طراحی و ساخت روبات پردازشگر تصویر در MATLAB7

امین جوادی نسب

a_javadi62@yahoo.com

***امیر حسین رحمانی –شکرالله گله داری – مهنوش واهبی**

ah_rahmani@yahoo.com

دانشکده فنی مهند سی دانشگاه آزاد اسلامی واحد دزفول

*عضو هیت علمی دانشگاه آزاد اسلامی واحد دزفول(دانشجو دوره دکترا کنترل علوم و تحقیقات تهران)

چکیده:

در این مقاله روباتی طراحی شده است که با استفاده از دوربینی که بر روی آن قرار گرفته است محیط اطراف را دیده وموتورهای خود راکنترل کند . برای این منظور از علم پردازش تصویر که در برنامه **MATLAB7** می باشد استفاده کرده ایم . برای این روبات هیچ چشم یا سنسوری طراحی نشده است و روبات فقط با استفاده از **web cam** کوچکی که در بالای آن قرار گرفته است از محیط اطراف خود فیلم گرفته و با استفاده از **MATLAB** فیلم گرفته شده را پردازش می کند و سپس از عکس های گرفته شده سیگنال کنترلی را به دست آورده و این سیگنالها را برای کنترل موتورها استفاده می کند . هدف از این کار معرفی علم پردازش تصویر در دنیای کنونی است.

کلمات کلیدی:

پردازش تصویر – فیلتر –MATLAB7 – روبات – WEB CAM-USB—PORT

1– مقدمه :

پردازش تصویر دانش جدیدی است که سابقه آن به پس از اختراع رایانه های رقمی باز می گردد با این حال این علم نوپا در چند دهه اخیر از هردو جنبه نظری و علمی پیشرفتهای چشمگیری داشته است.

سرعت این پیشرفت به اندازه ای بوده است که هم اکنون و پس از این مدت نسبتا کوتاه به راحتی می توان رد پای پردازش تصویر را در بسیاری از علوم مشاهده کرد .

هدف ما از نوشتن این مقاله نشان دادن کاربرد کوچکی از علم پردازش تصویر است که می تواند در کنترل روباتهای هوشمند مورد استفاده قرار گیرد.

بدون شک یکی از مهمترین مشخصات هوشمندی یک روبات واحد بینایی آن است که امروزه در روباتهای به جای استفاده از سنسورها در واحد بینایی از نرم افزارهایی همچون تشخیص الگو (پردازش تصویر) استفاده می نماید . واز طریق این علم اطلاعات دنیای خارج در اختیار روبات قرار گرفته و از طریق دسته بندی این اطلاعات دنیای خارج در اختیار روبات قرار گرفته واز طریق دسته بندی این اطلاعات روبات هدایت می شود.

2 – ساختار کلی روبات :

روباتی که طراحی شده است شامل سه قسمت زیر است که ما دراین پروژه بحث اصلی روی قسمت نرم افزاری که همان پردازش تصویر و **MATLAB** است می باشد و قسمت های دیگر به طور خلاصه گفته می شود.

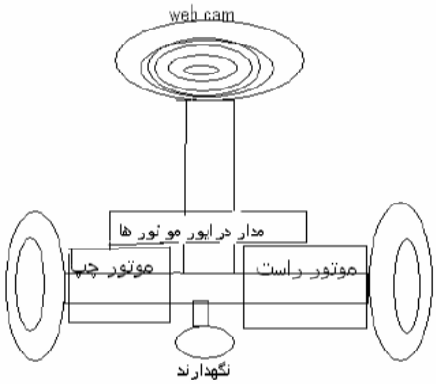
در یک دید کلی روبات شامل سه قسمت زیر است.

1. قسمت مکانیکی
2. قسمت الکترونیکی
3. قسمت نرم افزاری

2-1- قسمت مکانیکی:

قسمت مکانیکی این روبات بسیار ساده بوده ولی دارای ویژگیهای بسیار خوبی است که می تواند به راحتی حرکت کرده و در هر جهتی ازادانه حرکت کند. این طرح که به طرح تانک معروف است فقط دارای دو جرخ است که یکی در طرف راست و یکی در طرف چپ قرار دارد که با استفاده از یک دیفرانسیل ساده نیروی موتورها به چرخها انتقال می یابد و با استفاده از حرکت موتورها به جلو و عقب روبات به راحتی می تواند در هر جهتی که خواست حرکت کند. همانطور که گفته شد این روبات هیچ چشم یا سنسوری ندارد .و برای واحد بینایی آن از یک **web cam** کوچک استفاده شده است که در بالای آن قرار گرفته است و می تواند صفحه جلوی روبات را ببیند. و تصویر را به کامپیوتر منتقل کند.

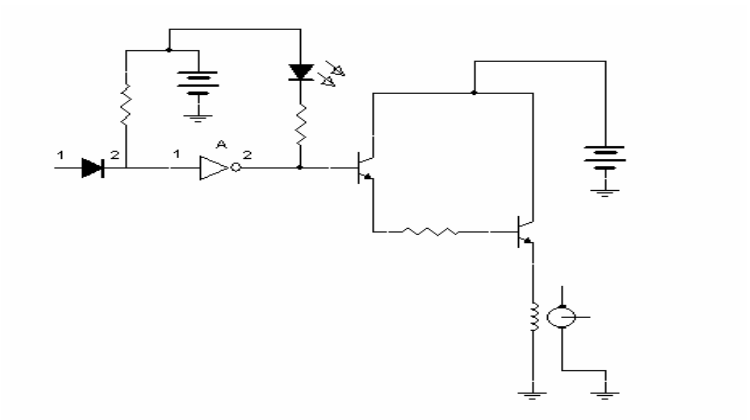
ساختار کلی قسمت مکانیکی در شکل زیر امده است.



2-2 – قسمت الکترونیکی

به علت اینکه این روبات فاقد هر چشمی است لذا مدار الکترونیکی آن بسیار کوچک است و فقط درایور کوچکی برای موتورها طراحی شده است که در بالای موتورها قرار دارد که وظیفه کنترل روبات رابر عهده دارد.

این درایور با استفاده از دو سیگنال کنترلی که از پورت پرینتر کامپیوتروارد آن می شود رله های مدار را قطع و وصل کرده و باعث می شود موتور در جهتهای مختلف بچرخد.



2-3 – قسمت کامپیوتری

نرم افزاری که برای اینکار استفاده شده است برنامه **MATLAB7** است که دارای ویژگی های جدیدی نسبت به نسخه های قبلی آن است و ما از ویژگی پردازش تصویر آن استفاده کرده ایم . برای این کار لازم است که ما تصویر گرفته شده را وارد کامپیوتر کنیم که این کار با استفاده از یک **web cam** کوچکی که در بالای روبات قرار دارد صورت می گیرد .

تصویر گرفته شده به صورت زنده از طریق پورت **USB** کامپیوتر وارد آن می شود و چون **MATLAB** می تواند **USB** را شناسایی کند می تواند تصویر گرفته شده را پردازش کند و در نهایت سیگنال کنترلی را برای کنترل روبات بوسیله پورت پرینتر فرستاده و این کار به صورت مداوم و با یک سرعت ثابت انجام می شود.

در یک دید کلی می توان کار روبات را اینگونه توصیف کرد که ابتد ا روبات با استفاده از دوربین تصویر جلوی خود را می بیند و این تصویر با استفاده پورت **USB** به کامپیوتر انتقال داده می شود. و سپس تصویر گرفته شده در کامپیوتر به عکس تبد یل شده ودر وقع کار اصلی نرم افزار تجزیه و تحلیل عکس هایی است که به صورت مداوم و تقریبا با سرعت **7** عکس در ثانیه گرفته می شود و سپس در کامپیوتر باید چشمهای فرضی برای آن طراحی کرد . یعنی چشمهای روبات را در نقاط مختلف صفحه در نظر گرفت و از آ نقاط بتوان سیگنالهای **0** یا **1** را بد ست آورد. البته باید توجه کنیم که تصویر گرفته شده یک تصویر واقعی است و ما باید این تصویر را با استفاده از فیلتر به یک تصویرسیاه و سفید تبدیل کرد . بعد از اینکار می توان مثلا رنگ سیاه را **0** و رنگ سفید را **1** در نظر گرفت.

در واقع کار اصلی این روبات دنبال کردن یک خط در صفحه می باشد که رنگ و اندازه آن متغیر می باشد است. بعد از اینکه فیلم گرفته شده به عکس تبد یل شد چشمهای فرضی را برای آن در نظر گرفت و سپس این سیگنالها را با استفاده از روابطی که رای آن در نظر می گیریم به دو سیگنال کنترلی تبدیل می کنیم و این دو سیگنال را به روبات منتقل می کنیم .

3- Image Processing Toolbox

Image Processing Toolbox از امکانات جنبی نرم افزار **matlab** است . برای اینکه مطمئن شوید که **Toolbox** روی **matlab** نصب است دستور **ver** را اجرا کنید .

این یک دستور لیست ا ست و هر چه که از **matlab** روی کامپیوتر نصب است را ارایه می دهد .

4- ایجاد یک تصویر در MATLAB

Matlab می‌تواند فایل‌های گرافیکی با فرمت‌های JPEG, TIFF, GIF, BMP, PNG, HDF, PCX, **XWD, ICO, CUR** را به عنوان فایل گرافیکی بخواند. مثلاً برای وارد کردن تصویری به نام**robot.tif** به فضای **Matlab** کافی است از دستور **imread** استفاده کنیم:

```
MyImage=imread('robot.tif','tif');
```

توجه داشته باشید که فایلی که دستور خواندنش را می‌دهید باید برای برنامه قابل دسترس باشد. یعنی یا باید در مسیر **Matlab (Path)** باشد یا اینکه در پرونده‌ای **(folder)** قرار داشته باشد که در حال حاضر برنامه به آن دسترسی دارد. برای اینکه بدانید که **Matlab** برای پیدا کردن فایلی که دستورش را دادید کجا را خواهد گشت اینکارها را بکنید: از دستور **path** برای اینکه بدانید کدام پرونده‌ها جزء مسیر پیش‌فرض **Matlab** است و از دستور **dir** برای اینکه بدانید که **Current Directory** چیست؛ استفاده کنید.

خب تا اینجا یک فایل تصویر را در محیط **Matlab** وارد کرده‌ایم. همانطور که می‌دانیم یک تصویر دیجیتال بر روی کامپیوتر در قالب یک ماتریس ذخیره می‌شود. پس **MyImage** مثل همه متغیرهای **Matlab** یک ماتریس است. برای اینکه بدانیم فایل خوانده شده از چه فرمتی است(سیاه سفید، یا **Gray Scale** یا رنگی) می‌نویسیم:

```
imfinfo('robot.tif')
```

این دستور را اجرا کنید و ببینید چه می‌نویسد... اما اگر بخواهید بدانید که ماتریس ذخیره شدهٔ **MyImage** از چه نوعی است کافی است بنویسد: **whos** و لیست متغییرهای مقیم شده در حافظه و نوع و اندازه آنها را ببینید.

برای اینکه تصویر ذخیره شده را به صورت ماتریس را مشاهده کنیم باید بنویسم

imshow(MyImage)

در **matlab** می توان دو تصویر را با هم مشاهده کرد . برای اینکار می توان از برنامه زیر استفاده کرد

```
YourImage=imread('tire.tif','tif');
figure
subplot(1,2,1), imshow(MyImage), title('MyImage')
subplot(1,2,2), imshow(YourImage), title('YourImage')
```

تصویر خوانده شده کمی نویز دارد یعنی جای **0** و **1** عوض شده است برای از بردن این نویز از یک فیلتر استفاده می کنیم .

5- هیستوگرام تصویر

تا به حال تصویر های **Gray scale** را بررسی کردیم . یعنی اگر تصویر ورودی رنگی بود اول آن را به سیاه و سفید تبدیل می کنیم و بعد کار را ادامه می دهیم

6- برنامه روبات

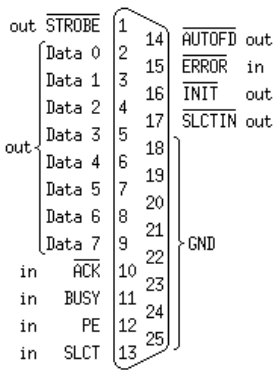
برنامه ای که برای این روبات نوشته شده است به صورت **M file** بوده که با استفاده از **MATLAB7** نوشته شده است

```
vid = videoinput('winvideo' , 1, 'RGB24_160x120');
preview(vid);
set(vid.source, 'Brightness',40);
for i=1:25
I = getsnapshot(vid);
I = rgb2ind(I,colorcube(300));
PSF = fspecial( 'gaussian',3,3);
I = imfilter(I,PSF,'symmetric' ,'conv');
BWs = edge(I, 'sobel', (graythresh(I) * .1));
se90 = strel('line',3,90);
se0 = strel('line',3,0);
BWsdil = imdilate(BWs, [se90 se0]);
w = imfill(BWsdil, 'holes');
w = ~w;
imshow(w);
a=w(11:30,73:88);b=w(31:50,57:72);c=w(51:70,41:56);d=w(71:90,25:40);e=w(91:110,9:24);
B=w(31:50,89:104);C=w(51:70,105:120);D=w(71:90,121:136);E=w(91:110,137:152);
fa=mean(mean(a));fb=mean(mean(b));fc=mean(mean(c));fd=mean(mean(d));fe=mean(mean(e));
fB=mean(mean(B));fC=mean(mean(C));fD=mean(mean(D));fE=mean(mean(E));
if fa>.3
fa=1;
else
fa=0;
end;
if fb>.3
fb=1;
else
fb=0;
end;
if fc>.3
fc=1;
else
fc=0;
end;
if fd>.3
fd=1;
else
fd=0;
end;
```

```
if fe>.3
fe=1;
else
fe=0;
end;
if fB>.3
fB=1;
else
fB=0;
end;
if fC>.3
fC=1;
else
fC=0;
end;
if fD>.3
fD=1;
else
fD=0;
end;
if fE>.3
fE=1;
else
fE=0;
end;
left=fa|fb|fc|fd|fe;
right=fa|fB|fC|fD|fE;
data=[left right]
dio=digitalio('parallel' ,'lpt1');
lines=addline(dio,2:3,0, 'out');
putvalue(dio,data);
delete(dio)
end
delete(vid)
```

این برنامه شامل چند قسمت است :

- شناساندن تصویر به **MATLAB**
- تبدیل تصویر رنگی به سیاه و سفید
- تبدیل فیلم به عکس
- تبدیل عکس به ماتریس
- به دست آوردن سیگنالهای **0** و **1**
- استفاده از روابطی و تبدیل آنها به دو سیگنال کنترلی
- ارسال سیگنالها به پورت



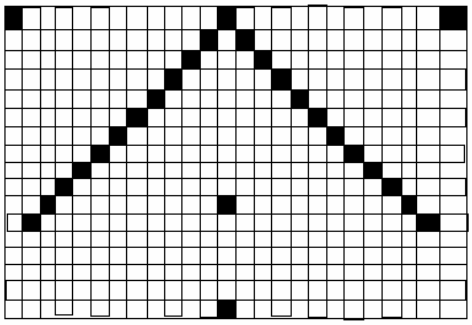
برای اجرای برنامه کافی است که آن را اجرا کنیم برای اینکار می توان آن را **RUN** یا کلید **F5** را فشار داد پس از اجرای برنامه دو تصویر گرافیکی دیده می شود که یکی از آنها تصویر واقعی از محیط اطراف و دیگری تصویری است که به صورت سیاه یا سفید است.

این یک برنامه **ON line** است یعنی به صورت مداوم کار می کند برای اینکار باید تصویر گرفته شده را به عکس تبدیل کرد و از هر عکس سیگنال کنترلی بدست آید البته هر چه سرعت عکس گرفتن بیشتر باشد

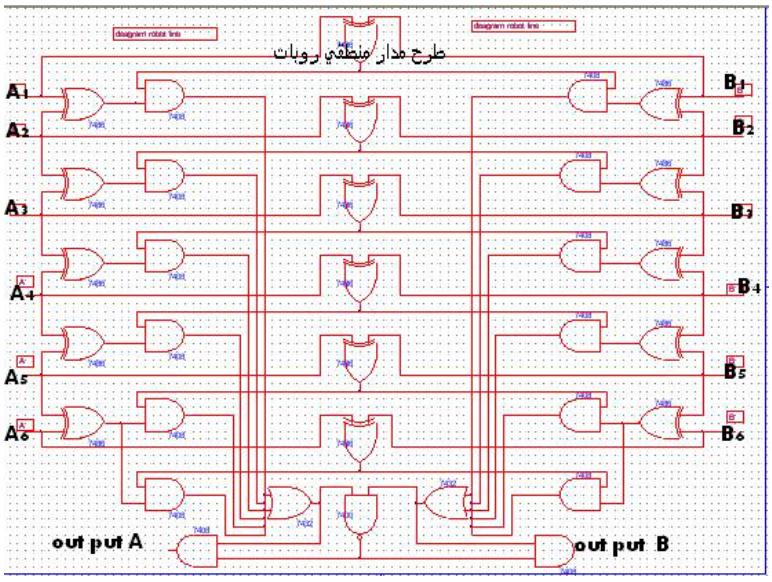
سرعت روبات هم بیشتر خواهد بود برای اینکه بتوان چشمهای فرضی روبات را به دست آورد باید عکس را به ماتریس تبدیل کرد در این پروژه برای اینکه احتمال درست بودن سیگنال بیشتر باشد ما هر سه نقطه را یک چشم در نظر گرفته ایم و میانگین این سه نقطه را در نظر گرفته ایم .

مثلا جایی که **2** نقطه سفید و یک نقطه سیاه است آن را سفید یا همان **1** در نظر گرفته ایم

طرز قرار گرفتن چشمهای فرضی در کامپیوتر مانند شکل زیر است



در این پروژه ما **6** چشم برای طرف راست و **6** چشم برای طرف چپ در نظر گرفته ایم . در شکل زیر طرح مدار منطقی را می بینید که می تواند هر نوع خط با هر رنگ و هر اندازه را دنبال کند



7-نتیجه گیری

در این مقاله روباتی طراحی شده است که می تواند قابلیتهای بیشتری دا شته باشد و با استفاده از علم پردازش تصویر می توان دنیای کنونی را راحتتر برای روباتها تشریح کرد و با استفاده از نرم افزارهای جدید می توان برنامه نویسی را راحتتر کرد . و پیشنهادی که در این مقاله داده می شود این است که بهتر است به جای استفاده از سنسور در اینگونه روباتها از دوربین استفاده کرد که امکانات بیشتری دارند.

8-سپاسگزاری

در آخر جا دارد از تمام اساتید و دانشجویان زیر که مرا در انجام این پروژه کمک کرده اند تشکر کنم

- آقای دکتر محسن رحمانی
- آقای مهندس محمد تقی اکبر زاده
- آقای مهندس داریوش زین العابدینی
- آقای دکتر حسن براتی

9- مراجع

1. **Gonzalez,Rafael C** (پردازش تصویر رقمی / تالیف رافائل سی. گونزالس / ترجمه مرتضی خادمی و داوود جعفری._ مشهد: دانشگاه فردوسی مشهد **1382**
2. خود آموز**6 MATLAB** سال انتشار **1380**
3. مهندسی کنترل/ کاتسو هیکو اوگاتا /ترجمه محمود دیانی
4. **HELP MATLAB7**
5. میکرو الکترونیک /عادل صدرا