

gesture_statistics

June 6, 2017

1 TIMIT Gestures Statistics

1.1 Load Db and Calculate Gestures

```
In [9]: import timit_stats as tist
        import gesture as ges
        import os
        import matplotlib.pyplot as plt

%pylab inline

trans_dir = "../USC-TIMIT/EMA/Data/F1/trans"
mat_dir = "../USC-TIMIT/EMA/Data/F1/mat"

gestures = {}

for fname in os.listdir(trans_dir):
    fname = os.path.splitext(fname)[0]
    t_fname = os.path.join(trans_dir, fname + ".trans")
    mat_fname = os.path.join(mat_dir, fname + ".mat")
    gest = tist.calc_gestures(mat_fname, t_fname)
    for g in gest:
        if g not in gestures:
            gestures[g] = ges.Gesture(g)
            gestures[g].extend(gest[g])
print "gestures calculation finished"
```

Populating the interactive namespace from numpy and matplotlib

WARNING: pylab import has clobbered these variables: ['colors', 'e']
`%matplotlib` prevents importing * from pylab and numpy

gestures calculation finished

1.2 Plot Gestures

In [10]: `from matplotlib.patches import Ellipse`

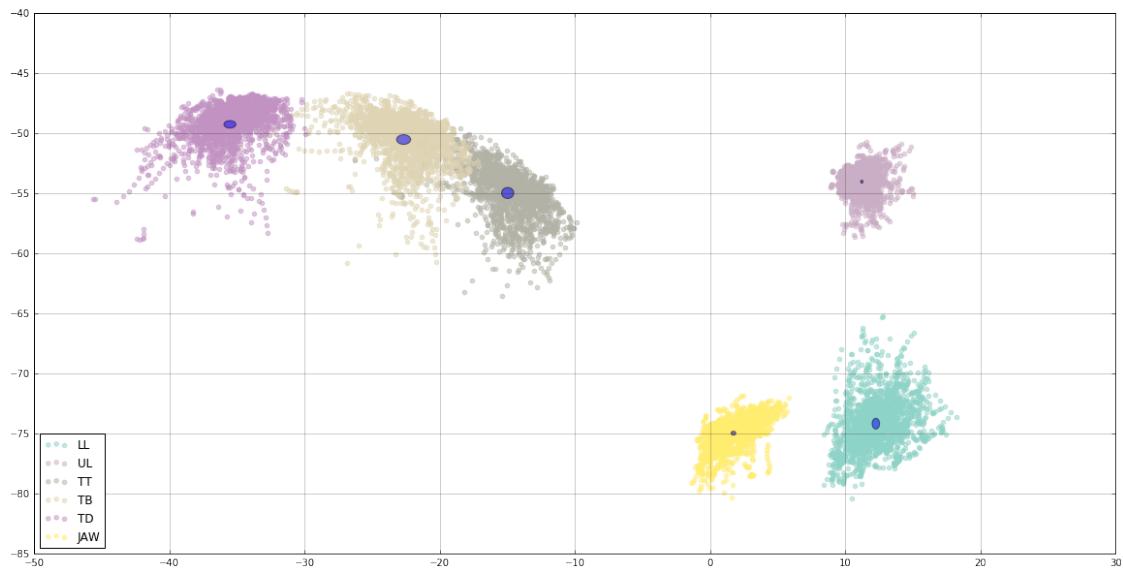
```
articulators = ["LL", "UL", "TT", "TB", "TD", "JAW"]

cmap = plt.get_cmap('Set3')
colors = [cmap(i) for i in np.linspace(0, 1, len(articulators))]
for g_name, g in gestures.items():
    fig1, ax1 = plt.subplots(figsize=(20, 10))
    fig1.suptitle("Gesture \"{}\" (samples = {})".format(g_name, len(g.params["LL_x"]))
                  , fontsize=20, fontweight='bold')
    #     ax1.set_xlim(-50, 40)
    #     ax1.set_ylim(-30, 30)
    ax1.grid(color='black', linestyle='-', linewidth=1, alpha=0.2)
    #     img = plt.imread("vt_bg.jpg")
    #     ax1.imshow(img, extent=[-100, 40, -70, 50], alpha=0.3)
    g_m = g.get_mean()
    g_v = g.get_variance()

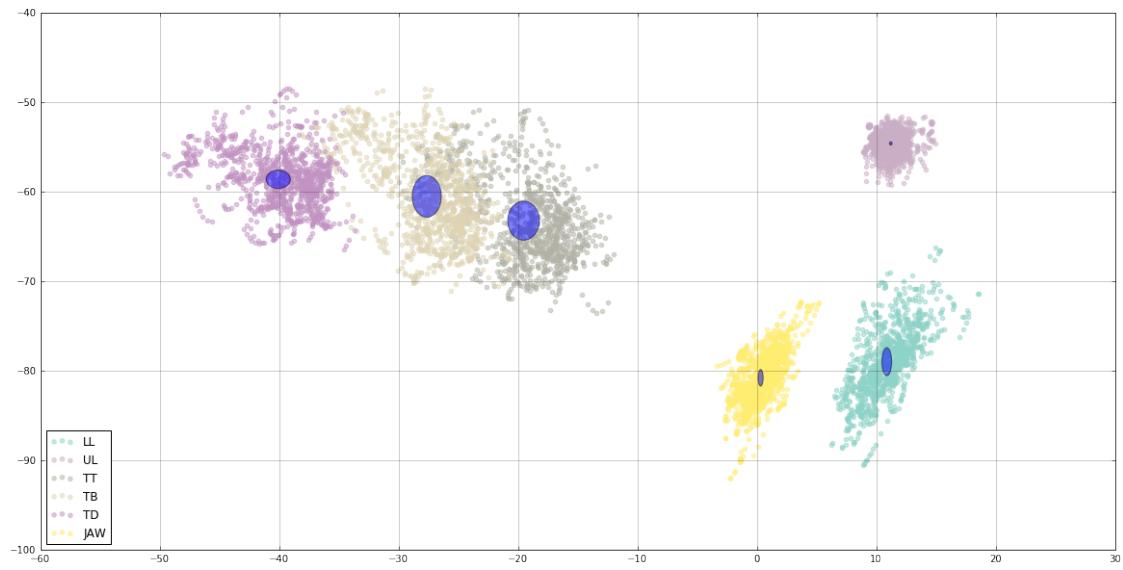
    for i in range(len(articulators)):
        a = articulators[i]
        a_x = g.params[a+"_x"]
        a_y = g.params[a+"_y"]
        ax1.scatter(a_x, a_y, color=colors[i], alpha=0.5, label=a)
    #         plot mean ellipse
        e = Ellipse(xy=[g_m[a+"_x"], g_m[a+"_y"]],
                    width=g_v[a+"_x"]/5, height=g_v[a+"_y"]/5, alpha=0.5)
        ax1.add_artist(e)
    plt.legend(loc='lower left')

plt.show()
```

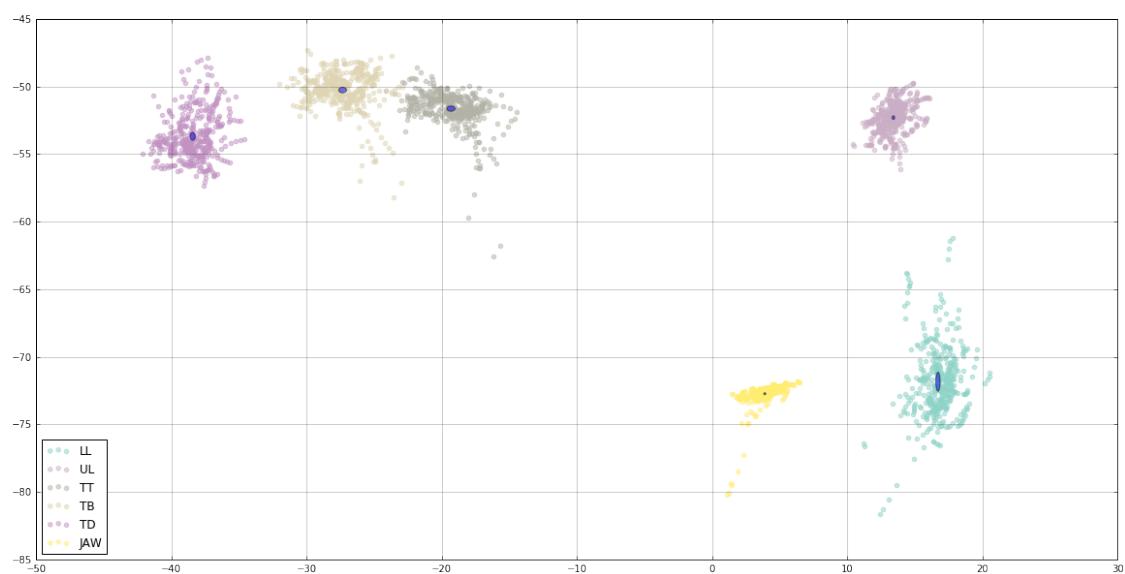
Gesture "iy" (samples = 2034)



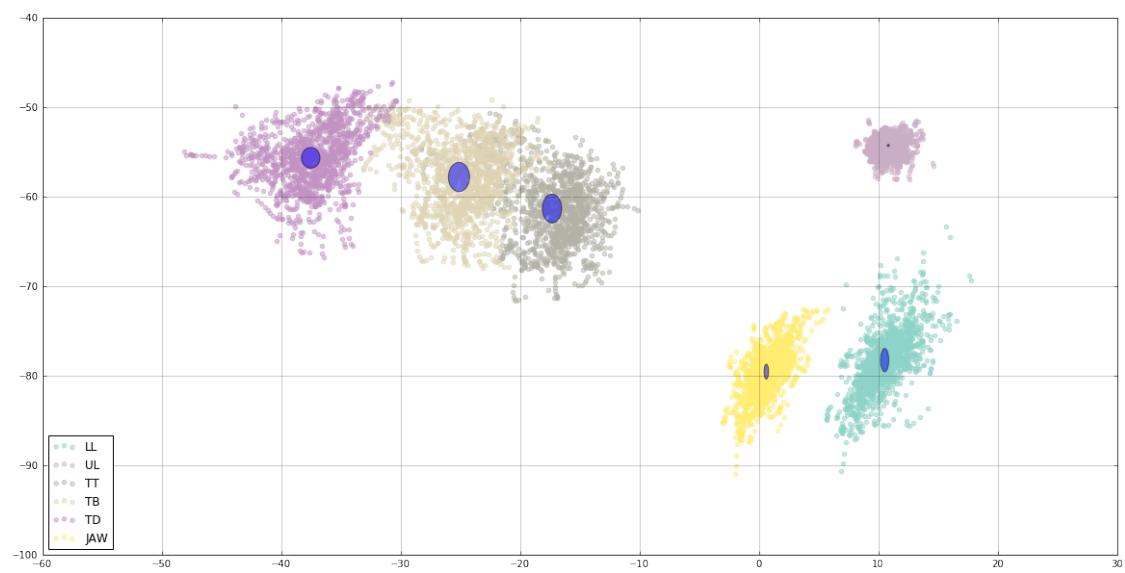
Gesture "aa" (samples = 1026)



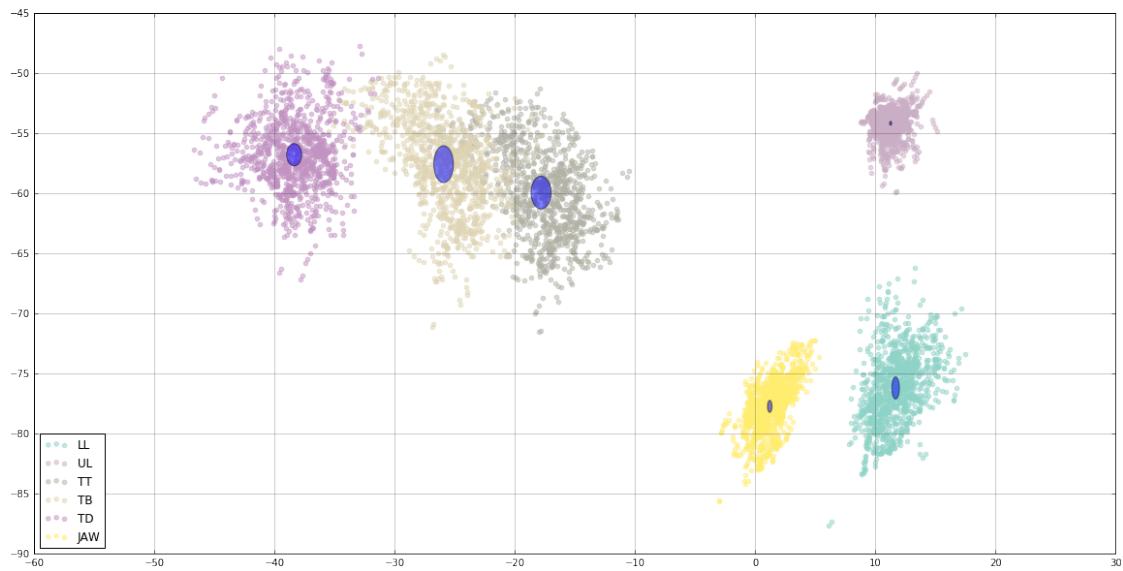
Gesture "ch" (samples = 424)



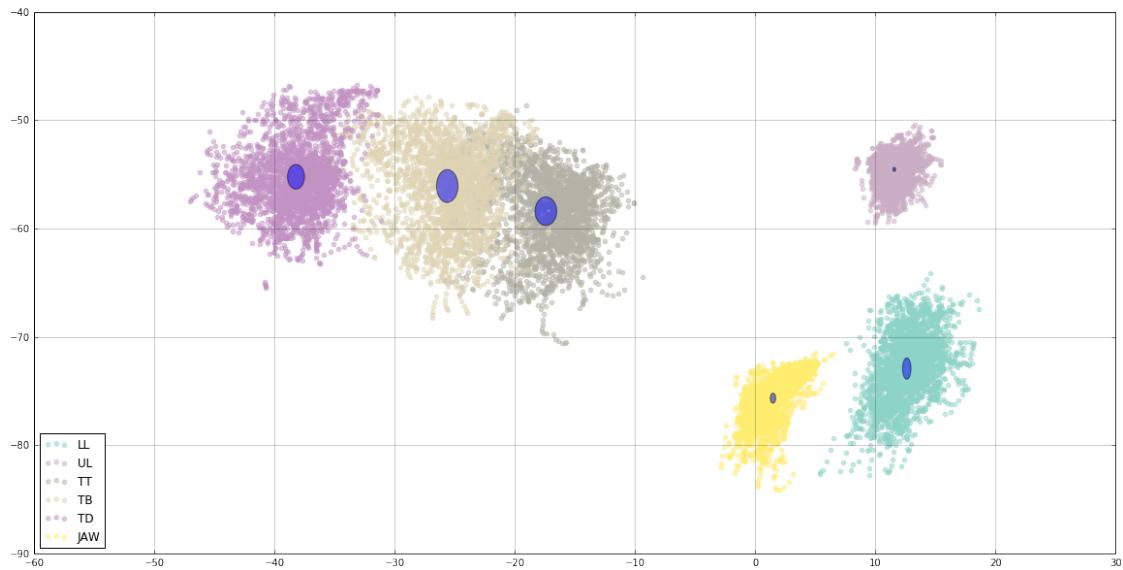
Gesture "ae" (samples = 1370)



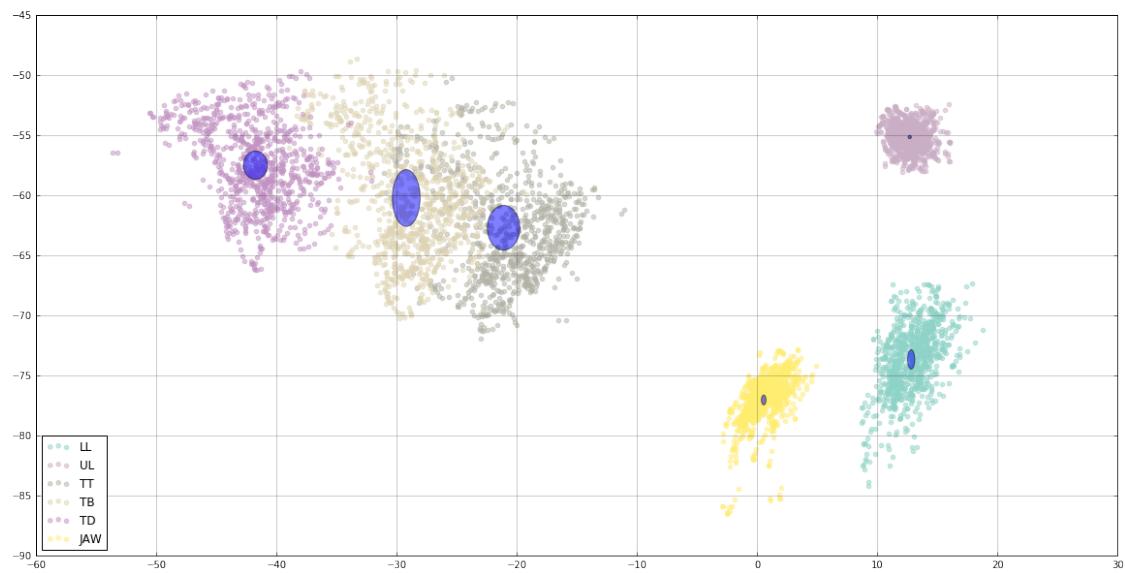
Gesture "eh" (samples = 1046)



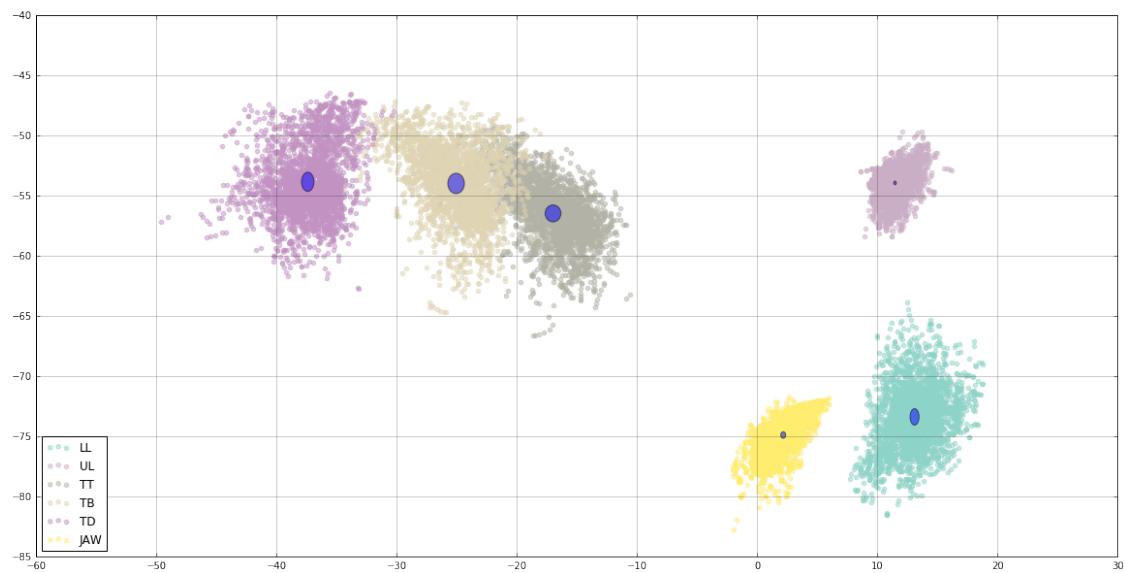
Gesture "ah" (samples = 2658)



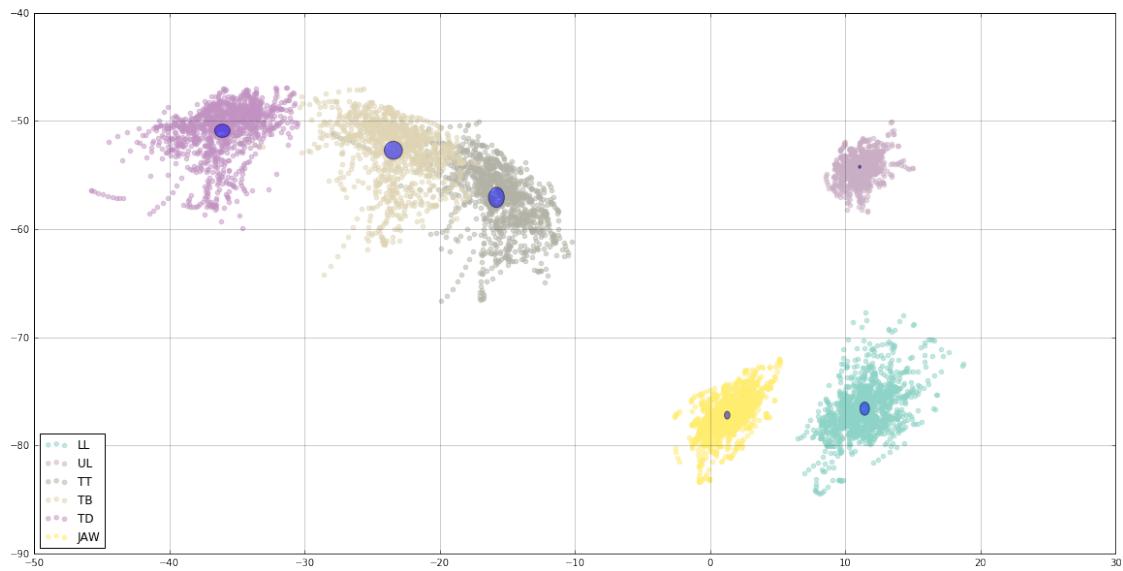
Gesture "ao" (samples = 890)



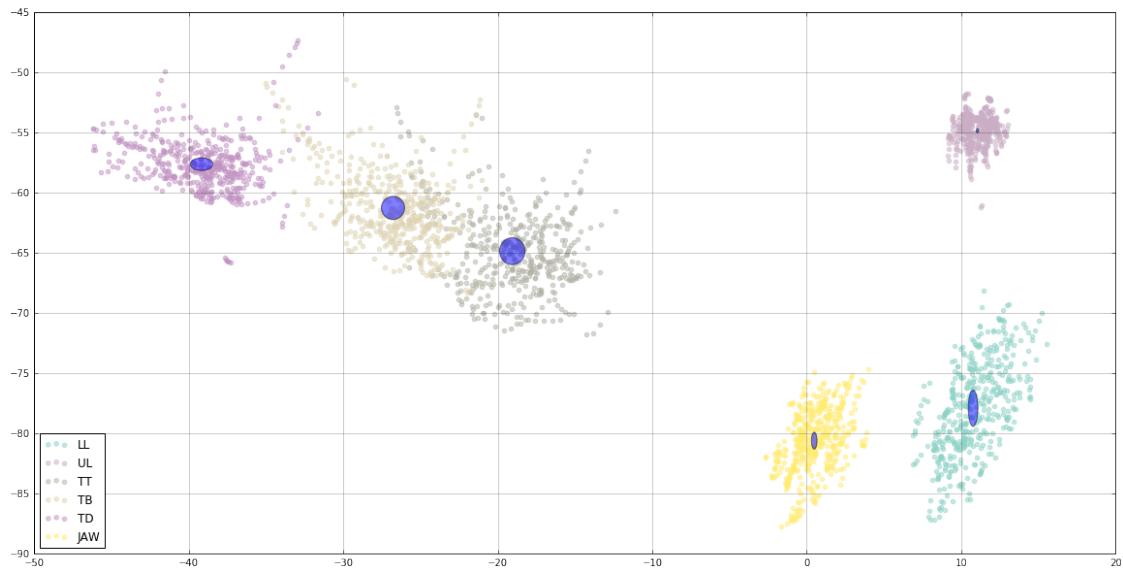
Gesture "ih" (samples = 2875)



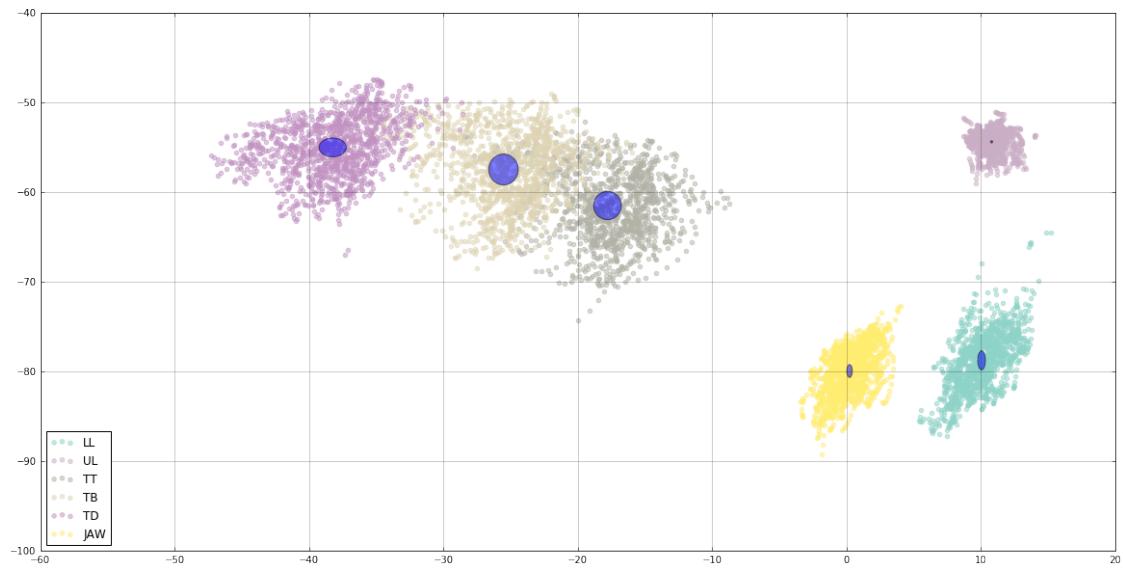
Gesture "ey" (samples = 1176)



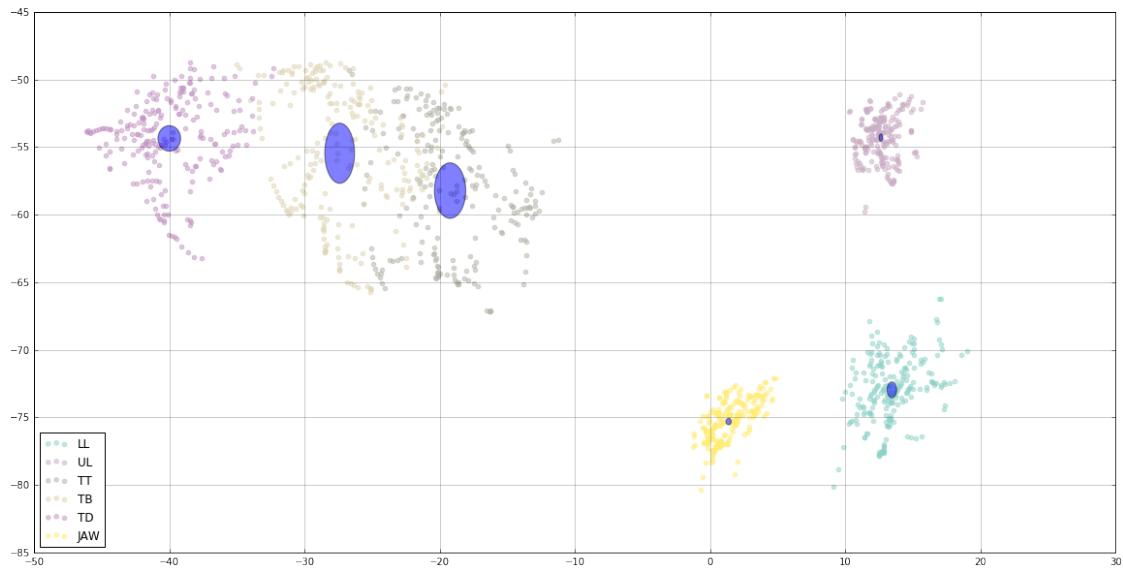
Gesture "aw" (samples = 456)



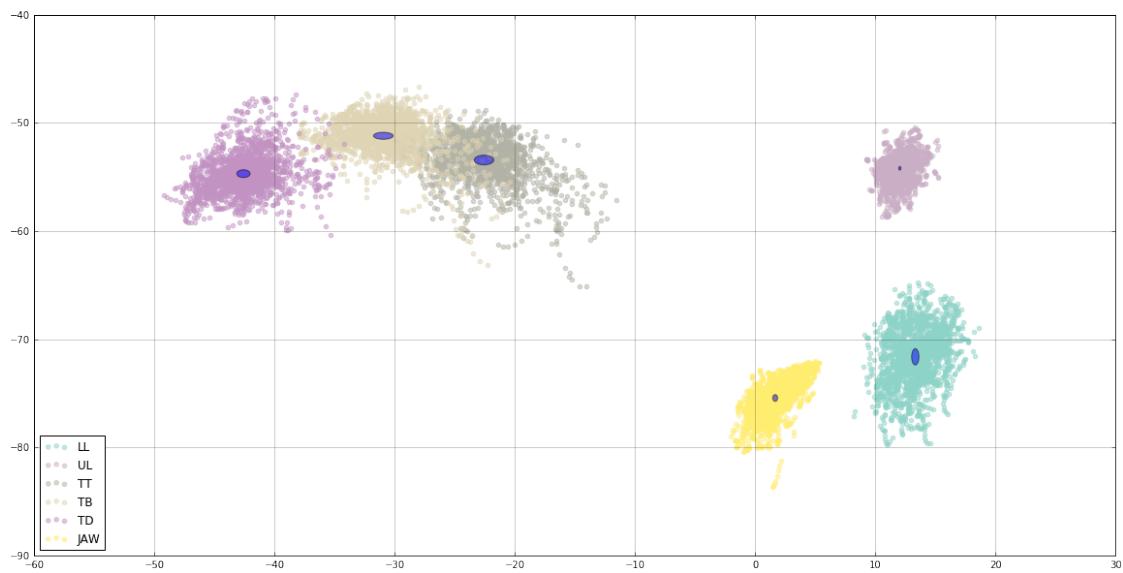
Gesture "ay" (samples = 1228)



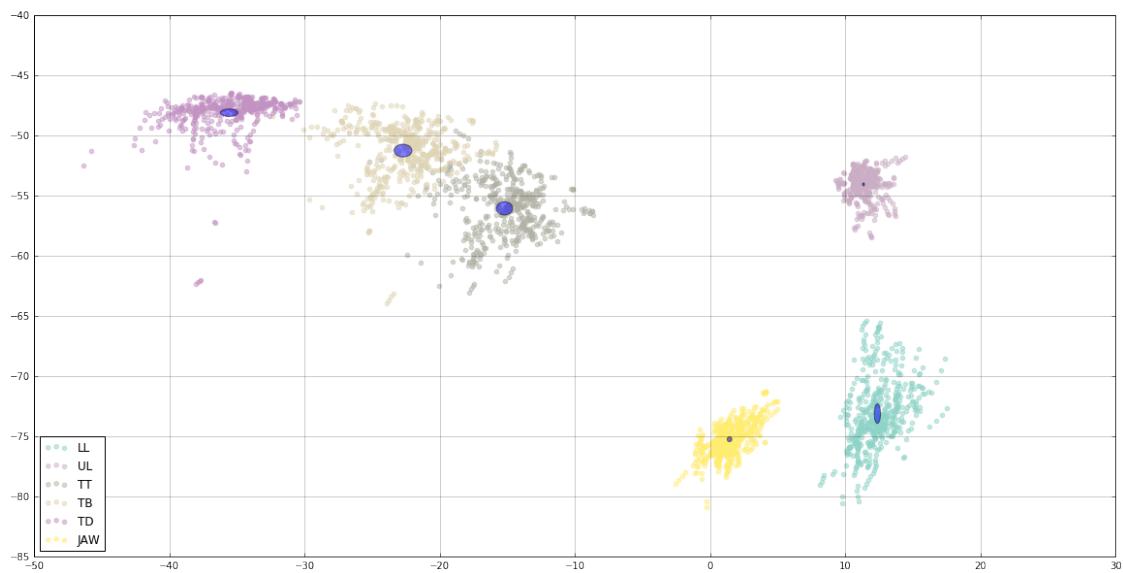
Gesture "uh" (samples = 220)



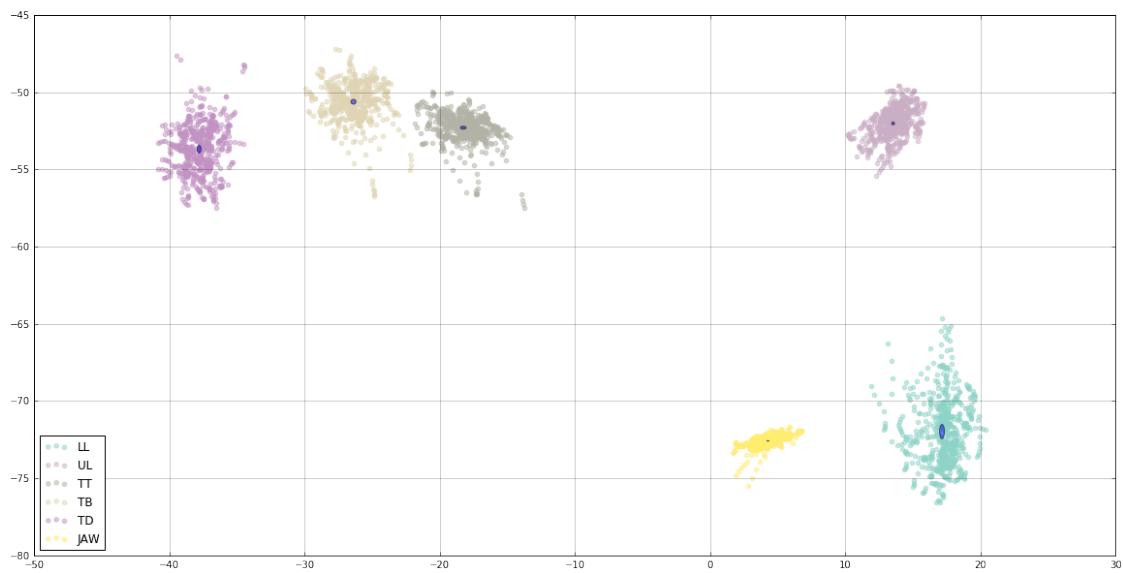
Gesture "er" (samples = 1888)



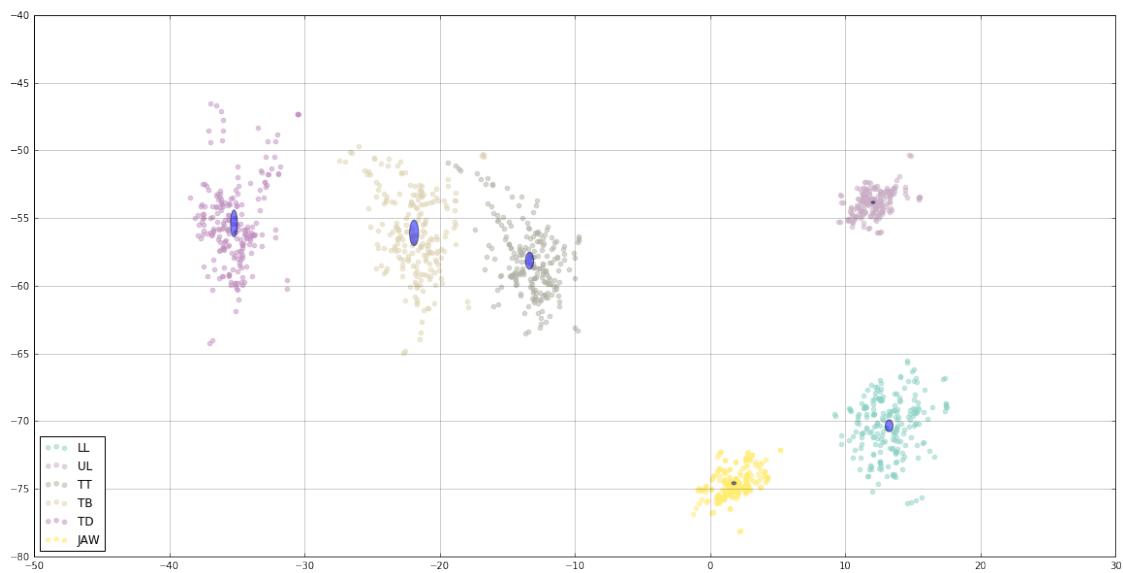
Gesture "ng" (samples = 498)



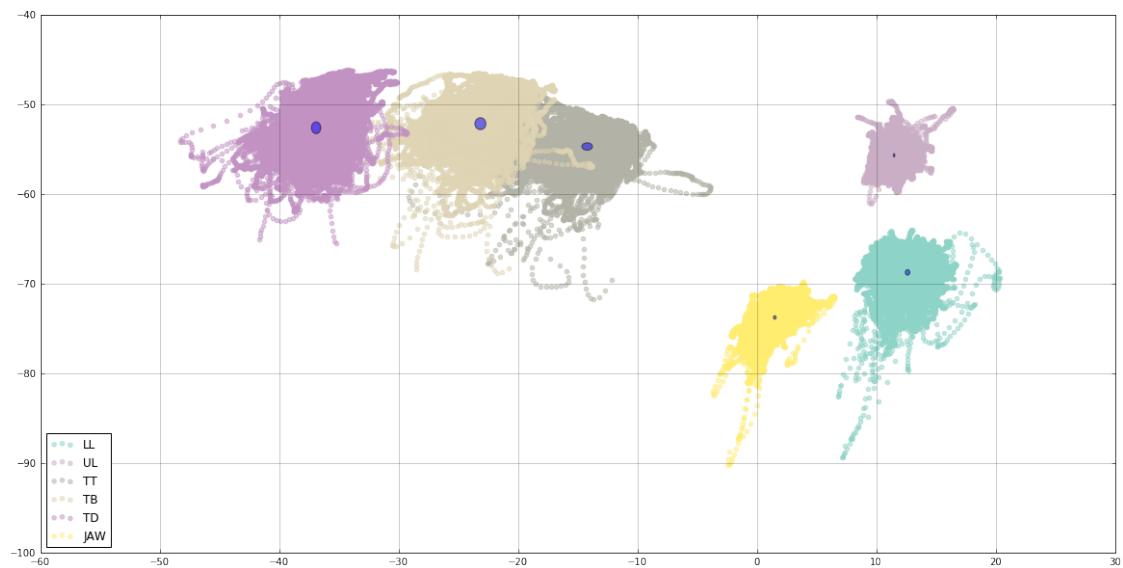
Gesture "sh" (samples = 610)



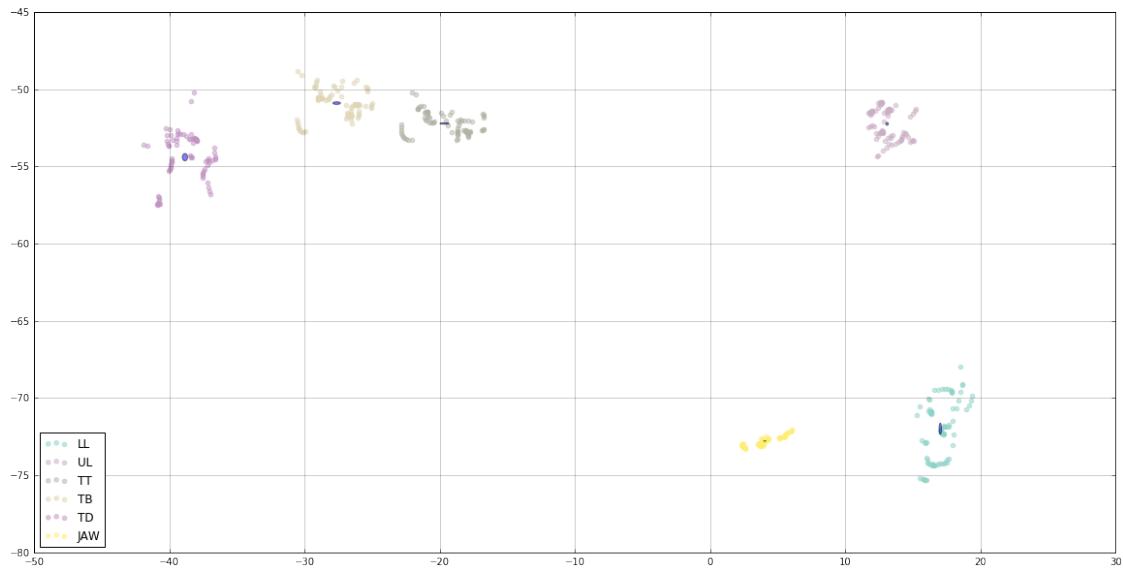
Gesture "th" (samples = 220)



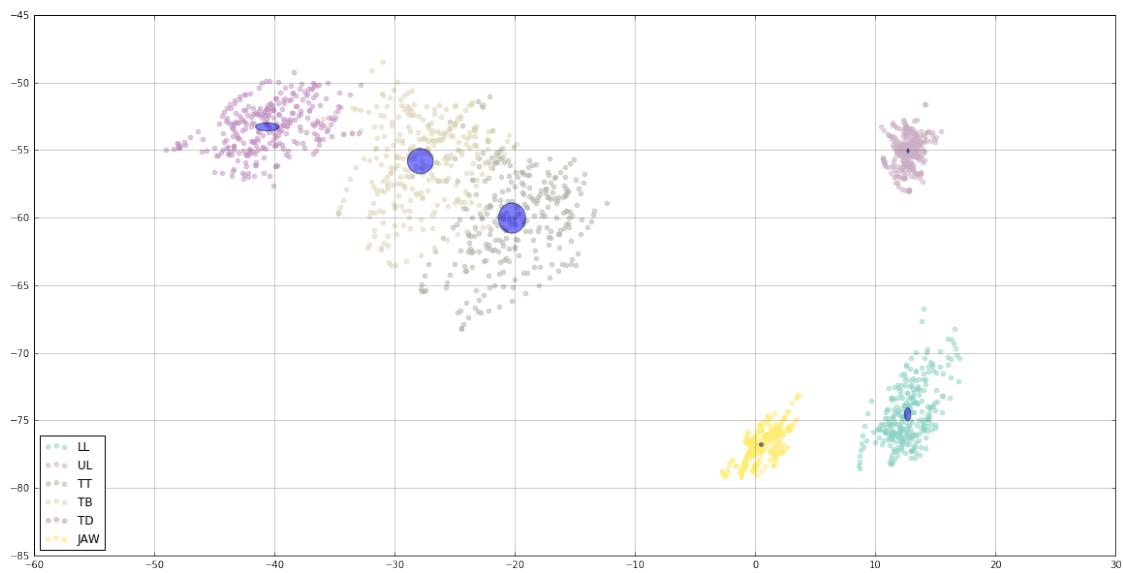
Gesture "sil" (samples = 46386)



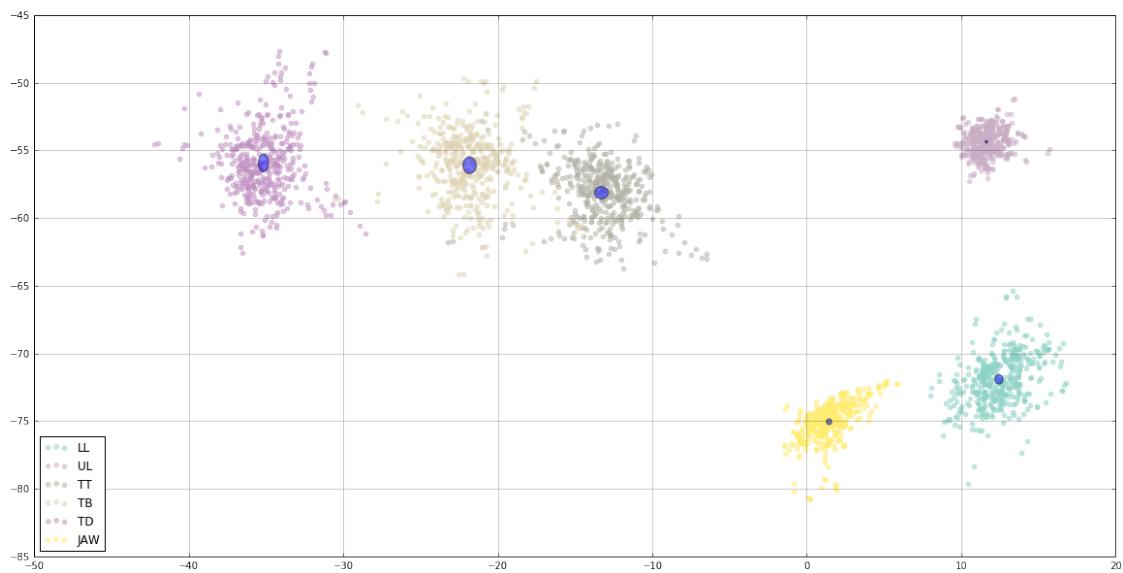
Gesture "zh" (samples = 82)



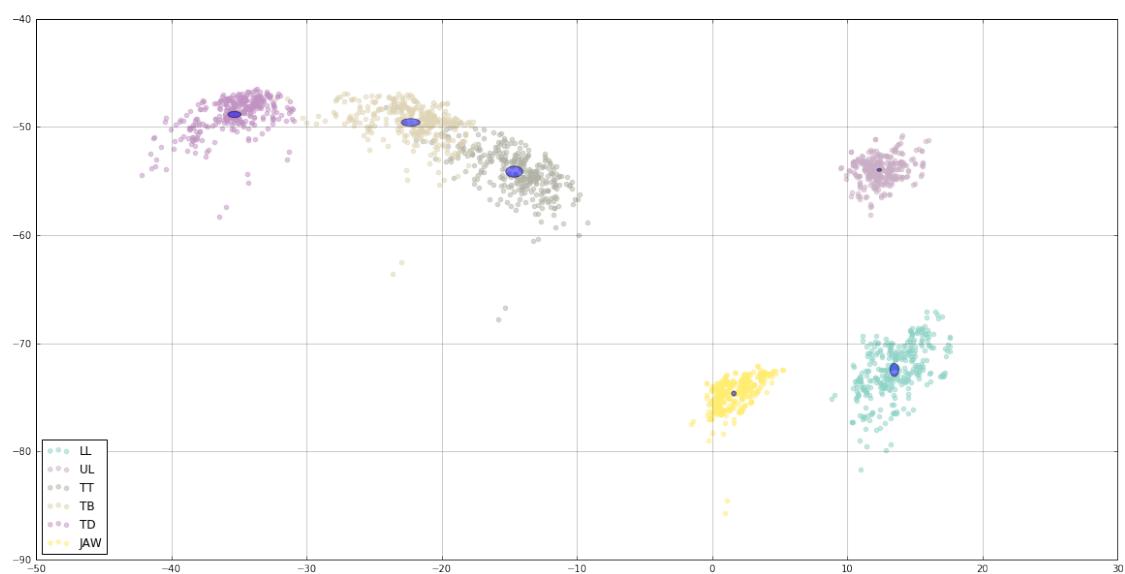
Gesture "oy" (samples = 290)



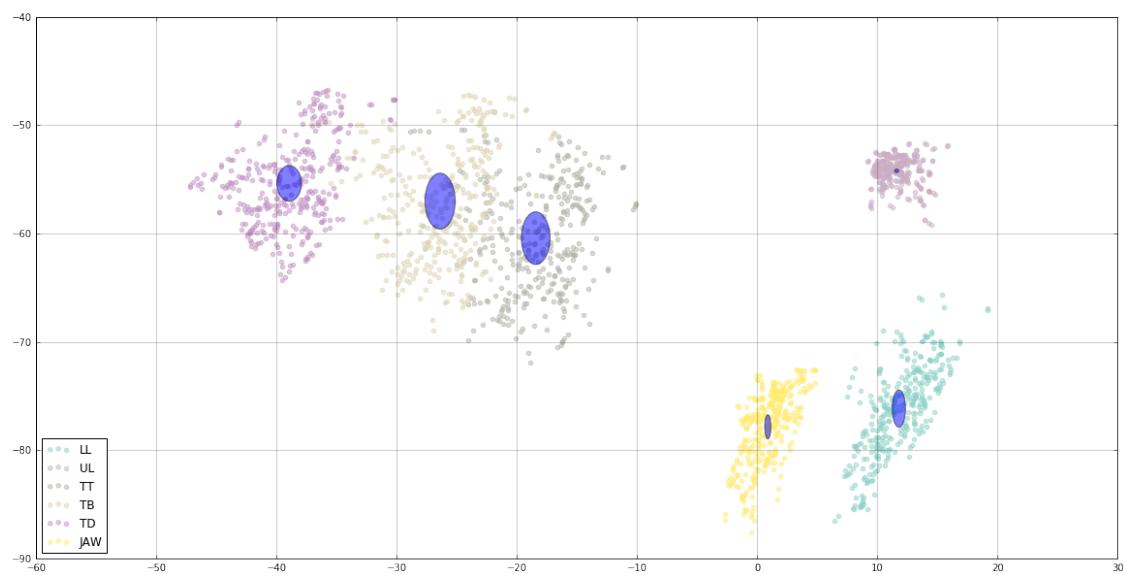
Gesture "dh" (samples = 432)



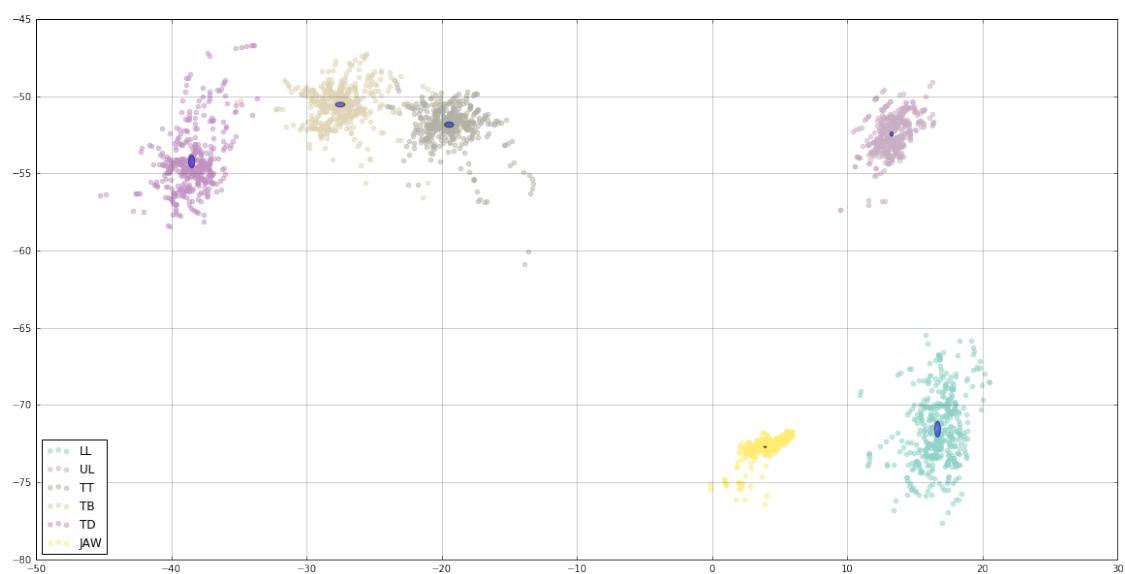
Gesture "y" (samples = 324)



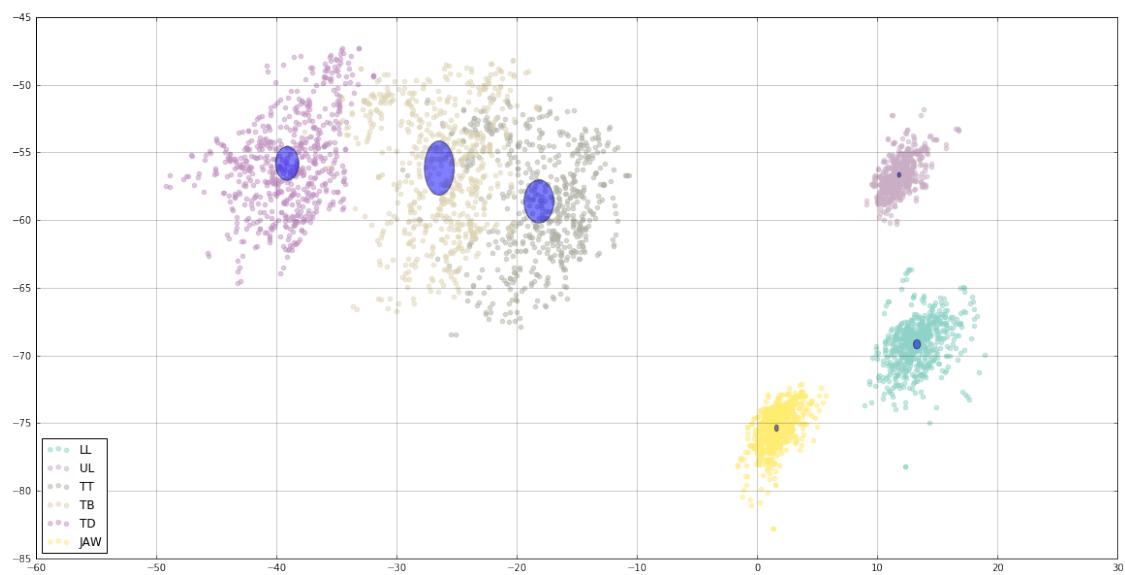
Gesture "hh" (samples = 346)



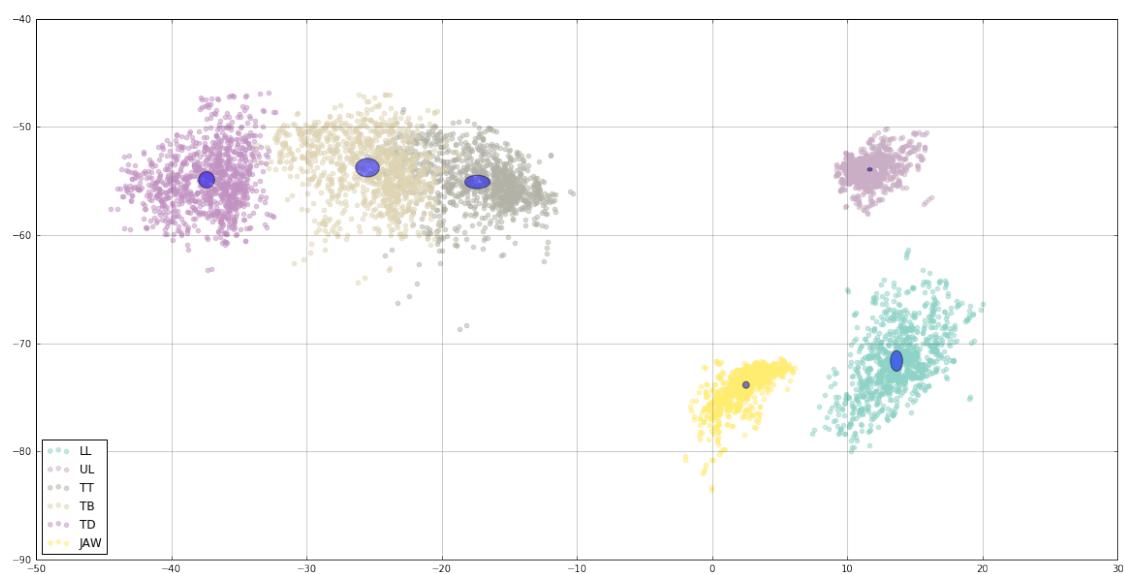
Gesture "jh" (samples = 460)



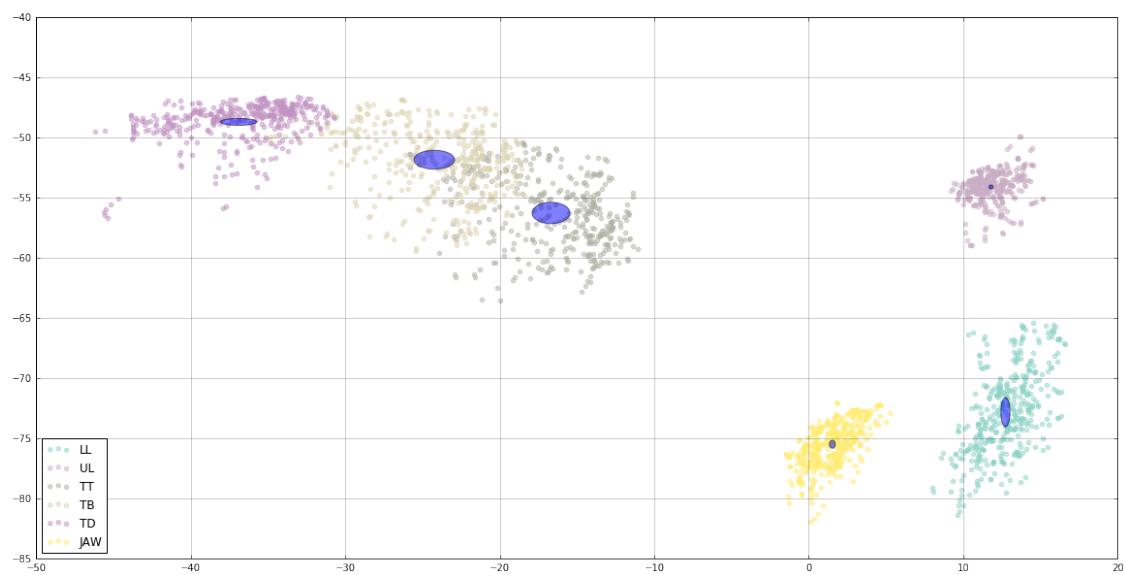
Gesture "b" (samples = 610)



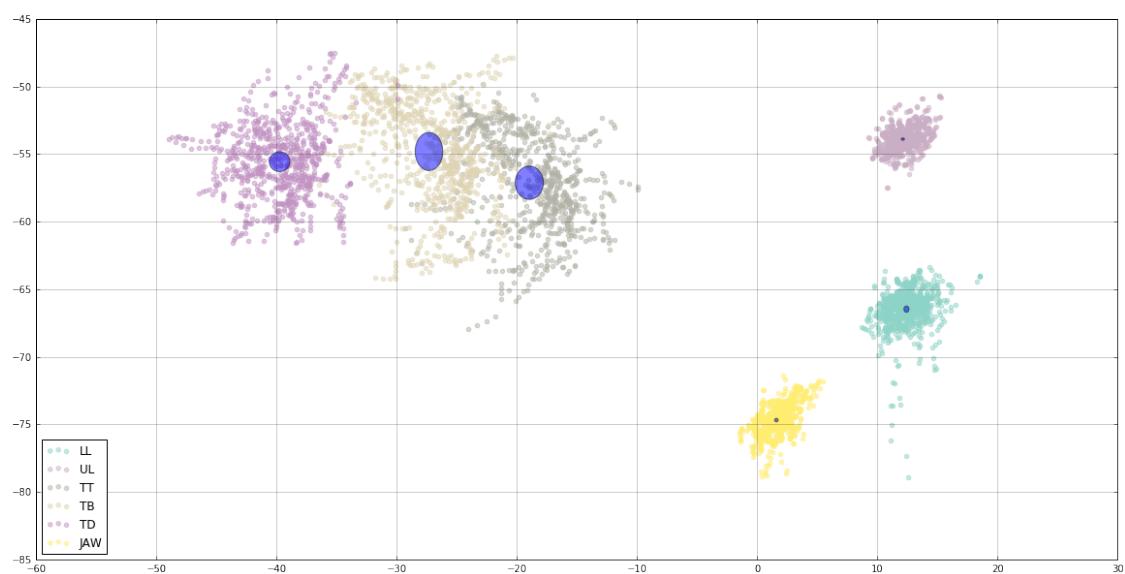
Gesture "d" (samples = 1076)



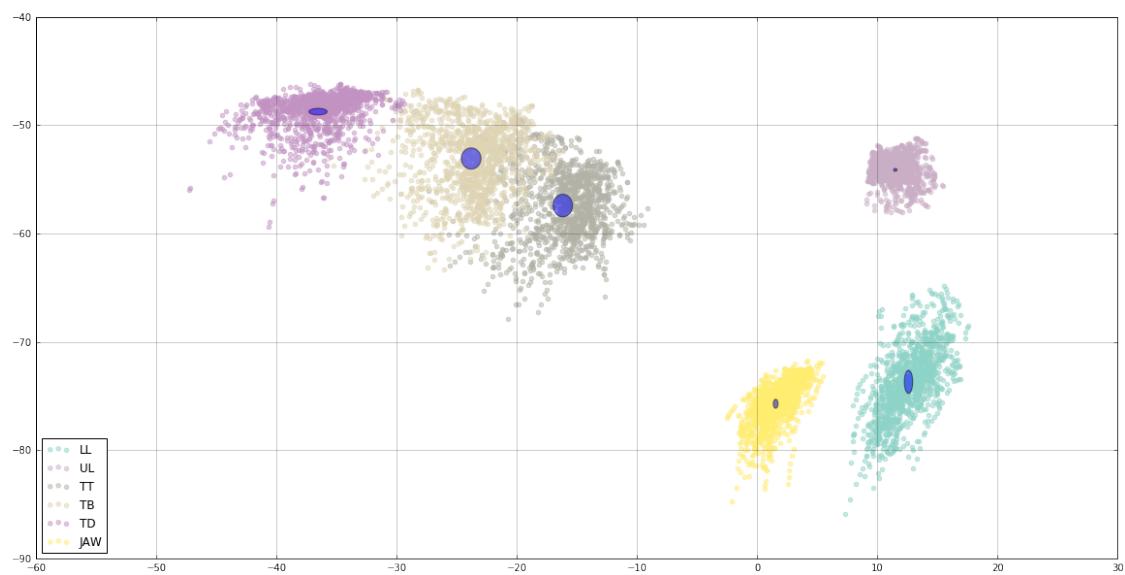
Gesture "g" (samples = 400)



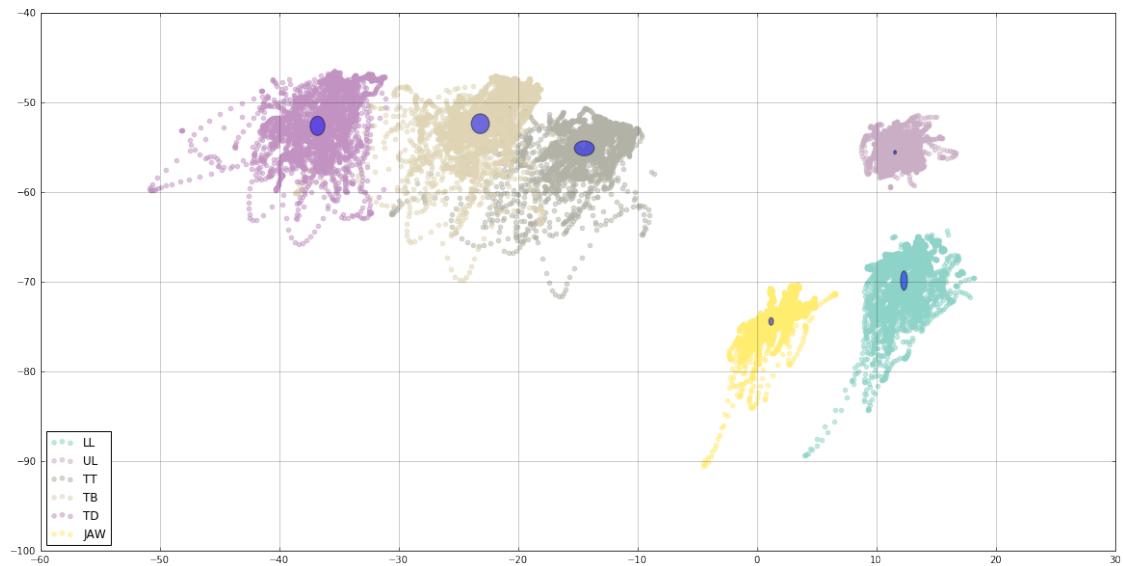
Gesture "f" (samples = 774)



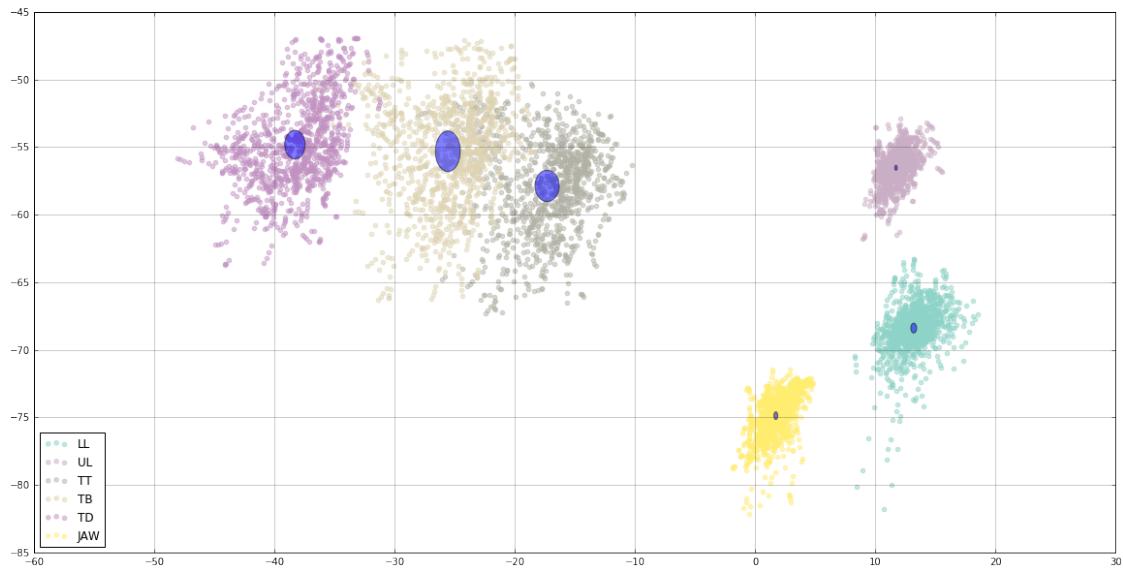
Gesture "k" (samples = 1282)



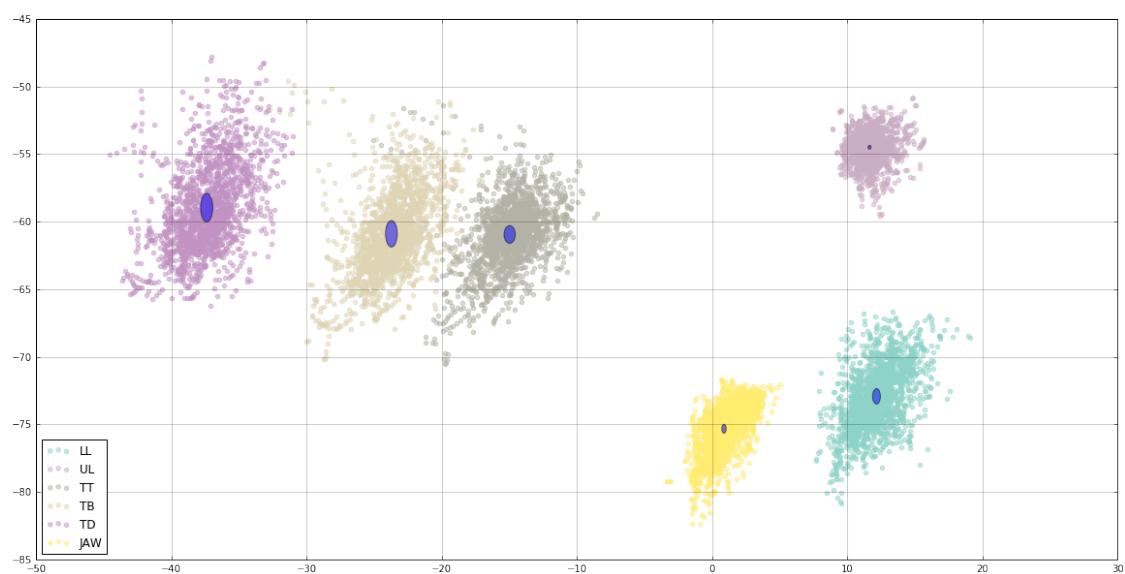
Gesture "sp" (samples = 4068)



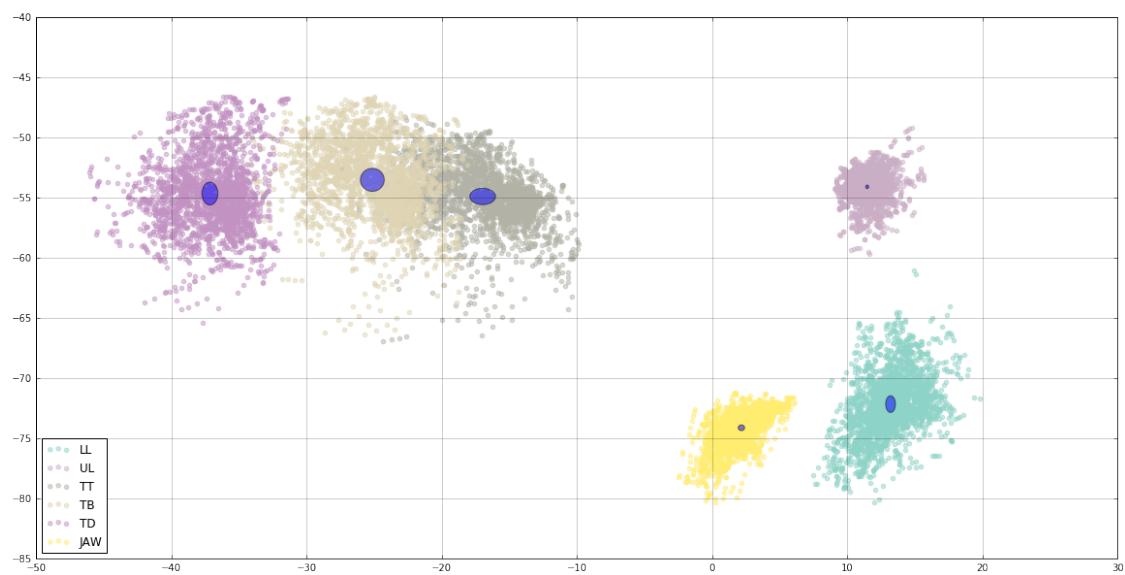
Gesture "m" (samples = 1114)



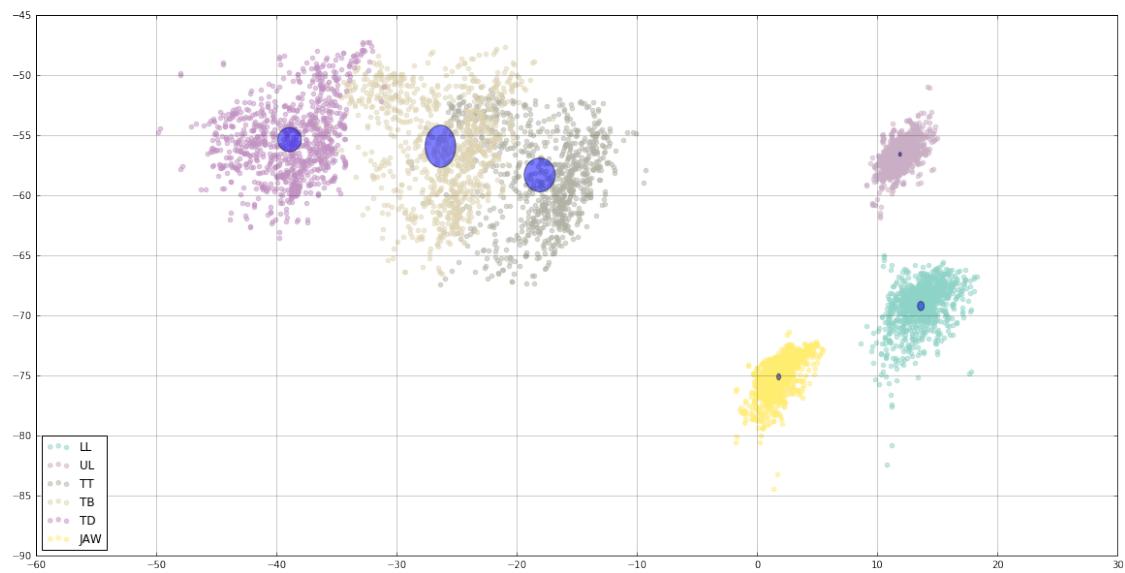
Gesture "I" (samples = 1822)



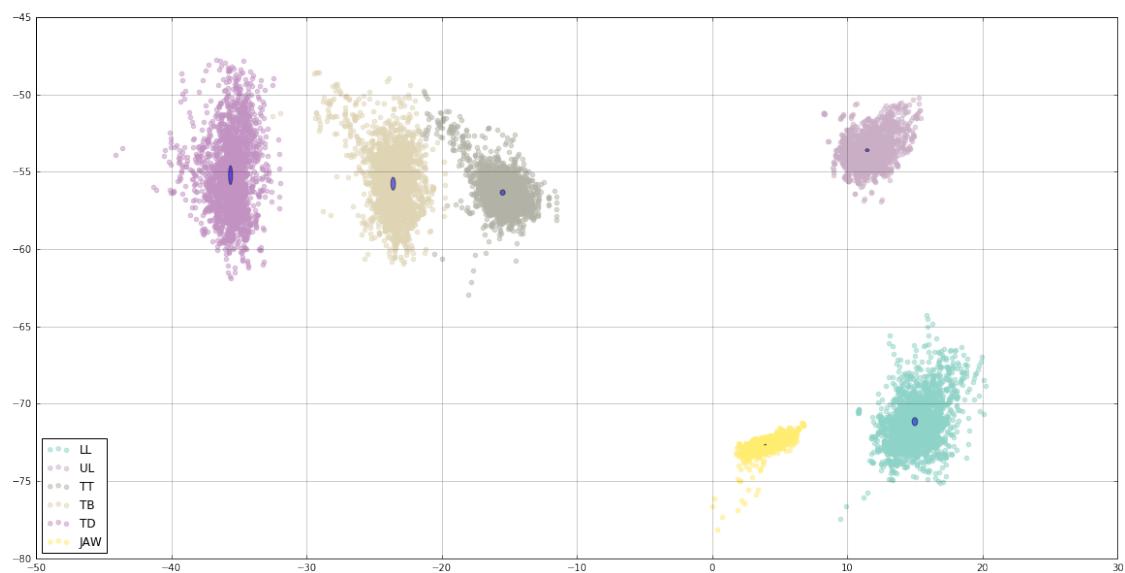
Gesture "n" (samples = 2408)



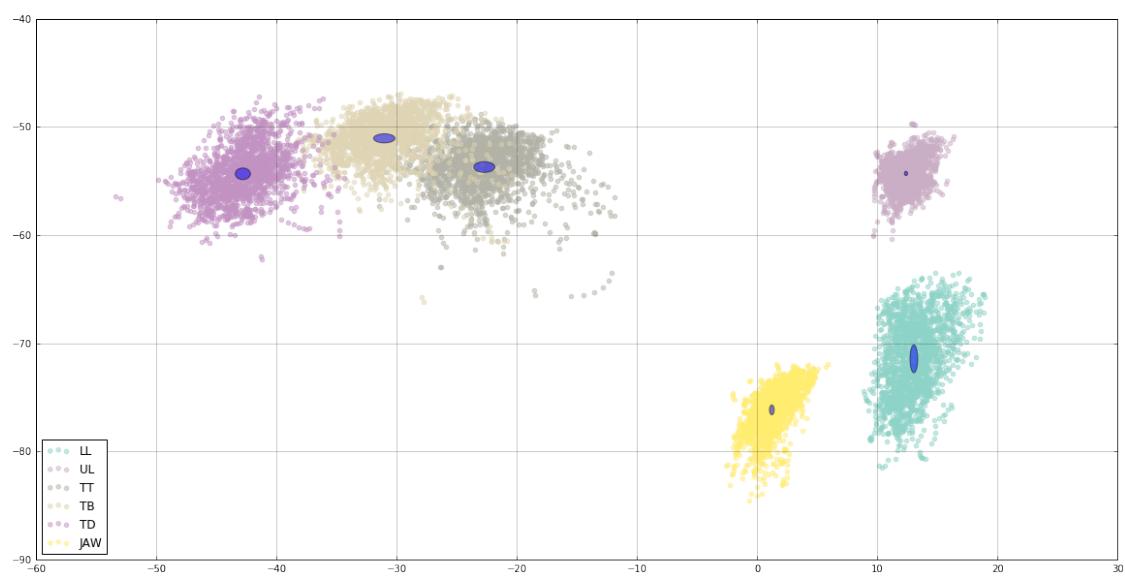
Gesture "p" (samples = 928)



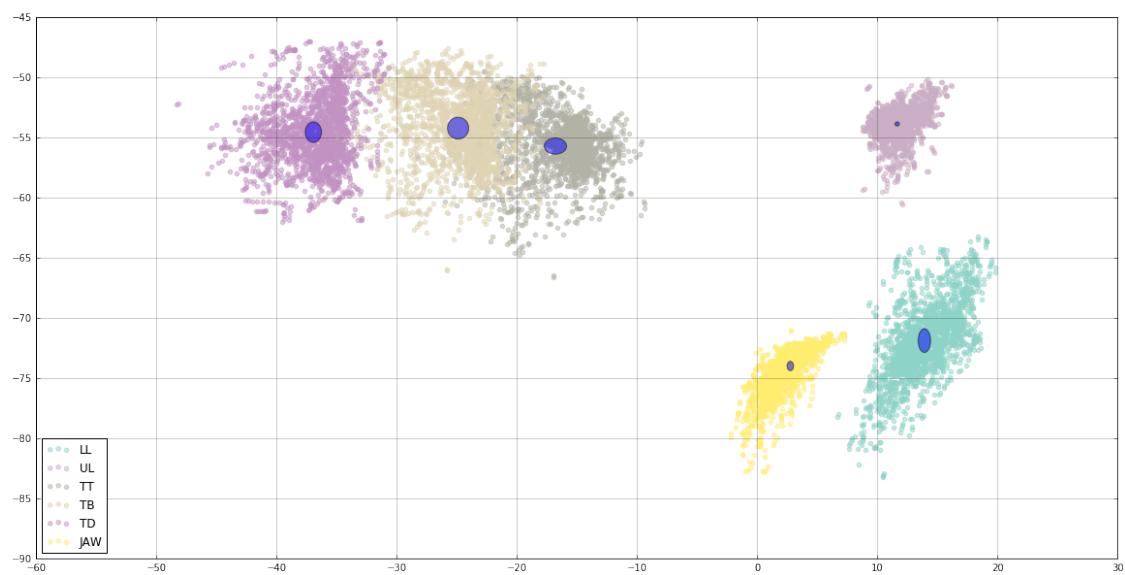
Gesture "s" (samples = 2328)



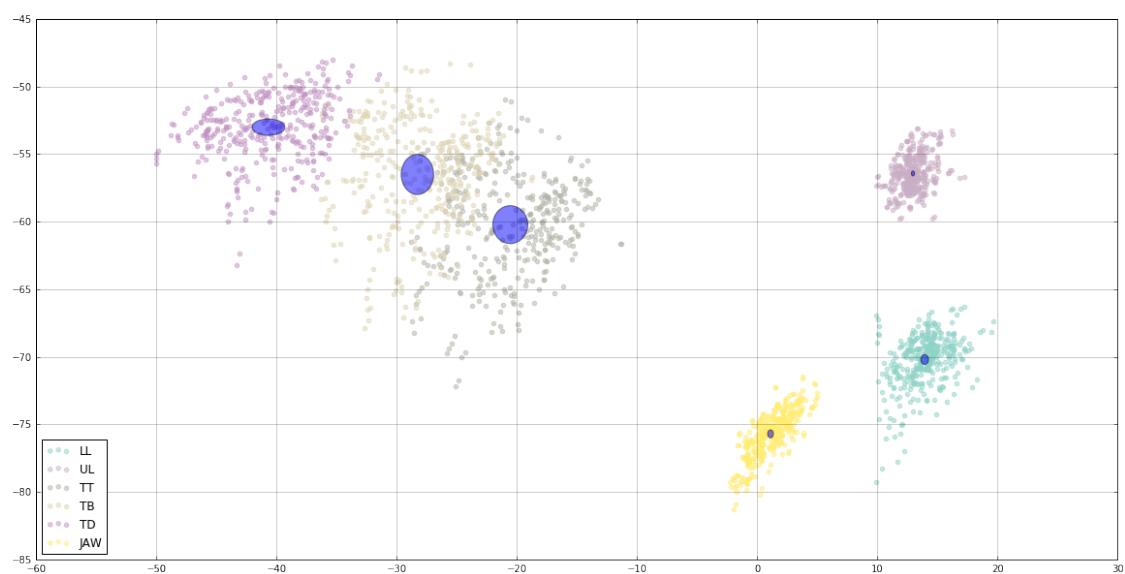
Gesture "r" (samples = 1862)



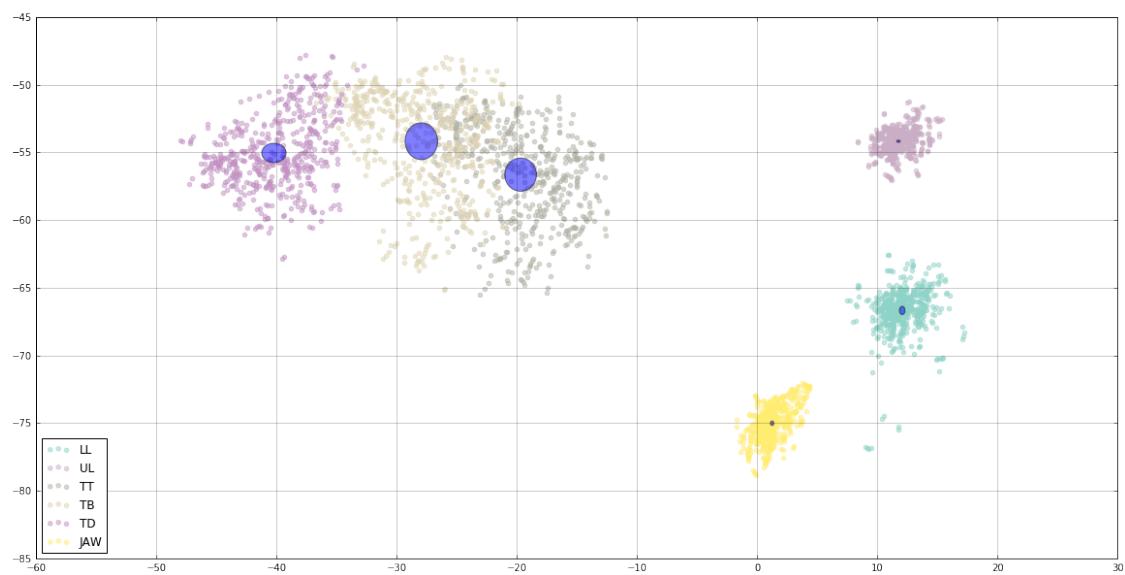
Gesture "t" (samples = 2084)



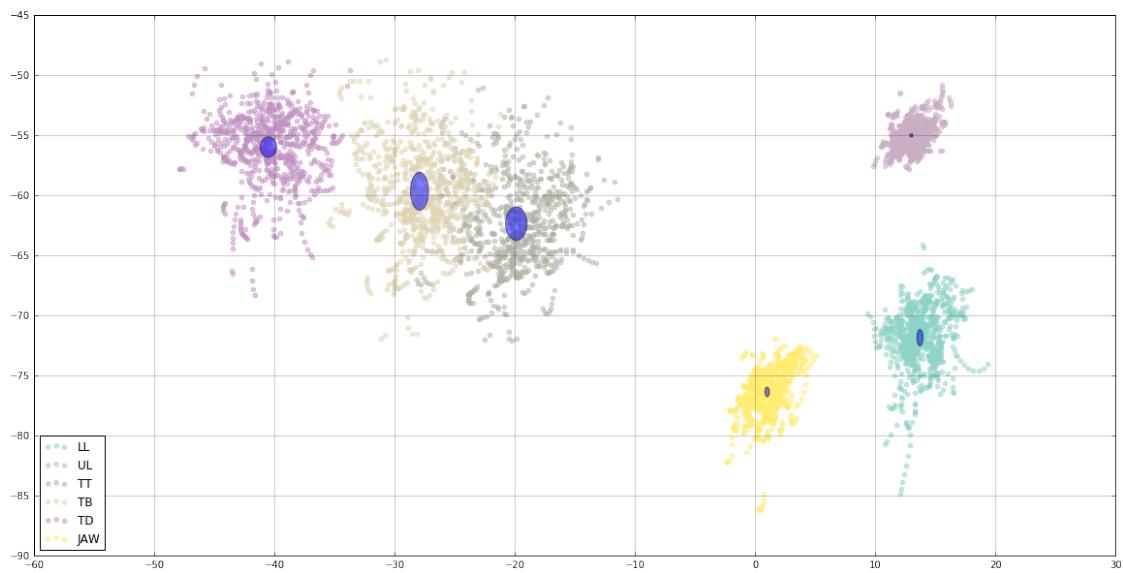
Gesture "w" (samples = 368)



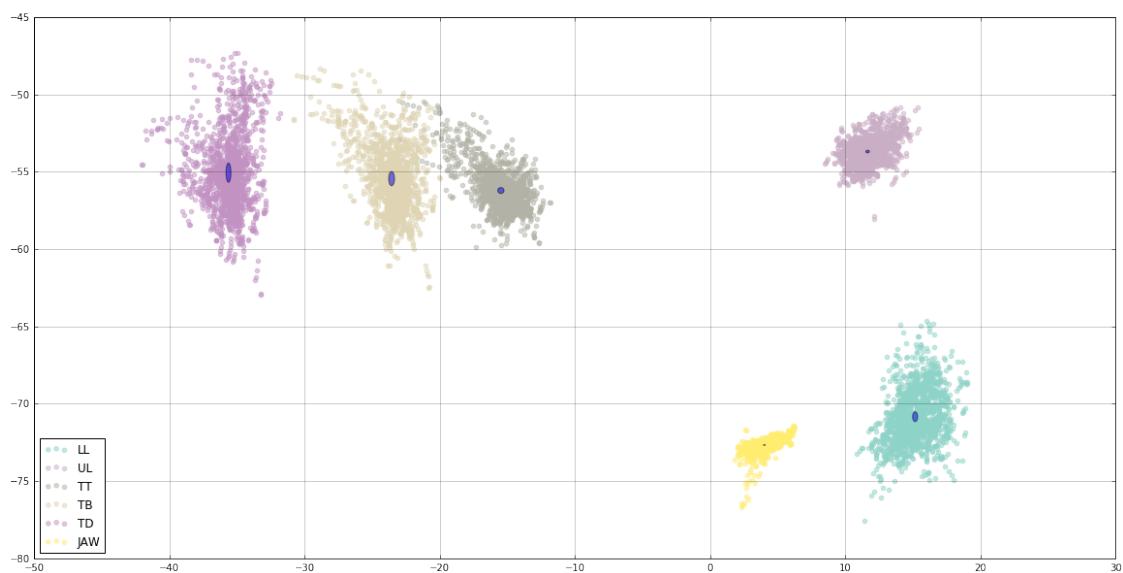
Gesture "v" (samples = 538)

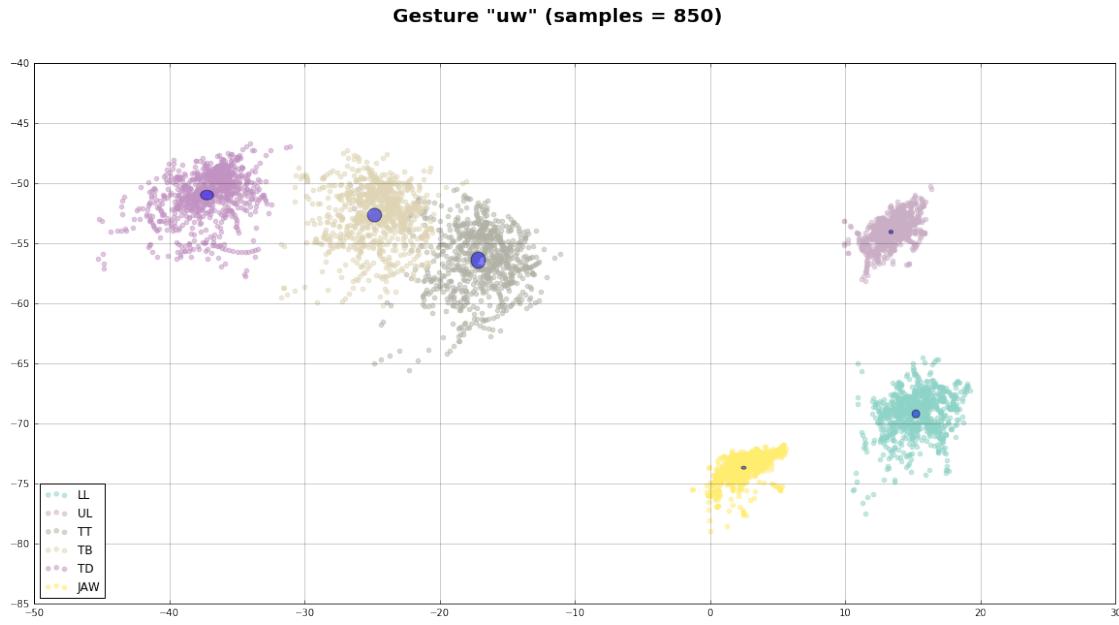


Gesture "ow" (samples = 820)



Gesture "z" (samples = 1570)





1.3 Normalize Gestures

```
In [11]: from matplotlib.patches import Ellipse

gestures, _, _ = tist.normalize_gestures(gestures)

articulators = ["LL", "UL", "TT", "TB", "TD", "JAW"]

cmap = plt.get_cmap('Set3')
colors = [cmap(i) for i in np.linspace(0, 1, len(articulators))]
for g_name, g in gestures.items():
#    f, (ax1, ax2) = plt.subplots(1, 2, sharey=True)
    fig1, ax = plt.subplots(1, len(articulators), sharey=True, figsize=(20, 3))
    fig1.suptitle("Gesture \"{}\" (samples = {})".format(g_name,
                                                          len(g.params["LL_x"])),
                  fontsize=20, fontweight='bold')

    g_m = g.get_mean()
    g_v = g.get_variance()

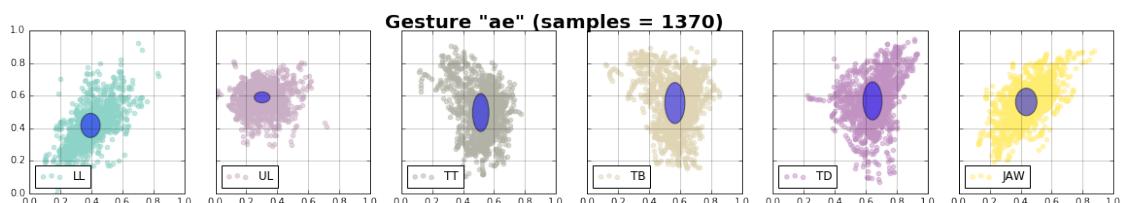
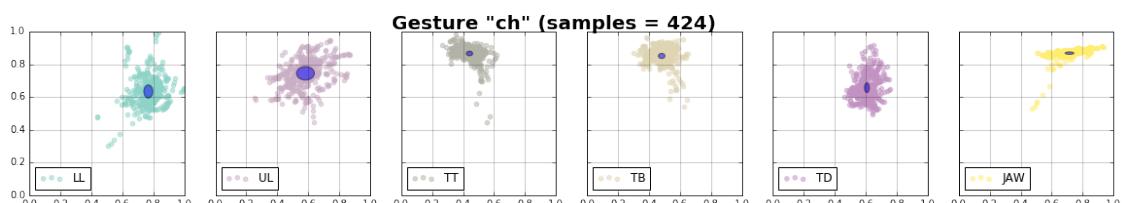
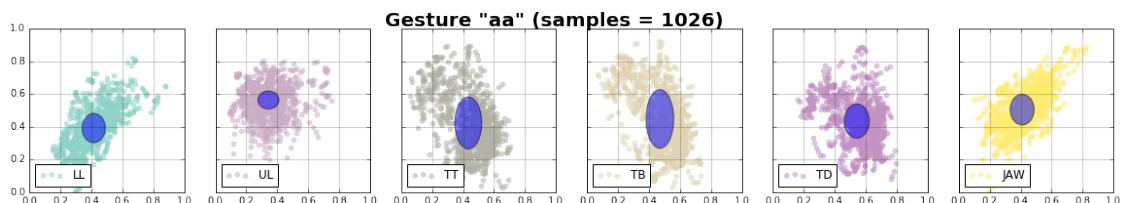
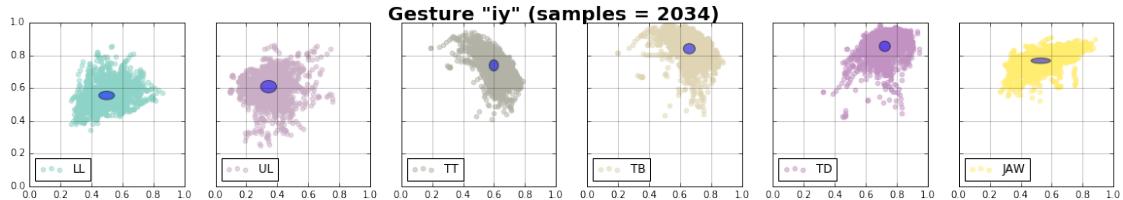
    for i in range(len(articulators)):
        ax[i].grid(color='black', linestyle='-', linewidth=1, alpha=0.2)
        ax[i].set_xlim(0, 1)
        ax[i].set_ylim(0, 1)
        a = articulators[i]
        a_x = g.params[a+"_x"]
        a_y = g.params[a+"_y"]
```

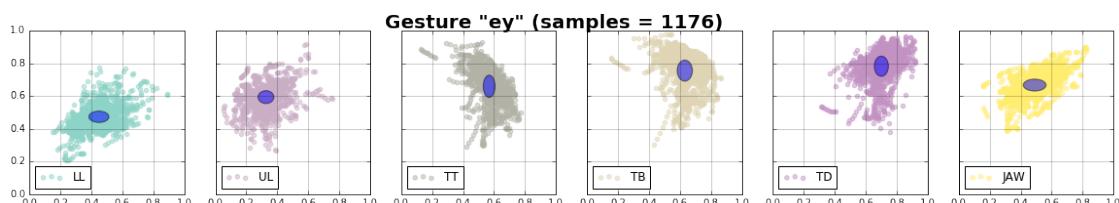
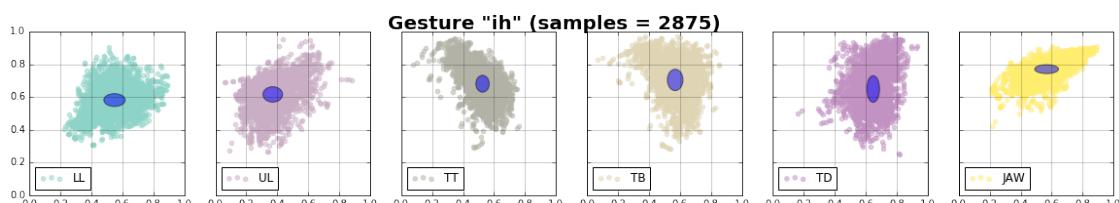
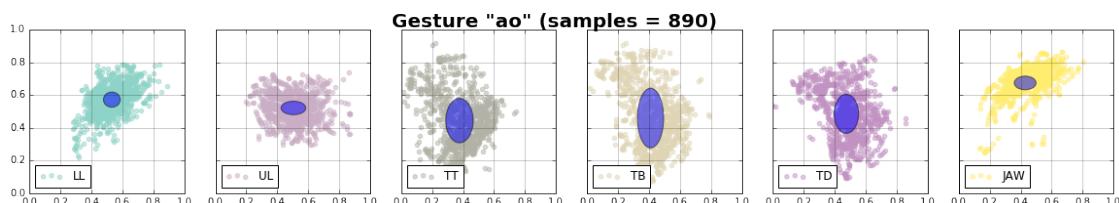
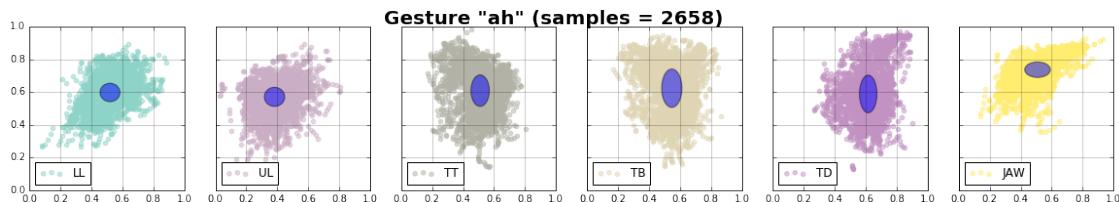
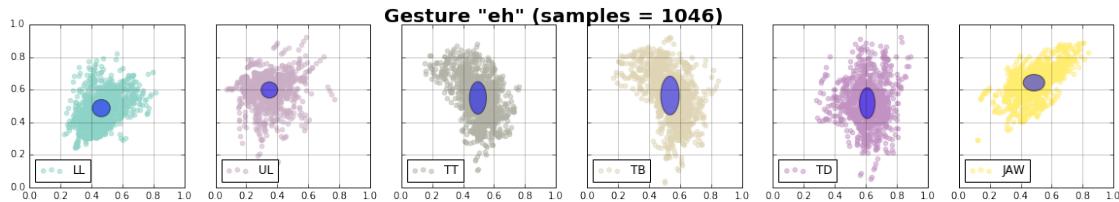
```

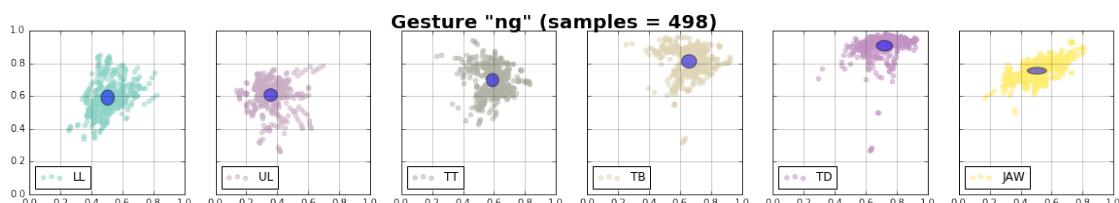
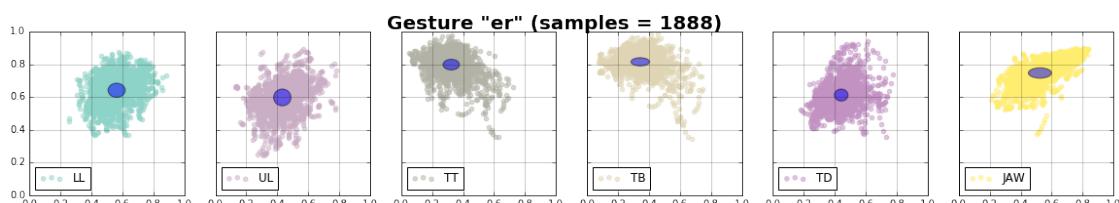
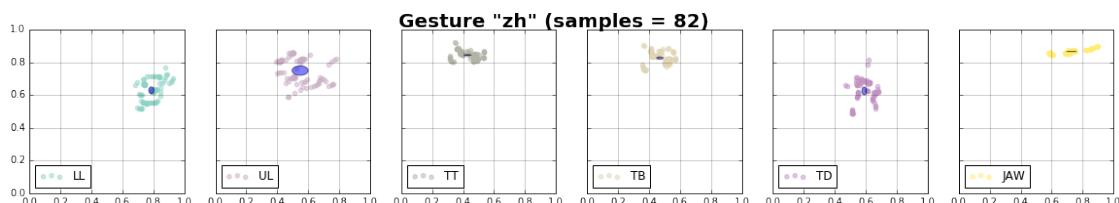
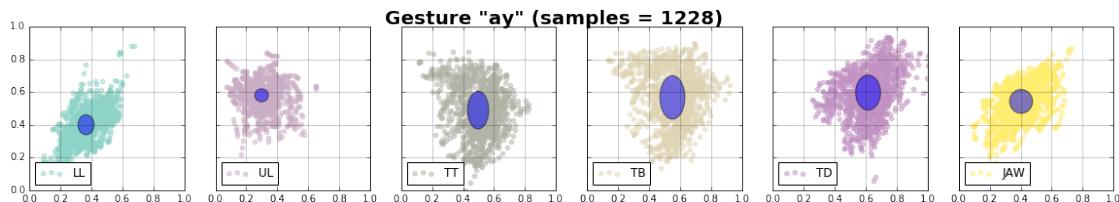
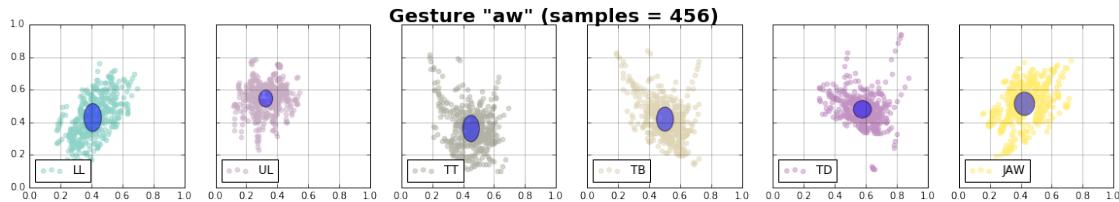
        ax[i].scatter(a_x, a_y, color=colors[i], alpha=0.5, label=a)
        # plot mean ellipse
        e = Ellipse(xy=[g_m[a+"_x"], g_m[a+"_y"]],
                    width=g_v[a+"_x"]*10, height=g_v[a+"_y"]*10, alpha=0.5)
        ax[i].add_artist(e)
        ax[i].legend(loc='lower left')

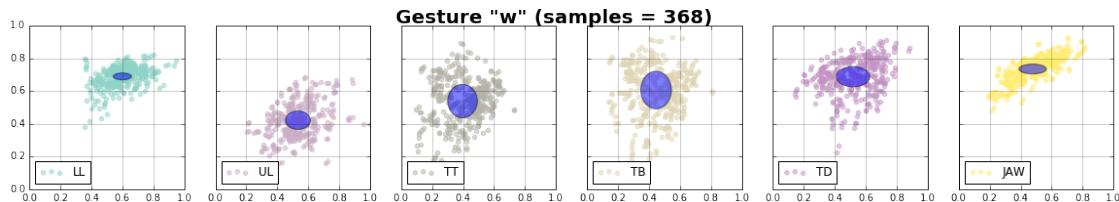
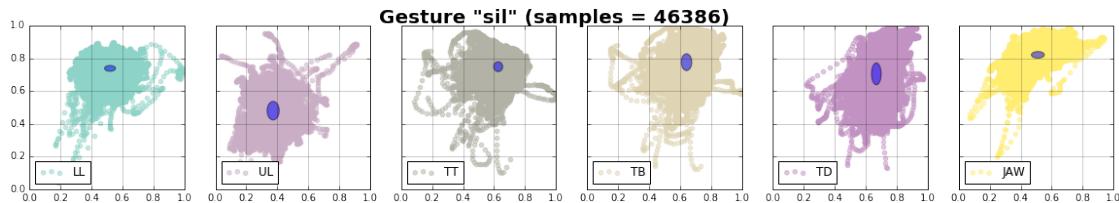
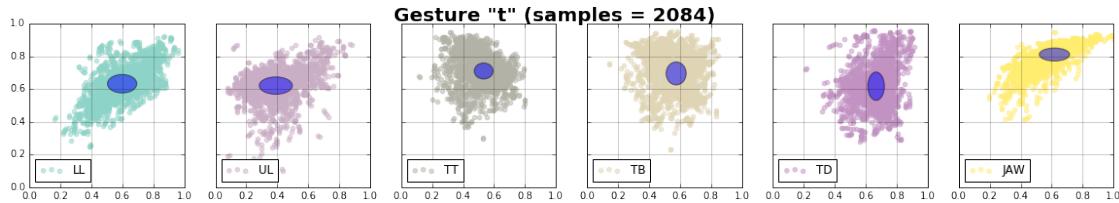
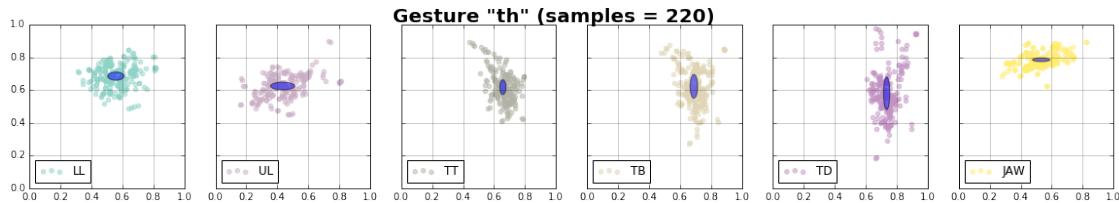
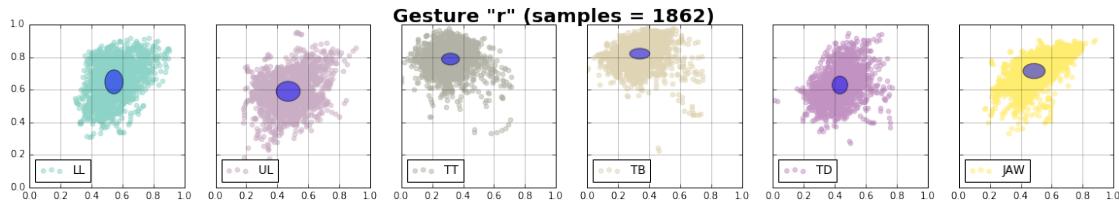
plt.show()

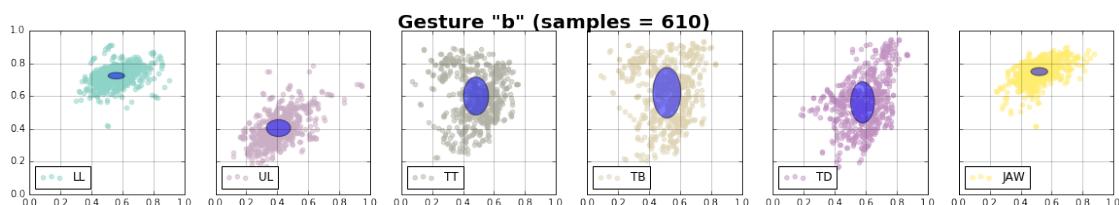
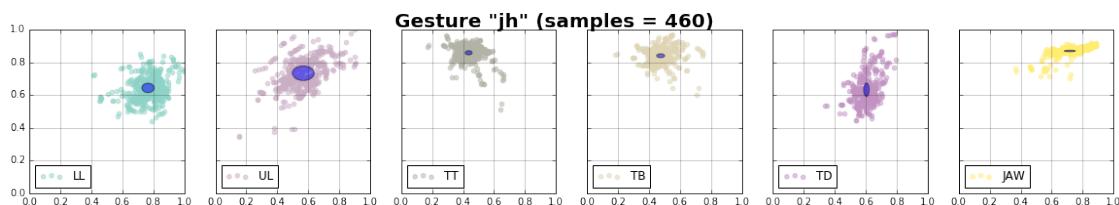
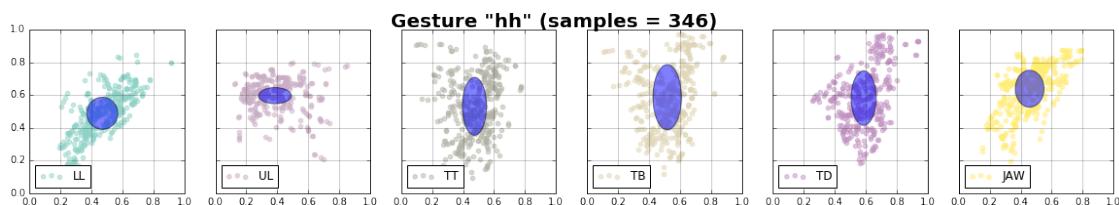
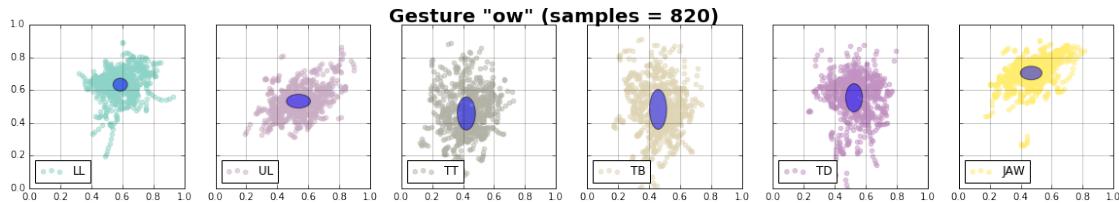
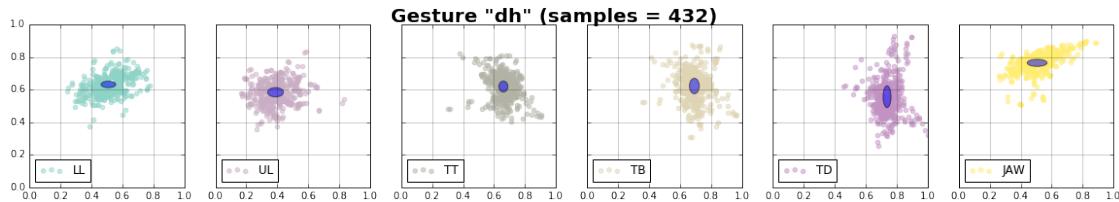
```

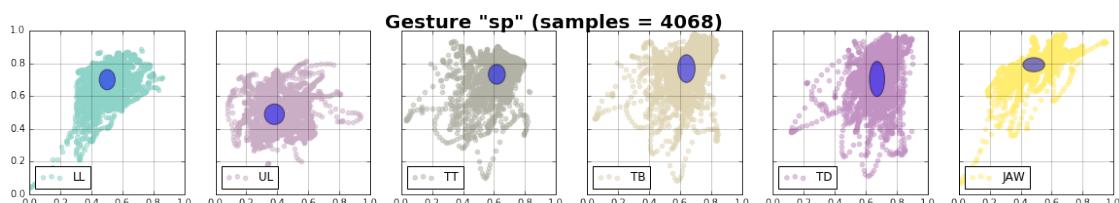
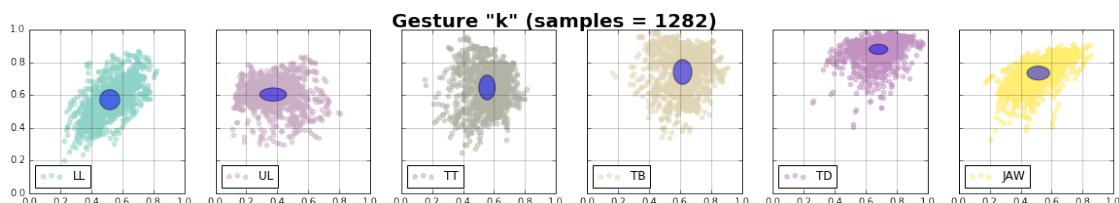
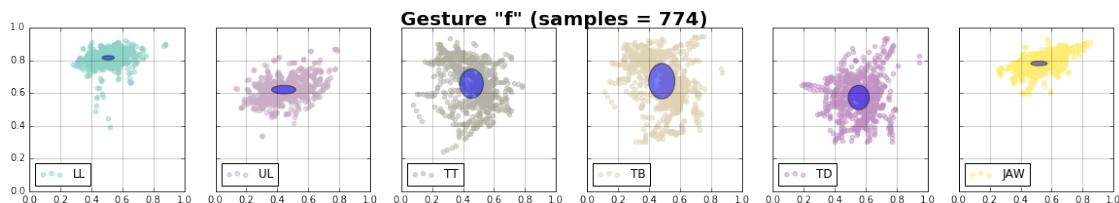
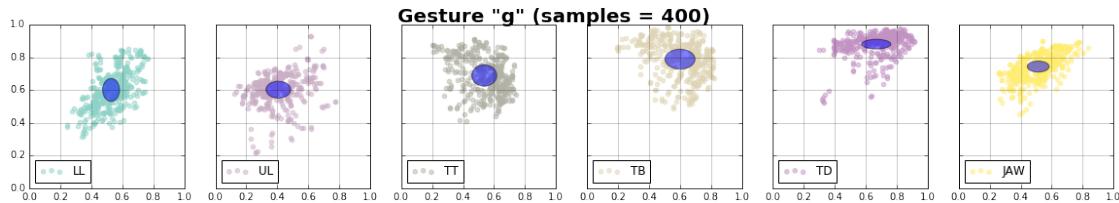
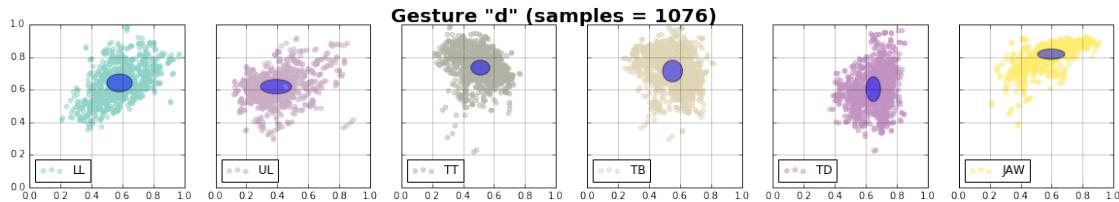


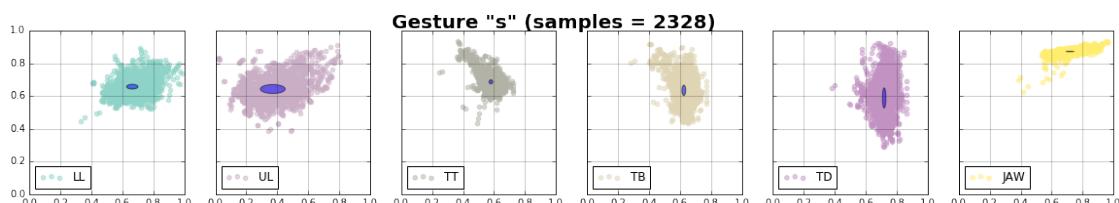
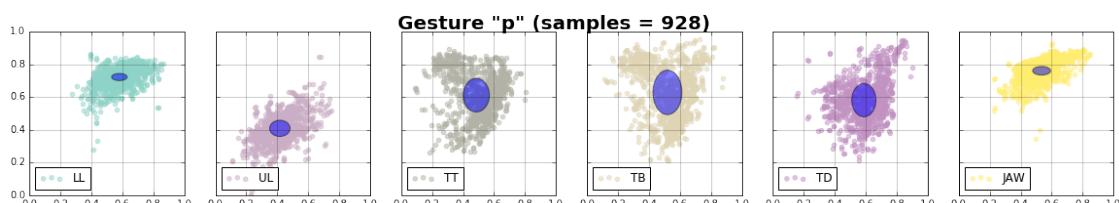
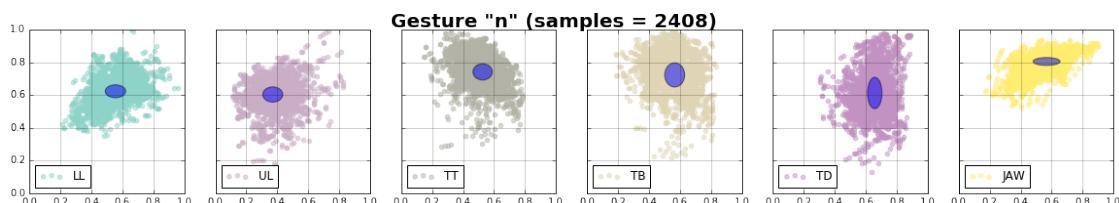
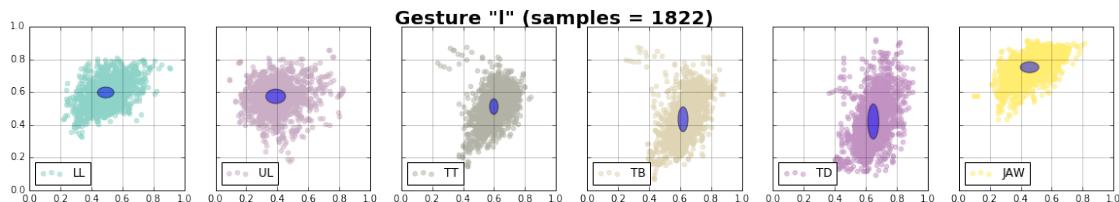
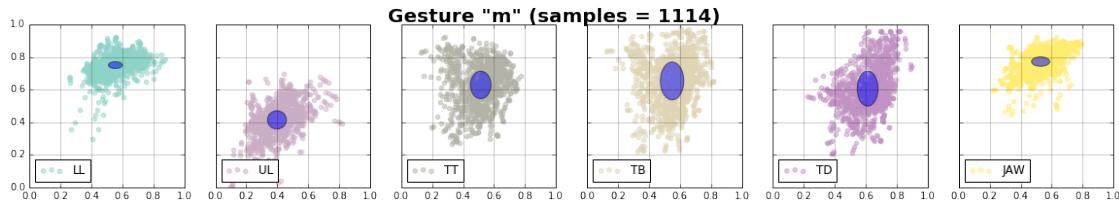


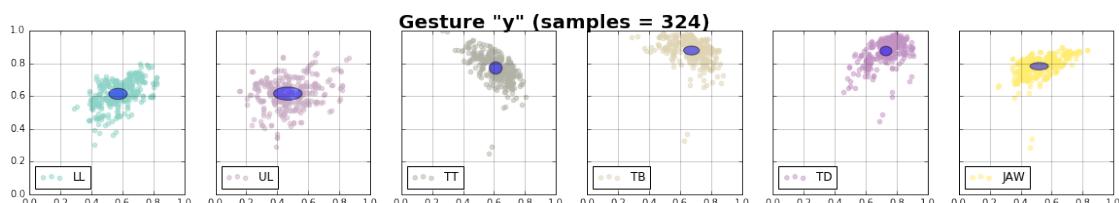
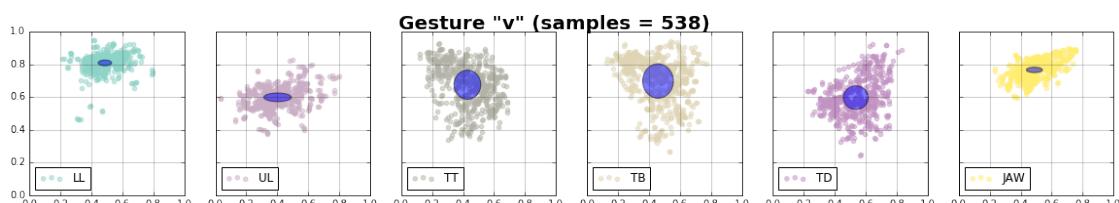
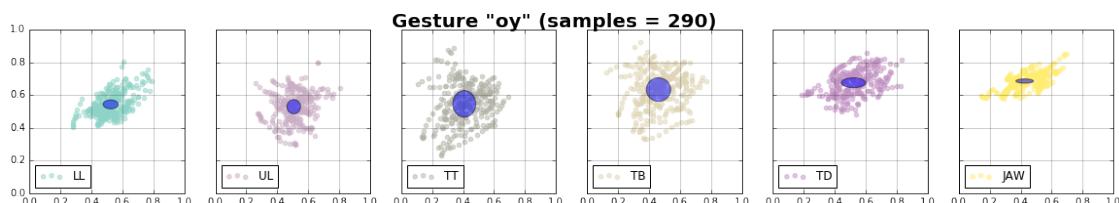
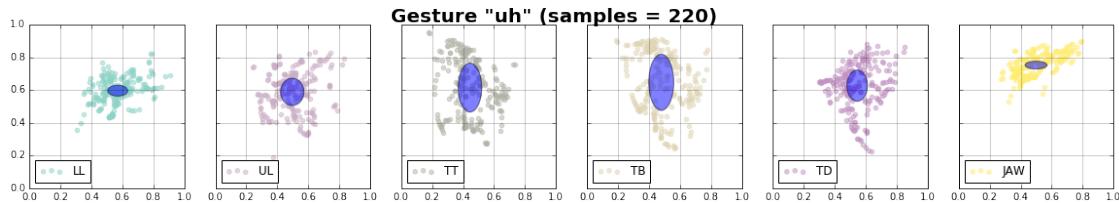
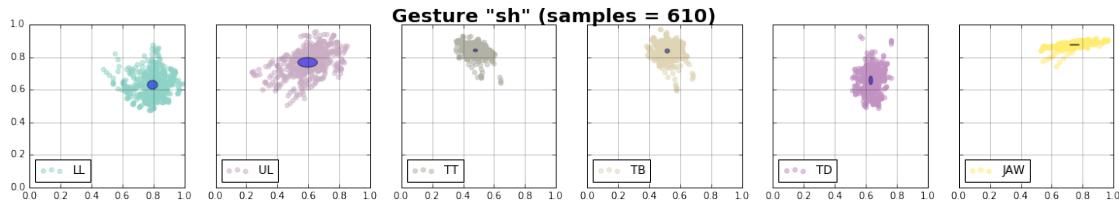


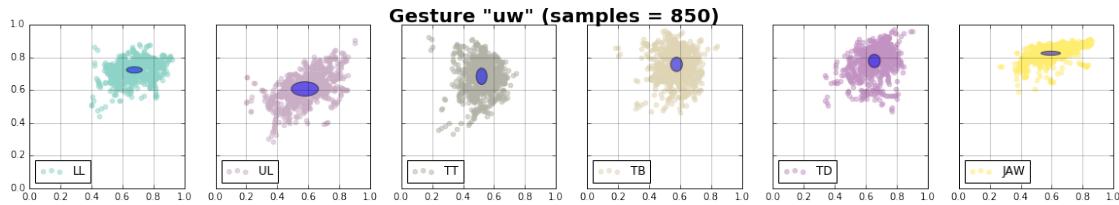
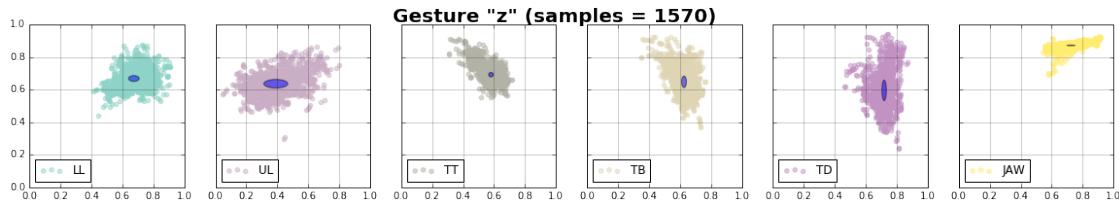












In []: