بسم الله الرحمن الرحیم

گزارش نهایی درس عملیات فتوگرامتری2

استاد راهنما : جناب آقای مهندس نیما زرین پنجه

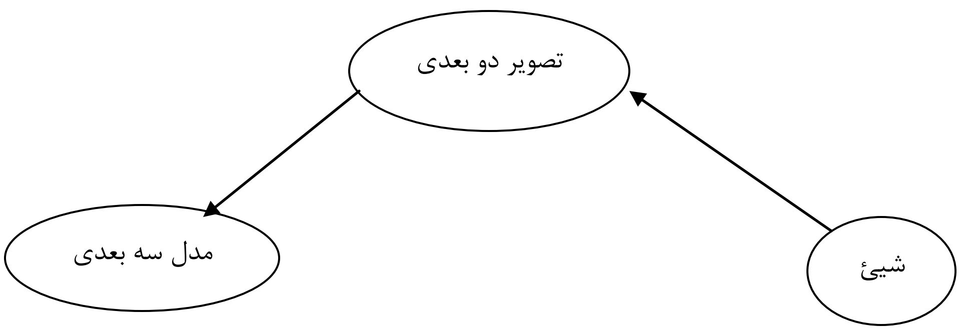
تهیه کننده : سپیده آبادپور

فهرست:

مقدمه:

فتوگرامتری بنا به تعریف عبارت است از هنر، علم و تکنولوژی تهیه ی اطلاعات قابل اعتماد درباره ی عوارض فیزیکی و محیط از طریق ثبت، اندازه گیری و تفسیر بر روی عکس و روی سایر مدارکی که در بر دارنده ی نقشی از انرژی الکترومغناطیس تابشی ثبت شده باشد.

بنابراین این علم به طور کلی بر تجزیه و تحلیل عکسی استوار است. در واقع در فتوگرامتری سعی می کنیم از طریق تصویر دو بعدی که از شی اخذ شده است به مدل سه بعدی آن دست یابیم.



هدف اصلی در درس عملیات فتوگرامتری 2 آشنایی با دستگاه های تبدیل (stereo plotter) می باشد که کار با نمونه های رقومی این دستگاه ها به صورت پروژه ی نهایی این درس دنبال خواهد شد. در چهار پروژه ی اول این درس سعی می کنیم با نوشتن برنامه هایی به زبان MATLAB چگونگی کار با این دستگاه ها را شبیه سازی کنیم.

پروژه ی یک:

توجیه سنسورها (دوربین ها) (sensor orientation) خود به دو بخش توجیه داخلی (interior orientation) و توجیه خارجی (exterior orientation) تقسیم می شود .

در توجیه داخلی هدف ، بازسازی هندسه ی داخل دوربین در لحظه ی عکسبرداری می باشد. (بازسازی هرم عکسی در لحظه ی عکسبرداری) در این قسمت از کار ما سعی می کنیم به پارامترهای زیر دست یابیم تا بتوانیم عکس را توجیه داخلی کنیم:

1 -فاصله ی کانونی
2 -مختصات مرکز تصویر
3 -اعوجاج عدسی
4 -خطای تغییر بعد و .....

ولی در توجیه خارجی هدف، تعیین موقعیت و وضعیت دوربین در لحظه ی عکسبرداری نسبت به شی می باشد. در این قسمت از کار سعی داریم که به مختصات مرکز تصویر نسبت به سیستم شیئی در لحظه ی عکسبرداری برسیم و همچنین وضعیت دوران های هرم عکسی را نسبت به سیستم شیئی مشخص کنیم.

پس از مرحله ی توجیه در واقع توانسته ایم رابطه ی بین عکس و شیئ را تعیین کنیم و می توان عملیات استخراج اطلاعات سه بعدی را به انجام رساند. در این مرحله عملیات گفته شده در زیر توسط یک دستگاه تبدیل (stereo plotter) انجام می گیرد.

1 -قرائت مختصات عکسی نقاط
2 -تعیین موقعیت سه بعدی نقاط
3 -ایجاد برجسته بینی در صورت نیاز
4 -ذخیره سازی اطلاعات
5 -ترسیم نقشه

این دستگاه های تبدیل در انواع زیر یافت می شوند:

1 -اپتیکی یا نوری

2 -مکانیکی
3 -تحلیلی
4 -رقومی

دستگاه های اپتیکی:

این دستگاه ها قدیمی ترین نسل دستگاه های تبدیل می باشند. در این دستگاه ها مشاهدات و محاسبات هر دو به روش دستگاهی صورت می گیرد. بازتابش تصویر بر روی میز فرآیند عکسبرداری را شبیه سازی می کند و از طریق دوران پروژکتورها امکان توجیه نسبی و سه بعدی بینی فراهم میگردد و عوارض مستقیما توسط عامل تبدیل با مداد ترسیم می گردد.

دستگاه های مکانیکی:

در این نوع دستگاه ها نیز مشاهدات و محاسبات هر دو به روش دستگاهی صورت می گیرد ولی به جای شعاع های نوری از دو میله (کاردان) استفاده می شود.

دستگاه های تحلیلی:

در این نوع دستگاه ها مشاهدات به روش دستگاهی و محاسبات توسط کامپیوتر صورت می گیرد. انجام محاسبات توسط کامپیوتر ممکن است به صورت تحلیلی یا عددی باشد. روش های عددی در کامپیوترهای قدیمی تر به علت قدرت پردازش کم این کامپیوترها به کار گرفته می شد. در این روش ها مسئله ی مورد نظر به صورت تقریبی و با تکرار حل می شود. در کامپیوتر های جدید تر از روش های تحلیلی بهره گرفته میشود که مسئله را به صورت دقیق و با معادلات تحلیلی حل می کند.

دستگاه های رقومی:

آخرین نسل دستگاه های تبدیل هستند که در آن ها مشاهدات و محاسبات هر دو به وسیله ی کامپیوتر صورت می گیرد و از روش های تحلیلی جهت حل مسئله استفاده می شود.

در فتوگرامتری 2 سعی بر این است که روش های توجیه نسبی و تهیه ی مدل سه بعدی توسط دستگاه ها فرا گرفته شود و برای نیل به این مهم تاکید بر دستگاه های اپتیکی است. بنابراین تمرکز ما در این گزارش بر روی دستگاه های اپتیکی خواهد بود.

همان طور که گفتیم در دستگاه های تبدیل نوری (optical stereo plotters) هدف، بازسازی معکوس فرآیند عکسبرداری می باشد که از طریق بازتابش تصویر از یک پروژکتور به این مهم دست می یابیم و با تقاطع بین شعاع های نوری حاصل از زوج عکس عوارض بر روی میز تشکیل میشود. سعی ما بر این است که با اعمال دوران هایی به پروژکتورها پارالاکس های y بین نقاط را صفر نماییم تا بتوان مدل سه بعدی را تشکیل داد. حال پس از تشکیل این مدل سه بعدی عامل تبدیل بایستی قادر باشد این مدل را به صورت سه بعدی ببیند تا بتواند اندازه گیری ها را انجام دهد. برای دستیابی به این هدف بایشتی چند شرط اساسی فراهم باشد:

1 -زوج تصویر باید موازی باشد (زاویه ی بین محورهای نوری دو عدسی نباید بیش از 7 درجه (نصف زاویه ی پارالاکتیک چشم انسان) باشد.

2 -زوج تصویر باید هم مقیاس باشد.

3 -زوج تصویر دارای باز باشد.

4 -تصویر چپ با چشم چپ و تصویر راست با چشم راست رویت شود (هر چشم تصویر مربوط به خود را ببیند.)

به منظور برقراری شرط 4 چند راه پیش رو داریم:

1 -استفاده از فیلترهای رنگی: بدین صورت که در مقابل پروژکتور چپ فیلتر قرمز و در مقابل پروژکتور راست فیلتری مخالف آن (سبز یا آبی) میگذارند. بنابراین از پروژکتور چپ فقط نور قرمز و از پروژکتور راست فقط نور آبی یا سبز عبور می کند سپس فیلتر هایی مشابه را برای چشم ها استفاده میکنند. در این صورت از فیلتر قرمز که به روی چشم چپ قرار گرفته فقط نور قرمز که مربوط به پروژکتور چپ (

عکس چپ) است گذر خواهد کرد و به چشم خواهد رسید و از فیلتر های سبز یا آبی که به روی چشم راست قرار گرفته اند فقط نور مربوط به عکس راست عبور خواهد کرد. بنابراین تصویر چپ با چشم چپ و تصویر راست با چشم راست رویت خواهد شد. مشکل این روش این است که در آن رنگ های مدل را از دست خواهیم داد و بنابراین عکس های رنگی با این روش قابل مشاهده نیستند. به این روش، روش آناگلیفیک گفته می شود.

2 -نمایش متناوب تصاویر با استفاده از شاترهای باز و بسته شونده : در این روش در مقابل پروژکتورها شاترهای باز و بسته شونده قرار می دهند، عینک ها هم حالت باز و بسته شونده دارند و در همان لحظه که شاتر پروژکتور راست باز و پروژکتور چپ بسته است عینک راست باز و عینک چپ بسته است و برعکس بدین ترتیب تصویر راست فقط با چشم راست و تصویر چپ با چشم چپ دیده میشود.

3 -استفاده از نور پلاریزه

یکی از مشکلات اصلی دستگاه های اپتیکی این است که در آن ها باید تصویر را فوکوس ببینیم تا بتوانیم عوارض را اندازه بگیریم یعنی رابطه ی نیوتن بایستی برقرار باشد:

$$\frac{1}{p}+\frac{1}{\mathrm{q}}=\frac{1}{\mathrm{f}}$$

اگر سطح میز نزدیک تر یا دورتر از میزان مشخص شده در این رابطه باشد، تصویر واضح نخواهد بود و برای هر نقطه دایره ی ابهام به وجود می آید.

البته آن چه در بالا گفته شد جنبه ی تئوریک مسئله است و در عمل اگر قطر دایره ی ابهام کمتر از 50 میکرون باشد باز هم تصویر فوکوس خواهد بود چون چشم انسان قادر به تفکیک فواصل کمتر از 50 میکرون نیست.

بنابراین در این جا دو اصطلاح تعریف خواهد شد:

عمق میدان نزدیک : فاصله ی نزدیک ترین فوکوس تصویر نسبت به عدسی

$$h_N=\frac{h}{1+(h-f)*c*\frac{f_{stop}}{f^2}}$$

عمق میدان دور : فاصله ی دورترین فوکوس تصویر نسبت به عدسی

$$h_f = \frac{h}{1-(h-f)*c*\frac{f_{stop}}{f^2}}$$

و عمق میدان از رابطه ی زیر بدست می آید:

Depth of field = h_f-h_n

در روابط بالا c قطر دایره ی ابهام، f فاصله ی کانونی پروژکتور و h فاصله ی پرده از عدسی می باشد.

F_{stop} یا f_{number} از رابطه ی زیر بدست می آید:

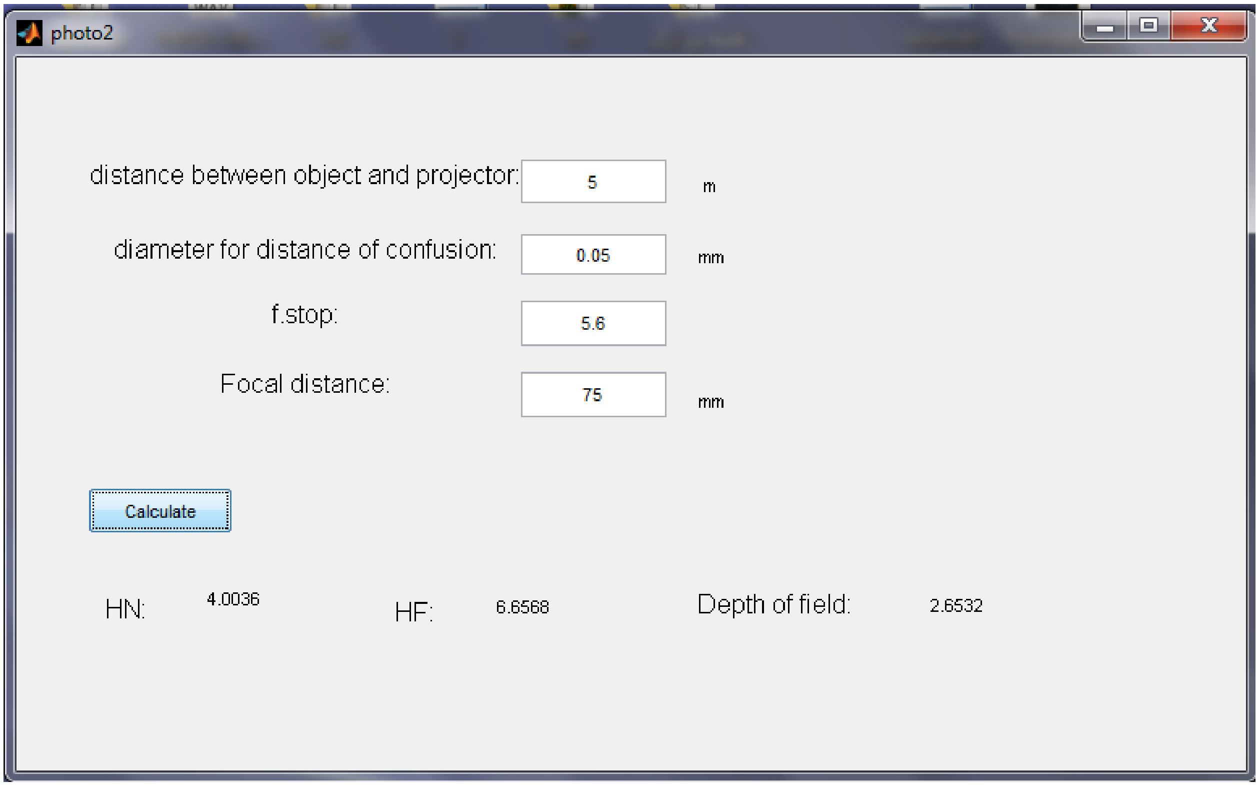
$$f_{stop} = \frac{f}{d}$$

که در آن f فاصله ی کانونی و d قطر دریچه ی دیافراگم است.

طبق روابط بالا به منظور افزایش عمق میدان بایستی h_f زیاد و h_n کم شود یعنی با فرض ثابت ماندن h و f و c ناگزیر هستیم که f_{stop} را افزایش دهیم پس باید d (قطر دریچه ی دیافراگم) را کم کنیم که البته کاهش بیش از حد قطر دریچه ی دیافراگم نیز باعث کاهش روشنایی مدل می شود پس در دستگاه های تبدیل اپتیکی ما همواره با محدوده ای از عمق میدان روبرو هستیم و نمی توان عمق میدان را تا هر حدی افزایش داد و تصویر را فوکوس دید.

در پروژه ی 1 از ما خواسته شد برنامه ای به زبان MATLAB بنویسیم که با داشتن h، f ، c و f_{stop} بتواند عمق میدان را محاسبه کند.

بدین منظور برنامه ای به صورت زیر طراحی گردید:



در برنامه ی بالا دیده می شود که کاربر بایستی h را بر حسب متر و f و c را بر حسب میلی متر وارد کند ولی f_{stop} بدون واحد می باشد چون اگر در رابطه ی مربوط به آن f و d را از یک واحد وارد کنیم واحدها در صورت و مخرج ساده خواهند شد و f_{stop} بدون واحد خواهد بود. حال به توضیح callback مربوط به pushbutton می پردازیم که در اینجا به آن نام calculate داده ایم.

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
h = str2num ( get ( handles.edit1 , 'string') );
c = str2num ( get ( handles.edit2 , 'string') )/1000;
fstop = str2num ( get ( handles.edit3,'string') );
f = str2num ( get ( handles.edit4,'string') )/1000;
```

توسط کدهای نوشته شده در box بالا آن چه را که کاربر به عنوان ورودی در editbox ها وارد کرده است کرده است برای محاسبات در اختیار برنامه می گذاریم و چون خاصیت string این box ها را get کرده ایم که در محاسبات قابل استفاده نیست با استفاده از تابع str2num آنها را به عدد تبدیل می کنیم.

در مورد c و f چون کاربر آن ها را بر حسب میلی متر وارد کرده ولی ما می خواهیم که به شکل متر در محاسبات شرکت کنند بعد از اعمال تابع str2num یک تقسیم بر 1000 انجام می گیرد.

```
hn = h / ( 1+((h-c)*c*fstop)/(f^2));
hf = h / ( 1-((h-c)*c*fstop)/(f^2));
dof = hf - hn;
```

توسط کدهای نوشته شده در box بالا معادلات مربوط به عمق میدان را بر ورودی ها اعمال کرده ایم.

```
HN = num2str ( hn );
HF = num2str ( hf );
set ( handles.text6,'string',HN );
set ( handles.text8,'string',HF);
DOF = num2str(dof);
set ( handles.text10,'string',DOF);
```

و در نهایت توسط کدهای نوشته شده در box بالا نتایج به دست آمده را به خروجی ارسال کرده ایم تا به عنوان string در textbox ها چاپ شود البته چون نتایج به صورت عدد بدست آمده اند ولی قرار است در خاصیت string این box ها set شوند ابتدا با تابع num2str نتایج را به string تبدیل و سپس آن ها را به خروجی ارسال کرده ایم.

نتایج بدست آمده در برنامه ی بالا نشان می دهد که با مشخصات تعیین شده برای دستگاه تبدیل اپتیکی یعنی f_{stop}=5.6 ، h=5 m و f=75 mm اگر بخواهیم قطر دایره ی ابهام از 0.05 mm فراتر نباشد میتوان میز تصویر را بین فاصله ی h=4.0036 m و h=6.6568 m یعنی در یک دامنه ی 2.6532 m بالا و پایین برد بدون آن که فوکوس تصویر به هم بخورد.

پروژه ی دو :

توجیهات در دستگاه های تبدیل اپتیکی:

1 -توجیه داخلی : هدف از آن بازسازی هندسه ی هرم دوربین در لحظه ی عکسبرداری در داخل پروژکتور تبدیل می باشد .

بدین منظور ابتدا از عکس مورد نظر یک دیاپوزیتیو تهیه می کنند که ابعاد این دیاپوزیتیو با توجه به نوع دستگاه اپتیکی مورد استفاده متفاوت است و مثلا در MULTIPLEX یک پنجم ابعاد عکس و در BALPLEX یک سوم ابعاد عکس است. کوچک بودن ابعاد دیاپوزیتیو به دلیل حل مشکل نوردهی یکنواخت به کل دیاپوزیتیو می باشد البته در دستگاه هایی مانند KELSH چون از روش نوردهی موضعی استفاده می کنیم نیازی به کوچک کردن ابعاد دیاپوزیتیو نیست. در مرحله ی آماده سازی دیاپوزیتیو سعی می شود که اعوجاجات عکسی (به عنوان مثال اعوجاج شعاعی عدسی دوربین هوایی) برطرف شود.

سپس هنگام استقرار دیاپوزیتیو از طریق انطباق نقاط فیدوشیال عکسی بر علائم موجود در قاب پروژکتور principal point عکس را بر محور اصلی دستگاه منطبق می گردانیم و از آنجایی که در این دستگاه ها فاصله اصلی با توجه به ابعاد دیاپوزیتیو ثابت است (نیازی به تنظیم فاصله ی اصلی نداریم) عملیات توجیه داخلی به اتمام می رسد.

2 -توجیه خارجی: هدف از آن تعیین مختصات زمینی سه بعدی نقاط می باشد که خود در دو مرحله ی توجیه نسبی و توجیه مطلق صورت می گیرد.

در مرحله ی توجیه نسبی موقعیت و وضعیت زوج عکس ها (در لحظه ی عکسبرداری) نسبت به یکدیگر بازسازی می شود و مدل سه بعدی تشکیل می گردد (بدون آنکه مقیاس اصلی و مختصات واقعی معلوم شود) و سپس در مرحله ی توجیه مطلق مدل سه بعدی تشکیل شده را به سیستم مختصات مشخص شده توسط کارفرما می بریم یعنی با اعمال دورانها، جابجایی ها و مقیاس به مدل به مختصات واقعی می رسیم.

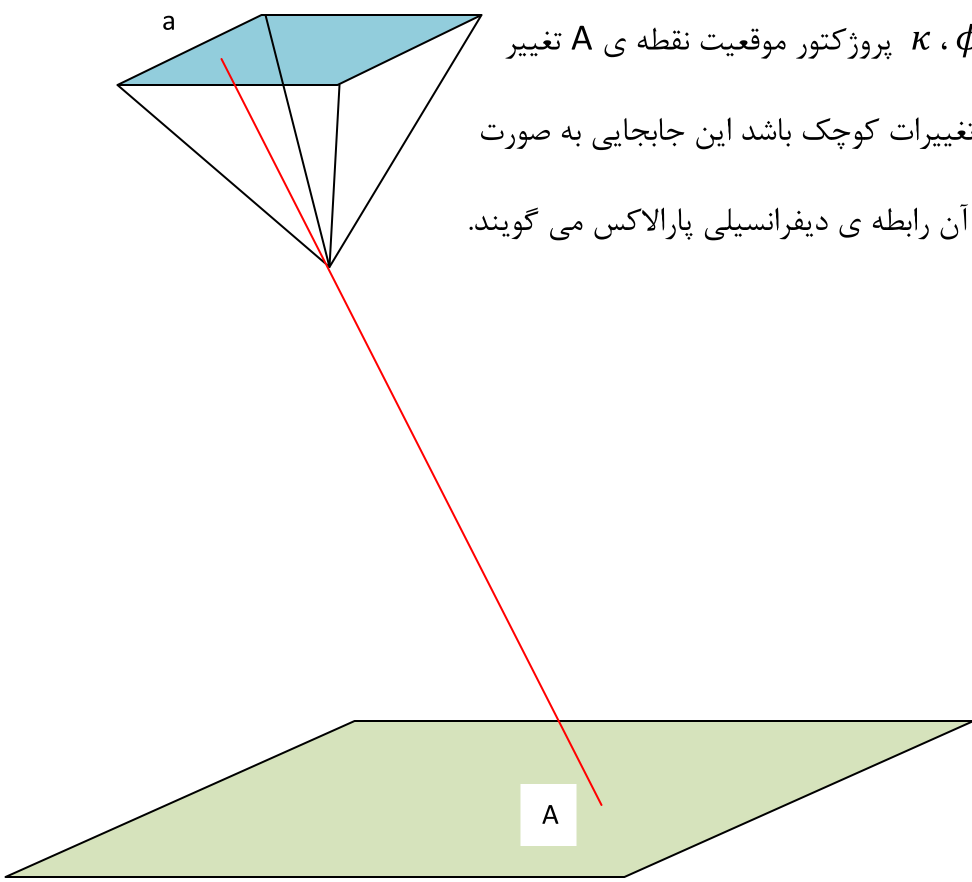
همان طور که گفتیم هدف از توجیه داخلی فقط بازسازی موقعیت و وضعیت دو دوربین نسبت به یکدیگر در لحظه ی عکسبرداری می باشد و اصلا به سیستم مختصات انتخابی ارتباط ندارد.

بنابراین سیستم مختصات مدل می تواند هرسیستم مختصاتی باشد ولی معمولا آن را به شکل زیر انتخاب می کنند.

- مبدا : مرکز عدسی پروژکتور چپ
- محور x : در راستای باز
- محور z : عمود بر میز به سمت بالا
- محور y : عمود بر هر دو محور و راستگرد

اگر موقعیت پروژکتور را به صورت زیر در نظر بگیریم:

با اعمال تغییرات در پارامترهای انتقالی x ، y ، z و همچنین در پارامترهای دورانی ω ، ϕ ، κ پروژکتور موقعیت نقطه ی A تغییر خواهد کرد که اگر این تغییرات کوچک باشد این جابجایی به صورت زیر فرموله می شود و به آن رابطه ی دیفرانسیلی پارالاکس می گویند.



$$dx_A = db_x + \frac{x}{z} db_z + z\left(1 + \frac{x^2}{z^2}\right) d\varphi - \frac{xy}{z} d\omega + y d\kappa$$

$$dy_A = db_y + \frac{y}{z} db_z + z\left(1 + \frac{y^2}{z^2}\right) d\omega - \frac{xy}{z} d\varphi + x d\kappa$$

معادلات بالا برای پروژکتور راست به صورت زیر در می آیند :

$$dx'_A = db_{x'} + \frac{x-b}{z} db_{z'} + z\left(1 + \frac{(x-b)^2}{z^2}\right) d\varphi' - \frac{(x-b)y}{z} d\omega' + y d\kappa'$$

$$dy'_A = db_{y'} + \frac{y}{z} db_{z'} + z\left(1 + \frac{y^2}{z^2}\right) d\omega' - \frac{(x-b)y}{z} d\varphi' + (x-b) d\kappa'$$

و پارالاکس برای نقطه ی A از رابطه ی زیر بدست می آید:

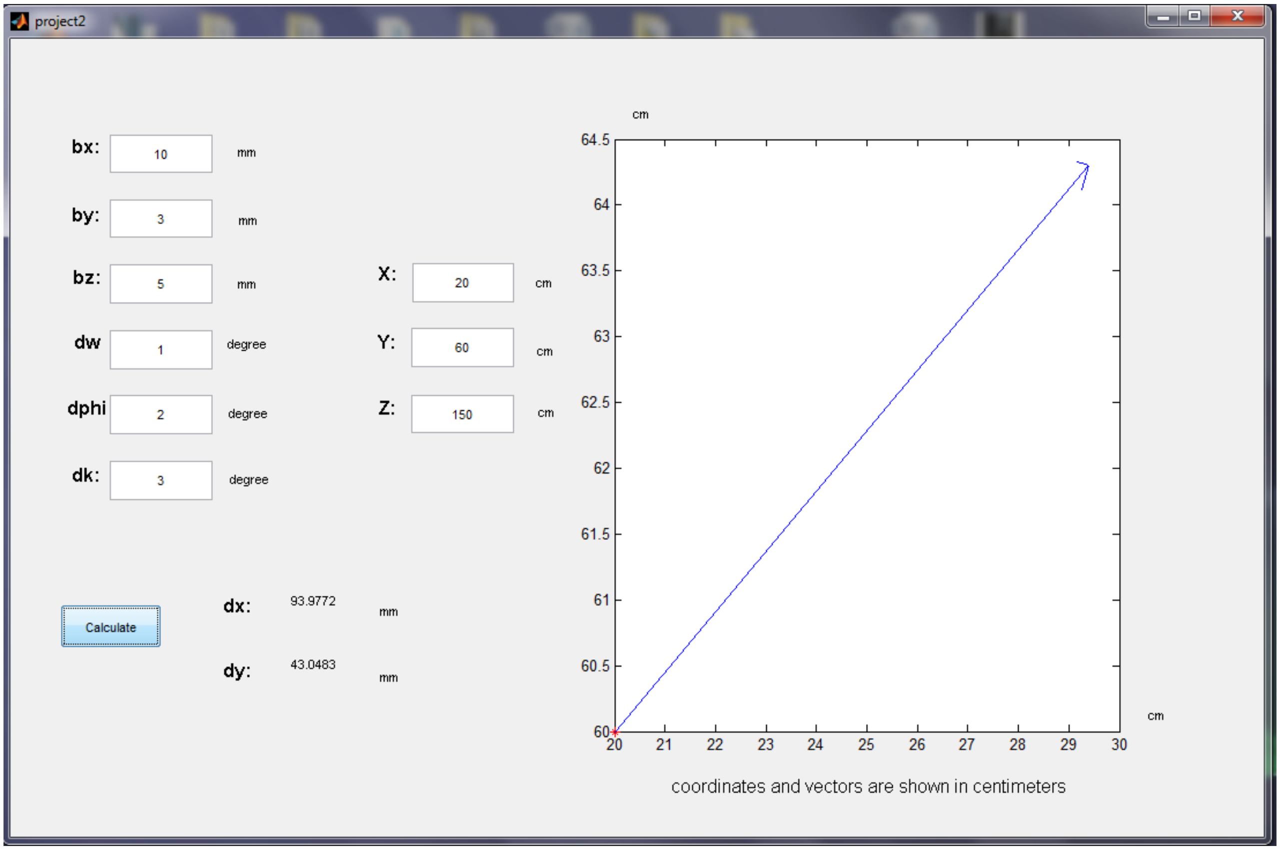
$$p_x = dx_A - dx'_A$$

$$p_y = dy_A - dy'_A$$

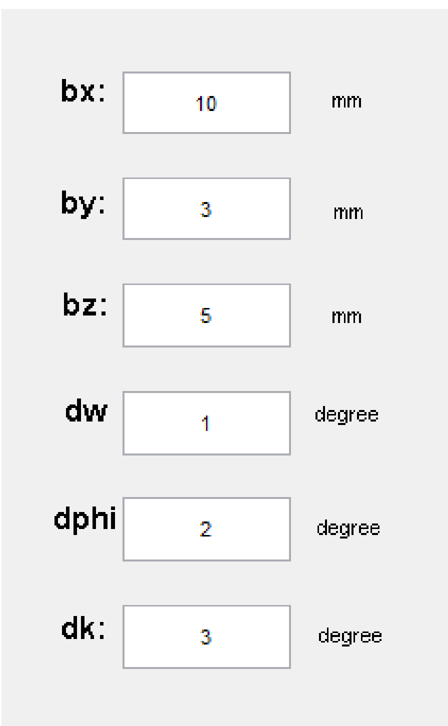
که p_x جهت اندازه گیری ارتفاع و p_y برای توجیه نسبی به کار می رود.

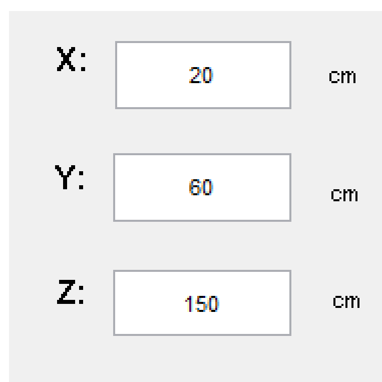
در پروژه ی دو از ما خواسته شد تا معادله ی دیفرانسیلی پارالاکس را برای پروژکتور چپ شبیه سازی کنیم یعنی با گرفتن تغییرات جزئی پارامترهای دورانی و انتقالی پروژکتور به عنوان ورودی و همچنین مختصات نقطه ی A ، جابجایی آن را در اثر اعمال این تغییرات بدست آوریم و بردار این جابجایی را رسم کنیم.

بدین منظور برنامه ای به صورت زیر طراحی گردید:



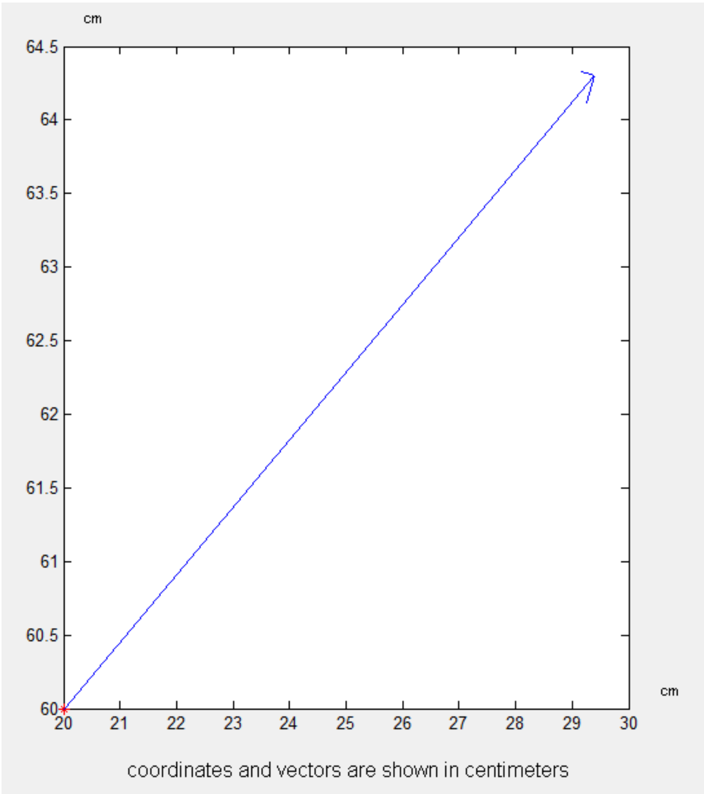
همان طور که دیده می شود کاربر در این برنامه تغییرات پارامترهای انتقالی را بر حسب میلی متر و تغییرات پارامترهای دورانی را بر حسب درجه وارد می کند همچنین مختصات نقطه ی مورد نظر بر حسب سانتی متر وارد خواهد شد.





که گفته شده بهتر است Z=150 و 0<X<100 و -100<Y<100 باشد. برنامه جابجایی نقطه ی A را بر حسب میلی متر بدست می دهد و بردار این جابجایی و مختصات نقطه ی A بر حسب

سانتی متر رسم خواهد شد.



حال به توضیح callback مربوط به pushbutton می پردازیم که در اینجا به آن نام calculate داده ایم.

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
pi=3.1416;
bx = str2num ( get ( handles.bx , 'string' ) )/1000;
by = str2num ( get ( handles.by , 'string' ) )/1000;
bz = str2num ( get ( handles.bz , 'string' ) )/1000;
dw = str2num ( get ( handles.dw , 'string' ) )*pi/180;
dphi = str2num ( get ( handles.dphi , 'string' ) )*pi/180;
dk = str2num ( get ( handles.dk , 'string' ) )*pi/180;
x = str2num ( get ( handles.x , 'string' ) )/100;
y = str2num ( get ( handles.y , 'string' ) )/100;
z = str2num ( get ( handles.z , 'string' ) )/100;
```

در کدهای نوشته شده در box بالا همان طور که در مورد پروژه ی یک گفته شد اطلاعات وارد شده توسط کاربر را گرفته و به عدد تبدیل می کنیم تا برای محاسبات در اختیار برنامه قرار دهیم. همچنین چون می خواهیم این اعداد بر حسب متر در محاسبات شرکت کنند در مورد عناصر انتقالی با ضریب $\frac{1}{1000}$ آنها را به متر و در مورد عناصر دورانی با ضریب $\frac{\pi}{180}$ آن ها را به رادیان تبدیل می کنیم همچنین مختصات نقطه ی مورد نظر را نیز با ضریب $\frac{1}{100}$ به متر تبدیل می کنیم.

```
dx = bx + ( x*bz/z ) + z*(1+(x/z)^2)*dphi - ( x*y*dw/z )+ y*dk
dy = by + ( y*bz/z ) + z*(1+(y/z)^2)*dw - ( x*y*dphi/z )+ x*dk
```

در box بالا همان فرمول های معادلات دیفرانسیلی پارالاکس پروژکتور چپ نوشته شده است.

```
set(handles.text21,'string',num2str(dx*1000))
set(handles.text22,'string',num2str(dy*1000))
```

در box بالا نتایج بدست آمده را که بر حسب متر بودند به میلی متر تبدیل و به خروجی ارسال کرده ایم.

همچنین در box روبرو مختصات نقطه ی A و جابجایی های آن را به سانتی متر تبدیل و با تابع arrow بردار را رسم کرده ایم.

تابع arrow یک m-file function است که به منظور رسم بردارها نوشته شده و من در پروژه ی 2 این m-file را ضمیمه کرده ام تا برنامه ها قابل اجرا باشند.

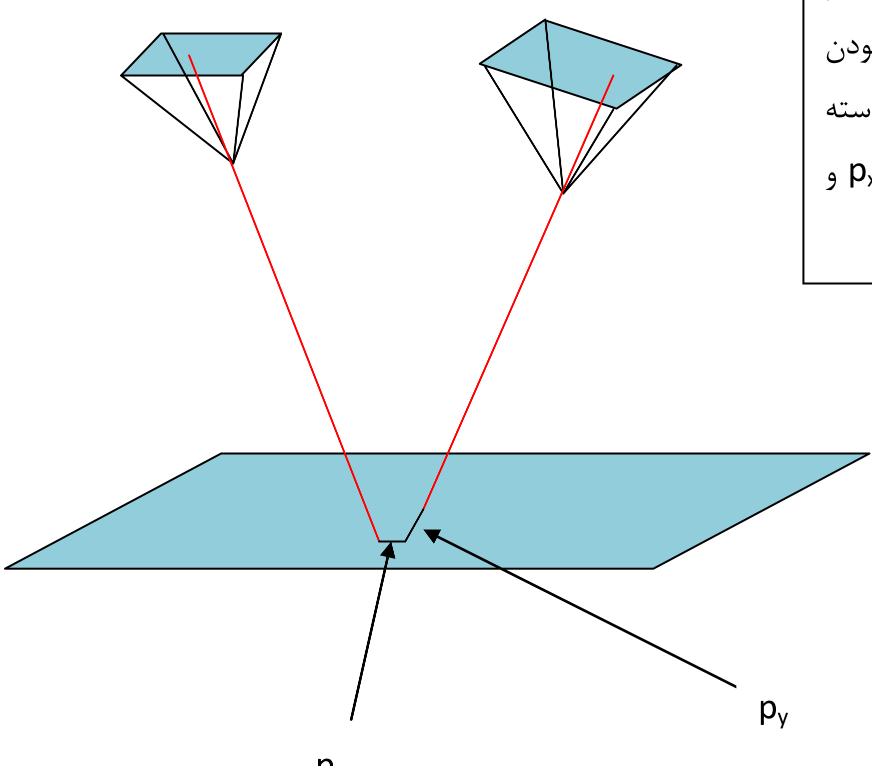
```
X=x*100;
Y=y*100;
plot(X,Y,'*r');
DX=dx*100;
DY=dy*100;
hold on;
p=[X,Y];
v=[DX,DY];
arrow(p,v);
```

پروژه ی سه:

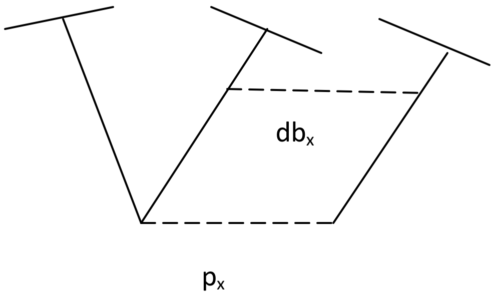
در این پروژه تصمیم داریم به طور خاص به انجام توجیه نسبی بپردازیم.

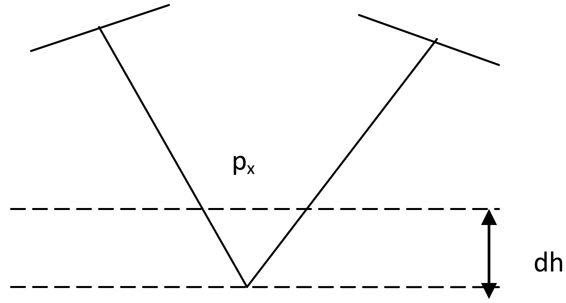
همان طور که گفتیم p_x فقط تابع ارتفاع صفحه ی ترسیم و یا باز دستگاه می باشد و با تغییر هریک از این پارامترها می توان p_x را تغییر داد البته تغییر باز دستگاه منجر به تغییر مقیاس مدل خواهد شد لذا بعد از انجام توجیه نسبی هنگام ترسیم عوارض با تغییر ارتفاع صفحه ی ترسیم سعی می کنند p_x را برای هر نقطه صفر کنند بدین ترتیب از تقاطع پرتوها نقطه ی مورد نظر تشکیل شده و با اندازه گیری میزان تغییر ارتفاع صفحه ی ترسیم ارتفاع نقطه ی مورد نظر بدست خواهد آمد. بدین ترتیب از p_x برای تعیین ارتفاع عوارض استفاده میشود.

در شکل روبرو دیده می شود که عدم تقاطع پرتوهای متناظر (متنافر بودن پرتوها) در مدل بعد از تشکیل دسته اشعه (توجیه داخلی) باعث ایجاد p_x و p_y می شود.



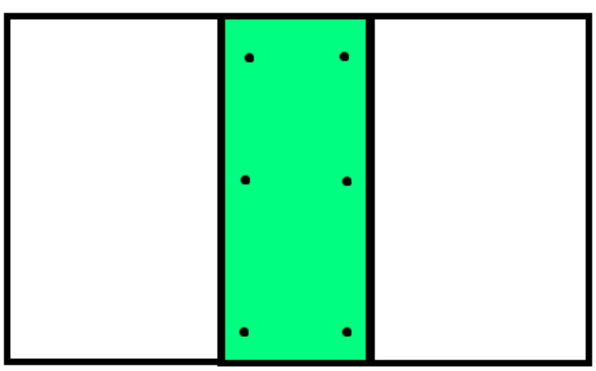
همچنین در شکل های زیر می بینیم که p_x فقط تابع باز دستگاه و ارتفاع صفحه ی ترسیم می باشد.





از p_y (منظور از y در این جا همان طور که در بحث انتخاب سیستم مختصات مدل گفتیم جهت عمود بر جهت باز می باشد) جهت انجام توجیه نسبی استفاده می شود ، هنگامی که p_y برای تمامی نقاط ناحیه ی مشترک بین زوج عکس صفر شود یعنی اینکه پرتوهای متناظر در یک صفحه (صفحه ی اپی پولار) قرار می گیرند و بنابراین دیگر متنافر نبوده ، بلکه متقاطع می باشند و با تغییر ارتفاع صفحه ی ترسیم می توان به نقطه ی تقاطع آن ها دست یافت و نقطه ی سه بعدی را تشکیل داد.(پس برای بازسازی شرایط دو دوربین در لحظه ی عکسبرداری و انجام توجیه نسبی لازم است پارالاکس در جهت عمود بر باز صفر شود.)

در شکل روبرو اگر ناحیه ی سبز رنگ ناحیه ی مشترک بین زوج عکس ها باشد ثابت شده است اگر برای 5 نقطه از 6 نقطه ی مشخص شده در این ناحیه $p_y=0$ باشد آن گاه برای تمام نقاط موجود در این ناحیه $p_y=0$ است و توجیه نسبی انجام شده است. از نقطه ی ششم عملا جهت سرشکنی و کنترل محاسبات استفاده می شود. به این نقاط، نقاط ون گروبر می گویند.



جهت صفر کردن پارالاکس y در این پنج لازم است که پروژکتورها را نسبت به هم دوران و انتقال دهیم.

با ادغام سه معادله ی زیر:

$$dy_A = db_y + \frac{y}{z}db_z + z\left(1 + \frac{y^2}{z^2}\right)d\omega - \frac{xy}{z}d\varphi + xd\kappa$$

$$dy'_A = db_{y'} + \frac{y}{z}db_{z'} + z\left(1+\frac{y^2}{z^2}\right)d\omega' - \frac{(x-b)y}{z}d\varphi' + (x-b)d\kappa'$$

$$p_y = dy_A - dy'_A$$

به معادله ی زیر می رسیم:

$$p_{\mathrm{y}} = db_y - db_{y'} + \frac{y}{z}db_z - \frac{y}{z}db_{z'} + z\left(1+\frac{y^2}{z^2}\right)d\omega - z\left(1+\frac{y^2}{z^2}\right)d\omega' - \frac{xy}{z}d\varphi + \frac{(x-b)y}{z}d\varphi' + xd\kappa - (x-b)d\kappa'$$

دیده می شود که از بین عناصر انتقالی و دورانی دو پروژکتور p_y اصلا ارتباطی به db_x یا $db_{x'}$ ندارد بنابراین جابجایی پروژکتورها در راستای باز هیچ اثری بر توجیه نسبی نخوهد داشت.

ولی جابه جایی پروژکتورهای چپ و راست در راستاهای عمود بر باز (db_y ، $db_{y'}$ ، db_z و $db_{z'}$) و همچنین دوران این پروژکتورها ($d\omega$ ، $d\omega'$ ، $d\varphi$ ، $d\varphi'$ ، $d\kappa$ و $d\kappa'$) همگی دز معادله ی p_y موجودند و برتوجیه نسبی تاثیر خواهند گذاشت.

از بین ده پارامتر موجود در معادله ی p_y پنج تای آن ها وابسته اند (یعنی تغییرات یکسانی را اعمال می کنند)

بنابراین با انتخاب پنج پارامتر از این ده پارامتر و انجام تغییرات روی آن ها توجیه نسبی را حل می کنیم. ولی آیا می توان این 5 پارامتر را به هر صورتی انتخاب کرد؟ یعنی آیا می توان

$$\binom{10}{5} = \frac{10!}{5!\,5!} = 252$$

انتخاب داشت؟ با نگاهی به معادله ی بالا و چگونگی وابستگی بین پارامترها معلوم می شود که نمی توان بعضی از پارامترها را با یکدیگر انتخاب کرد. در این مورد توضیح می دهیم:

1 -دیده میشود که db_y ضریب 1 و $db_{y'}$ ضریب 1- دارد و چون این دو ضریب مضرب یکدیگر می باشند از هم مستقل نیستند یعنی db_y و $db_{y'}$ اثرات یکسانی دارند و با هم انتخاب نمی شوند.

2 -dbz ضریب $\frac{y}{z}$ و dbz' ضریب $-\frac{y}{z}$ دارد و به همان دلیل بالا این دو نیز با هم انتخاب نمی شوند.

3 -ضرایب $d\omega$ و $d\omega'$ نیز قرینه (مضرب) همدیگرند و این دو پارامتر هم با هم انتخاب نمی شوند.

4 -$d\varphi$ دارای ضریب $-\frac{xy}{z}$ و $d\varphi'$ دارای ضریب $+\frac{(x-b)y}{z}$ است. این دو ضریب مضرب هم نمی باشند پس $d\varphi$ و $d\varphi'$ را می توان با هم انتخاب کرد ولی اگر آن ها را باهم انتخاب کنیم (در صورتی که $d\varphi=d\varphi'$ یعنی دوران ها هم اندازه و هم جهت باشند) به ضریب $-\frac{by}{z}$ میرسیم که مضربی است از ضرایب db_z یا $db_{z'}$.

بنابراین $d\varphi$ و $d\varphi'$ را می توان با هم انتخاب کرد ولی اگر این دو را با هم انتخاب کنیم دیگر نمی توانیم db_z یا $db_{z'}$ را انتخاب کنیم.

5 -به همان دلیل بالا $d\kappa$ و $d\kappa'$ را نیز می توان با هم انتخاب کرد ولی اگر این دو را با هم انتخاب کنیم و داشته باشیم $d\kappa=d\kappa'$ به ضریب b می رسیم که مضربی است از ضرایب db_y یا $db_{y'}$ ، بنابراین در صورت انتخاب $d\kappa$ و $d\kappa'$ با هم دیگر نمی توان db_y یا $db_{y'}$ را انتخاب کرد.

حال ببینیم از 252 حالت بالا چند حالت باقی می ماند. بدین منظور با استفاده از ترکیبات به حالت های ممکن زیر می رسیم.

1 -از بین db_y و $db_{y'}$ یکی انتخاب شده (2 حالت) ، از بین db_z و $db_{z'}$ نیز یکی انتخاب شده (2 حالت) و در مورد $d\omega$ و $d\omega'$ (2 حالت) $d\varphi$ و $d\varphi'$ (2 حالت $d\kappa$ و $d\kappa'$ (2 حالت) نیز مسئله همین طور است.

اصل ضرب : 2*2*2*2*2 = 32 حالت

2 -db_y یا $db_{y'}$ اصلا انتخاب نشده اند (1 حالت) ، از بین db_z و $db_{z'}$ یکی را (2 حالت) و از بین $d\omega$ و $d\omega'$ یکی را (2 حالت) و از بین $d\varphi$ و $d\varphi'$ نیز یکی را (2 حالت) انتخاب کرده ایم ولی مجبوریم هر دو $d\kappa$ و $d\kappa'$ را انتخاب کنیم.(1 حالت)

1*2*2*2*1 = 8 حالت

3 -از بین db_y و $db_{y'}$ یکی را انتخاب می کنیم (2 حالت) ولی از بین db_z و $db_{z'}$ هیچ کدام را انتخاب نمی کنیم.(1 حالت) از بین $d\omega$ و $d\omega'$ نیز یکی را انتخاب می کنیم (2 حالت) ولی در مورد $d\varphi$ و $d\varphi'$ مجبوریم هر دو را انتخاب کنیم (1 حالت) و از بین $d\kappa$ و $d\kappa'$ هم یکی را انتخاب می کنیم. (2 حالت)

2*1*2*1*2 = 8 حالت

4- db_y یا $db_{y'}$ اصلا انتخاب نمی شوند (1 حالت) همچنین db_z و $db_{z'}$ نیز اصلا انتخاب نمی شوند (1 حالت) ولی از بین $d\omega$ و $d\omega'$ یکی را انتخاب می کنیم (2 حالت) و از $d\varphi$ و $d\varphi'$ مجبوریم هر دو را انتخاب کنیم (1 حالت) و همچنین در مورد $d\kappa$ و $d\kappa'$ هم مجبوریم هر دو را انتخاب کنیم. (1 حالت)

1*1*2*1*1 = 2 حالت

بنابراین 32+8+8+2=50 حالت ممکن خواهیم داشت.

در عمل تنها 4 حالت مورد استفاده قرار می گیرد که دو حالت آن به صورت زیر است:

$$\begin{bmatrix} db_y \\ db_z \\ d\omega \\ d\varphi \\ d\kappa \end{bmatrix} \qquad \begin{bmatrix} db_{y'} \\ db_{z'} \\ d\omega' \\ d\varphi' \\ d\kappa' \end{bmatrix}$$

یعنی تنها از پارامترهای یک پروژکتور برای توجیه استفاده می شود که در این حالت به آن توجیه نسبی یکطرفه می گویند . این روش در اتصال مدل ها برای ساخت نوار مورد استفاده قرار می گیرد.

دو حالت بعدی به صورت زیر می باشند:

$$\begin{bmatrix} d\varphi \\ d\varphi' \\ d\kappa \\ d\kappa' \\ d\omega \end{bmatrix} \qquad \begin{bmatrix} d\varphi \\ d\varphi' \\ d\kappa \\ d\kappa' \\ d\omega' \end{bmatrix}$$

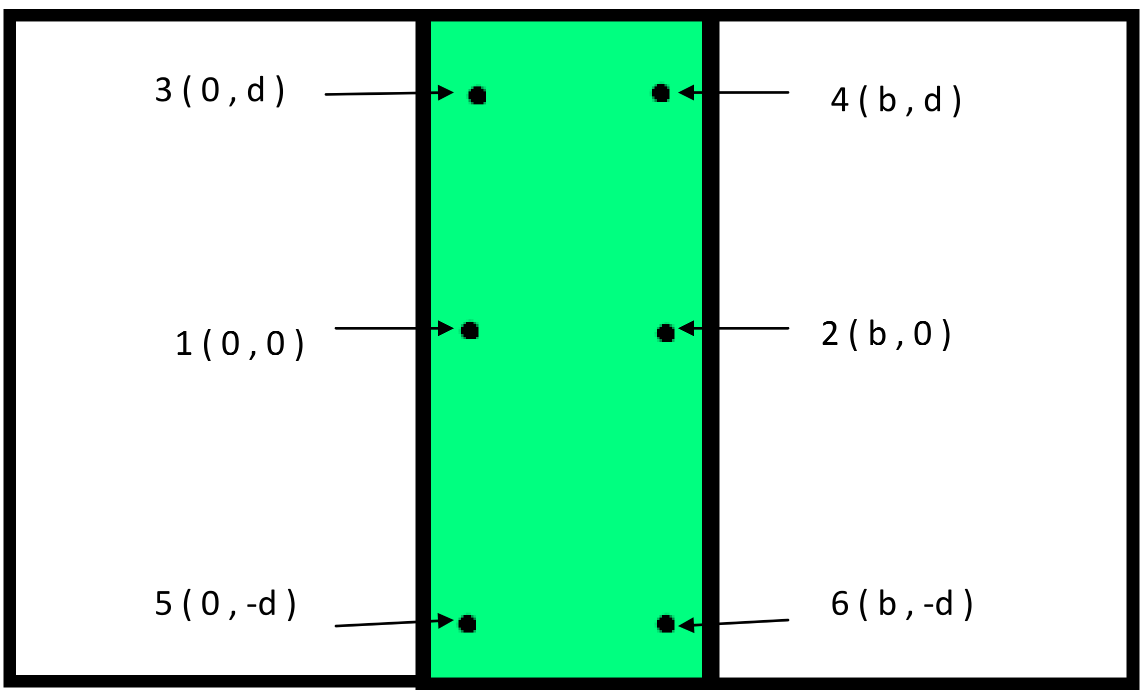
در این روش تنها از پارامترهای دورانی هر دو پروژکتور استفاده میشود و به ان توجیه نسبی دوطرفه می گویند. از این روش عموما در تشکیل مدل مستقل استفاده می شود. همچنین این روش از نظر ساخت پروژکتورها و دستگاه های تبدیل روش بهتری است.

ولی در روش محاسباتی توجیه نسبی از هر 50 حالت می توان بهره گرفت. روش محاسباتی به این صورت است که ابتدا برای 6 نقطه ی استاندارد p_y را اندازه می گیریم و سپس 5 پارامتر انتخابی را از طریق یک دستگاه 6 معادله – 5 مجهول به روش کمترین مربعات محاسبه می کنیم.

مثلا فرض کنید پارامترهای $d\omega$ ، $d\kappa$ ، $db_{y'}$ ، $d\varphi'$ و db_z انتخاب شده اند و پارامترهای دیگر در محاسبات شرکت نداشته و برابر صفر فرض شده اند. برای این حاالت از انتخاب ها معادله ی p_y را می توان به شکل ماتریسی به صورت زیر نوشت:

$$[p_y] = \begin{bmatrix} -1 & \frac{y}{z} & z\left(1+\frac{y^2}{z^2}\right) & \frac{(x-b)y}{z} & x \end{bmatrix} \begin{bmatrix} db_{y'} \\ db_z \\ d\omega \\ d\varphi' \\ d\kappa \end{bmatrix}$$

حال اگر مختصات نقاط استاندارد را به صورت زیر در نظر بگیریم:



دستگاه معادلات ماتریسی (برای 6 نقطه ی استاندارد) به صورت زیر درمی آید:

$$\underbrace{\begin{vmatrix} py1 \\ py2 \\ py3 \\ py4 \\ py5 \\ py6 \end{vmatrix}}_{L} = \underbrace{\begin{vmatrix} -1 & 0 & z & 0 & 0 \\ -1 & 0 & z & 0 & b \\ -1 & d/z & z(1+(d^2)/(z^2)) & -bd/z & 0 \\ -1 & d/z & z(1+(d^2)/(z^2)) & 0 & b \\ -1 & -d/z & z(1+(d^2)/(z^2)) & bd/z & 0 \\ -1 & -d/z & z(1+(d^2)/(z^2)) & 0 & b \end{vmatrix}}_{A} \underbrace{\begin{vmatrix} db_{y'} \\ db_z \\ d\omega \\ d\varphi' \\ d\kappa \\ \; \end{vmatrix}}_{X}$$

و سپس از طریق رابطه ی کمترین مربعات داریم:

$$X = (A^T A)^{-1} A^T L$$

و پنج پارامتر محاسبه خواهند شد.

حال اگر دستگاه مدرج باشد می توان نتایج را اعمال کرده و سریعا توجیه نسبی را انجام داد ولی اگر دستگاه مدرج نباشد به روش تجربی و با اعمال تغییرات جزئی و مشاهده ی نتیجه به جواب می رسیم.

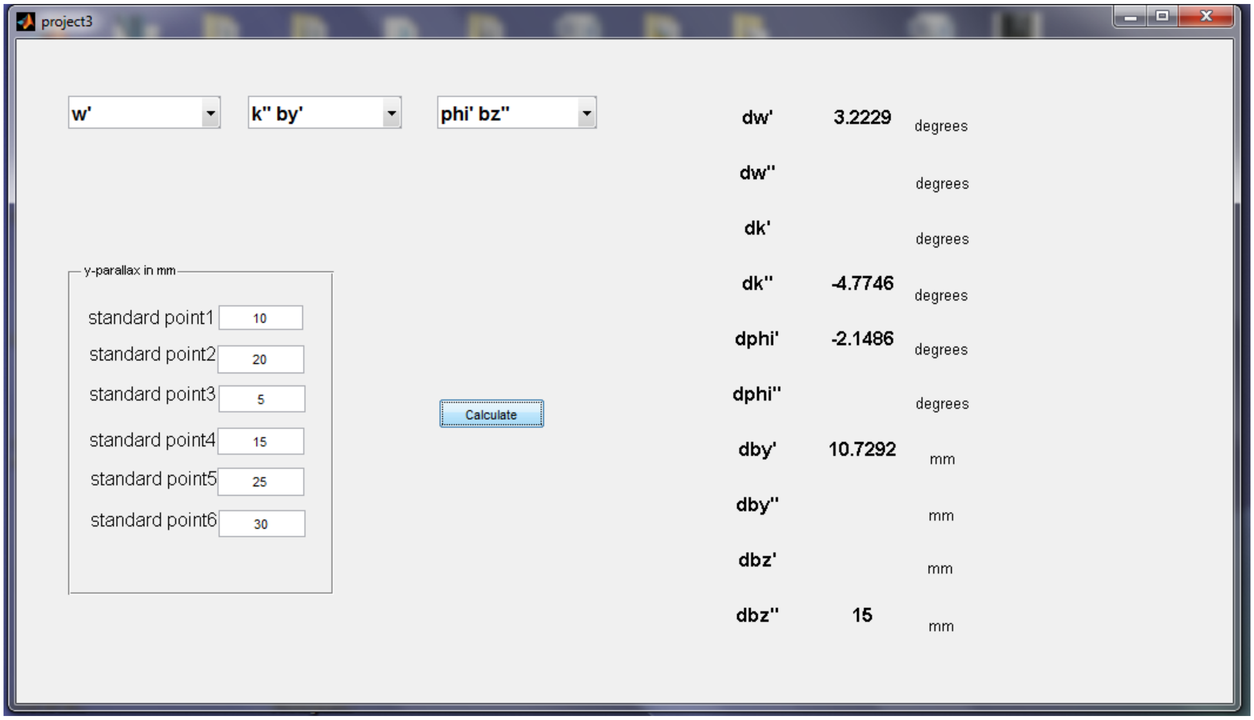
در مورد b و z و d در دستگاه بالا باید گفت در اغلب موارد z=150mm (فاصله ی میز از پروژکتور) است. و در مورد دوربین های wide angle داریم :

$$\frac{b}{z} = \frac{2}{3}$$

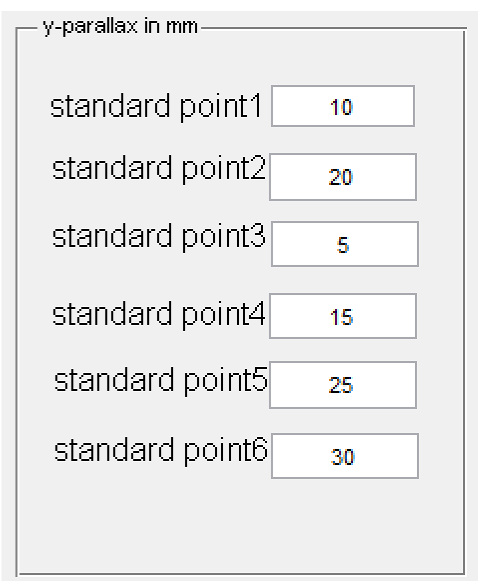
b=100mm و همچنین b=d می باشد.

در پروژه ی 3 از ما خواسته شد که توجیه نسبی محاسباتی را شبیه سازی کنیم به این صورت که برنامه پارالاکس 6 نقطه را بر حسب میلی متر از کاربر گرفته و همچنین کاربر از بین 50 حالت ممکن یک حالت را

انتخاب کرده و سپس برنامه از همان روش محاسباتی که در بالا شرح داده شد استفاده می کند و پارامترهای انتخابی را چاپ می کند. بدین منظور برنامه ای به شکل زیر طراحی شد:

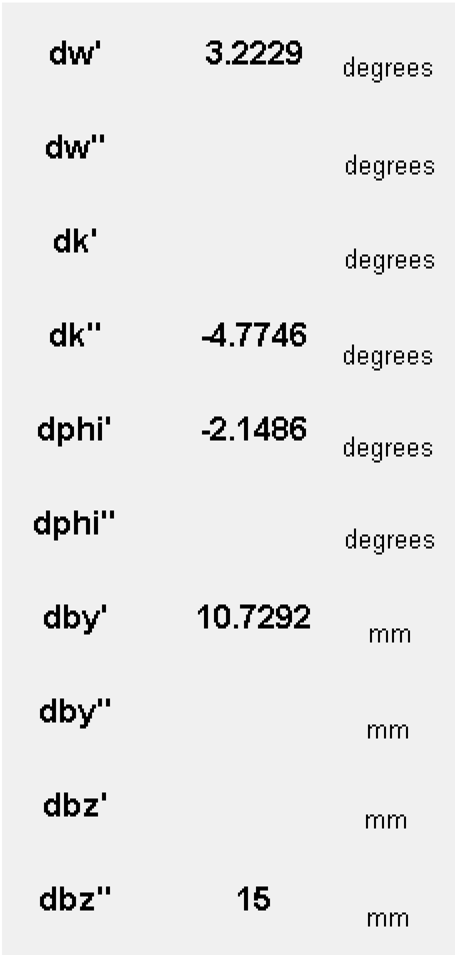


همچنان که دیده می شود کاربر در این برنامه پارالاکس y نقاط استاندارد را بر حسب میلی متر وارد می کند.



نتایج حاصل از برنامه نشان می دهند که :

1 -پروژکتور چپ بایستی حول محور x در جهت عقربه های ساعت به اندازه ی 3.2229 درجه دوران کند.



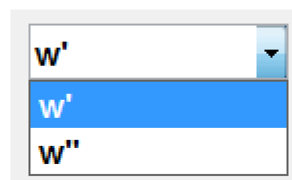
2 -پروژکتور راست بایستی حول محور z در خلاف جهت عقربه های ساعت به اندازه ی 4.7746 درجه دوران کند.

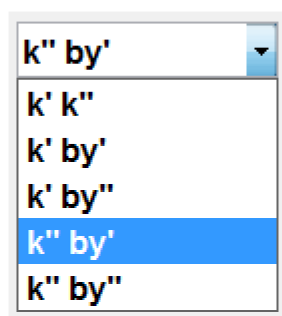
3 -پروژکتور چپ بایستی حول محور y در خلاف جهت عقربه های ساعت به اندازه ی 2.1486 درجه دوران کند.

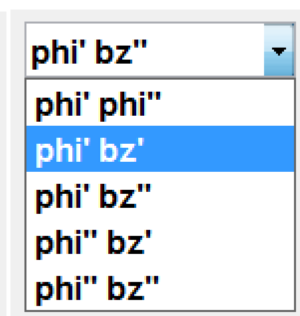
4 -پروژکتور چپ بایستی در راستای محور y به اندازه ی 10.7292mm جابه جا شود.

5 -پروژکتور راست بایستی در راستای محور z به اندازه ی 15mm جابه جا شود.

حال ممکن است سئوالی بدین صورت مطرح شود که چرا popupmenu ها را به صورت زیر در نظر گرفته ایم؟







جواب این است که هنگام صحبت در مورد 50 حالت ممکن دیدیم که :

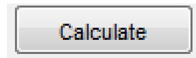
1 همواره $d\omega$ یا $d\omega'$ حضور دارند پس آن دو را در یک popupmenu جدا جا داده ایم.

2 انتخاب db_y و $db_{y'}$ به انتخاب $d\kappa$ و $d\kappa'$ وابسته است پس این چهار ا لمان را در یک popupmen جدا میگذاریم. db_y و $db_{y'}$ نمیتوانند با هم انتخاب شوند و به همین دلیل در popupmenu های بالا گزینه ی (by' by'') دیده نمیشود ولی $d\kappa$ و $d\kappa'$ میتوانند با هم انتخاب شوند ولی در صورت انتخاب آن دو با هم دیگر نمی توانیم db_y یا $db_{y'}$ را انتخاب کنیم.

3 -توضیح در مورد popupmenu سوم همانند بند 2 است.

با اندک توجهی می توان دریافت که popupmenu های بالا همه ی 50 حالت را پوشش می دهند.

50= 2*5*5 اصل ضرب

حال به توضیح callback مربوط 

به می پردازیم:

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

```
set(handles.domegaprim,'string','')
set(handles.domegazegond,'string','')
set(handles.dkapaprim,'string','')
set(handles.dkapazegond,'string','')
set(handles.byprim,'string','')
set(handles.byzegond,'string','')
set(handles.dfiprim,'string','')
set(handles.dfizegond,'string','')
set(handles.bzprim,'string','')
set(handles.bzzegond,'string','')
```

توسط کدهای نوشته شده در box بالا با هر بار فشار calculate، textbox های مربوط به نمایش نتایج را null می کنیم تا اگر کاربر بخواهد بدون باز کردن مجدد برنامه اطلاعات جدید وارد کرده و نتایج جدید بگیرد با مشکلی مواجه نشود.

کد های نوشته شده در box روبرو مقادیر ثابت pi، b، d و z را به وجود می آورند (توجه شود که b، d و z بر حسب متر نوشته شده اند)

```
pi=3.1416;
b=0.1;
d=0.1;
z=0.15;
```

```
if get(handles.popupmenu1,'value')==1
    A(1:6,1)=[z;z;z*(1+(d/z)^2);z*(1+(d/z)^2);z*(1+(d/z)^2);z*(1+(d/z)^2)]
else
    A(1:6,1)=[-z;-z;-z*(1+(d/z)^2);-z*(1+(d/z)^2);-z*(1+(d/z)^2);-
z*(1+(d/z)^2)]
end
```

توسط کدهای نوشته شده در box بالا popupmenu1 را مورد ارزیابی قرار می دهیم. این popupmenu قرار است ستون 1 ماتریس A را تشکیل دهد و از سطر اول تا ششم مختصات نقاط استاندارد 1 تا 6 را در ضریب $d\omega$ یا $d\omega'$ با توجه به انتخاب کاربر جایگذاری کرده ایم.

توسط کدهای نوشته شده در box روبرو popupmenu2 را مورد ارزیابی قرار می دهیم. این popupmenu قرار است ستون های2 و 3 ماتریس A را تشکیل دهد و از سطر اول تا ششم مختصات نقاط استاندارد 1 تا 6 را در ضریب $d\kappa$، $d\kappa'$، db_y یا $db_{y'}$ با توجه به انتخاب کاربر جایگذاری می کنیم.

```
if get(handles.popupmenu2,'value')==1
    A(1:6,2)=[0;b;0;b;0;b]
    A(1:6,3)=[b;0;b;0;b;0]
elseif get(handles.popupmenu2,'value')==2
    A(1:6,2)=[0;b;0;b;0;b]
    A(1:6,3)=[1;1;1;1;1;1]
elseif get(handles.popupmenu2,'value')==3
    A(1:6,2)=[0;b;0;b;0;b];
    A(1:6,3)=[-1;-1;-1;-1;-1;-1]
elseif get(handles.popupmenu2,'value')==4
    A(1:6,2)=[b;0;b;0;b;0]
    A(1:6,3)=[1;1;1;1;1;1]
else
    A(1:6,2)=[b;0;b;0;b;0]
    A(1:6,3)=[-1;-1;-1;-1;-1;-1]
end
```

توسط کدهای نوشته شده در box روبرو popupmenu3 را مورد ارزیابی قرار می دهیم. این popupmenu قرار است ستون های4 و 5 ماتریس A را تشکیل دهد و از سطر اول تا ششم مختصات نقاط استاندارد 1 تا 6 را در ضریب $d\varphi$، $d\varphi'$ db_z یا $db_{z'}$ با توجه به انتخاب کاربر جایگذاری می کنیم.

```
if get(handles.popupmenu2,'value')==1
    A(1:6,2)=[0;b;0;b;0;b]
    A(1:6,3)=[b;0;b;0;b;0]
elseif get(handles.popupmenu2,'value')==2
    A(1:6,2)=[0;b;0;b;0;b]
    A(1:6,3)=[1;1;1;1;1;1]
elseif get(handles.popupmenu2,'value')==3
    A(1:6,2)=[0;b;0;b;0;b];
    A(1:6,3)=[-1;-1;-1;-1;-1;-1]
elseif get(handles.popupmenu2,'value')==4
    A(1:6,2)=[b;0;b;0;b;0]
    A(1:6,3)=[1;1;1;1;1;1]
else
    A(1:6,2)=[b;0;b;0;b;0]
    A(1:6,3)=[-1;-1;-1;-1;-1;-1]
end
```

توسط کدهای نوشته شده در box روبرو ورودی های کاربر را که به عنوان پارالاکس نقاط استاندارد وارد کرده است گرفته به متر تبدیل کرده و در اختیار برنامه می گذاریم. سپس ماتریس مشاهدات (L) را تشکیل می دهیم و از روش کمترین مربعات ماتریس مجهولات محاسبه می شود.

```
py1=str2num(get(handles.py1,'string'))/1000;
py2=str2num(get(handles.py2,'string'))/1000;
py3=str2num(get(handles.py3,'string'))/1000;
py4=str2num(get(handles.py4,'string'))/1000;
py5=str2num(get(handles.py5,'string'))/1000;
py6=str2num(get(handles.py6,'string'))/1000;
L=[py1;py2;py3;py4;py5;py6]
X=inv((A'*A))*A'*L
```

توسط کدهای نوشته شده در box زیر نیز به ترتیب popupmenu های 1 و 2 و 3 بررسی شده و نتایج با توجه به آن به خروجی ارسال می شوند چون هر بار با توجه به انتخاب کاربر از میان 50 حالت ممکن نتایج متفاوتی تولید و به شکل متفاوتی چاپ می شوند.

```
if get(handles.popupmenu1,'value')==1
   set(handles.domegaprim,'string',num2str(X(1,1)*180/pi))
else
   set(handles.domegazegond,'string',num2str(X(1,1)*180/pi))
end

if get(handles.popupmenu2,'value')==1
   set(handles.dkapaprim,'string',num2str(X(2,1)*180/pi))
   set(handles.dkapazegond,'string',num2str(X(3,1)*180/pi))
elseif get(handles.popupmenu2,'value')==2
   set(handles.dkapaprim,'string',num2str(X(2,1)*180/pi))
   set(handles.byprim,'string',num2str(X(3,1)*1000))
elseif get(handles.popupmenu2,'value')==3
   set(handles.dkapaprim,'string',num2str(X(2,1)*180/pi))
   set(handles.byzegond,'string',num2str(X(3,1)*1000))
elseif get(handles.popupmenu2,'value')==4
   set(handles.dkapazegond,'string',num2str(X(2,1)*180/pi))
   set(handles.byprim,'string',num2str(X(3,1)*1000))
else
   set(handles.dkapazegond,'string',num2str(X(2,1)*180/pi))
   set(handles.byzegond,'string',num2str(X(3,1)*1000))
end

if get(handles.popupmenu3,'value')==1
   set(handles.dfiprim,'string',num2str(X(4,1)*180/pi))
   set(handles.dfizegond,'string',num2str(X(5,1)*180/pi))
elseif get(handles.popupmenu3,'value')==2
   set(handles.dfiprim,'string',num2str(X(4,1)*180/pi))
   set(handles.bzprim,'string',num2str(X(5,1)*1000))
elseif get(handles.popupmenu3,'value')==3
   set(handles.dfiprim,'string',num2str(X(4,1)*180/pi))
   set(handles.bzzegond,'string',num2str(X(5,1)*1000))
elseif get(handles.popupmenu3,'value')==4
   set(handles.dfizegond,'string',num2str(X(4,1)*180/pi))
   set(handles.bzprim,'string',num2str(X(5,1)*1000))
else
   set(handles.dfizegond,'string',num2str(X(4,1)*180/pi))
   set(handles.bzzegond,'string',num2str(X(5,1)*1000))
end
```

دقت شود با توجه به اینکه popupmenu1 ستون اول ماتریس A را تشکیل می داد پس نتیجه ی مربوط به آن سطر اول ماتریس X است که با توجه به انتخاب کاربر به عنوان $d\omega'$ یا $d\omega''$ چاپ می شود و در مورد سایر popupmenu ها هم وضع به همین منوال است.

پروژه ی چهار:

پس از اینکه توجیه نسبی به طور کامل انجام شده و مدل سه بعدی تشکیل شد از 12 پارامتر توجیه توانسته-ایم 5 پارامتر را حل کنیم ولی 7 پارامتر باقی مانده هنگام توجیه مطلق حل خواهند شد.

به طور کلی لازم است به مدل تشکیل شده 1 مقیاس ، 3 دوران حول محورهای x ، y و z همچنین 3 جابجایی در راستای محورهای x ، y و z اعمال شود تا به مختصات زمینی سه بعدی دست یابیم. مقیاس λ ، دوران های Ω، Φ و κ همچنین جابجایی های X_T ، Y_T و Z_T پارامترهای توجیه مطلق را تشکیل می دهند. ابتدا این پارامترها را با بررسی ارتباط بین چند نقطه ی کنترل زمینی (GCP) که مختصات آن ها روی زمین و مدل مشخص است بدست می آوریم و سپس همه ی نقاط مدل را با توجه به این پارامترها منتقل می کنیم.

7 پارامتر توجیه مطلق را به پارامترهای مقیاس گذاری (scaling) و ترازیابی (leveling) تقسیم می کنیم.

Scaling: λ κ X_T Y_T

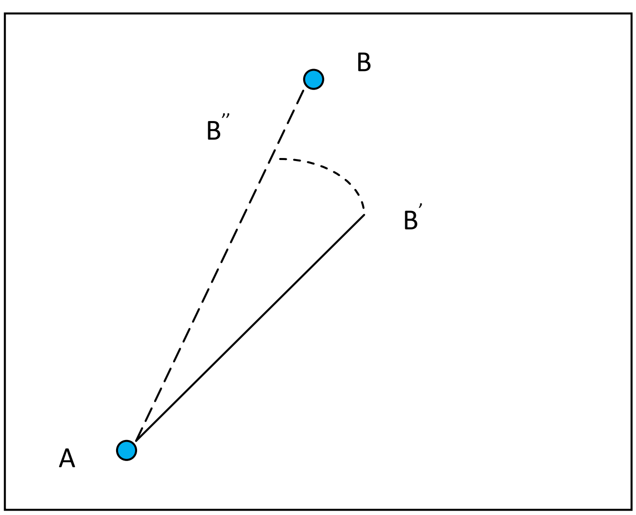
Leveling: Ω Φ Z_T

و بر این اساس حل توجیه مطلق به روش دستگاهی در دو مرحله انجام می گیرد:

جهت scaling ما نیاز به حداقل دو نقطه ی کنترل مسطحاتی داریم (در عمل 3 نقطه) و جهت leveling ما نیاز به حداقل سه نقطه ی کنترل ارتفاعی غیر هم امتداد داریم (در عمل 4-5 نقطه)

مرحله ی scaling :

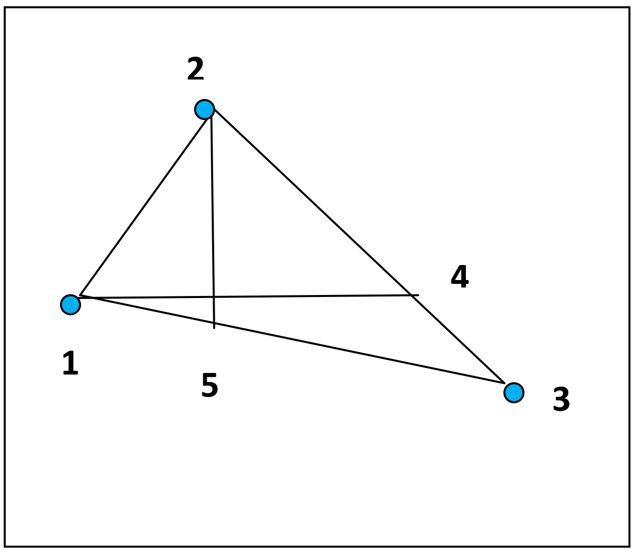
ابتدا شیت نقشه را که حداقل دو نقطه ی کنترل مسطحاتی روی آن مشخص است روی میز ترسیم قرار می دهیم و سپس مداد را روی یکی از نقاط کنترل مثلا نقطه ی A قرار می دهیم (انتقال X_T و Y_T) سپس شیت نقشه را حول نقطه ی A دوران میدهیم تا امتداد حرکت AB در مدل با نقشه یکی شود (دوران κ) سپس باز دستگاه را تغییر میدهیم



تا نقطه ی کنترل B در روی نقشه و مدل انطباق یابد. (مقیاس λ)

با انجام این عمل (تغییر باز دستگاه) اگر باز دارای ا لمان های b_y و b_z باشد b_y و b_z تغییر خواهند کرد. در نتیجه توجیه نسبی دستگاه به هم خواهد ریخت بنابراین دوباره توجیه نسبی را انجام داده و به توجیه مطلق میپردازیم و این دو عمل را آن قدر تکرار می کنیم تا توجیه نسبی و مطلق هر دو درست باشند.

مرحله ی leveling :



ابتدا مقیاس مدل را به نشانگر ارتفاع میز معرفی می کنیم سپس نقطه ی شناور را روی یک نقطه ی کنترل ارتفاعی مثلا نقطه ی 1 قرار داده از آن خطی به موازات یکی از محورهای x یا y دستگاه خارج می کنیم تا ضلع 23 را در نقطه ی 4 قطع کند. سپس نقطه ی شناور را روی نقطه ی 2 قرار داده و آن خطی عمود بر خط قبلی اخراج می کنیم تا ضلع 13 را در نقطه ی 5 قطع کند. حال با توجه به اینکه طول های 23 و 24 و 13 و 15 روی مدل معلوم است و ارتفاع زمینی نقاط کنترل 1 و 2 و 3 مشخص می باشد برای ارتفاع نقاط 4 و 5 داریم:

$$h_4 = h_2 + (h_3 - h_2)\frac{L_{24}}{L_{23}}$$

$$h_5 = h_1 + (h_3 - h_1)\frac{L_{15}}{L_{13}}$$

و در مورد دوران ها خواهیم داشت:

$$\Omega = \tan^{-1}\frac{\mathrm{h}_5 - \mathrm{h}_2}{\mathrm{L}_{25}}$$

$$\Phi = \tan^{-1}\frac{h_4 - h_1}{L_{14}}$$

حال این دوران ها را به دستگاه اعمال می کنیم همچنین از طریق محاسبه ی میانگین Z نقاط Z_T نیز محاسبه شده و به دستگاه اعمال می شود.

از آن جایی که Ω و Φ دوران هایی حول محورهای X و Y می باشند و امتداد AB در مدل یک امتداد مسطحاتی نیست با اعمال دوران های Ω و Φ، scaling به هم می خورد و مراحل بالا را باید با تکرار آن قدر انجام داد تا توجیه نسبی ، scaling و leveling هر سه درست باشد.

در روش محاسباتی توجیه مطلق 7 پارامتر به صورت زیر در محاسبه شرکت می کنند:

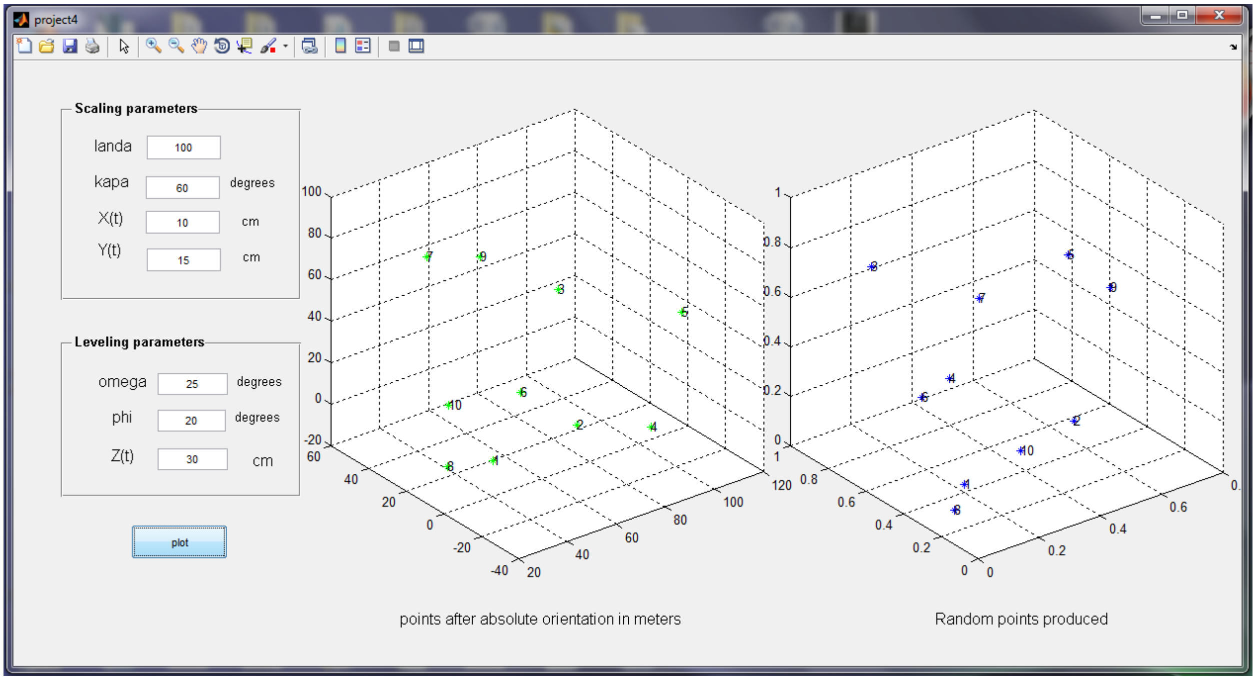
$$\begin{bmatrix} X_{ground} \\ Y_{ground} \\ Z_{ground} \end{bmatrix} = \lambda\, R\, \begin{bmatrix} X_{model} \\ Y_{model} \\ Z_{model} \end{bmatrix} + \begin{bmatrix} X_T \\ Y_T \\ Z_T \end{bmatrix}$$

که در آن R ماتریس دوران است و از ضرب ماتریسی زیر بدست می آید:

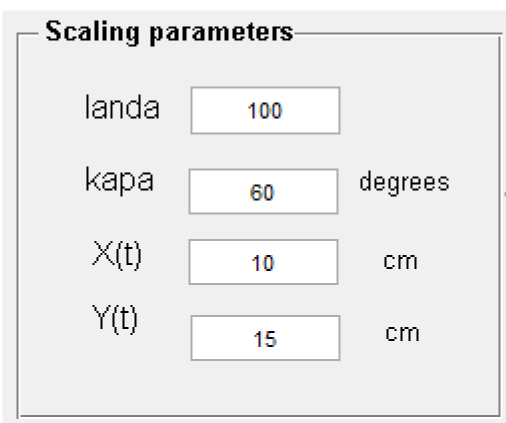
$$R = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Omega & sin\Omega \\ 0 & -\sin\Omega & \cos\Omega \end{bmatrix}$$

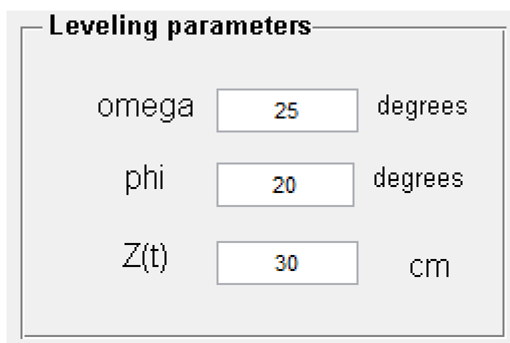
در پروژه ی 4 از ما خواسته شد که برنامه ای بنویسیم که ابتدا 10 نقطه ی تصادفی تولید و آن ها را رسم کند سپس پارامترهای scaling و leveling را از کاربر گرفته به نقاط مورد نظر اعمال کرده و نقاط بدست آمده را نیز رسم کند.

بدین منظور برنامه ای به صورت زیر طراحی گردید:

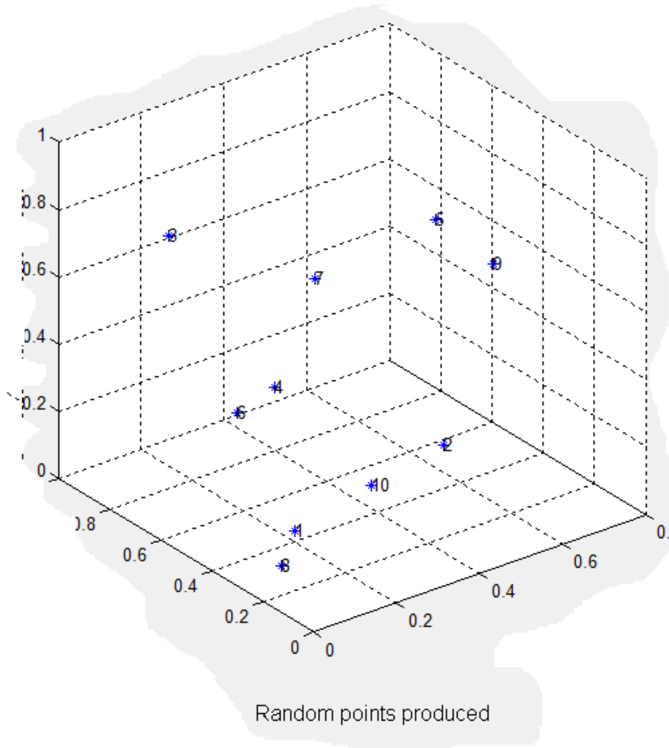


واحدهایی که کاربر پارامترهای توجیه را برحسب آنان وارد میکند در شکل های زیر دیده می شوند:





در ابتدای اجرای برنامه نقاط random تولید شده و در نمودار مربوط نمایش داده می شوند:



بدین منظور کدهای زیر در project4_OpeningFcn اضافه شده اند تا همزمان با باز شدن برنامه نقاط رندوم نیز تولید و نمایش داده شوند.

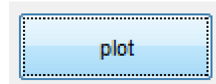
```
set(hObject,'toolbar','figure')
axes(handles.axes1)
for i=1:10
    handles.x(i)=rand;
    handles.y(i)=rand;
    handles.z(i)=rand;
plot3(handles.x(i),handles.y(i),handles.z(i),'*b');
hold on
grid on
text(handles.x(i),handles.y(i),handles.z(i),num2str(i))
end
```

نوار ابزار مربوط به figure را به GUI اضافه می کند.

چون در GUI دو نمودار داریم این کد axes مورد نظر را که میخواهیم ترسیمات در آن انجام شود فعال می کند.

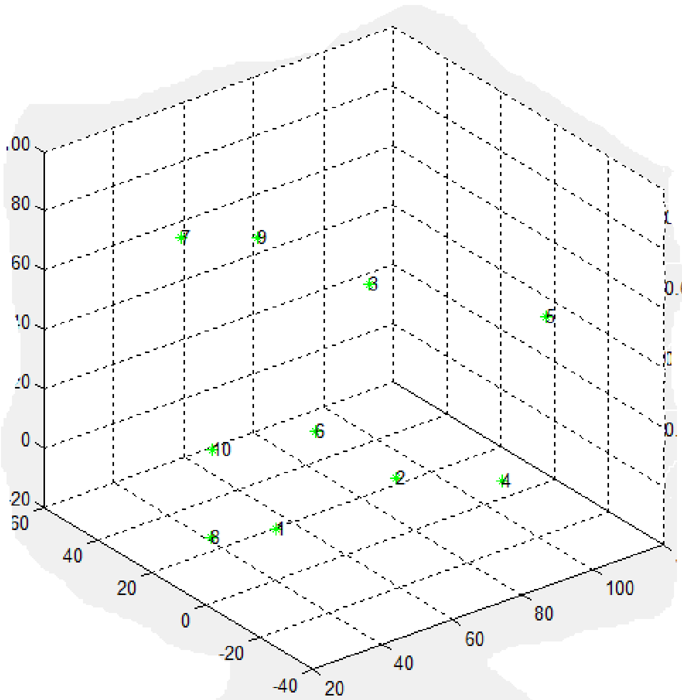
این سه کد نقاط تصادفی را تولید و در ماتریس های x و y و z ذخیره می کنند و چون می خاهیم از این ماتریس ها در function های دیگر GUI نیز استفاده کنیم آن ها را از نوع handles تعریف کرده ایم.

در کنار هر نقطه شماره ی آن را نیز نمایش می دهد.

حال به توضیح 

می پردازیم با فشار plot نمودار زیر رسم می شود:

که نقاط متناظر با نقاط موجود در نمودار قبل را در یک سیستم مختصات جدید نمایش می دهد.



points after absolute orientation in meters

در callback مربوط به این pushbutton داریم:

```
axes(handles.axes2)
pi=3.141592654;
```

Axes2 را فعال کرده و ثابت pi را تعریف می کنیم.

```
landa=str2num(get(handles.landa,'string'));
kapa=str2num(get(handles.kapa,'string'))*pi/180;
xt=str2num(get(handles.xt,'string'))/100;
yt=str2num(get(handles.yt,'string'))/100;
omega=str2num(get(handles.omega,'string'))*pi/180;
phi=str2num(get(handles.phi,'string'))*pi/180;
zt=str2num(get(handles.zt,'string'))/100;
```

توسط کدهای نوشته شده در box بالا ورودی های کاربر را در اختیار برنامه می گذاریم. دقت شود کاربر زوایا را بر حسب درجه وارد کرده که آن ها را به رادیان تبدیل می کنیم. مقادیر انتقالی را که بر حسب سانتی متر وارد شده به متر تبدیل می کنیم و مقیاس نیز واحدی ندارد.

```
omega_matrix=[1 0 0;0 cos(omega) sin(omega);0 -sin(omega) cos(omega)];
phi_matrix=[cos(phi) 0 -sin(phi);0 1 0;sin(phi) 0 cos(phi)];
kapa_matrix=[cos(kapa) sin(kapa) 0;-sin(kapa) cos(kapa) 0;0 0 1];
t_matrix=[xt;yt;zt];
R=kapa_matrix*phi_matrix*omega_matrix;
```

ماتریس های دوران حول محورهای x و y و z را تشکیل داده آنان را در هم ضرب می کنیم تا ماتریس دوران کلی R بدست آید.

```
for i=1:10
    X=[handles.x(i);handles.y(i);handles.z(i)];
    temp1=R*X;
    temp2=landa*temp1;
    temp3=temp2+t_matrix;
    plot3(temp3(1,1),temp3(2,1),temp3(3,1),'*g')
    hold on
    grid on
    text(temp3(1,1),temp3(2,1),temp3(3,1),num2str(i))
end
```

هر بار با یک سطر از ماتریس های x ، y و z که در OpeningFcn تشکیل شدند یی از نقاط random را در نظر گرفته و معادله ی توجیه مطلق را روی آن اعمال می کنیم و سپس آن را رسم می کنیم.