In [14]:
```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
from scipy import optimize
from scipy.optimize import curve_fit
import scienceplots
```

In [15]:
```python
plt.style.use(['science', 'notebook', 'grid'])
```

O4.e Nicolò De Masi

Task 1

In [16]:
```python
#input constant values

A_g=30 #lamp position [mm]
d_g=40 #correction value lamp position [mm]
d_b=30 #correction value eyepiece position [mm]
d_l=235 #distance between lenses [mm]
G = 7 #size of our object [mm]

# input measurements and adjust them

A_I = np.array([215, 220, 225, 230, 235, 240, 245, 250, 255, 260]) #system of lenses positions [mm]

A_I = A_I- A_g  #adjusted for lamp position

g_p = A_I-d_g #adjusted for correction value of lamp position -> g prime = distance pointer-object

A_b = np.array([1112, 973, 864, 776, 691, 633.5, 583.5, 536.5, 497, 459]) #eyepiece positions [mm]

A_b = A_b-A_g#adjusted for lamp position

A_b = A_b+d_b #adjusted for correction value of eyepiece position

b_p = A_b-g_p #b prime = distance pointer image

print(g_p)

print(b_p)

B = np.array([6.63, 6.11, 5.22, 4.91, 4.34, 4.14, 3.87, 3.64, 3.36, 2.62]) #size of the image [ticks for each mi

B= 7 *B #size of the image [in mm]

gamma=B/G #magnification, gamma

#we want a linear fit, as such we plot g_p against (1+1/gamma), and b_p against (1+gamma)


gamma1=1+1/gamma
gamma2=1+gamma

print(gamma)
print(gamma2)
```

```
[145 150 155 160 165 170 175 180 185 190]
[967.  823.  709.  616.  526.  463.5 408.5 356.5 312.  269. ]
[6.63 6.11 5.22 4.91 4.34 4.14 3.87 3.64 3.36 2.62]
[7.63 7.11 6.22 5.91 5.34 5.14 4.87 4.64 4.36 3.62]
```

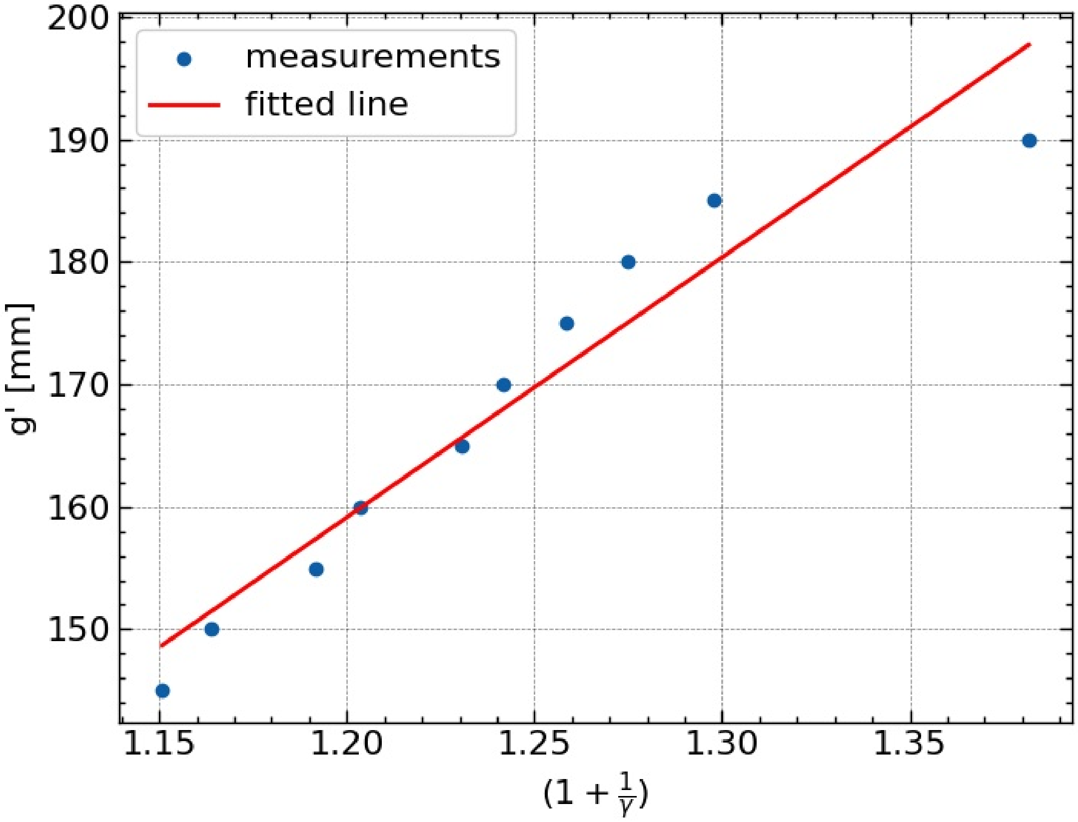
In [22]:
```python
#g_prime = g+h = f(1+1/gamma)+h

res = stats.linregress(gamma1,g_p)

plt.scatter(gamma1,g_p, label="measurements",)
plt.plot(gamma1, res.intercept + res.slope*gamma1, 'r', label='fitted line') #y=x*m+n
plt.xlabel(r"$(1+\frac{1}{\gamma})$")
plt.ylabel("g' [mm]")
plt.legend()
plt.show()

print(f"The focal lenght of the system will then be at: {res.slope:.6f} +/- {res.stderr:.6f} mm")

print(f"The first principal plane of the system will then be at: {res.intercept:.6f} +/- {res.intercept_stderr:
```




```
The focal lenght of the system will then be at: 212.282564 +/- 20.909411 mm
The first principal plane of the system will then be at: -95.605448 +/- 25.951166 mm
```

In [26]:

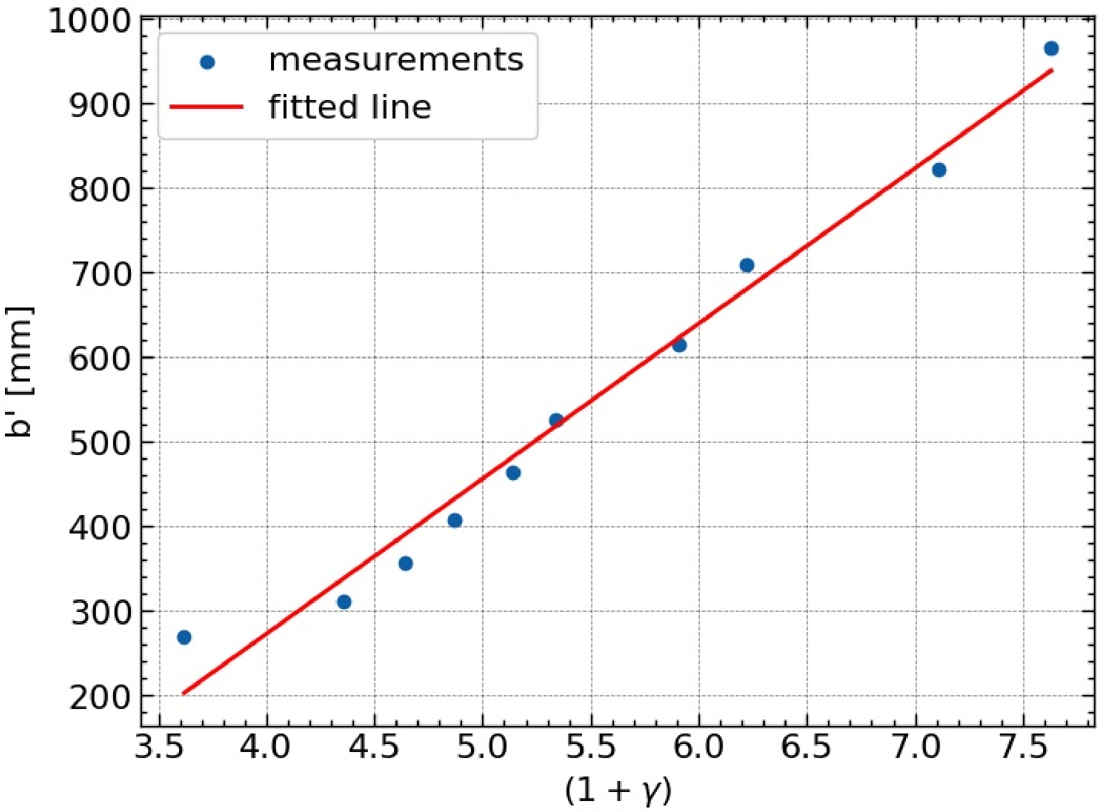
```python
#b_prime = b+h = f(1+gamma)+h_prime

res2 = stats.linregress(gamma2,b_p)

plt.scatter(gamma2,b_p, label="measurements")
plt.plot(gamma2, res2.intercept + res2.slope*gamma2, 'r', label='fitted line') #y=x*m+n
plt.xlabel(r"$(1+\gamma)$")
plt.ylabel("b' [mm]")
plt.legend()
plt.show()

print(f"The focal lenght of the system will then be at: {res2.slope:.6f} +/- {res2.stderr:.6f} mm")

print(f"The first principal plane of the system will then be at: {res2.intercept:.6f} +/- {res2.intercept_stder
```




```
The focal lenght of the system will then be at: 183.563318 +/- 9.116899 mm
The first principal plane of the system will then be at: -461.611233 +/- 51.142572 mm
```

Task 2

In [28]:

```
# we use bessels eq to find focal lenght for both individual lenses:

# f = (a^2-e^2)/4a a being the distance between lamp and screen, and e being the distance between the two point:

#lens1

a1= np.array([500, 750, 1000]) #distance between first lens and screen [mm]

P1_1 = np.array([152,144,142]) #first point of sharp image [mm]
P2_1 = np.array([405.5,665.5,910]) #second point of sharp image [mm]

a1=a1-30-40 #a1 adjusted by lamp position and correction value of lamp position [mm]

e1= P2_1-P1_1 #distance between lenses [mm]

f1 = (a1**2-e1**2)/(4*a1)

print(f1)

#lens 2

a2= np.array([1250, 1500, 1750]) #distance between first lens and screen [mm]

P1_2 = np.array([382,358,339]) #first point of sharp image [mm]
P2_2 = np.array([912,1209,1461.5]) #second point of sharp image [mm]

a1=a1-30-40 #a1 adjusted by lamp position and correction value of lamp position [mm]

e2= P2_2-P1_2 #distance between lenses [mm]

f2 = (a2**2-e2**2)/(4*a2)

print(f2)
```

```
[70.13822674 70.01387868 73.94516129]
[256.32       254.29983333 257.49910714]
```

In [29]:

```
# use system of equations formula to find the focus of the system:

#1/f_s = 1/f1 + 1/f2 + d/f1f2 where d is the distance between lenses -> d_l=235 [mm]

f_s = (1/f1 + 1/f2 - d_l/(f1*f2))**(-1)

# f_s = (f1*f2)/(f1+f2+d_l)

f1=sum(f1) / len(f1)
print(f1)

f2=sum(f2) / len(f2)
print(f2)

f_s=sum(f_s) / len(f_s)
print(f_s)
```

```
71.3657555703264
256.0396468253968
197.7816643681085
```