a1,a2,a3,b1,b2,b3,c1,c2 لازم است که حداقل ۴ نقطه معلوم در دو سیستم مختصات موجود باشند.

با طرفین وسطین کردن رابطه داریم:

$$X = a_1 x + a_2 y + a_3 - c_1 xX - c_2 yX$$

$$Y = b_1 x + b_2 y + b_3 - c_1 xY - c_2 yY$$

در صورتی که رابطه بالا را به صورت ماتریسی بنویسم، فرم ماتریسی آن به صورت زیر خواهد بود:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -xX & -yX \\ 0 & 0 & 0 & x & y & 1 & -xY & -yY \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \\ c_1 \\ c_2 \end{bmatrix}$$

و در صورت تعمیم رابطه برای ۴ نقطه داریم:

$$\begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \\ X_4 \\ Y_4 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -y_1X_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1Y_1 & -y_1Y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -y_2X_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2Y_2 & -y_2Y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -y_3X_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3Y_3 & -y_3Y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -y_4X_4 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4Y_4 & -y_4Y_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \\ b_2 \\ b_3 \\ c_1 \\ c_2 \end{bmatrix}$$

گفتم که برای استفاده از این تبدیل حداقل ۴ نقطه نیاز است که مختصات آنها در هر ۲ سیستم و یا هر ۲ عکس (در این پروژه) مشخص باشد تا بتوان بعد از تشکیل سیستم معادلات و حل آنها ضرایب را محاسبه کرد. در صورتی که نقاط کنترل بیش از ۴ تا باشد با استفاده از روش کمترین مربعات این ضرایب قابل محاسبه خواهند بود. پس از تعیین ضرایب به ازای مختصات عکسی هر نقطه یا پیکسل (x,y)، مختصات تصحیح شده (X,Y) را در عکس تبدیل یافته با استفاده از معادلات فوق می توان محاسبه کرد. به بیان ساده تر با حل معادلات بالا و محاسبه ضرایب a1,a2,a3,b1,b2,b3,c1,c2، یک نگاشتی بین دو فضای مورد نظرمان خواهیم داشت که با قرار دادن مختصات هر نقطه از فضای اول، مختصات معادل آن را در فضای دوم بدست خواهیم آورد.

حال اگر بخواهیم تبدیل معکوس انجام دهیم و از فضای دوم به فضای اول برسیم کافی است که در معادلات بالا جای مختصات نقاط کنترل بین ۲ فضا را عوض کرده و پارامتر های تبدیل جدید معکوس را بدست آوریم. در حقیقت در این حالت یک تبدیل پروژکتیو ۲ بعدی دیگر بین ۲ فضای جدید انجام میدهیم.

از هر دو تبدیل مستقیم و معکوس در این پروژه استفاده خواهم کرد.

۱-۳- تبدیلات هندسی در تصویر

تبدیلات هندسی عموماً روابط مکانی بین پیکسل های تصویر را تغییر می دهند. تبدیلات هندسی را تبدیلات صفحه لاستیکی نیز می نامند. به این دلیل که می توان این تبدیلات را همانند فرآیند چاپ تصویر روی صفحه لاستیکی و سپس کشش این صفحه طبق مجموعه ای از قانون های از پیش تعریف شده در نظر گرفت.

از نظر پردازش رقومی تصاویر ، تبدیل هندسی یک تصویر از دو عمل پایه ای تشکیل می شود:

۱- یک تبدیل مکانی که باز آرایی پیکسل ها روی صفحه تصویر را تعریف میکند.

۲- یک درون یابی سطح خاکستری که سطوح خاکستری را به پیکسلهای تصویر حاصل از تبدیل مکانی نسبت می دهد.

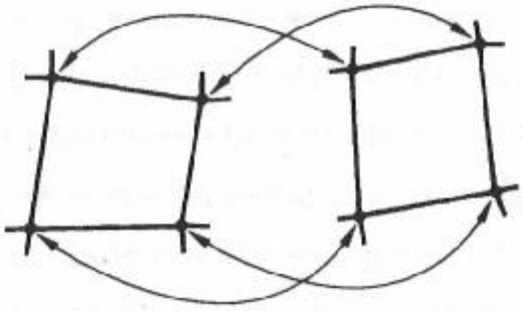
۱-۳-۱- تبدیل های مکانی

فرض کنید که تصویر f با مختصات پیکسلی (x,y) تحت اعوجاج هندسی قرار گیرد تا تصویر g با مختصات $(\hat{x},\hat{y})$ تولید شود. این تبدیل را می توان به صورت زیر بیان کرد:

$$\hat{x} = r(x,y)$$
$$\hat{y} = s(x,y)$$

r(x,y) و s(x,y) تبدیلات مکانی هستند که تصویر اعوجاج هندسی یافتهٔ $g(\hat{x},\hat{y})$ را تولید کرده اند. برای مثال اگر r(x,y)=x/2 و s(x,y)=y/2 باشد، اعوجاج صرفاً کوچک کردن ابعاد f(x,y) به یک دوم در هر دو جهت مکانی است. اگر r(x,y) و s(x,y) به طور تحلیلی معلوم باشند، ممکن است به طور نظری بازیابی تصویر f(x,y) از تصویر اعوجاج یافته $g(\hat{x},\hat{y})$ با اعمال تبدیلات به ترتیب عکس مقدور باشد. اما در عمل عموماً بیان تحلیلی یک مجموعه توابع r(x,y) و s(x,y) که فرآیند اعوجاج هندسی را روی تمام صفحه تصویر توصیف نمایند، ممکن نیست. روشی که اغلب برای غلبه بر این مشکل استفاده می شود، بیان تغییر مکان پیکسلها با استفاده از نقاط گره (tiepoint) است. نقاط گره زیر مجموعه ای از پیکسل هایی هستند که محل آنها در تصاویر ورودی (اعوجاج یافته) و خروجی (تصحیح شده) به دقت معلوم است.

شکل زیر نواحی چهار ضلعی در تصویر اعوجاج یافته و متناظر تصحیح شده آنرا نشان می دهد. گوشه های چهار ضلعی ها نقاط گره متناظر هستند.



نقاط گره متناظر در دو قطعه تصویر

فرض کنید که فرآیند اعوجاج هندسی درون نواحی چهار ضلعی با یک زوج معادلات دو خطی مدل شود به طوری که:

$$r(x,y) = c_1x + c_2y + c_3xy + c_4$$
$$s(x,y) = c_5x + c_6y + c_7xy + c_8$$

آنگاه همانطور که در بالا نیز اشاره شد داریم:

$$** \quad \hat{x} = c_1x + c_2y + c_3xy + c_4$$
$$\hat{y} = c_5x + c_6y + c_7xy + c_8$$

از آنجا که جمعاً ۸ نقطه گره معلوم وجود دارد ، به سادگی می توان معادلات را برای ۸ ضریب c_i $(i=1,2,\ldots,8)$ حل کرد. این ضرایب مدلی را میسازند که برای تبدیل تمام پیکسل های درون چهار ضلعی مشخص شده با گره ها به کار میرود. توجه شود که ضرایب از همین گره ها بدست آمده اند. به طور کلی، برای تولید مجموعه ای از چهار ضلعی

ها که تمام تصویر را بپوشانند، تعداد کافی گره مورد نیاز است که هر یک از این چهار ضلعی ها مجموعه ضرایب خود را دارد.

روال مورد استفاده برای تولید تصویر تصحیح شده ساده است. مثلاً، برای تولید f(0,0)، (x,y)=(0,0) را در معادلات بالا (**) قرار می دهیم و یک زوج مختصات $(\hat{x},\hat{y})$ را بدست می آوریم. آنگاه f(0,0) و $g(\hat{x},\hat{y})$ را با هم مساوی قرار می دهیم =f(0,0) $g(\hat{x},\hat{y})$ [ ] که در این رابطه $\hat{x}$ و $\hat{y}$ مقادیر مختصاتی هستند که اخیراً بدست آمده اند. آنگاه برای ستون بعدی در همان سطر (x,y)=(0,1) را در معادلات بالا (**) قرار می دهیم، زوج دیگری از مقادیر ($\hat{x},\hat{y}$) را بدست می آوریم، و برای آن مقادیر مختصات ، از تساوی=f(0,0) $g(\hat{x},\hat{y})$ استفاده می کنیم. این روال پیکسل به پیکسل و سطر به سطر ادامه می یابد تا آرایه یا ماتریسی که ابعاد آن از ابعاد تصویر g تجاوز نمی کند، بدست آید. به همین صورت برای پوشش ستونی نیز می توان این روند را اجرا نمود. همچنین به منظور استفاده از ضرایب مناسب یک روال ثبت اطلاعات (Bookkeeping procedure) برای تعیین اینکه کدام چهار ضلعی در یک نقطه معین تصویر به کار رود الزامی است.

در این پروژه همانطور که در بخش قبل نیز توضیح دادم تبدیل مکانی مورد نظر ما، تبدیل پروژکتیو ۲ بعدی است که از نظر هندسی کمی با آنچه بیان شد متفاوت است که در ادامه توضیح می دهم.

۱-۳-۲- درون یابی سطح خاکستری و معرفی مهمترین انواع روشهای آن (Nearest Neighbour ، Bilinear Interpolation ، Bicubic Interpolation)

روش تبدیل مکانی ای که در بالا تشریح کردم با مجموعه مقادیر صحیح مختصات (x,y) پیش میرود تا تصویر تصحیح شده f(x,y) را بدست آورد. اما بسته به ضرایب c_i و نوع معادلات مورد استفاده (همچون تبدیل ما در این پروژه) ممکن است مقادیر غیر صحیحی برای $\hat{x}$ و $\hat{y}$ بدست آید. چون تصویر اعوجاج یافته g رقومی است، مقدار پیکسل های آن تنها در مختصات صحیح تعریف می شوند. بنابراین استفاده از مقادیر غیر صحیح برای $\hat{x}$ و $\hat{y}$ ، نگاشتی را به نقاطی از g که هیچ سطح خاکستری برایشان تعریف نشده است، ایجاد می نماید. تحت این شرایط اطلاع از مقدار سطوح خاکستری در محل هایی با مختصات صحیح الزامی می شود. روش مورد استفاده برای انجام این کار *درون یابی سطح خاکستری*[1] نامیده میشود.

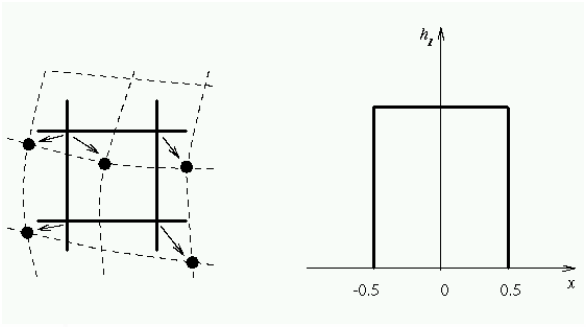
برای اینکار روشهای درون یابی مختلفی وجود دارد که به آنها *روشهای نمونه برداری مجدد* نیز میگویند. معروفترین روشهای نمونه برداری مجدد عبارت اند از:

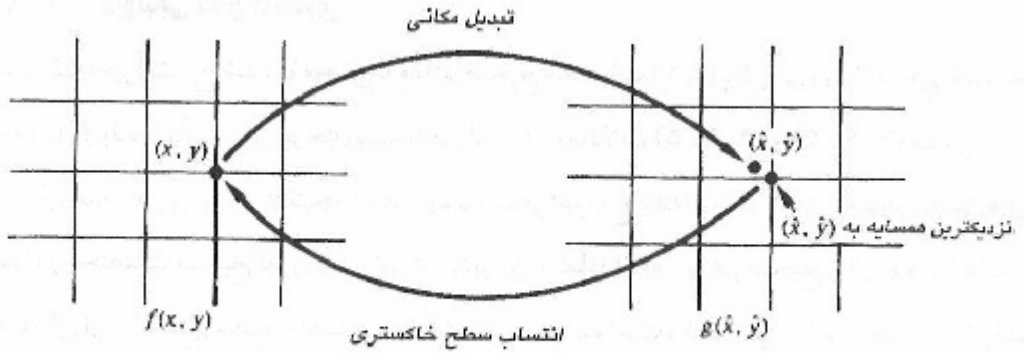
- روش نمونه برداری نزدیکترین همسایه (Nearest Neighbour)
- روش درون یابی دو خطی (Bilinear Interpolation)
- و روش برآورد مکعبی (Bicubic Interpolation یا Cubic Convolution)

[1] Gray-level interpolation

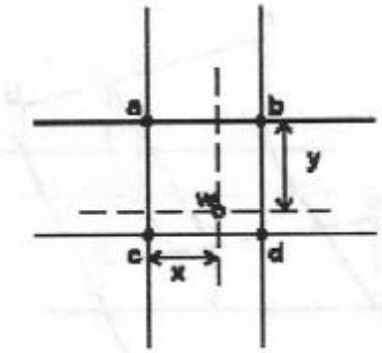
۱-۳-۲-۱- روش نزدیکترین همسایه (Nearest Neighbour resampling)

در روش نزدیکترین همسایه عملاً هیچگونه محاسبه انجام نمیپذیرد. فقط مقدار درجه خاکستری پیکسلی انتخاب می شود که به موقعیت نقطه در تصویر نزدیکترین باشد. (شکل های زیر)





درونیابی سطح خاکستری براساس مفهوم نزدیکترین همسایه



Nearest neighbour resampling

درون یابی نزدیکترین همسایه : a,b,c,d چهار پیکسل همسایه و w پیکسل مورد نظر

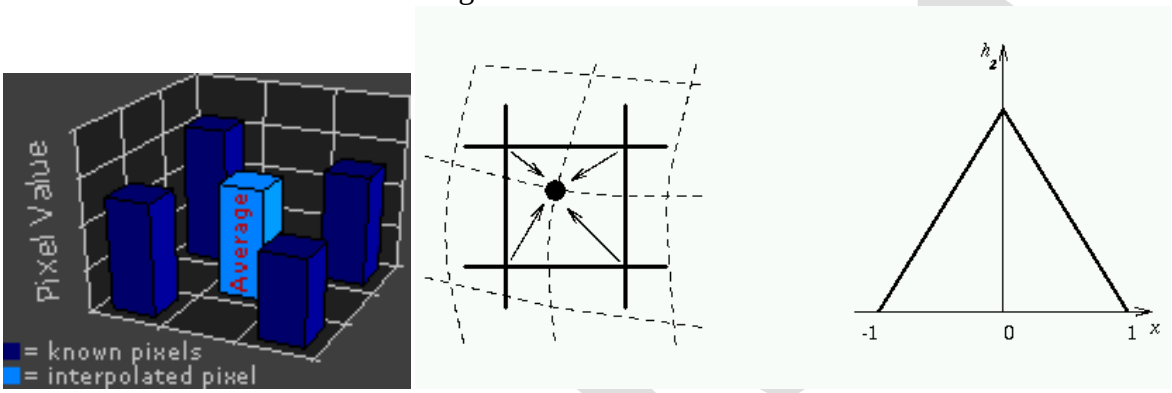
پس از اینکار ، این درجه خاکستری به موقعیت نقطه در شبکه منظم اصلی نسبت داده می شود. این روش از آنجایی که محاسبات خاصی انجام نمیدهد، سریع ترین روش نمونه برداری مجدد محسوب میشود. علاوه بر این، از مقادیر حقیقی موجود در تصویر استفاده میکند و بنابراین ارقام جدیدی که گاه ممکن است غیر واقعی باشند را تولید نمی نماید. این مسئله به خصوص برای موقعیکه قرار است تصویر طبقه بندی گردد، بسیار مفید خواهد بود. مهمترین عیب این روش این است که در نتایج این روش، گاه بعضی از مقادیر تکرار شده و اثر بلوک بلوک شدن (Blocky Effect) ظاهر میشود.

به بیان دیگر گرچه روش درون یابی نزدیکترین همسایه پیاده سازی ساده ای دارد، اما اغلب با تولید اثرات نا مطلوبی نظیر اعوجاج دادن لبه های مستقیم الخط در تصاویر با تفکیک بالا، مشکل ایجاد می کند.

۱-۳-۲-۲- روش درون یابی دو خطی (Bilinear Interpolation)

در این روش از چهار پیکسل همسایه نقطه در تصویر استفاده شده و میان آنها درون یابی برای یافتن درجه خاکستری نقطه مورد نظر انجام می گردد. مطابق شکل زیر ابتدا میان هر دو پیکسل مقابل یک درون یابی خطی انجام می شود. سپس با استفاده از محل نقطه در این درون یابی های انجام شده درجه خاکستری پیکسل محاسبه می گردد. در عمل یک صفحه به چهار درجه خاکستری همسایه برازش داده شده و سپس درجه خاکستری نقطه مورد نظر محاسبه می شود.

- Bilinear interpolation explores four points neighboring the point (x,y), and assumes that the brightness function is bilinear in this neighborhood.



Linear interpolation is given by

$$f_2(x,y) = (1-a)(1-b)g_s(l,k) + a(1-b)g_s(l+1,k)$$
$$(1-a)bg_s(l,k+1) + abg_s(l+1,k+1)$$
$$= g_s(l,k)$$
$$+a(g_s(l+1,k) - g_s(l,k))$$
$$+b(g_s(l,k+1) - g_s(l,k))$$
$$+(g_s(l,k) + g_s(l+1,k+1) - g_s(l+1,k) - g_s(l,k+1))ab$$

where

$l = \text{floor}(x)$, $a = x - l$
$k = \text{floor}(y)$, $b = y - k$

از معایب و مزایای این روش نیز می توان به موارد زیر اشاره کرد:

Linear interpolation can cause a small decrease in resolution and blurring due to its averaging nature.

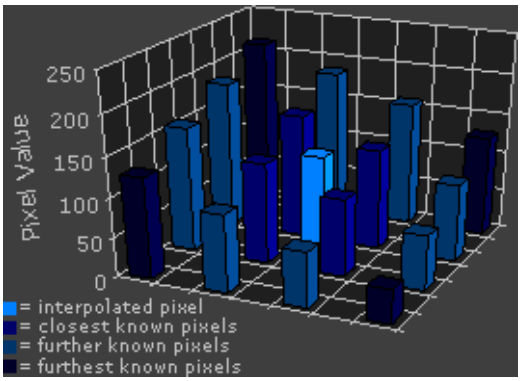
The problem of step like straight boundaries with the nearest neighborhood interpolation is reduced.

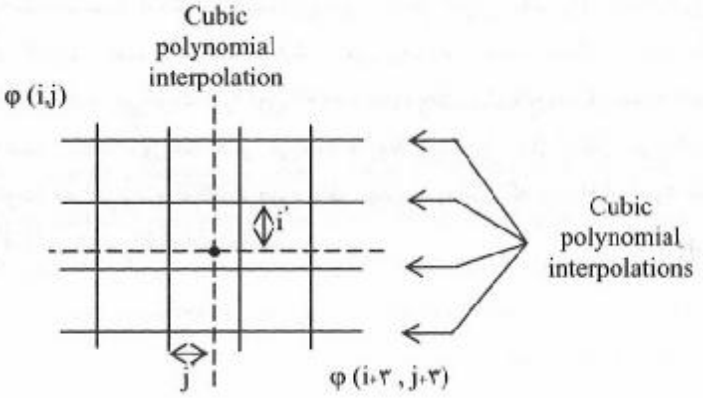
به بیان دیگر با توجه به محاسبات بیشتر این روش نسبت به روش نزدیکترین همسایه، زمان بیشتری را نیاز دارد. علاوه بر این، این روش درون یابی باعث می شود تا تصویری نرم تر ایجاد شود و بنابراین بعضی از لبه های بسیار بارز در تصویرِ تصحیح نشده، در تصویر نهایی ممکن است کمی مبهم (Blur) شوند.

از رابطه ریاضی این روش که در بالا ذکر شد، عیناً در برنامه نویسی این پروژه استفاده خواهم کرد.

۱-۳-۲-۳- برآورد مکعبی (Bicubic Interpolation یا Cubic Convolution)

این روش از ۱۶ پیکسل همسایه نقطه مطابق شکل زیر استفاده می کند. این ۱۶ پیکسل که در ۴ ردیف قرار دارند مبنای درون یابی خواهند بود. ابتدا به هر چهار پیکسل در یک ردیف یک چند جمله ای درجه ۳ برازش داده می شود. در نهایت با استفاده از موقعیت نقطه یک چند جمله ای دیگر نیز در محل نقطه مورد نظر به چهار نقطه ای که در این محل از چهار چند جمله ای بدست می آیند برازش داده می شود و درجه خاکستری نقطه مورد نظر محاسبه می گردد.





بر آورد مکعبی

شکل های مختلفی از چند جمله ای ها میتوانند در درون یابی استفاده شوند. بر اساس یکی از مناسبترین آنها، درجه خاکستری مورد نظر به صورت زیر محاسبه می شود.

$$l = \text{floor}(u)\ ,\quad a = u - l$$
$$k = \text{floor}(v)\ ,\quad b = v - k$$

$$b(x,y) = \sum_{m=l-1}^{l+2} \sum_{n=k-1}^{k+2} A(m,n)h(u-m)h(v-n)$$

$$h(t) = \begin{cases} 1-2|t|^2+|t|^3 & |t|<1 \\ 4-8|t|+5|t|^2-|t|^3 & 1\le|t|<2 \\ 0 & otherwise \end{cases}$$

در رابطه بالا A تصویر اولیه، h تابع وزن و B تصویر تبدیل یافته است. از این رابطه نیز عیناً در این پروژه و در برنامه استفاده می کنم.

Bi-cubic interpolation improves the model of the brightness function by approximating it locally by a bicubic polynomial surface; 16 neighboring points are used for interpolation.
Bicubic interpolation does not suffer from the step-like boundary problem of nearest neighborhood interpolation, and copes with linear interpolation blurring as well.
Bicubic interpolation is often used in raster displays that enable zooming to an an arbitrary scale.
Bicubic interpolation preserves fine details in the image very well.

محاسبات پیچیده این روش نیاز به زمان بیشتری نسبت به دو روش قبلی دارد که در عمل مخصوصاً وقتی ابعاد تصویر بزرگ باشد، کاملاً محسوس است. این نوع درون یابی که از اطلاعات همسایگان بیشتری استفاده می کند، در نهایت

تصویری با نمای طبیعی بیشتر و همچنین نرم شدگی بیشتری تولید مـی کنـد و در نتیجـه گرچـه مقـداری از اطلاعـات و جزئیات را محو کرده و یا از بین می برد و لی برای تفسیر بصری مناسبتر است.

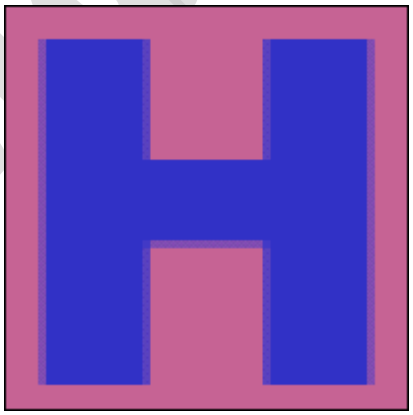
از کاربرد های فراوان تبدیلات هندسی برای بازیابی تصویر می توان به ثبت تصویر[۱] یا پیدا کردن تناظر بین دو تصـویر، تصحیح اعوجاجات نمایشگر، نگاشتهای نقشه[۲]، و نگاشت های کارتوگرافیک[۳] اشاره کرد.

در ادامه و در فصول بعدی مراحل اجرای پروژه را با کمک مطالبی که تا اینجا بیان شد شرح خـواهم داد و در حقیقـت یک همبستگی بین مفاهیمی که تا کنون گفته شد ایجاد خواهم کرد. در نهایت پس از توضیح بخش بخش برنامه، نتیجه گیری خواهم نمود.

۱ Image registeration
۲ Map projection
۳ Cartographic projection

فصل دوم

مراحل انجام پروژه و توضیح بخش های مختلف برنامه



تصاویر تزئینی هستند

2 - 1 - توضیح مراحل انجام پروژه

تا اینجا با مفاهیم مورد نیاز کاملاً آشنا شدیم. به طور کلی کاری را که در این پروژه قصد انجام آن را داریم به سه مرحله اصلی می توان تقسیم بندی نمود.

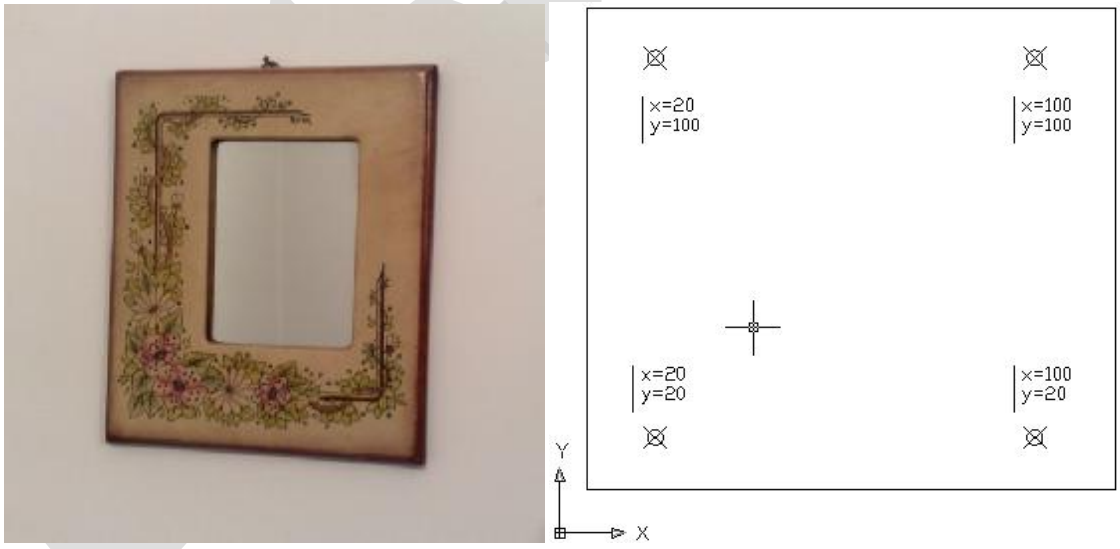
۱- تهیه یک عکس رقومی تیلت دار از یک سطح صاف

۲- ایجاد پارامتر های تبدیل پروژکتیو

۳- اعمال تبدیل پروژکتیو و resampling با هر یک از سه روش درون یابی توضیح داده شده

به عنوان مثال تصویری گرفته ایم که که عوارض در آن شکل صحیح هندسی خود را ندارند و همچنین ممکن است در جای صحیح خود نیز قرار نگرفته و یا توجیه مناسبی نداشته باشند. برای رفع این مشکلات (ومشکلات مشابه در سایر انواع تصاویر که در بخش تبدیلات توضیح دادم) باید تصویر جدیدی تشکیل دهیم و با انتخاب مدل تبدیل مورد نظر، پیکسل ها را بر اساس مختصات صحیح آنها (مثلاً مختصات زمینی صحیح) در کنار یکدیگر قرار دهیم.

در این پروژه تصویر تیلت داری که تهیه می شود، به دلیل اینکه تبدیل ما ۲ بعدی است نسبتاً مسطح می باشد. این تصویر را در زیر مشاهده می فرمایید. ابتدا ۴ گوشه از یک عارضه مشخص از این تصویر را انتخاب می کنیم. با داشتن مختصات عکسی این نقاط، و مختصات آنها در فضای دوم که در این پروژه به طور فرضی (۲۰و۲۰) ؛ (۱۰۰و۲۰)؛ (۱۰۰و۱۰۰)؛ (۲۰و۱۰۰) در نظر گرفته می شود، پارامتر های تبدیل مستقیم را (که در این پروژه از پروژکتیو ۲ بعدی استفاده می کنیم) بدست می آوریم.

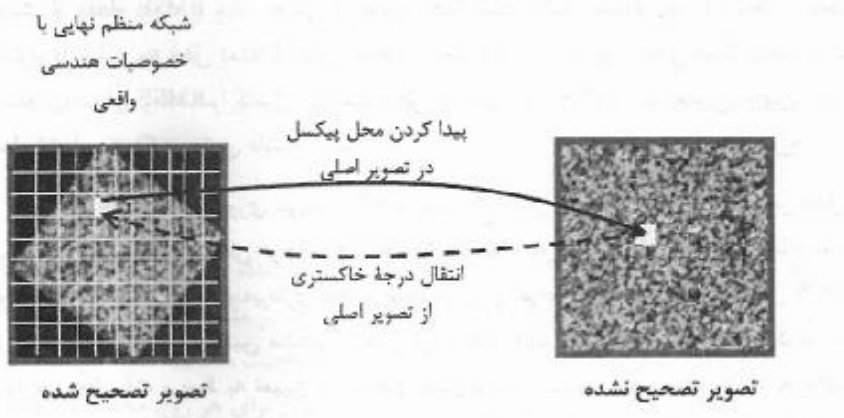


تصویر اولیه مورد استفاده در این پروژه و قاب مورد نظر با مختصات نقاط کنترل

پس از یافتن پارامتر های تبدیل مستقیم ۴ نقطه از گوشه های عکسمان را که تهیه کرده ایم انتخاب کرده و این ۴ نقطه را با استفاده از معادلات بدست آمده با پارامتر های معلوم، به فضای جدیدمان منتقل می کنیم. این ۴ نقطه با توجه به اینکه گوشه های تصویر اولیه هستند، پس از تبدیل، به عنوان گوشه های فضای جدید یا قابمان نیز در نظر گرفته می شوند. حال یک شبکه منظم از پیکسل ها(قابمان) با مختصات زمینی و ابعاد معلوم (ابعاد قابمان) تعیین می گردد. در

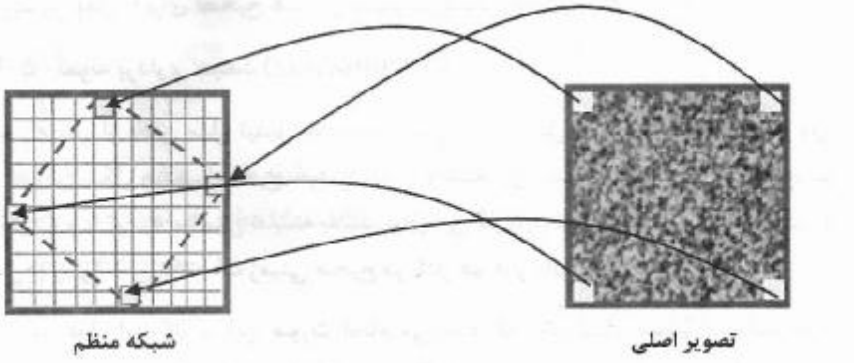
مرحله بعد معادلات تبدیل از فضای نقشه به فضای تصویر (معادلات تبدیل معکوس) با استفاده از همان نقاط کنترل اولیه حل می شوند. یعنی پارامتر های تبدیل معکوس پروژکتیو ۲ بعدی را بدست می آوریم.

حال می توان برای هر پیکسل از این شبکه منظم یک محل متناظر در تصویر یافت. سپس درجه خاکستری مناسب از تصویر اصلی استخراج شده و به پیکسل در این شبکه نسبت داده می شود. پس از اینکه تمام پیکسل های این شبکه دارای درجه خاکستری شدند، تصویر تصحیح شده بدست آمده است.



انتقال درجه خاکستری از تصویر اصلی به پیکسل های قاب

همانطور که گفتم این شبکه منظم را معمولاً با استفاده از نقاط گوشه ای تصویر اصلی بدست می آورند. به این معنی که مختصات زمینی چهار پیکسل گوشه ای تصویر اصلی یافت شده و بر اساس آنها شبکه ای منظم که اندازه آنرا ابعاد زمینی پیکسل های تصویر مشخص می کنند[1](که در این پروژه در هر دو تصویر این ابعاد یکی است)، تولید می شود.در شکل زیر به طور شماتیک فرآیند این کار را نشان داده ام.



تشکیل شبکه منظم برای ایجاد تصویر تصحیح شده

در پروژه هایی که مختصات قابمان یا نقشه عکسی مان مختصات زمینی پیکسل ها است، پس از تشکیل شبکه منظم، هر پیکسل دارای یک مختصات زمینی معلوم است. چراکه مختصات زمینی پیکسل اول مشخص بوده و با استفاده از ابعاد زمینی پیکسل ها (که کاربر مشخص می کند) می توان مختصات زمینی مابقی پیکسل ها را نیز به راحتی بدست آورد.

می دانیم که در یک تصویر جدا از ماهیت هندسی آن، درجات خاکستری نیز بعد دیگری از آن را تشکیل میدهند. بنابراین برای تبدیل شبکه جدید ایجاد شده به یک تصویر، باید درجات خاکستری آن را نیز یافت. پس نیاز به تبدیل معکوس از فضای این شبکه به فضای تصویر است، چراکه درجات خاکستری در تصویر اصلی قرار دارند. بنابراین

[1] برای تصاویر هوایی و ماهواره ای

معادلات تبدیل معکوس حل شده و برای هر پیکسل در شبکه، یک مختصات تصویری بدست می آید و به اصطلاح از فضای قاب یا فضای نقشه ای به فضای تصویر باز می گردیم. ولی همانطور که انتظار داریم با استفاده از این تبدیل معکوس نمی توان یک مختصات صریح تصویری بر اساس سطر و ستون مشخص بدست آورد. بنابر این باید تصمیم گرفت کدام درجه خاکستری را به فضای قابمان (یا نقشه در تهیه نقشه عکسی) برگردانیم. برای این کار نیز همانطور که قبلاً هم گفتم از روشهای درون یابی استفاده می کنیم که سه روش از مهمترین آنها را که در این پروژه از آنها استفاده کرده ام یعنی روش نمونه برداری نزدیکترین همسایه (Nearest Neighbour)، روش درون یابی دو خطی (Bilinear Interpolation) و روش برآورد مکعبی (Bicubic Interpolation یا Cubic Convolution) در فصل های قبل کاملاً تشریح کردم. خوب، با استفاده از هر یک از این سه روش (جداگانه) درجات خاکستری را نیز به قابمان یا تصویر تبدیل یافته مان نسبت دادیم. تصویر مورد نظرمان پس از تبدیل و resampling بدست خواهد آمد.

حال در ادامه بخشهای مختلف برنامه نوشته شده را با توجه به مراحل و بخشهایی که توضیح داده شد به ترتیب معرفی می کنم.

2 - 2 - معرفی بخشهای مختلف برنامه نوشته شده برای این پروژه

ابتدا پس از معرفی برنامه، عکس مورد نظر را فراخوانی نموده و محل ۴ نقطه را روی آن کلیک کرده و مختصات آنها را در ماتریس input_points ۸*۱ به صورت ستونی ذخیره می کنیم. ماتریس input_points همانطور که گفتم یک ماتریس ۸*۱ است که مختصات xوy ۴ نقطهٔ کلیک شده به ترتیب نقاط از بالا به پایین در آن قرار دارد. از آنجا که سیستم مختصات تصویر در نرم افزار مطلب بر خلاف سیستم مورد نظر ما است، لذا جای مختصات x,y حاصل از کلیک کردن، در ماتریس input_points عوض شده است. در این قسمت نقاط مورد نظر را برای کلیک کردن با رنگ **زرد** و شماره ترتیب کلیک کردن روی عکس مشخص نموده ام.

سپس مختصات نقاط در فضای دوم را معرفی می کنیم.

```
%by farid esmaeili                    sh.d:
%----tabdile projective yek tasvir ba 3 raveshe daron yabiye
%nearest neighbour,bilinear interpolation,bicubic interpolation---------
%========================================================================
clc
clear all
q=imread('farid.jpg');
figure(1);imshow(q);
[u,v]=ginput(4);
b(1:2:8)=v;
b(2:2:8)=u;
input_points=b';
base_points=[20;20;20;100;100;100;100;20];
```

حال معادلات تبدیل مستقیم پروژکتیو ۲ بعدی را با ۴ نقطه انتخاب شده روی عکس و ۴ نقطه مبنا حل کرده و پارامتر های آن را بدست می آوریم. ملاحظه می شود که برنامه را طوری نوشته ام که در صورتی که تعداد نقاط کنترل بیش از ۴ تا باشد نیز بتوان تبدیل پروژکتیو را با روش کمترین مربعات حل نمود:

```
%---------------mohasebeye parametr haye tabdile 2d-projective ba
%4 noghteye entekhabi az tasvire asli (tabdile mostaghim) direct-----------
a=size(input_points,1)/2;
for i=1:a
    s(2*i-1,1)=input_points(2*i-1);
    s(2*i-1,2)=input_points(2*i);
    s(2*i-1,3)=1;
    s(2*i-1,4:6)=0;
    s(2*i-1,7)=-(input_points(2*i-1)*base_points(2*i-1));
    s(2*i-1,8)=-(input_points(2*i-1)*base_points(2*i));

    s(2*i,4)=input_points(2*i-1);
    s(2*i,5)=input_points(2*i);
    s(2*i,6)=1;
    s(2*i,1:3)=0;
    s(2*i,8)=-(input_points(2*i)*base_points(2*i));
    s(2*i,7)=-(input_points(2*i-1)*base_points(2*i));
end
projective_direct=inv(s)*base_points;
%projective_direct=inv(s'*s)*s'*base_points;
p=projective_direct;
```

اکنون برای تعیین ابعاد قابمان، روی ۴ نقطه گوشه تصویرمان کلیک کرده و با معادلات و پارامتر های بدست آمده در بالا، این ۴ گوشه از فضای اول و تصویر اولمان را به فضای دوم یا قابمان می بریم. این ۴ نقطه را به ترتیب شماره کلیک با رنگ **قرمز** روی تصویر مشخص نموده ام. بعد از تبدیل،x,y ماکزیمم و مینیمم قابمان را پیدا می کنیم و در صورتی که مینیمم این مقادیر از تبدیل، منفی بدست آمده باشد، آنها را مساوی ۱ قرار می دهیم تا در محاسباتمان با مشکل مواجه نشویم، در مرتبه دوم کلیک روی نقاط گوشه تصویر نیز جای مختصات x,y حاصله عوض شده است.

dx و dy در زیر در حقیقت ابعاد قابمان است. نکته مربوط به ۲ پارامتر را در پایین توضیح خواهم داد.

```
%----tabdile projective tasvire aval be dovom ba 4 noghteye entekhabi------
figure(2);imshow(q);
[u2,v2]=ginput(4);
b2(1:2:8)=v2;
b2(2:2:8)=u2;
e=b2';
for z=1:4
    x1(z)=(p(1)*e(2*z-1)+p(2)*e(2*z)+p(3))/(p(7)*e(2*z-1)+p(8)*e(2*z)+1);
    y1(z)=(p(4)*e(2*z-1)+p(5)*e(2*z)+p(6))/(p(7)*e(2*z-1)+p(8)*e(2*z)+1);
end
xmin=min(x1);
xmax=max(x1);
ymin=min(y1);
ymax=max(y1);
dx=xmax-xmin;
dy=ymax-ymin;
```

حال پارامتر های تبدیل معکوسمان را در تبدیل پروژکتیو معکوس از قاب به عکس ، با همان ۴ نقطه اولیه انتخابی از تصویر در تبدیل مستقیم و ۴ نقطه کنترل فضای قابمان بدست می آوریم:

```
%-----mohasebeye parameter haye projective dar tabdile makoos(invers)------
a=size(input_points,1)/2;
for i=1:a
    s(2*i-1,1)=base_points(2*i-1);
    s(2*i-1,2)=base_points(2*i);
    s(2*i-1,3)=1;
    s(2*i-1,4:6)=0;
```

```
    s(2*i-1,7)=-(base_points(2*i-1)*input_points(2*i-1));
    s(2*i-1,8)=-(base_points(2*i-1)*input_points(2*i));

    s(2*i,4)=base_points(2*i-1);
    s(2*i,5)=base_points(2*i);
    s(2*i,6)=1;
    s(2*i,1:3)=0;
    s(2*i,8)=-(base_points(2*i)*input_points(2*i));
    s(2*i,7)=-(base_points(2*i-1)*input_points(2*i));
end
c=inv(s)*input_points;
%c=inv(s'*s)*s'*input_points;
```

در اینجا روش کار مورد نظر را به کاربر توضیح داده و از او می خواهم تا بین ۳ روش انتخاب کند. در صورتی که کاربر عددی کلیک نکند و enter کند به طور پیش فرض حالت nearest neighbour اجرا خواهد شد. با enter کردن عدد۱ روش nearest neighbour ، عدد۲ روش bilinear interpolation، عدد۳ روش bicubic interpolation اجرا خواهد شد.

```
%-------------------------------------------------------------------------
disp('raveshe daron yabiye morede nazare khod ra')
disp('ba enter kardane adade marbote entekhab konid:')
disp('1- nearest neighbour')
disp('2- bilinear interpolation')
disp('3- bicubic interpolation')
ravesh=input('1/2/3? [1]:');
if isempty(ravesh)
    ravesh=1;
end
```

حال بخش نهایی برنامه، مربوط به اختصاص مقادیر پیکسل ها از تصویر اصلی به قابمان با هر یک از روشهای بیان شده درون یابی با انتخاب کاربر اجرا می گردد. نکته ای که در رابطه با dx و dy باید بیان کنم این است که پس از اعمال تبدیل مستقیم به ۴ نقطه اولیه، الزاماً مختصات این نقاط در فضای جدیدمان مثبت نخواهند بود. از طرفی برای یک ماتریس نمی توان سطرو ستون منفی در نظر گرفت. لذا در محاسبات زیر مقادیر f,g هستند که از کوچکترین ابعاد قابمان شروع می شوند(xmin,ymin) و با شمرده شدن در حلقه، مختصات خانه های خالی قابمان را می سازند. f,g در محاسبات می توانند منفی نیز باشند که پس از اعمال تبدیل معکوس بر روی آنها مقادیر x,y متناظر آنها در تصویر اول بدست می آیند. اما در نهایت درایه های ماتریسی که تصویر نهایی ما را تشکیل می دهد، i,j هستند که تا dx و dy (ابعاد قابمان) شمرده می شوند و مقادیر مثبتی را دارا می باشند.

نکته دیگر اینکه هروقت مختصاتی که با تبدیل معکوس از قابمان به تصویر اصلی نسبت داده می شوند خارج از تصویر اصلی بیافتند، مقادیر درجه خاکستری آنها صفر در نظر گرفته می شود. برای سایر پیکسلها نیز با توجه به فرمولهای هر روش که نوشته شده اند و قبلاً نیز هریک را جداگانه توضیح دادم، درجه خاکستری نسبت داده می شود.

یک نکته هم اینکه از آنجا که ماتریس های نهایی حاصله double هستند، آنها را با uint8 (۸ بیتی) نمایش داده و ذخیره می کنیم.

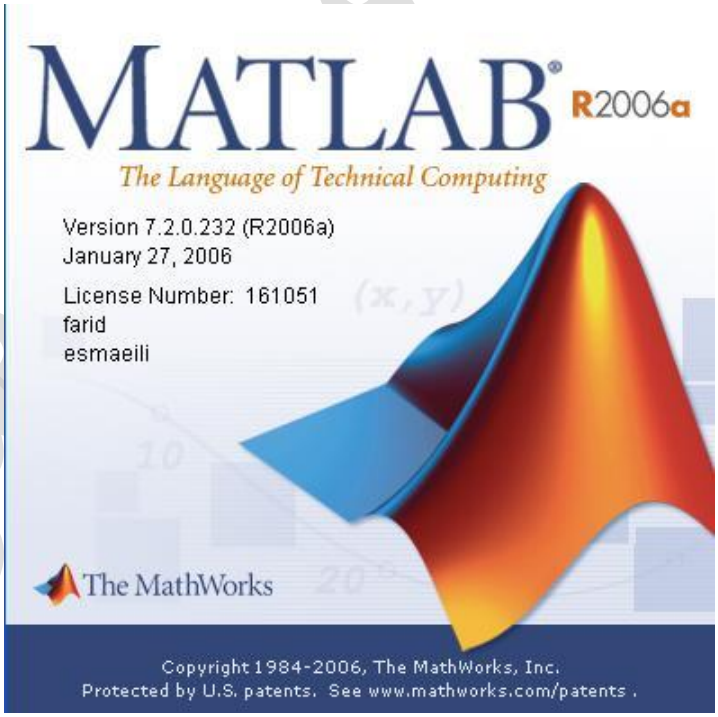
در نهایت در هر روش از برنامه، تصویر خروجی در پنجره ای جدید نشان داده شده و با فایلی با نام روش درون یابی مربوطه ذخیره می گردد:

```
%----------------------------nearest neighbour-----------------------------
switch ravesh
    case 1
        f=xmin;
        g=ymin;
        for i=1:dx
            for j=1:dy
                x=(c(1)*f+c(2)*g+c(3))/(c(7)*f+c(8)*g+1);
                y=(c(4)*f+c(5)*g+c(6))/(c(7)*f+c(8)*g+1);
                g=g+1;
                if x<1 | x>size(q,1)
                    output_image(i,j,:)=0;
                    elseif y<1 | y>size(q,2)
                    output_image(i,j,:)=0;
                else
                    output_image(i,j,1)=q(round(x),round(y),1);
                    output_image(i,j,2)=q(round(x),round(y),2);
                    output_image(i,j,3)=q(round(x),round(y),3);
                end
            end
            f=f+1;
            g=ymin;
        end
figure,imshow(uint8(output_image));
imwrite(uint8(output_image),'resampled_nearest.jpg');
%--------------------------bilinear interpolation-------------------------
    case 2
        f=xmin;
        g=ymin;
        for i=1:dx
            for j=1:dy
                x=(c(1)*f+c(2)*g+c(3))/(c(7)*f+c(8)*g+1);
                y=(c(4)*f+c(5)*g+c(6))/(c(7)*f+c(8)*g+1);
                g=g+1;
                if x<1 | x>size(q,1)
                    output_image(i,j,:)=0;
                    elseif y<1 | y>size(q,2)
                    output_image(i,j,:)=0;
                else
l=floor(x);
k=floor(y);
aa=x-l;
bb=y-k;
output_image(i,j,1)=q(l,k,1)+aa*(q(l+1,k,1)-q(l,k,1))+bb*(q(l,k+1,1)-
q(l,k,1))+aa*bb*(q(l,k,1)+q(l+1,k+1,1)-q(l,k+1,1)-q(l+1,k,1));
output_image(i,j,2)=q(l,k,2)+aa*(q(l+1,k,2)-q(l,k,2))+bb*(q(l,k+1,2)-
q(l,k,2))+aa*bb*(q(l,k,2)+q(l+1,k+1,2)-q(l,k+1,2)-q(l+1,k,2));
output_image(i,j,3)=q(l,k,3)+aa*(q(l+1,k,3)-q(l,k,3))+bb*(q(l,k+1,3)-
q(l,k,3))+aa*bb*(q(l,k,3)+q(l+1,k+1,3)-q(l,k+1,3)-q(l+1,k,3));
                end
            end
            f=f+1;
            g=ymin;
        end
figure,imshow(uint8(output_image));
imwrite(uint8(output_image),'resampled_bilinear.jpg');
%-------------------------bicubic interpolation-----------------------------
    case 3
        f=xmin;
        g=ymin;
```

```
        for i=1:dx
            for j=1:dy
                x=(c(1)*f+c(2)*g+c(3))/(c(7)*f+c(8)*g+1);
                y=(c(4)*f+c(5)*g+c(6))/(c(7)*f+c(8)*g+1);
                g=g+1;
                if x<2 | x>size(q,1)
                    output_image(i,j,:)=0;
                    elseif y<2 | y>size(q,2)
                    output_image(i,j,:)=0;
                else
                l=floor(x);
                k=floor(y);
                mm=l-1;
                nn=k-1;
                mmm=l+2;
                nnn=k+2;
                sum2=0;
                sum1=0;
                for m=mm:mmm
                     for n=nn:nnn
                        if (abs(x-m))<1
                            hx=1-2*(abs(x-m))^2+(abs(x-m))^3;
                        elseif 1<=(abs(x-m))<2
                            hx=4-8*(abs(x-m))+5*(abs(x-m))^2-(abs(x-m))^3;
                        else
                            hx=0;
                        end
                        if (abs(y-n))<1
                            hy=1-2*(abs(y-n))^2+(abs(y-n))^3;
                        elseif 1<=(abs(y-n))<2
                            hy=4-8*(abs(y-n))+5*(abs(y-n))^2-(abs(y-n))^3;
                        else
                            hy=0;
                        end
                    sum1=sum1+(q(m,n)*hx*hy);
                    end
                sum2=sum2+sum1;
                end
                output_image(i,j,1)=sum2;
                output_image(i,j,2)=sum2;
                output_image(i,j,3)=sum2;
                end
            end
            f=f+1;
            g=ymin;
        end
figure,imshow(uint8(output_image));
imwrite(uint8(output_image),'resampled_bicubic.jpg');
end
```

فصل سوم

متن برنامه نوشته شده به طور یکجا ، ورودی ها، خروجی ها و مقایسه خروجی ها



تصویر تزئینی است

3 - 1 - متن برنامه نوشته شده به زبان برنامه نویسی مطلب به طور یکجا

شروع

```
%by farid esmaeili                    sh.d:
%----tabdile projective yek tasvir ba 3 raveshe daron yabiye
%nearest neighbour,bilinear interpolation,bicubic interpolation---------
%=========================================================================
clc
clear all
q=imread('farid.jpg');
figure(1);imshow(q);
[u,v]=ginput(4);
b(1:2:8)=v;
b(2:2:8)=u;
input_points=b';
base_points=[20;20;20;100;100;100;100;20];
%---------------mohasebeye parametr haye tabdile 2d-projective ba
%4 noghteye entekhabi az tasvire asli (tabdile mostaghim) direct-----------
a=size(input_points,1)/2;
for i=1:a
    s(2*i-1,1)=input_points(2*i-1);
    s(2*i-1,2)=input_points(2*i);
    s(2*i-1,3)=1;
    s(2*i-1,4:6)=0;
    s(2*i-1,7)=-(input_points(2*i-1)*base_points(2*i-1));
    s(2*i-1,8)=-(input_points(2*i-1)*base_points(2*i));

    s(2*i,4)=input_points(2*i-1);
    s(2*i,5)=input_points(2*i);
    s(2*i,6)=1;
    s(2*i,1:3)=0;
    s(2*i,8)=-(input_points(2*i)*base_points(2*i));
    s(2*i,7)=-(input_points(2*i-1)*base_points(2*i));
end
projective_direct=inv(s)*base_points;
%projective_direct=inv(s'*s)*s'*base_points;
p=projective_direct;
%----tabdile projective tasvire aval be dovom ba 4 noghteye entekhabi------
figure(2);imshow(q);
[u2,v2]=ginput(4);
b2(1:2:8)=v2;
b2(2:2:8)=u2;
e=b2';
for z=1:4
    x1(z)=(p(1)*e(2*z-1)+p(2)*e(2*z)+p(3))/(p(7)*e(2*z-1)+p(8)*e(2*z)+1);
    y1(z)=(p(4)*e(2*z-1)+p(5)*e(2*z)+p(6))/(p(7)*e(2*z-1)+p(8)*e(2*z)+1);
end
xmin=min(x1);
xmax=max(x1);
ymin=min(y1);
ymax=max(y1);
dx=xmax-xmin;
dy=ymax-ymin;
%-----mohasebeye parameter haye projective dar tabdile makoos(invers)------
a=size(input_points,1)/2;
for i=1:a
    s(2*i-1,1)=base_points(2*i-1);
    s(2*i-1,2)=base_points(2*i);
    s(2*i-1,3)=1;
    s(2*i-1,4:6)=0;
    s(2*i-1,7)=-(base_points(2*i-1)*input_points(2*i-1));
    s(2*i-1,8)=-(base_points(2*i-1)*input_points(2*i));

    s(2*i,4)=base_points(2*i-1);
```

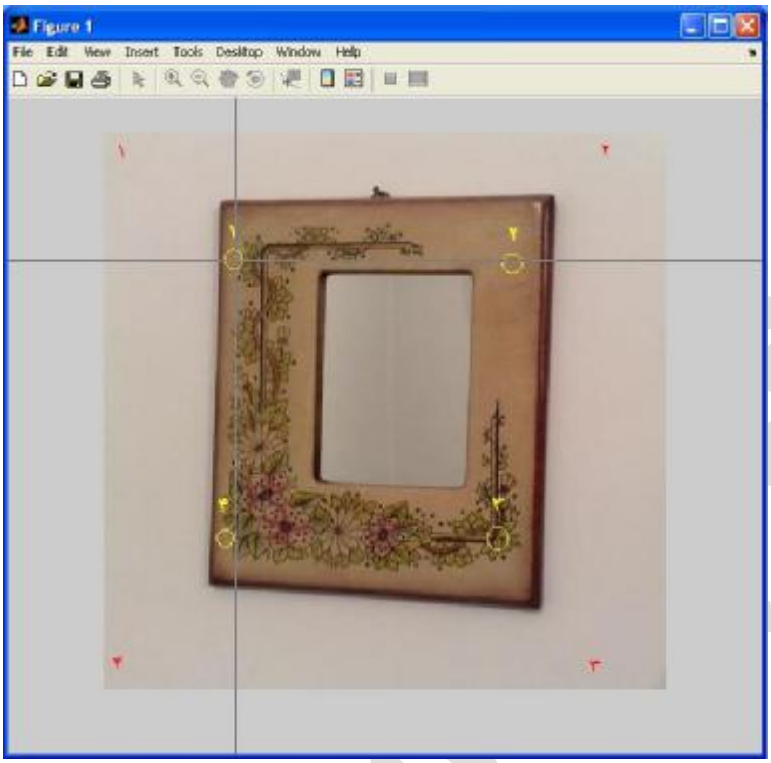
```
    s(2*i,5)=base_points(2*i);
    s(2*i,6)=1;
    s(2*i,1:3)=0;
    s(2*i,8)=-(base_points(2*i)*input_points(2*i));
    s(2*i,7)=-(base_points(2*i-1)*input_points(2*i));
end
c=inv(s)*input_points;
%c=inv(s'*s)*s'*input_points;
%-------------------------------------------------------------------------
disp('raveshe daron yabiye morede nazare khod ra')
disp('ba enter kardane adade marbote entekhab konid:')
disp('1- nearest neighbour')
disp('2- bilinear interpolation')
disp('3- bicubic interpolation')
ravesh=input('1/2/3? [1]:');
if isempty(ravesh)
    ravesh=1;
end
%----------------------------nearest neighbour-----------------------------
switch ravesh
    case 1
        f=xmin;
        g=ymin;
        for i=1:dx
            for j=1:dy
                x=(c(1)*f+c(2)*g+c(3))/(c(7)*f+c(8)*g+1);
                y=(c(4)*f+c(5)*g+c(6))/(c(7)*f+c(8)*g+1);
                g=g+1;
                if x<1 | x>size(q,1)
                    output_image(i,j,:)=0;
                    elseif y<1 | y>size(q,2)
                    output_image(i,j,:)=0;
                else
                    output_image(i,j,1)=q(round(x),round(y),1);
                    output_image(i,j,2)=q(round(x),round(y),2);
                    output_image(i,j,3)=q(round(x),round(y),3);
                end
            end
            f=f+1;
            g=ymin;
        end
figure,imshow(uint8(output_image));
imwrite(uint8(output_image),'resampled_nearest.jpg');
%--------------------------bilinear interpolation------------------------
    case 2
        f=xmin;
        g=ymin;
        for i=1:dx
            for j=1:dy
                x=(c(1)*f+c(2)*g+c(3))/(c(7)*f+c(8)*g+1);
                y=(c(4)*f+c(5)*g+c(6))/(c(7)*f+c(8)*g+1);
                g=g+1;
                if x<1 | x>size(q,1)
                    output_image(i,j,:)=0;
                    elseif y<1 | y>size(q,2)
                    output_image(i,j,:)=0;
                else
l=floor(x);
k=floor(y);
aa=x-l;
bb=y-k;
output_image(i,j,1)=q(l,k,1)+aa*(q(l+1,k,1)-q(l,k,1))+bb*(q(l,k+1,1)-
q(l,k,1))+aa*bb*(q(l,k,1)+q(l+1,k+1,1)-q(l,k+1,1)-q(l+1,k,1));
output_image(i,j,2)=q(l,k,2)+aa*(q(l+1,k,2)-q(l,k,2))+bb*(q(l,k+1,2)-
q(l,k,2))+aa*bb*(q(l,k,2)+q(l+1,k+1,2)-q(l,k+1,2)-q(l+1,k,2));
```

```
output_image(i,j,3)=q(l,k,3)+aa*(q(l+1,k,3)-q(l,k,3))+bb*(q(l,k+1,3)-
q(l,k,3))+aa*bb*(q(l,k,3)+q(l+1,k+1,3)-q(l,k+1,3)-q(l+1,k,3));
                end
            end
            f=f+1;
            g=ymin;
        end
figure,imshow(uint8(output_image));
imwrite(uint8(output_image),'resampled_bilinear.jpg');
%-----------------------bicubic interpolation-----------------------------
    case 3
        f=xmin;
        g=ymin;
        for i=1:dx
            for j=1:dy
            x=(c(1)*f+c(2)*g+c(3))/(c(7)*f+c(8)*g+1);
            y=(c(4)*f+c(5)*g+c(6))/(c(7)*f+c(8)*g+1);
            g=g+1;
            if x<2 | x>size(q,1)
                output_image(i,j,:)=0;
                elseif y<2 | y>size(q,2)
                output_image(i,j,:)=0;
            else
            l=floor(x);
            k=floor(y);
            mm=l-1;
            nn=k-1;
            mmm=l+2;
            nnn=k+2;
            sum2=0;
            sum1=0;
            for m=mm:mmm
                 for n=nn:nnn
                    if (abs(x-m))<1
                        hx=1-2*(abs(x-m))^2+(abs(x-m))^3;
                    elseif 1<=(abs(x-m))<2
                        hx=4-8*(abs(x-m))+5*(abs(x-m))^2-(abs(x-m))^3;
                    else
                        hx=0;
                    end
                    if (abs(y-n))<1
                        hy=1-2*(abs(y-n))^2+(abs(y-n))^3;
                    elseif 1<=(abs(y-n))<2
                        hy=4-8*(abs(y-n))+5*(abs(y-n))^2-(abs(y-n))^3;
                    else
                        hy=0;
                    end
                sum1=sum1+(q(m,n)*hx*hy);
                end
            sum2=sum2+sum1;
            end
            output_image(i,j,1)=sum2;
            output_image(i,j,2)=sum2;
            output_image(i,j,3)=sum2;
            end
        end
        f=f+1;
        g=ymin;
       end
figure,imshow(uint8(output_image));
imwrite(uint8(output_image),'resampled_bicubic.jpg');
end
```

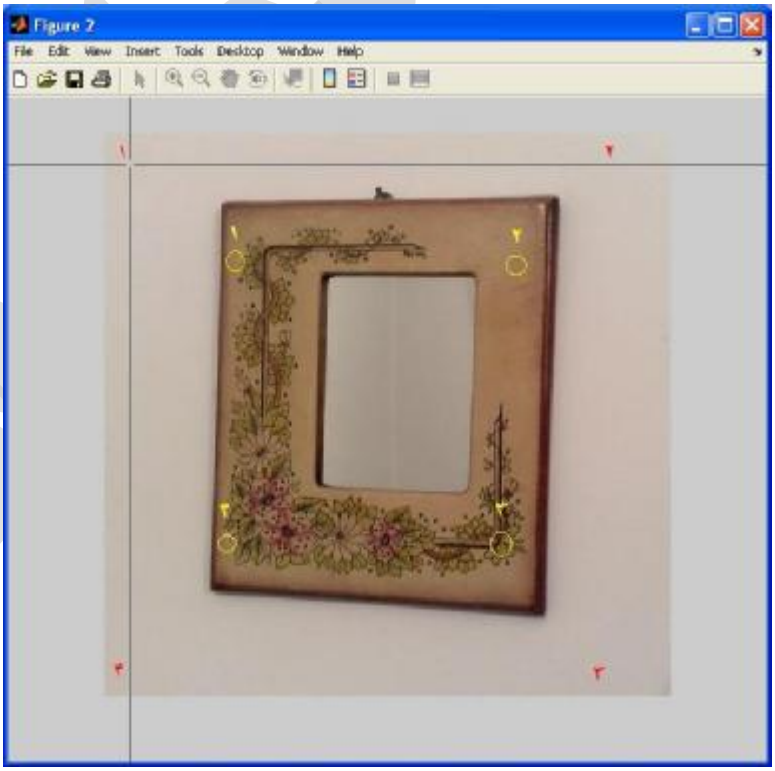
پایان

3 - 2 - تصویر ورودی، نحوه انتخاب گوشه ها و تصاویر خروجی برنامه با هر یک از سه روش

ابتدا ۴ نقطه اول (زرد رنگ) را کلیک می کنیم:



سپس ۴ نقطه گوشه عکس (قرمز رنگ) را کلیک می کنیم:



روش مورد نظرمان را انتخاب می کنیم:

```
Command Window
raveshe daron yabiye morede nazare khod ra
ba enter kardane adade marbote entekhab konid:
1- nearest neighbour
2- bilinear interpolation
3- bicubic interpolation
1/2/3? [1]:2
```

و در نهایت تصویر اولیه و خروجی های برنامه:

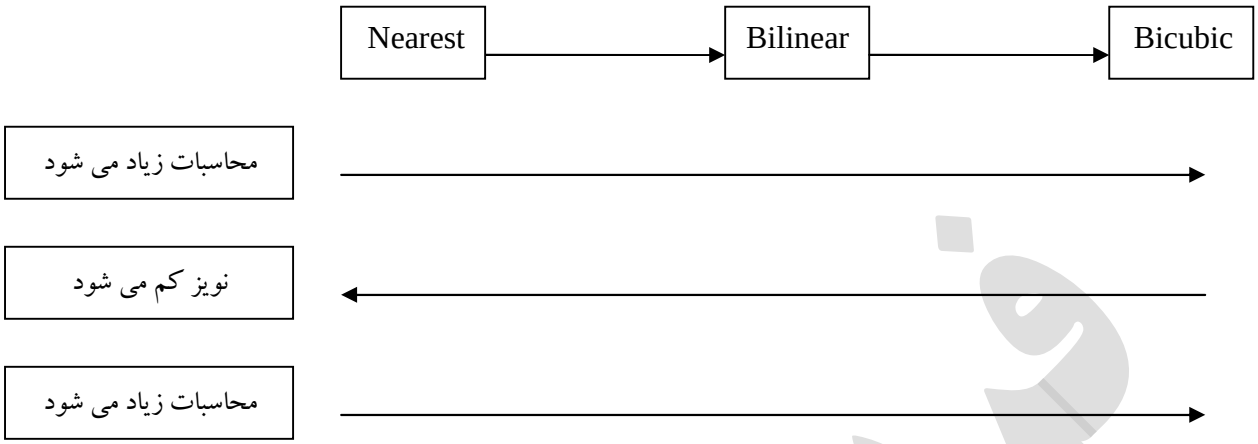
تصویر اولیه

Nearest

Bilinear

3 - 3 - مقایسه بین سه روش و نتایج آنها؛ نتیجه گیری و پیشنهادات

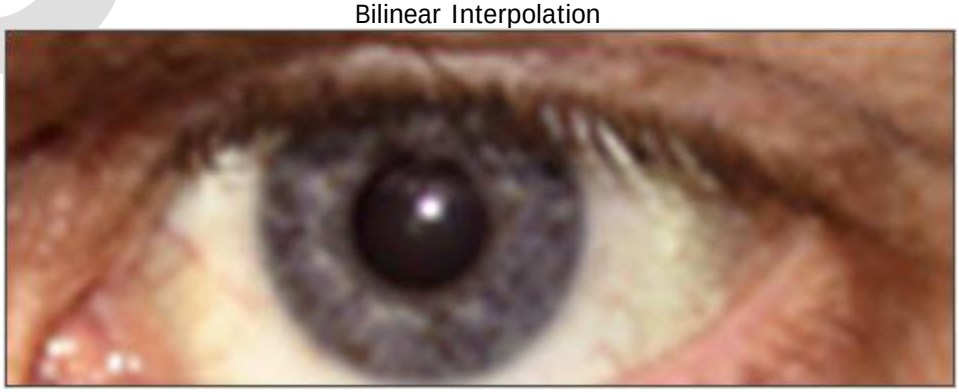
برای مقایسه ابتدا نموداری را که در کلاس توضیح داده شد را اینجا نیز می کشم:



نمونه سه تصویر که به طور جداگانه با هر یک از سه روش توضیح داده شده درون یابی شده اند را در زیر ملاحظه می فرمایید:

Nearest Neighbor Interpolation

Bilinear Interpolation



Bicubic Interpolation

مزایا و معایب هر یک از روشها را در کنار توضیحات هر یک در بخش های قبلی جداگانه بیان کردم. اینکه کدام روش نمونه برداری مناسب تر است، معیار های مختلفی دارد. اولین مسئله کاربرد تصویر است. اگر قرار باشد تصویر برای طبقه بندی به کار رود، آنگاه استفاده از روشی نظیر نزدیکترین همسایه که کمترین اثر را روی درجات خاکستری دارد، می تواند انتخاب مناسبتری نسبت به روش های درون یابی دیگر باشد. چراکه ایجاد درجات خاکستری غیر واقعی گاه روش های طبقه بندی را به اشتباه می اندازد. ولی اگر قرار است فقط از تصویر تصحیح شده در نفسیر بصری کمک گرفته شود، می توان از روش های درون یابی دیگر نظیر بر آورد مکعبی نیز استفاده نمود. امکانات کامپیوتری نیز از دیگر عوامل مؤثر روی انتخاب روش بهینه می باشد. اگر کامپیوترمان از سرعت پردازش کمی برخوردار است، استفاده از روشهای کندتر و حجیم درون یابی اصلاً توصیه نمی گردد. زیرا زمان فوق العاده زیادی برای انجام کار مصرف خواهد شد. این موضوع خود مشکلات زیادی را به دنبال خواهد داشت و روند پروژه را دچار اشکال می کند. بنابر این همیشه با توجه به کاربرد و امکانات مورد نظر بهترین روشی که نرم افزارِ مورد استفاده در اختیار ما قرار می دهد را انتخاب می کنیم.

با تشکر

فرید اسماعیلی

3 - 4 - منابع و مراجع

۲. کتاب پردازش تصویر رقمی، تألیف رافائل سی. گونزالس و ریچارد ای.وودز، ترجمه دکتر مرتضی خادمی و مهندس داوود جعفری، انتشارات دانشگاه فردوسی مشهد، چاپ سوم بهار ۱۳۸۵

۳. کتاب فتوگرامتری تحلیلی تألیف دکتر جلال امینی، انتشارات دانشگاه تهران، چاپ اول ۱۳۸۵

۴. کتاب مبانی سنجش از دور، تألیف مهندس سید باقر فاطمی و مهندس یوسف رضایی، انتشارات آزاده، چاپ اول اسفند ۸۴

۵. کتاب پردازش تصاویر ماهواره ای با نرم افزار ژئوماتیکا، ترجمه و تألیف مهندس مجتبی جورین سر، انتشارات سازمان جغرافیایی نیروهای مسلح، چاپ اول ۱۳۸۵

ضمائم

لوح فشرده پروژه شامل فایل های برنامه نوشته شده، تصویر اولیه و تصاویر خروجی برنامه به انضمام متن گزارش کار حاضر که به پیوست تقدیم می گردد.

نمایی از Autorun لوح فشرده پروژه و مطالب موجود در این CD