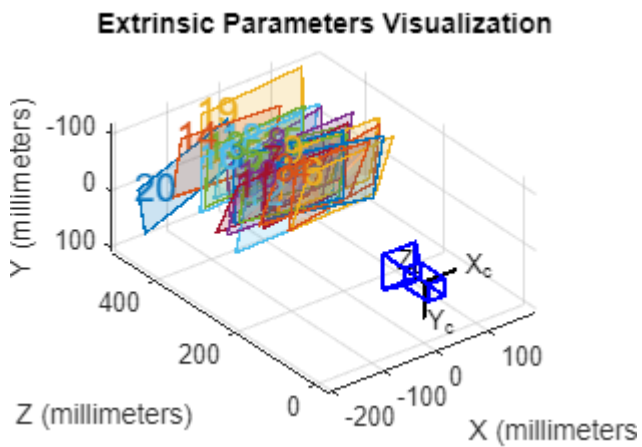
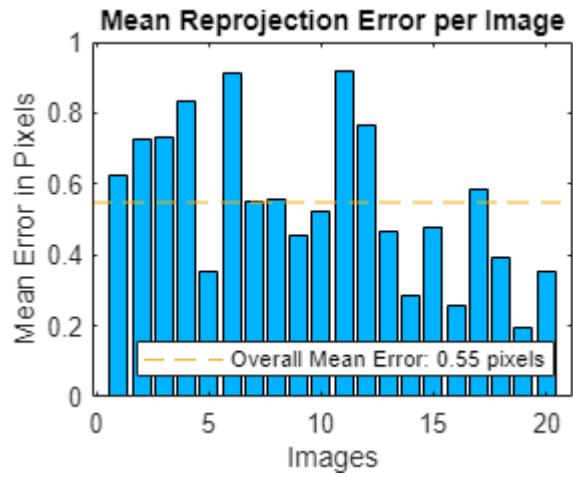


```
calibrationParams;
```



Standard Errors of Estimated Camera Parameters

Intrinsics

```
-----  
Focal length (pixels): [ 447.9294 +/- 6.5245    449.6906 +/- 6.4970 ]  
Principal point (pixels): [ 317.0547 +/- 1.3225    192.9172 +/- 2.4815 ]  
Radial distortion: [ 0.0215 +/- 0.0100    -0.4400 +/- 0.0387 ]
```

Extrinsics

Rotation vectors:

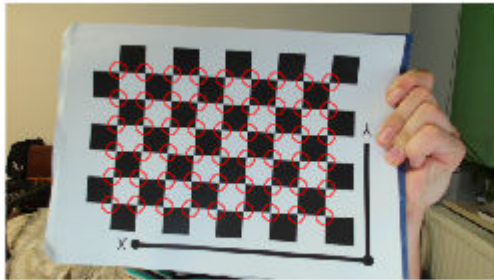
```
[ -0.0408 +/- 0.0064    0.2080 +/- 0.0049    0.0103 +/- 0.0011 ]  
[ 0.1371 +/- 0.0074    0.1460 +/- 0.0057    0.0734 +/- 0.0010 ]  
[ 0.3090 +/- 0.0062   -0.0483 +/- 0.0041    0.1086 +/- 0.0008 ]  
[ 0.1738 +/- 0.0055   -0.1452 +/- 0.0047   -0.0305 +/- 0.0010 ]  
[ 0.0468 +/- 0.0071    0.0588 +/- 0.0049    0.1030 +/- 0.0008 ]  
[ 0.2752 +/- 0.0058   -0.1273 +/- 0.0049    0.0700 +/- 0.0011 ]  
[ 0.0161 +/- 0.0056   -0.1749 +/- 0.0049    0.0462 +/- 0.0010 ]  
[ 0.0800 +/- 0.0058   -0.1727 +/- 0.0049    0.1192 +/- 0.0011 ]  
[ 0.1479 +/- 0.0063   -0.0504 +/- 0.0050    0.0442 +/- 0.0009 ]  
[ 0.2349 +/- 0.0066   -0.0204 +/- 0.0045    0.0447 +/- 0.0008 ]  
[ 0.4447 +/- 0.0062   -0.0449 +/- 0.0039    0.0310 +/- 0.0010 ]  
[ 0.2257 +/- 0.0060   -0.0485 +/- 0.0044    0.0126 +/- 0.0011 ]  
[ 0.1515 +/- 0.0101    0.0019 +/- 0.0068   -0.0643 +/- 0.0010 ]  
[ 0.1883 +/- 0.0090    0.0455 +/- 0.0065    0.0105 +/- 0.0011 ]  
[ 0.2541 +/- 0.0068   -0.0676 +/- 0.0053    0.1873 +/- 0.0010 ]  
[ 0.2022 +/- 0.0087    0.0557 +/- 0.0062   -0.1178 +/- 0.0010 ]
```

[0.2153 +/- 0.0076	-0.0217 +/- 0.0061	-0.2203 +/- 0.0011
[0.2159 +/- 0.0090	-0.0240 +/- 0.0067	0.0144 +/- 0.0010
[0.2096 +/- 0.0124	0.0694 +/- 0.0093	-0.1599 +/- 0.0013
[0.2721 +/- 0.0091	-0.0605 +/- 0.0070	-0.3431 +/- 0.0013

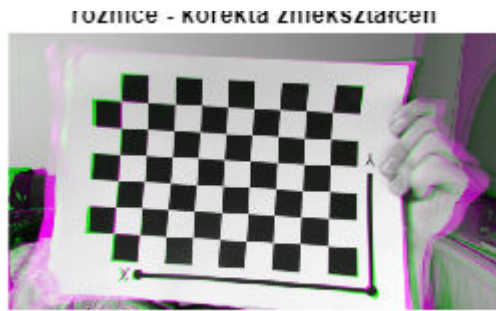
Translation vectors (millimeters):

[-81.4522 +/- 1.0560	-79.7337 +/- 2.0032	370.0562 +/- 5.0883
[-21.1303 +/- 0.9996	-73.2852 +/- 1.8626	336.0385 +/- 4.7466
[-49.2942 +/- 0.8953	-77.5639 +/- 1.6757	297.0599 +/- 4.3528
[-11.6173 +/- 0.9172	-57.1659 +/- 1.6875	305.8991 +/- 4.4424
[-76.5266 +/- 1.0042	-90.8180 +/- 1.9011	341.7324 +/- 4.9714
[11.1153 +/- 0.9158	-46.9980 +/- 1.6716	304.0233 +/- 4.3931
[-46.9304 +/- 0.9963	-74.2207 +/- 1.8370	332.2349 +/- 4.7901
[-30.7693 +/- 1.0237	-94.4822 +/- 1.9039	340.8544 +/- 4.9796
[-8.7552 +/- 0.9897	-79.1276 +/- 1.8351	330.4443 +/- 4.7758
[-95.1533 +/- 1.0081	-63.7775 +/- 1.8860	334.9009 +/- 4.9335
[-112.6491 +/- 0.9558	-63.1283 +/- 1.7876	317.6185 +/- 4.8213
[-59.3737 +/- 1.0879	-5.4723 +/- 2.0184	365.4259 +/- 5.3675
[-108.5963 +/- 1.1939	-90.1107 +/- 2.2477	398.3159 +/- 5.8568
[-148.2700 +/- 1.2321	-114.5244 +/- 2.3394	413.7216 +/- 6.0094
[-21.5939 +/- 1.1119	-77.5613 +/- 2.0821	370.2403 +/- 5.4507
[-117.7017 +/- 1.1870	-74.9516 +/- 2.2572	401.6004 +/- 5.8823
[-61.6750 +/- 1.2290	-36.6586 +/- 2.2945	412.0391 +/- 6.0856
[-67.0852 +/- 1.2322	-84.1221 +/- 2.3163	410.7124 +/- 6.0699
[-52.1968 +/- 1.4544	-88.5897 +/- 2.7490	488.9553 +/- 7.1690
[-209.4567 +/- 1.3442	-33.0120 +/- 2.4752	441.8242 +/- 6.5337

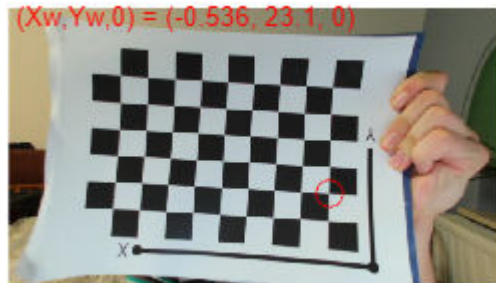
```
figure;
imshow(IMG);
hold on
plot(imagePoints(:,1),imagePoints(:,2),'ro')
hold off
```



```
figure;
imshowpair(IMG, undistortedImage, 'falsecolor');
title('różnice - korekta zniekształceń')
```

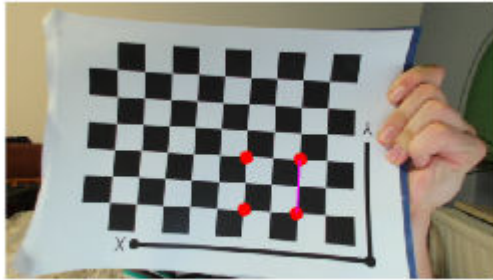


```
figure;
imshow(undistortedImage)
hold on
plot(myPoint1(1),myPoint1(2),'ro','MarkerSize',10)
hold off
text(10,10, ...
    ['(Xw,Yw,0) = (' num2str(myWorldPoint1(1),3) ' , ' num2str(myWorldPoint1(2),3)
    ', 0)'],...
    'Color','red')
```



```
figure;
imshow(undistortedImage);
hold on
plot(projectedPoints1(:,1), projectedPoints1(:,2),'r.','MarkerSize',15);
plot([projectedPoints1(1,1) projectedPoints1(2,1)],...
    [projectedPoints1(1,2) projectedPoints1(2,2)],'m-')
hold off
title('punkty 3D rzutowane na obraz')
```

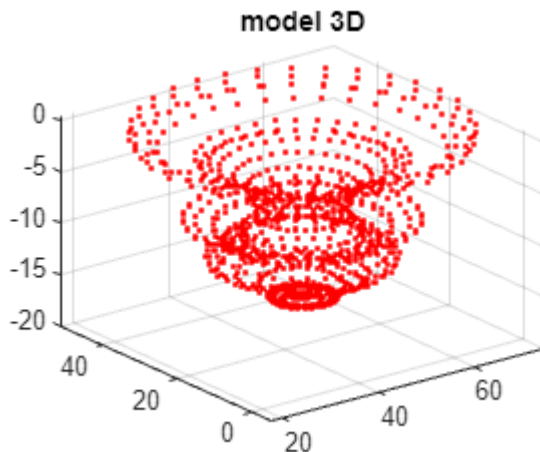
punkty 3D rzutowane na obraz



```
myWorldPoints.Z = [1 2 2 1];  
pts1 = [myWorldPoints.X(:) myWorldPoints.Y(:) myWorldPoints.Z(:)]; %  
(X, Y, Z - wektory kolumnowe)  
projectedPoints1 = worldToImage(cameraParams, R, t, pts1);
```

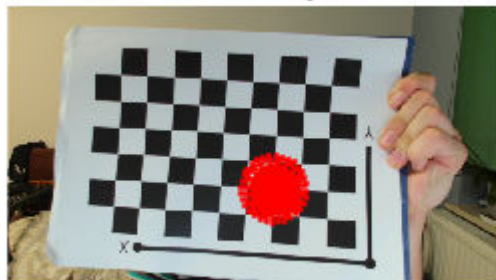
```
r1 = 0.5 + sin(t1);
```

```
figure;  
plot3(X,Y,Z,'r.')  
hold off  
grid on  
title('model 3D')
```



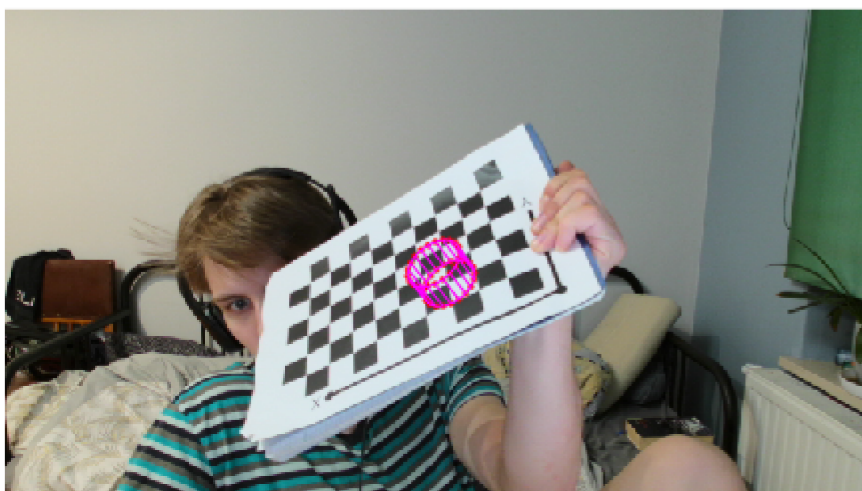
```
figure;  
imshow(IMG);  
hold on  
plot(projectedPoints1(:,1), projectedPoints1(:,2),'r.');  
hold off  
title('model 3D rzutowany na obraz')
```

model 3D rzutowany na obraz



```
figure;  
imshow('img1.png');
```

Ramka nr 56

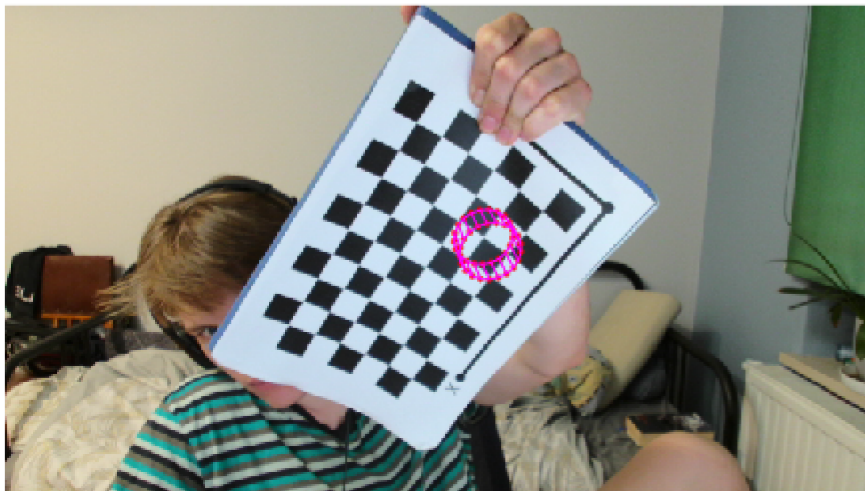


stop

continue

```
figure;  
imshow('img2.png');
```

Ramka nr 97

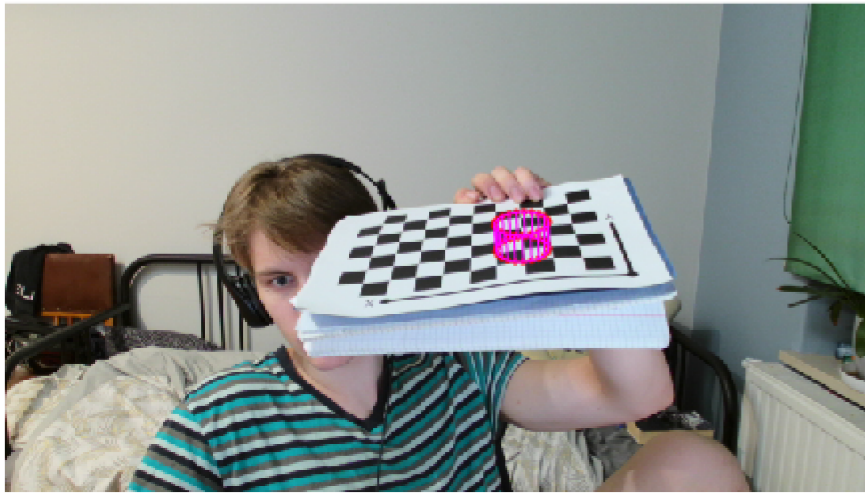


stop

continue

```
figure;  
imshow('img3.png');
```

Ramka nr 140



stop

continue

Nakładanie działa bardzo dobrze, przestaje funkcjonować dopiero przy prawie płaskim ułożeniu planszy względem kamery

Podsumowanie

Rezultaty: wyżej

Analiza:

1. Wyżej
2. zewnętrzne: lokalizacja kamery w scenie 3D: rotacja i przesunięcie odpowiadają za przejście parametrów 3D ze świata w parametry 3D kamery, wewnętrzne: centrum optyczne i ogniskowa odpowiadają za transformację z koordynatów 3D kamery w obraz 2D

Pytania

1. Pozbywanie się zaburzeń soczewki - usprawnianie i polepszanie jakości zdjęć, pomiary rzeczywistych obiektów - nie trzeba do tego wykorzystywać ludzi, odtwarzanie obiektów 3D - modelowanie np. do gier czy symulacji.

2. umieszczanie cylindra działało zaskakująco dobrze, sposób w jaki za pomocą zestawu parametrów przeliczamy obraz po koeli w 3 sferach i na ich podstawie możemy zrozumieć głębie wczytywanego obrazu
3. Jak interpretować wykres Extrinsic parameters visualization