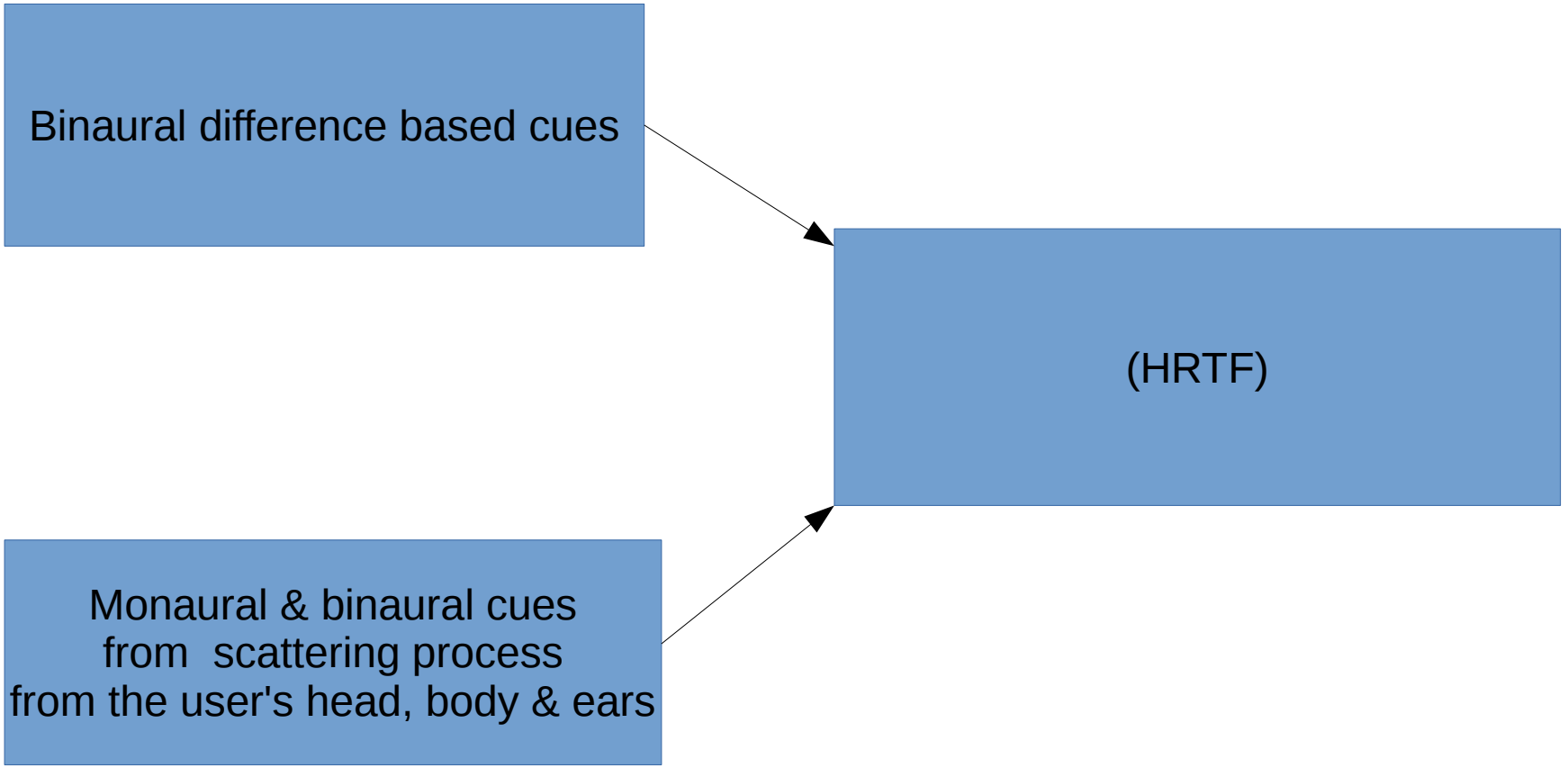
HRTF based binaural rendering with head tracking

-Pradeep Kumar Govindaraju

Head Related Transfer Function



HRTF – Dependency on source direction & frequency

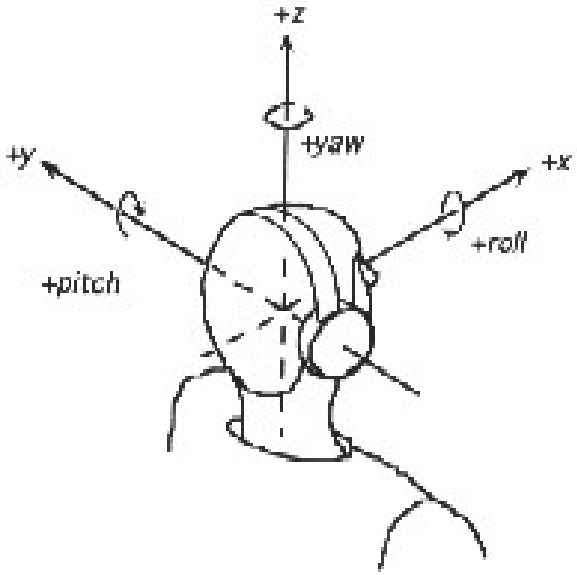
- Relatively distant sources – weaker dependence on distance of sound source – ignored.

- HRTFs H_l and H_r are defined as the frequency-dependent ratio of the SPL at the corresponding eardrum Φ_l , Φ_r to the free-field SPL at the center of the head as if the listener is absent Φ_f

$$H_l(\omega, \varphi, \theta) = \frac{\Phi_l(\omega, \varphi, \theta)}{\Phi_f(\omega)}, \quad H_r(\omega, \varphi, \theta) = \frac{\Phi_r(\omega, \varphi, \theta)}{\Phi_f(\omega)}.$$

Rendering Algorithm - Head Tracker

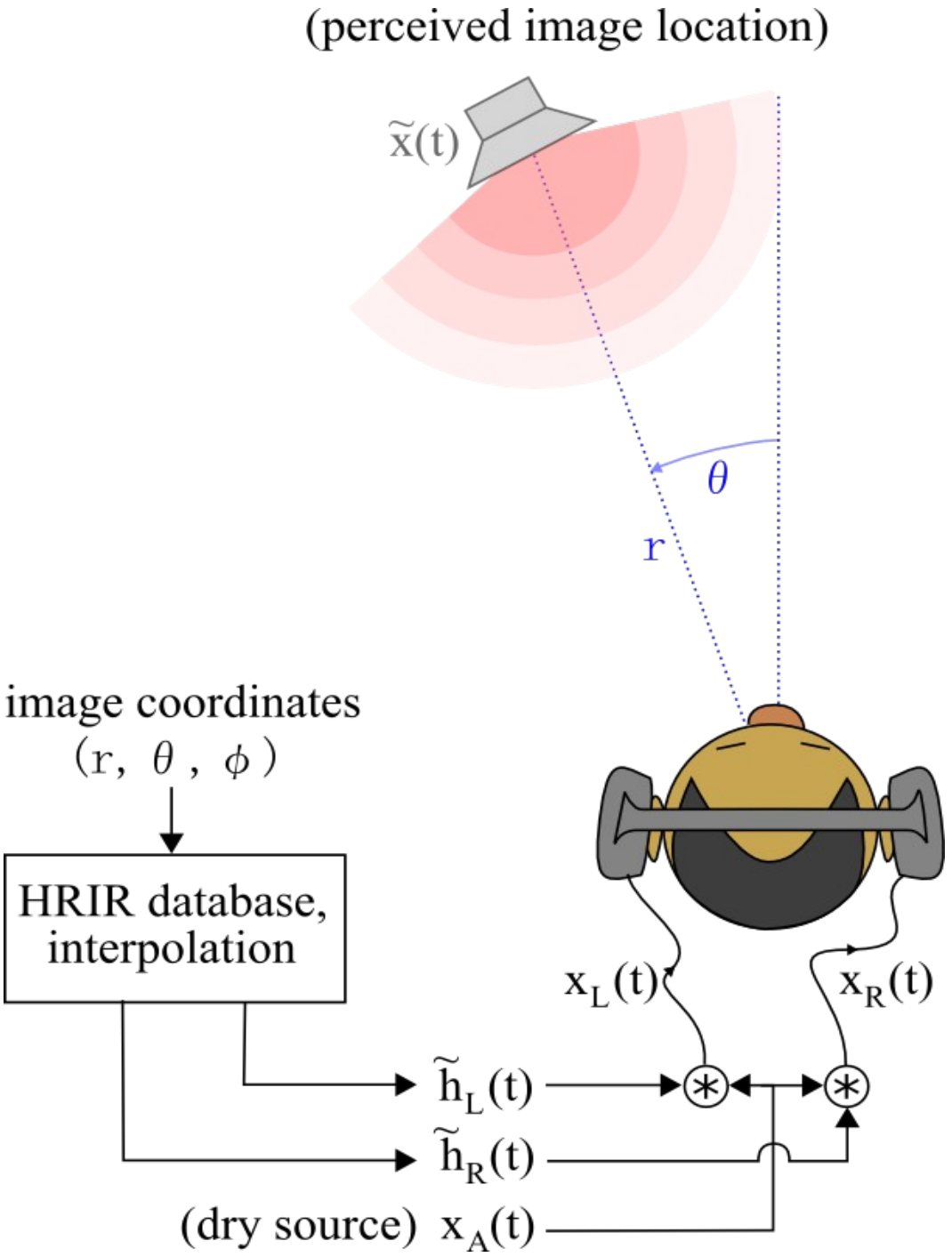
$$R = \begin{bmatrix} \cos\varphi\cos\theta & \cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi & \cos\varphi\sin\theta\cos\psi + \sin\varphi\sin\psi \\ \sin\varphi\cos\theta & \sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi & \sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi \\ -\sin\theta & \cos\theta\sin\psi & \cos\theta\cos\psi \end{bmatrix}.$$



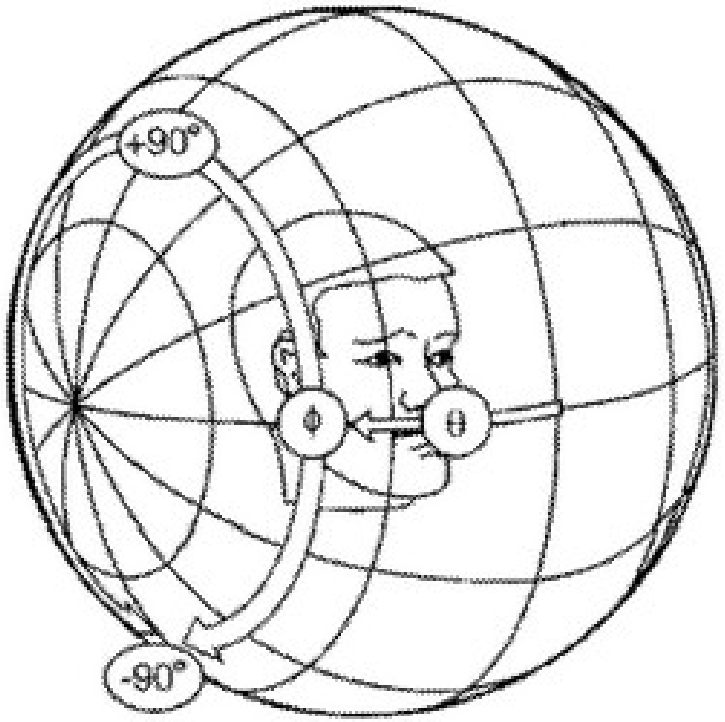
$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R^{-1} \begin{bmatrix} X - X_r \\ Y - Y_r \\ Z - Z_r \end{bmatrix}$$

$$\theta' = \arcsin(\frac{Y'}{R'}), \varphi' = \arcsin(\frac{X'}{R'\cos\theta'}), R'^2 = X'^2 + Y'^2 + Z'^2$$

Rendering Algorithm –Convolution

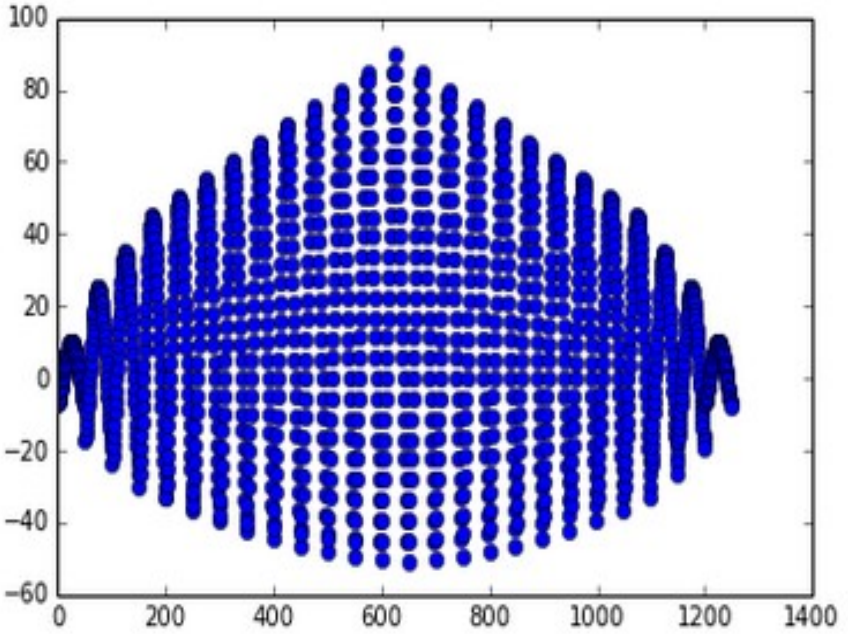


Interpolation of HRIR



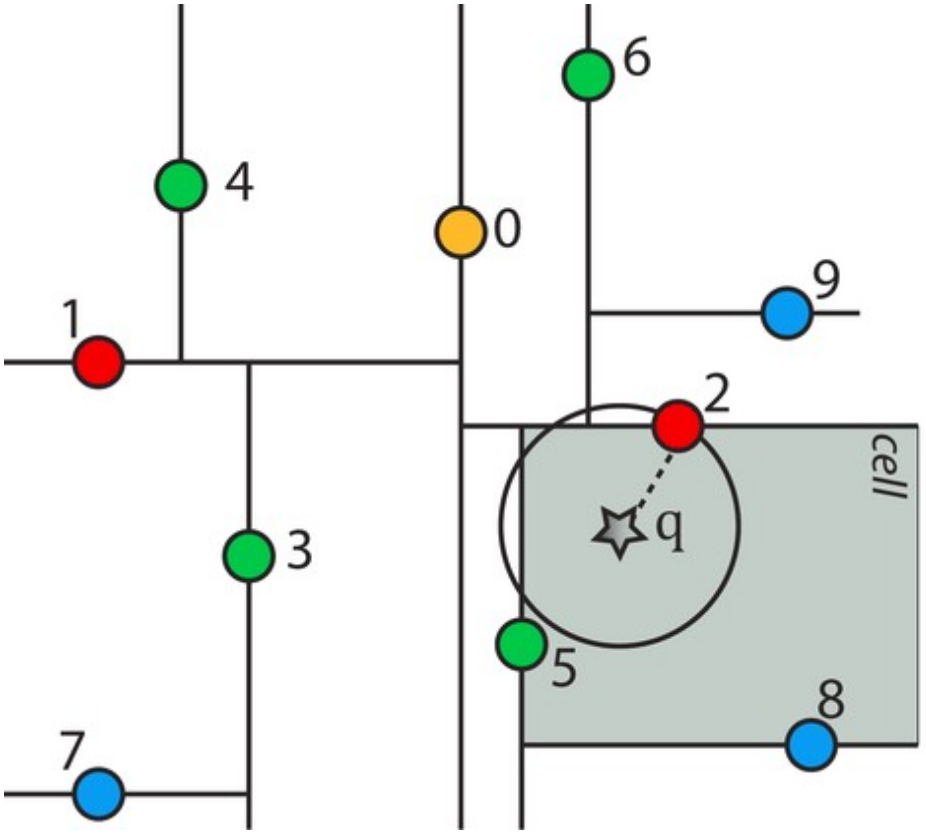

```
plot(f.variables['SourcePosition'][:][:, 1], 'o')
```

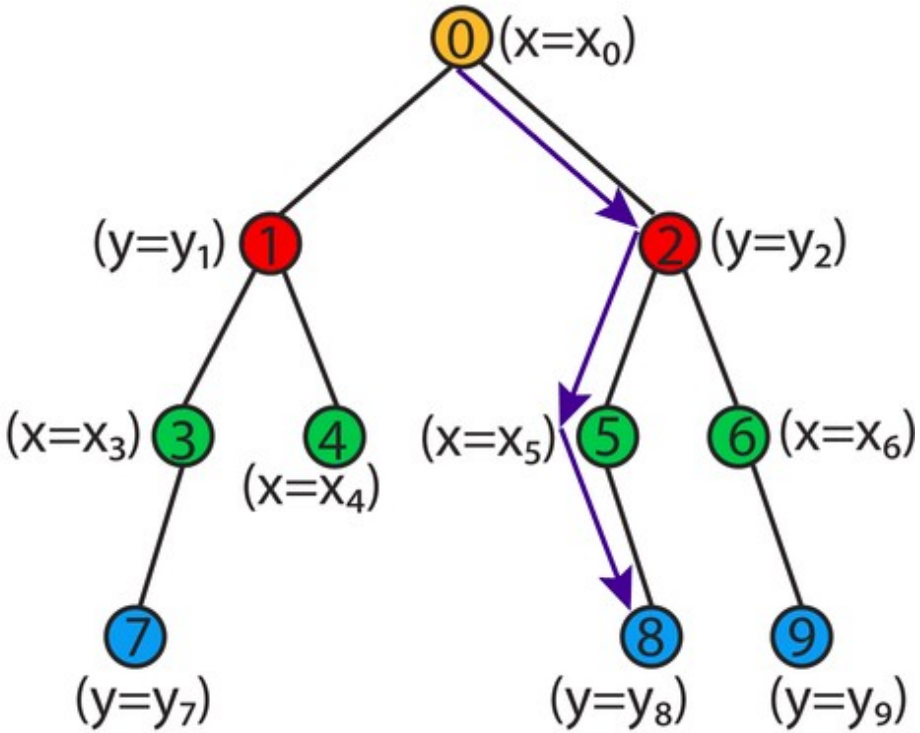
```
[<matplotlib.lines.Line2D at 0x7f179ebfeb50>]
```



Binary Space Partition – KD Trees

- To find 3 nearest neighbor HRTF points in the database





Screenshot

Original Source Position

Source Position – compensated for head rotation