In [46]:
```
from astropy.io import fits
from PIL import Image as PILImage
import pylab as pl
from astropy.modeling.polynomial import Polynomial1D
from astropy.modeling.models import Gaussian1D, Linear1D
from astropy.modeling.fitting import LinearLSQFitter
from IPython.display import Image
from astroquery.nist import Nist
import numpy as np
from astropy import units as u
pl.rcParams['image.origin'] = 'lower'
pl.matplotlib.style.use('dark_background')
```

In [47]:
```
hg_filename = "\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 2\\BestGroup_Aug
hy_filename = "\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 2\\BestGroup_Aug
he_filename = "\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 2\\BestGroup_Aug
ne_filename = "\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 2\\BestGroup_Aug
sun_filename = "\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 2\\BestGroup_Au
hy1_filename = "\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 2\\BestGroup_Au
hy2_filename = "\\Users\\Sydnee O'Donnell\\OneDrive\\UF\\Obs Tech 2\\BestGroup_Au
```

In [48]:
```
ne_image = fits.getdata(ne_filename)
he_image = fits.getdata(he_filename)
h_image = fits.getdata(hy_filename)
lights_image = fits.getdata(hg_filename)
```

In [49]:
```
ne_spectrum = ne_image[350:450,:].mean(axis=0)
he_spectrum = he_image[350:450,:].mean(axis=0)
h_spectrum = h_image[350:450,:].mean(axis=0)
lights_spectrum = lights_image[350:450,:].mean(axis=0)
```

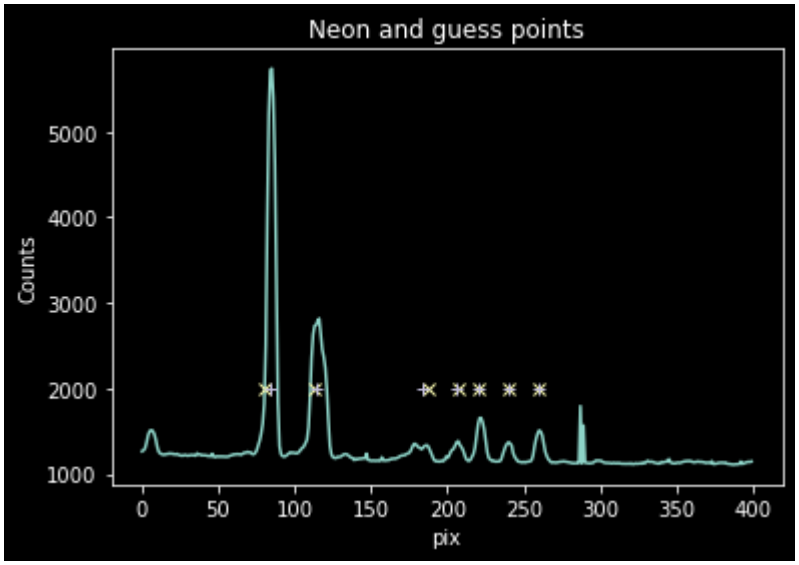
In [50]:
```
xaxis = np.arange(he_image.shape[1])
```

In [51]:
```
guessed_wavelengths_ne = [540, 535, 520, 518, 510, 507, 504]
guessed_xvals_ne = [80, 113, 187, 207, 220, 240, 260]

npixels = 10
improved_xval_guesses_ne = [np.average(xaxis[g-npixels:g+npixels],
                                       weights=ne_spectrum[g-npixels:g+npixels] - np
                            for g in guessed_xvals_ne]
improved_xval_guesses_ne
```

Out[51]:
```
[83.99439688492227,
 114.74226128407567,
 184.077164601371,
 206.42057442506083,
 220.5523215145684,
 239.62171566961243,
 260.0977554664349]
```

In [52]:
```
pl.plot(xaxis[0:400], ne_spectrum[0:400])
pl.plot(guessed_xvals_ne[0:400], [2000]*7, 'x')
pl.plot(improved_xval_guesses_ne[0:400], [2000]*7, '+');
pl.ylabel("Counts")
pl.xlabel("pix")
pl.title("Neon and guess points")
```

Out[52]:
```
Text(0.5, 1.0, 'Neon and guess points')
```

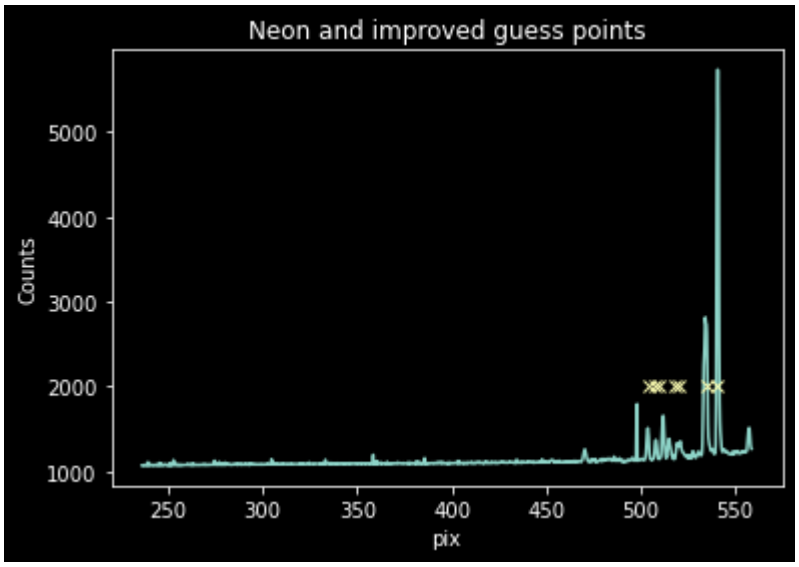


In [53]:
```
linfitter = LinearLSQFitter()
wlmodel = Linear1D()
linfit_wlmodel = linfitter(model=wlmodel, x=improved_xval_guesses_ne, y=guessed_v
wavelengths = linfit_wlmodel(xaxis) * u.nm
linfit_wlmodel
```

Out[53]:
```
<Linear1D(slope=-0.2110209, intercept=558.61902513)>
```

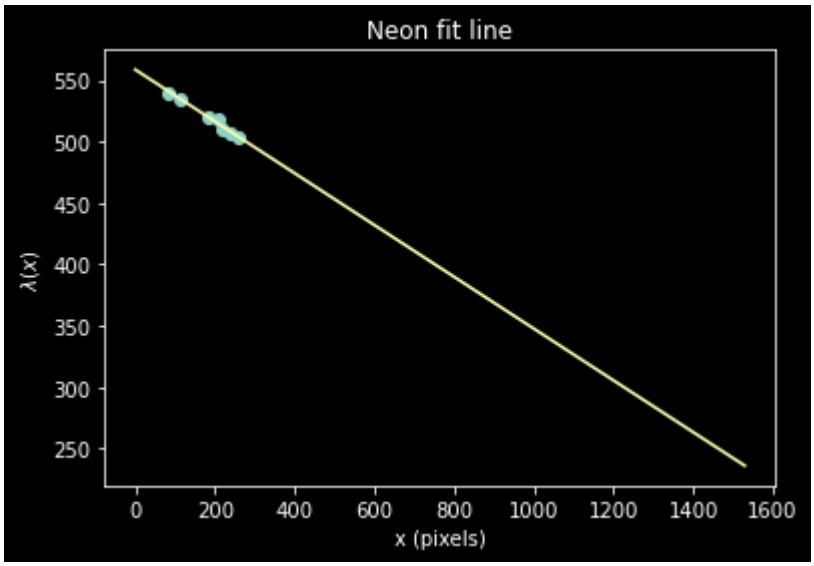
In [54]:
```
pl.plot(wavelengths, ne_spectrum)
pl.plot(guessed_wavelengths_ne, [2000]*7, 'x');
pl.ylabel("Counts")
pl.xlabel("pix")
pl.title("Neon and improved guess points")
```

Out[54]:
```
Text(0.5, 1.0, 'Neon and improved guess points')
```



In [55]:
```
pl.plot(improved_xval_guesses_ne, guessed_wavelengths_ne, 'o')
pl.plot(xaxis, wavelengths, '-')
pl.ylabel("$\lambda(x)$")
pl.xlabel("x (pixels)")
pl.title("Neon fit line")
```

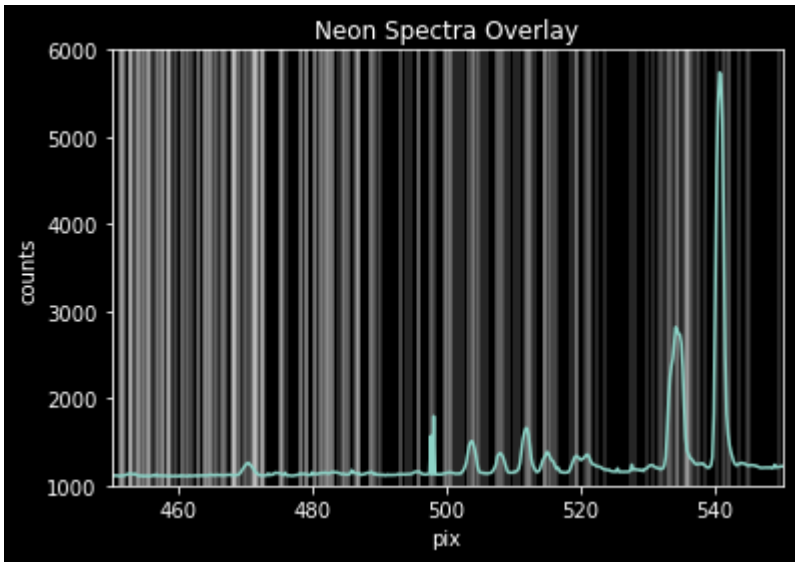
Out[55]:
```
Text(0.5, 1.0, 'Neon fit line')
```



In [15]:
```
minwave = wavelengths.min()
maxwave = wavelengths.max()

hydrogen_lines = Nist.query(minwav=minwave,
                            maxwav=maxwave,
                            linename='H I')
helium_lines = Nist.query(minwav=minwave,
                            maxwav=maxwave,
                            linename='He I')
neon_lines = Nist.query(minwav=minwave,
                        maxwav=maxwave,
                        linename='Ne I')
```

In [16]:
```
pl.plot(wavelengths, ne_spectrum)
pl.vlines(neon_lines['Observed'], 6000, 250, 'w', alpha=0.20);
pl.axis([450, 550, 1000, 6000])
pl.xlabel("pix")
pl.ylabel('counts')
pl.title('Neon Spectra Overlay')
```

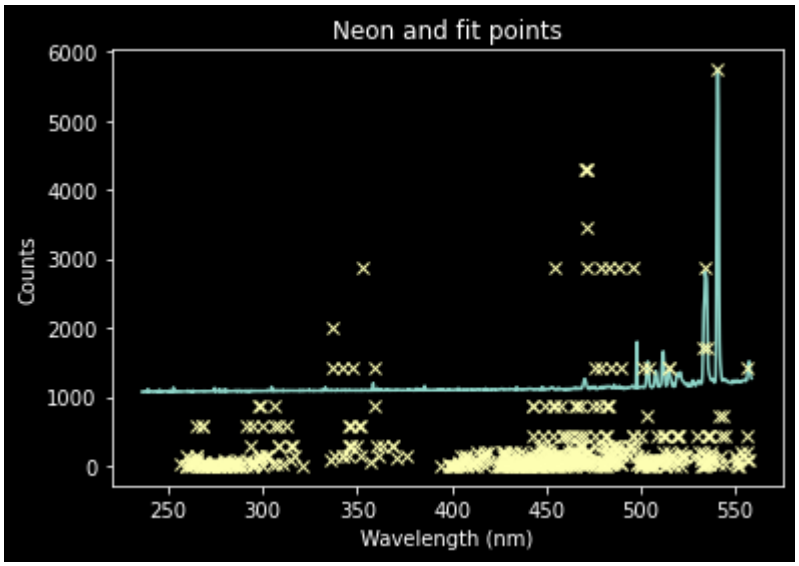
Out[16]:
```
Text(0.5, 1.0, 'Neon Spectra Overlay')
```



In [17]:
```
ne_keep = np.array([('*' not in x) and ('f' not in x) for x in neon_lines['Rel.']
ne_wl_tbl = neon_lines['Observed'][ne_keep]
ne_rel_tbl = np.array([float(x) for x in neon_lines['Rel.'][ne_keep]])
```

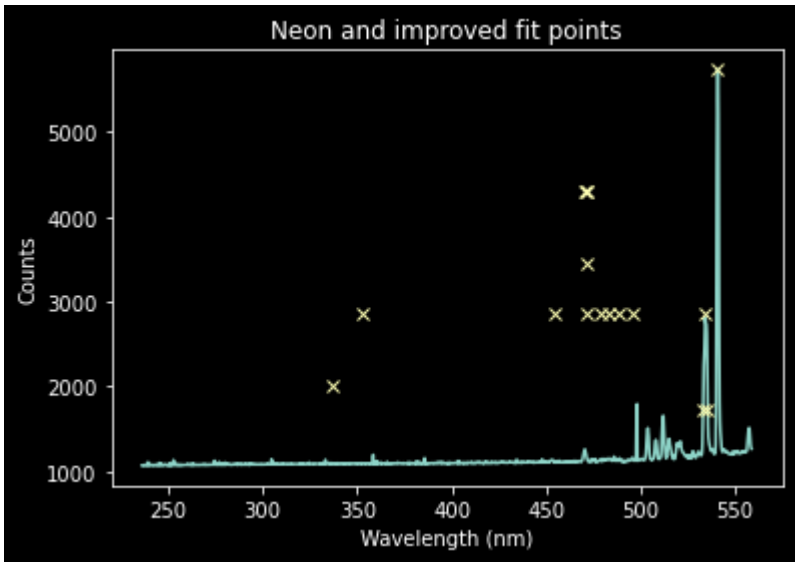
In [56]:
```
ne_rel_intens = ne_rel_tbl / ne_rel_tbl.max() * ne_spectrum.max()
pl.plot(wavelengths, ne_spectrum)
pl.plot(ne_wl_tbl, ne_rel_intens, 'x')
pl.xlabel('Wavelength (nm)');
pl.ylabel("Counts")
pl.title("Neon and fit points")
```

Out[56]:
```
Text(0.5, 1.0, 'Neon and fit points')
```



In [57]:
```
ne_keep_final = ne_rel_intens > 1500
pl.plot(wavelengths, ne_spectrum)
pl.plot(ne_wl_tbl[ne_keep_final], ne_rel_intens[ne_keep_final], 'x')
pl.xlabel('Wavelength (nm)');
pl.ylabel("Counts")
pl.title("Neon and improved fit points")
```

Out[57]:
```
Text(0.5, 1.0, 'Neon and improved fit points')
```



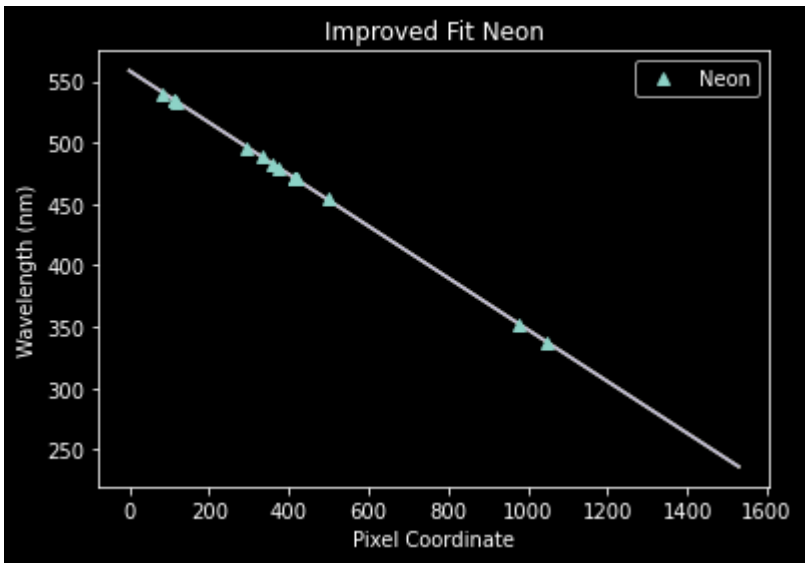
In [58]:
```
ne_wl_final = ne_wl_tbl[ne_keep_final]
ne_pixel_vals = linfit_wlmodel.inverse(ne_wl_final)
```

In [59]:
```
npixels = 10
improved_xval_guesses_ne = [np.average(xaxis[g-npixels:g+npixels],
                                       weights=ne_spectrum[g-npixels:g+npixels] - np
                            for g in map(int, ne_pixel_vals)]
improved_xval_guesses_ne
```

Out[59]:
```
[1048.1054780089544,
 978.4126258005489,
 497.89341692789964,
 417.7665161828031,
 417.2371734517502,
 416.7041664617049,
 416.2423245772903,
 415.56976338276144,
 375.51753347761394,
 358.1301508804359,
 331.41186991104627,
 291.56066428326767,
 116.06114000964114,
 115.11796428586625,
 114.9452765013603,
 84.65107849301077]
```

In [60]:
```
pl.plot(improved_xval_guesses_ne, ne_wl_final, '^', label='Neon')
#pl.plot(improved_xval_guesses, guessed_wavelengths, '+', label='Hydrogen')
pl.plot(xaxis, wavelengths, zorder=-5)
pl.plot(xaxis, linfit_wlmodel(xaxis), zorder=-5)
pl.legend(loc='best')
pl.xlabel("Pixel Coordinate")
pl.ylabel("Wavelength (nm)")
pl.title("Improved Fit Neon")
```

Out[60]: Text(0.5, 1.0, 'Improved Fit Neon')



In [ ]:

In [ ]:

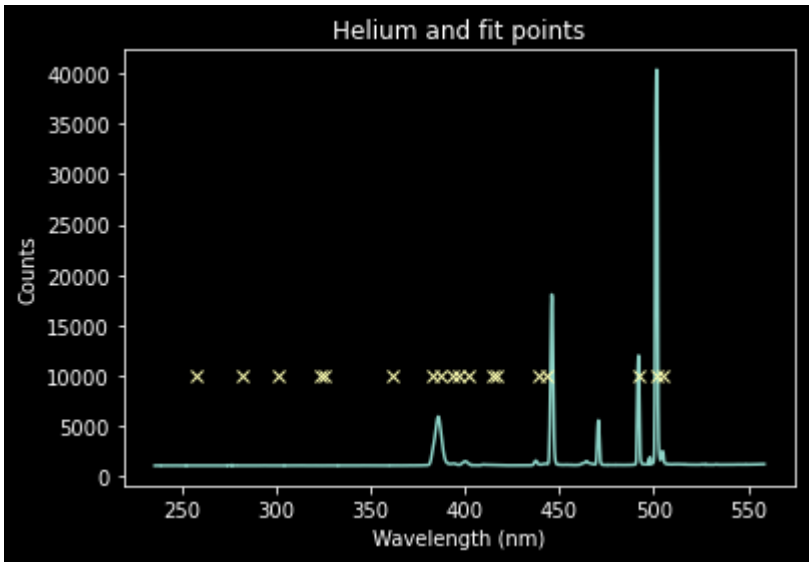
Helium

In [24]:
```
he_keep1 = np.array(['*' not in x for x in helium_lines['Rel.']])
he_keep2 = (~helium_lines['Rel.'].mask)
```

In [26]:
```
he_keep = []
for i in range(len(he_keep1)):
    if he_keep2[i] == 1 & he_keep1[i] == 1:
        he_keep.append(1)
    else:
        he_keep.append(0)
he_keep = np.array(he_keep)
```

In [44]:
```
helium_lines['Rel.'] = helium_lines['Rel.'][he_keep]
he_keep_true = np.array(['*' not in x for x in helium_lines['Rel.']])
he_wl_tbl = helium_lines['Observed'][he_keep_true]
he_rel_tbl = np.array([float(x) for x in helium_lines['Rel.'][he_keep_true]])
he_rel_intens = he_rel_tbl / he_rel_tbl.max() * 10000
he_keep_final = he_rel_intens > 9999

pl.plot(wavelengths, he_spectrum)
pl.plot(he_wl_tbl[he_keep_final], he_rel_intens[he_keep_final], 'x')
pl.xlabel('Wavelength (nm)')
pl.ylabel("Counts")
pl.title("Helium and fit points")
```

Out[44]:
```
Text(0.5, 1.0, 'Helium and fit points')
```



In [30]:
```
he_wl_final = he_wl_tbl[he_keep_final]
he_pixel_vals = linfit_wlmodel.inverse(he_wl_final)
```

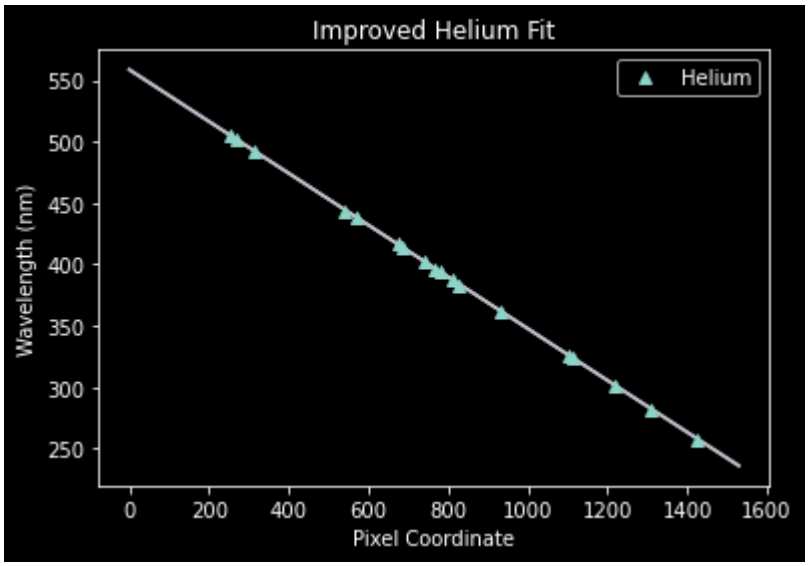
In [31]:
```
he_pixel_remove = (~he_pixel_vals.mask)
he_pixel_vals = he_pixel_vals[he_pixel_remove]
he_wl_remove = (~he_wl_final.mask)
he_wl_final = he_wl_final[he_wl_remove]
```

In [32]:
```
npixels = 5
improved_xval_guesses_he = [np.average(xaxis[g-npixels:g+npixels],
                                       weights=he_spectrum[g-npixels:g+npixels] - np
                            for g in map(int, he_pixel_vals)]
improved_xval_guesses_he
```

Out[32]:
```
[1424.5229157434956,
 1310.2973044397463,
 1217.5799264246548,
 1114.8137563668884,
 1101.5242100798755,
 933.2120090669382,
 828.539626890745,
 811.3429468594837,
 780.3101168693187,
 766.5242928511118,
 740.3484761055448,
 683.245026525199,
 675.9616451016634,
 568.240468301561,
 539.2740680836008,
 315.1338081072584,
 269.98213362268183,
 254.61648807504866]
```

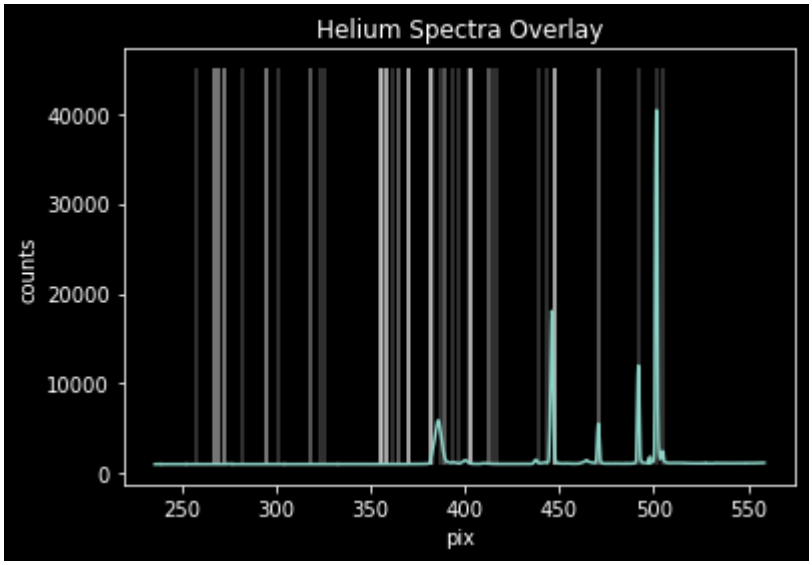
In [38]:
```
pl.plot(improved_xval_guesses_he, he_wl_final, '^', label='Helium')
pl.plot(xaxis, wavelengths, zorder=-5)
pl.plot(xaxis, linfit_wlmodel(xaxis), zorder=-5)
pl.legend(loc='best')
pl.xlabel("Pixel Coordinate")
pl.ylabel("Wavelength (nm)")
pl.title("Improved Helium Fit")
```

Out[38]: Text(0.5, 1.0, 'Improved Helium Fit')



In [37]:
```
pl.plot(wavelengths, he_spectrum)
pl.vlines(helium_lines['Observed'], 1000, 45000, 'w', alpha=0.25);
pl.xlabel("pix")
pl.ylabel('counts')
pl.title('Helium Spectra Overlay')
```

Out[37]:
```
Text(0.5, 1.0, 'Helium Spectra Overlay')
```

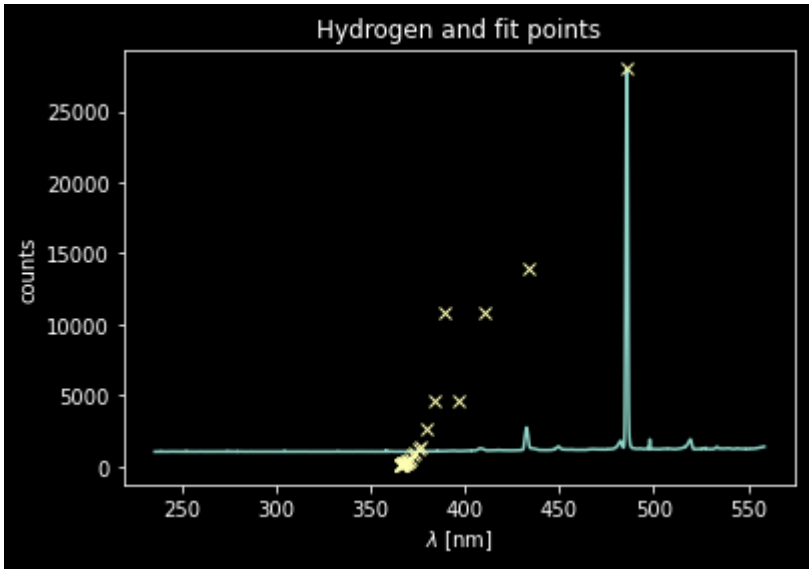


In [ ]:
```
```

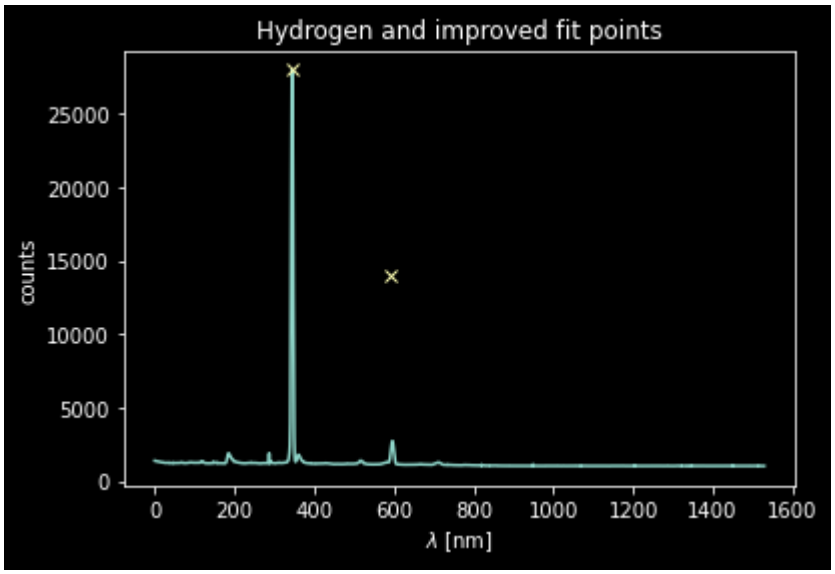
Hydrogen

In [40]:
```
h_keep = (~hydrogen_lines['Rel.'].mask) & (hydrogen_lines['Rel.'] != "700bl") & (
h_wl_tbl = hydrogen_lines['Observed'][h_keep]
h_rel_tbl = np.array([float(x) for x in hydrogen_lines['Rel.'][h_keep]])
```

In [62]:
```
pl.plot(wavelengths, h_spectrum)
pl.plot(h_wl_tbl, h_rel_tbl / h_rel_tbl.max() * h_spectrum.max(), 'x')
pl.xlabel("$\lambda$ [nm]")
pl.ylabel("counts")
pl.title("Hydrogen and fit points");
```



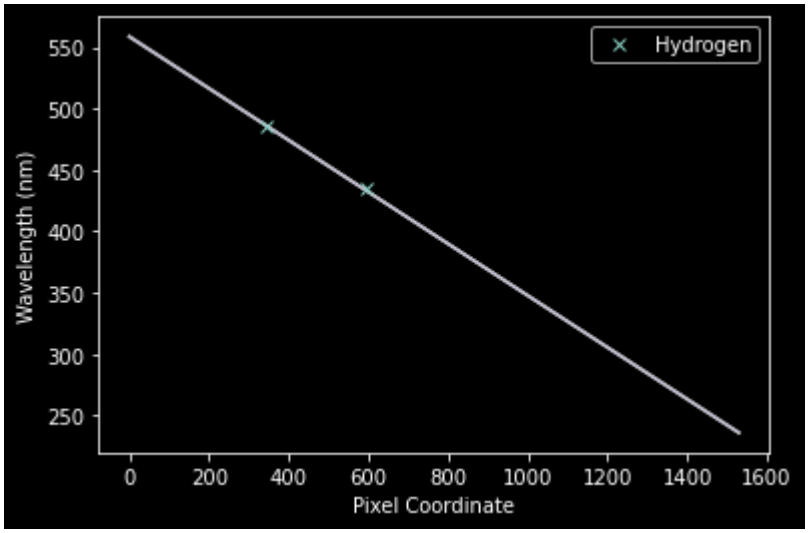
In [63]:
```
h_rel_intens =  h_rel_tbl / h_rel_tbl.max() * h_spectrum.max()
h_keep_final = h_rel_intens > 12500
h_wl_final = h_wl_tbl[h_keep_final]
h_pixel_vals = linfit_wlmodel.inverse(h_wl_final)

pl.plot(xaxis, h_spectrum)
pl.plot(h_pixel_vals, h_rel_intens[h_keep_final], 'x');
pl.xlabel("$\lambda$ [nm]")
pl.ylabel("counts")
pl.title("Hydrogen and improved fit points");
```



In [64]:
```
npixels = 15
improved_xval_guesses_h = [np.average(xaxis[g-npixels:g+npixels],
                                      weights=h_spectrum[g-npixels:g+npixels] - np.
                           for g in map(int, h_pixel_vals)]
improved_xval_guesses_h
```

Out[64]:
```
[593.817694714745, 344.59213802359596]
```

In [65]:
```
pl.plot(improved_xval_guesses_h, h_wl_final, 'x', label='Hydrogen')
pl.plot(xaxis, wavelengths, zorder=-5)
pl.plot(xaxis, linfit_wlmodel(xaxis), zorder=-5)
pl.legend(loc='best')
pl.xlabel("Pixel Coordinate")
pl.ylabel("Wavelength (nm)");
```

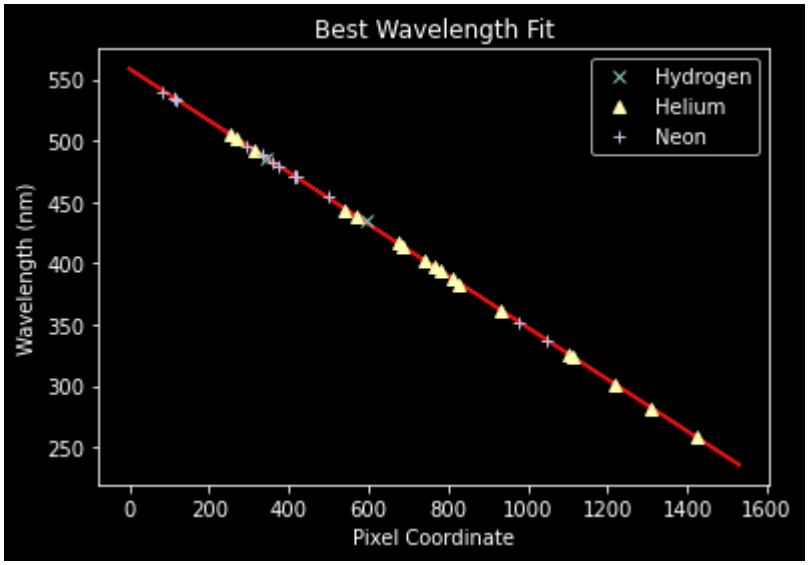


In [66]:
```
xvals_ne_plus_he_plus_h = list(improved_xval_guesses_ne) + list(improved_xval_gue
waves_ne_plus_he_plus_h = list(ne_wl_final) + list(he_wl_final) + list(h_wl_final
linfit_wlmodel_neheh = linfitter(model=wlmodel, x=xvals_ne_plus_he_plus_h, y=wave
linfit_wlmodel_neheh
```

Out[66]:
```
<Linear1D(slope=-0.21109674, intercept=558.63214298)>
```

In [70]:
```
pl.plot(improved_xval_guesses_h, h_wl_final, 'x', label='Hydrogen')
pl.plot(improved_xval_guesses_he, he_wl_final, '^', label='Helium')
pl.plot(improved_xval_guesses_ne, ne_wl_final, '+', label='Neon')
pl.plot(xaxis, wavelengths, zorder=-5)
# Plotting the same thing but with the final wavelength solution!
pl.plot(xaxis, linfit_wlmodel_neheh(xaxis), zorder=-5, color='r')
pl.legend(loc='best')
pl.xlabel("Pixel Coordinate")
pl.ylabel("Wavelength (nm)");
pl.title("Best Wavelength Fit")
```

Out[70]:
```
Text(0.5, 1.0, 'Best Wavelength Fit')
```



In [93]:
```
h_lam = np.median(h_wl_final)
```

In [94]:
```
he_lam = np.median(he_wl_final)
```

In [95]:
```
ne_lam = np.median(ne_wl_final)
```

In [104]:
```
D = 830
```

In [105]:
```
ne_ang = np.arcsin(ne_lam/D)
ne_ang = ne_ang*u.rad
ne_ang.to(u.deg)
```

Out[105]: $34.939174\ ^\circ$

In [106]:
```
he_ang = np.arcsin(he_lam/D)
he_ang = he_ang*u.rad
he_ang.to(u.deg)
```

Out[106]: $28.429491\ ^\circ$

In [107]:
```
h_ang = np.arcsin(h_lam/D)
h_ang = h_ang*u.rad
h_ang.to(u.deg)
```

Out[107]: $33.675047\ ^\circ$

In [ ]:
```

```