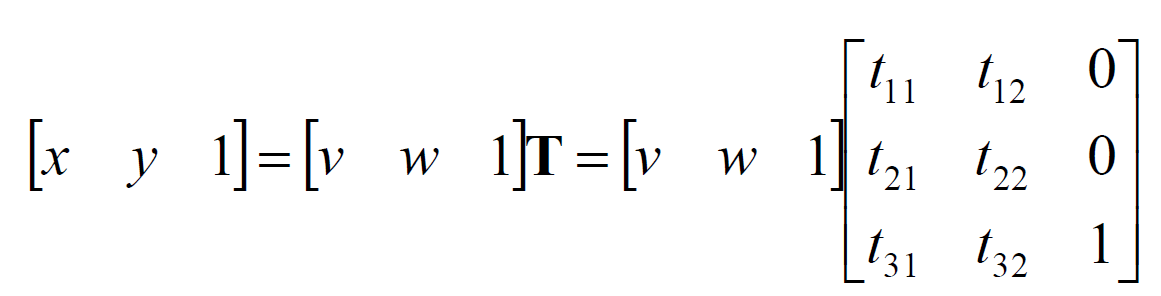
گزارش تمرین اول

مجید عسکری

|  |  |  |
| --- | --- | --- |
| اطلاعات گزارش |  | چکیده |
| **تاریخ: 27/12/97** |  | در اين گزارش، به توضیح و پیاده‌سازی تبدیل هندسی Affine با دو الگورتم درونیابی نزدیک‌ترین همسایه و دوخطی خواهیم پرداخت. در انتها با استفاده از الگوریتم‌های پیاده شده چند تبدیل خواسته شده را بر روی تصویر تست انجام می‌دهیم. |
| **واژگان كليدي:**  تبدیل هندسی  درون‌یابی  دوخطی  نزدیک‌ترین همسایه |  |

1-مقدمه[[1]](#footnote-1)

تبدیلات هندسی کاربرد فراوانی در پردازش تصاویر دارند. از این تبدیلات علی‌الخصوص برای اصلاح و استاندارد سازی تصاویر استفاده می‌شود. از تبدیلات پر استفاده هندسی تبدیل Affine می‌باشد که به صورت زیر تعریف می‌شود:



2-روش‌های پیاده‌سازی

پیاده سازی تبدیل affine به دو شیوه امکان پذیر است:

* روش مستقیم
* روش معکوس

در روش مستقیم به ازای هر پیکسل تصویر اصلی، موقعیت تبدیل شده آن‌را در تصویر جدید پیدا می‌کنیم و مقدار پیکسل را در آن قرار می‌دهیم.

در روش معکوس به ازای هر پیکسل در تصویر مقصد، سعی میکنیم پیکسل متناظر آن در تصویر اصلی را پیدا کنیم. اما از آنجا که انجام تبدیل معکوس از تصویر مقصد ممکن است مختصاتی صحیح به ما ندهد، نیاز داریم تا با استفاده از روش‌هایی، مقداری تقریبی برای نقطه خواسته شده به‌دست آوریم.

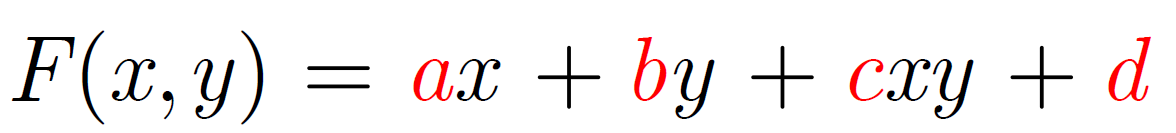
در این پیاده‌سازی از روش معکوس به همراه دو روش درون‌یابی نزدیک‌ترین همسایه و دو‌خطی استفاده کردیم..

**1-2 روش درون‌یابی نزدیک‌ترین همسایه**

در این روش از مقدار نزدیک‌ترین نقطه‌ی دارای مقدار به نقطه مجهول به عنوان مقدار نقطه مجهول استفاده می‌شود.

**2-2 روش درون‌یابی دوخطی**

در این روش از مقدار وزن‌دار نزدیک‌ترین 4 نقطه به نقطه مجهول به عنوان مقدار نقطه مجهول استفاده می‌شود. معمولا تابع درون‌یاب در این روش به صورت زیر در نظر گرفته می‌شود:

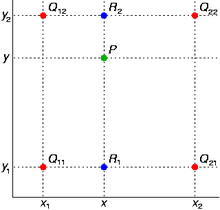
****

با حل 4 معادله (یک معادله برای هر یک از 4 نقطه) و 4 مجهول می‌توان ضرایب معادله را به‌دست آورده و سپس با قرار دادن مختصات نقطه مجهول مقدار آن را مشخص کرد.

3- پیاده‌سازی

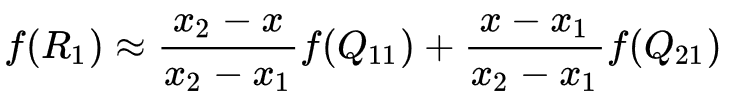
در پیاده‌سازی تبدیل affine از نرم‌افزار متلب استفاده کردیم. ابتدا یک تصویر خالی ایجاد کرده و برای هر پیکسل آن، با ضرب موقعیت پیکسل در ماتریس معکوس تبدیل، موقعیت پیکسل مرجع در تصویر اصلی را به دست آوردیم. برای مشاهده کد‌های پیاده‌سازی شده به فایل پیوست مراجعه کنید. توضیحات مربوط به کد به صورت کامنت در همان فایل‌ها موجود می‌باشد.

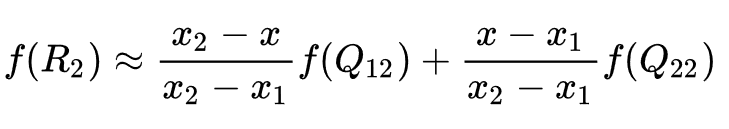
برای پیاده‌سازی درون‌یابی دوخطی از روش زیر استفاده کردیم:



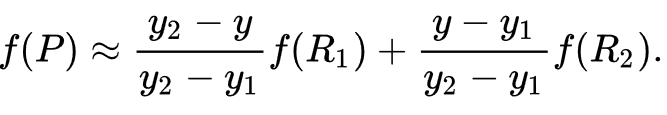
فرض کنید مقدار ناشناخته تابع f در نقطه P = (x، y) مدنظر باشد. اینگونه فرض خواهد شد که مقادیر f در نقاط Q11 = (x1، y1)، Q12 = (x1، y2)، Q21 = (x2، y1) و Q22 = (x2، y2) مشخص هستند.

ابتدا درون‌یابی در سمت x انجام خواهد شد و نتیجه زیر بدست می‌آید:





با درون‌یابی در سمت y ادامه خواهیم داد:

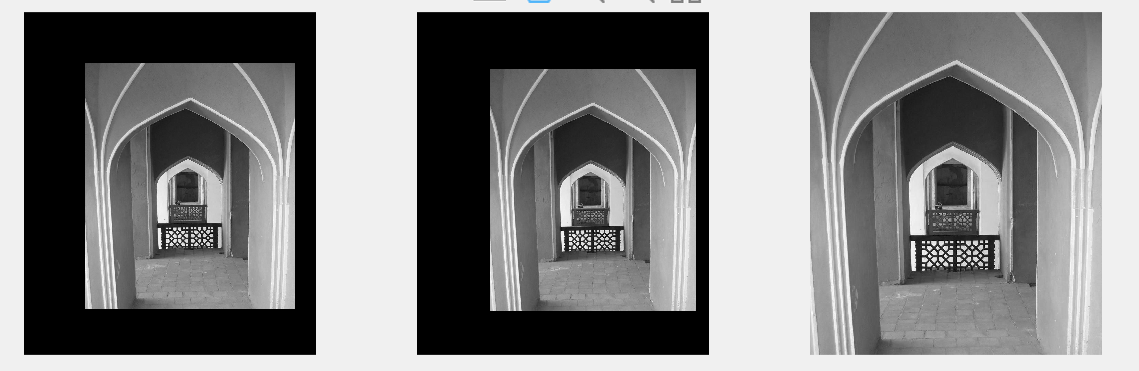
 بدین ترتیب مقدار تقریبی f(x,y) به‌دست خواهد آمد.

**4-خروجی**

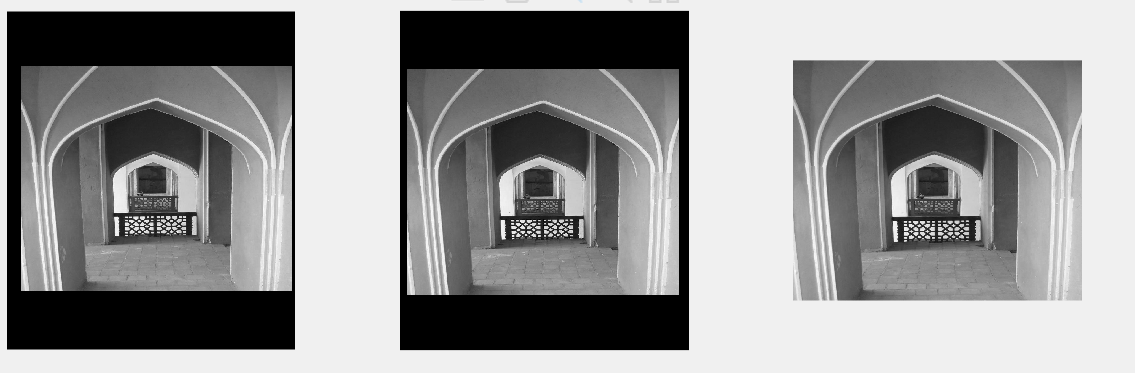
در این قسمت با انجام هر یک از تبدیلات خواسته شده به بررسی خروجی خواهیم پرداخت. در هر یک از تصاویر، تصویر سمت چپ مربوط به تبدیل با درون‌یابی دوخطی، تصویر وسط مربوط به درون‌یابی نزدیک‌ترین همسایه و تصویر سمت راست مربوط به خروجی تابع imwarp متلب می‌باشد.

1-4 انتقال به اندازه 102.5 و 180.2 پیکسل در جهات x و y

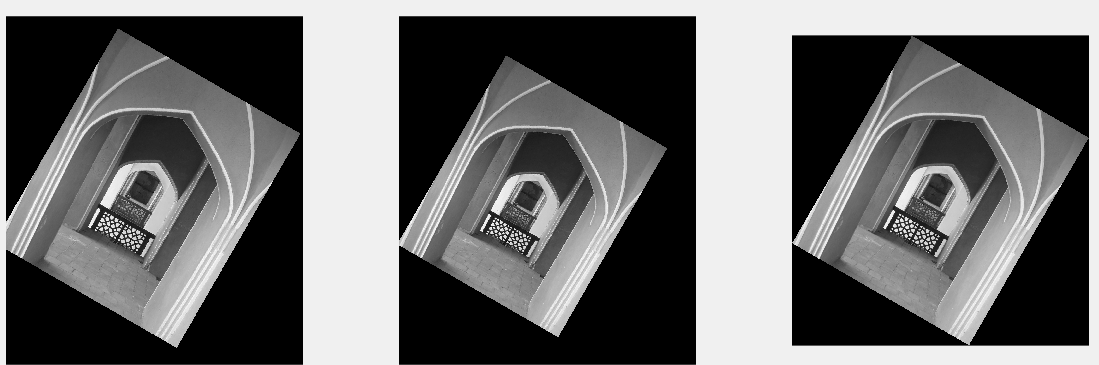
در این تبدیل خروجی تابع متلب همان تصویر اصلی است. برای مشاهده انتقال در خروجی الگوریتم خود نیز باید تصویر مقصد را بزرگتر از تصویر اصلی در نظر بگیریم.



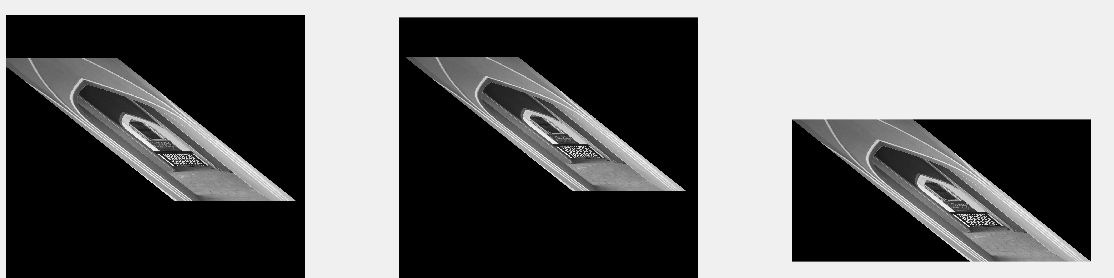
2-4 اسکیل به اندازه 1.2 و 0.85 در جهات x و y



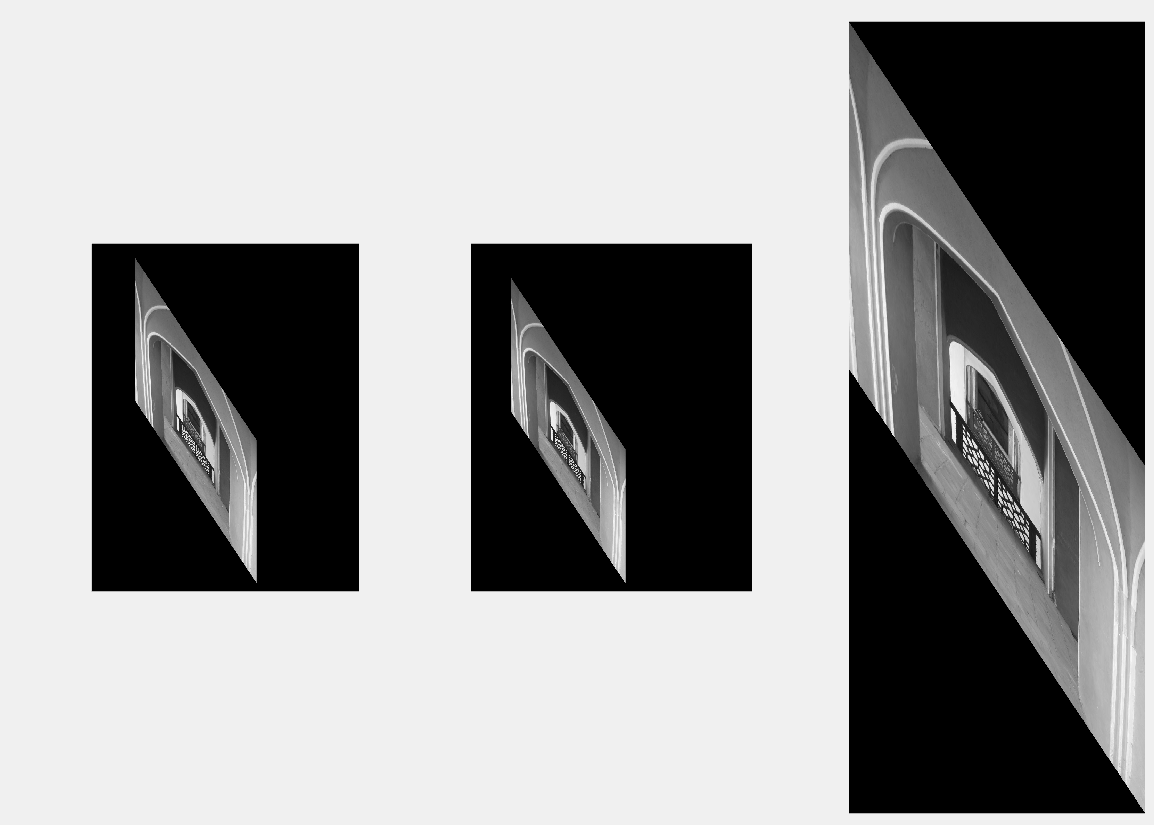
3-4 دوران نسبت به مرکز تصویر به اندازه 30 درجه



4-4 shear عمودی با مقدار sv=1.25

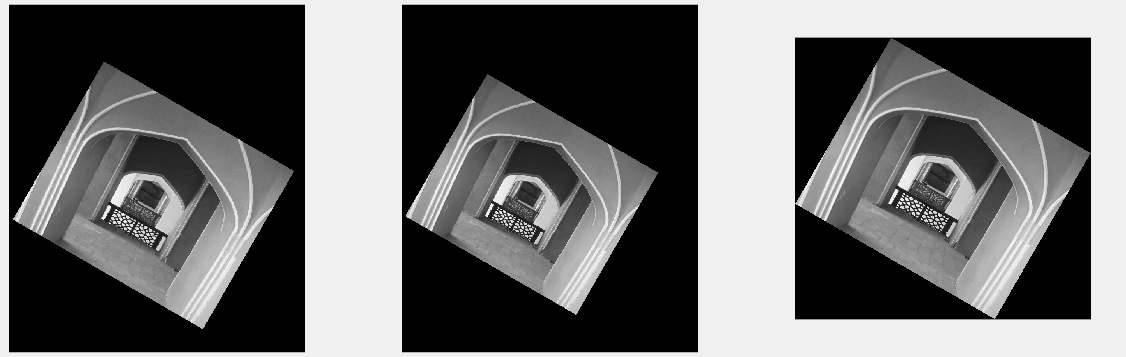


5-4 shear افقی با مقدار sh=1.5

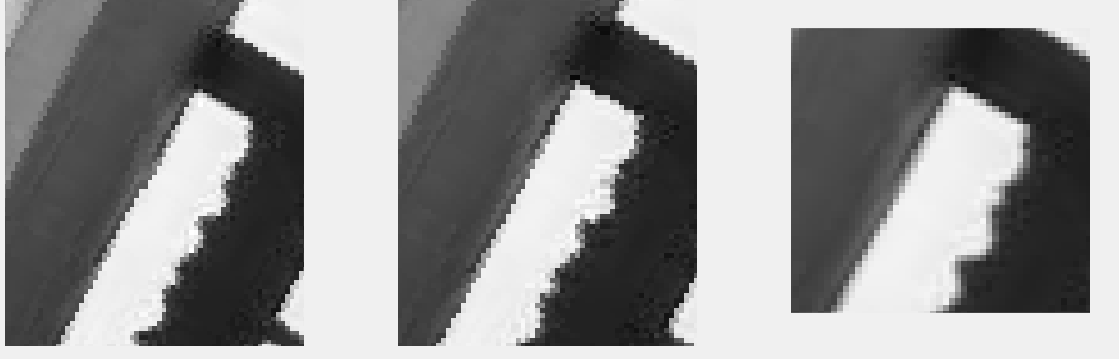


6-4 انتقال به اندازه 102.5 و 180.2 پیکسل در جهات x و y، اسکیل به اندازه 1.2 و 0.85 در جهات x و y و دوران نسبت به مرکز تصویر به اندازه 30 درجه

در این تبدیل می‌توان هر یک از تبدیلات را یک به یک اعمال کرد. اما روش بهتر ضرب ماتریس های تبدیل در یکدیگر و سپس اعمال آن بر روی تصویر اصلی است.



5- مقایسه دو روش درون‌یابی و نتیجه‌گیری



همانطور که در تصویر بالا مشاهده می‌شود، روش دوخطی (تصویر چپ) نتیجه‌ی بهتری مخصوصا در لبه‌های تصویر نسبت به روش نزدیک‌ترین همسایه به وجود می‌آورد. علت این پدیده استفاده از 4 پیکسل‌ به‌جای یک پیکسل نزدیک برای تخمین مقدار مجهول است.

6-پیوست

7-مراجع

1. <https://en.wikipedia.org/wiki/Bilinear_interpolation>
2. <https://en.wikipedia.org/wiki/Invertible_matrix>
3. <http://mathworld.wolfram.com/MatrixInverse.html>

1. [↑](#footnote-ref-1)