A high-throughput stream-oriented GPU environment for simultaneous processing and visualization of EEG measurements

Zoltan Juhasz
Dept. of Electrical Engineering and Information Systems
University of Pannonia, Veszprem, Hungary

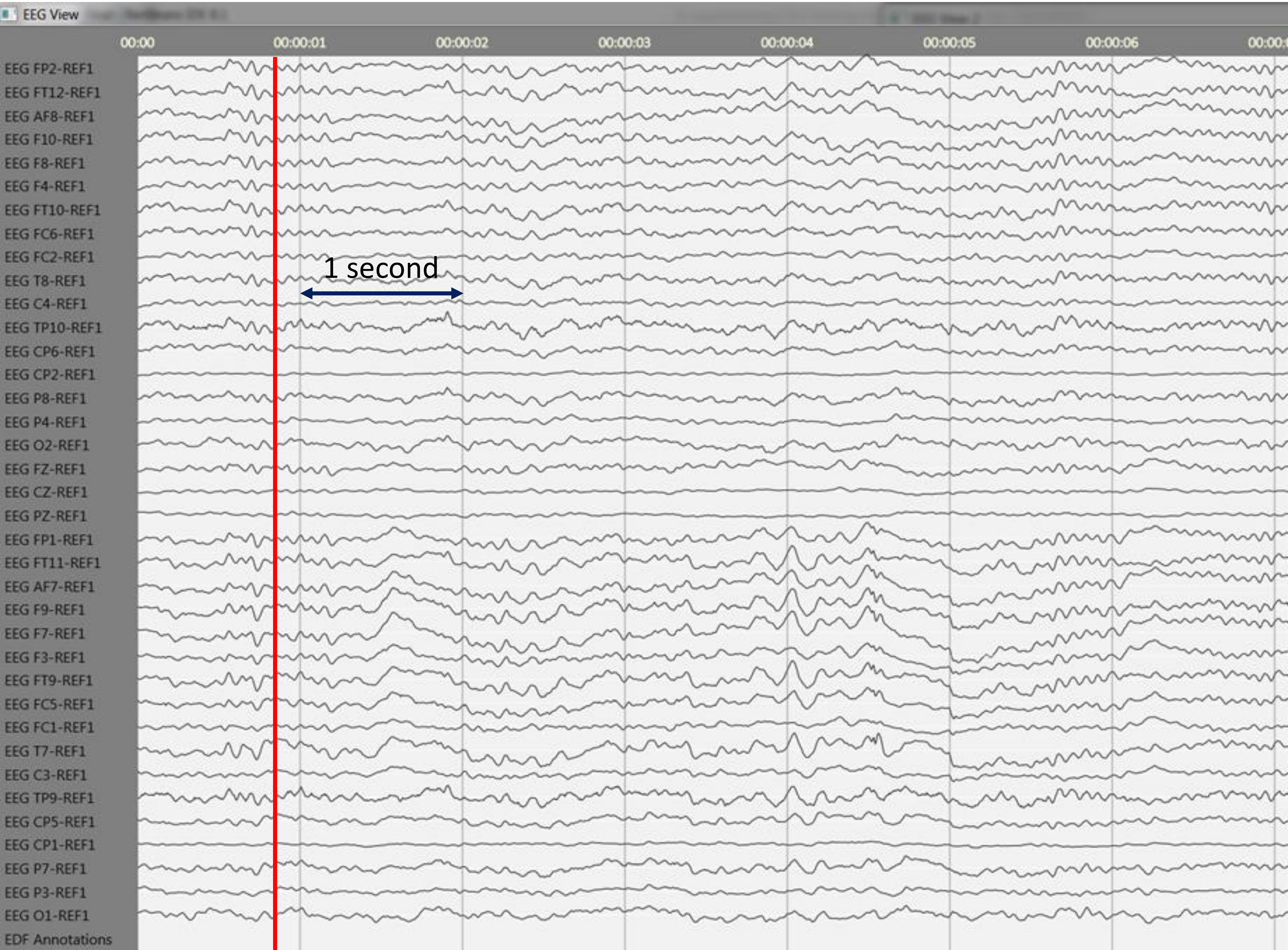
High resolution EEG imaging

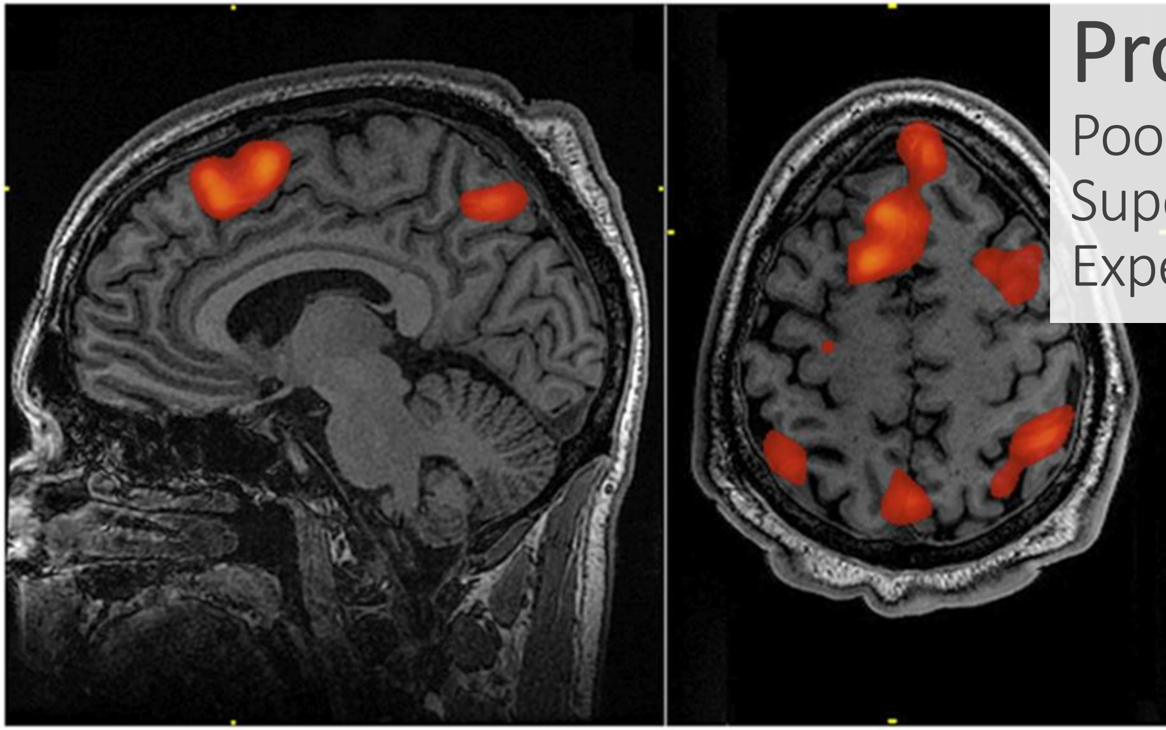
EEG fundamentals

- Cortical neurons as current sources
- The electrical field generates potentials on the scalp

Primary tool in

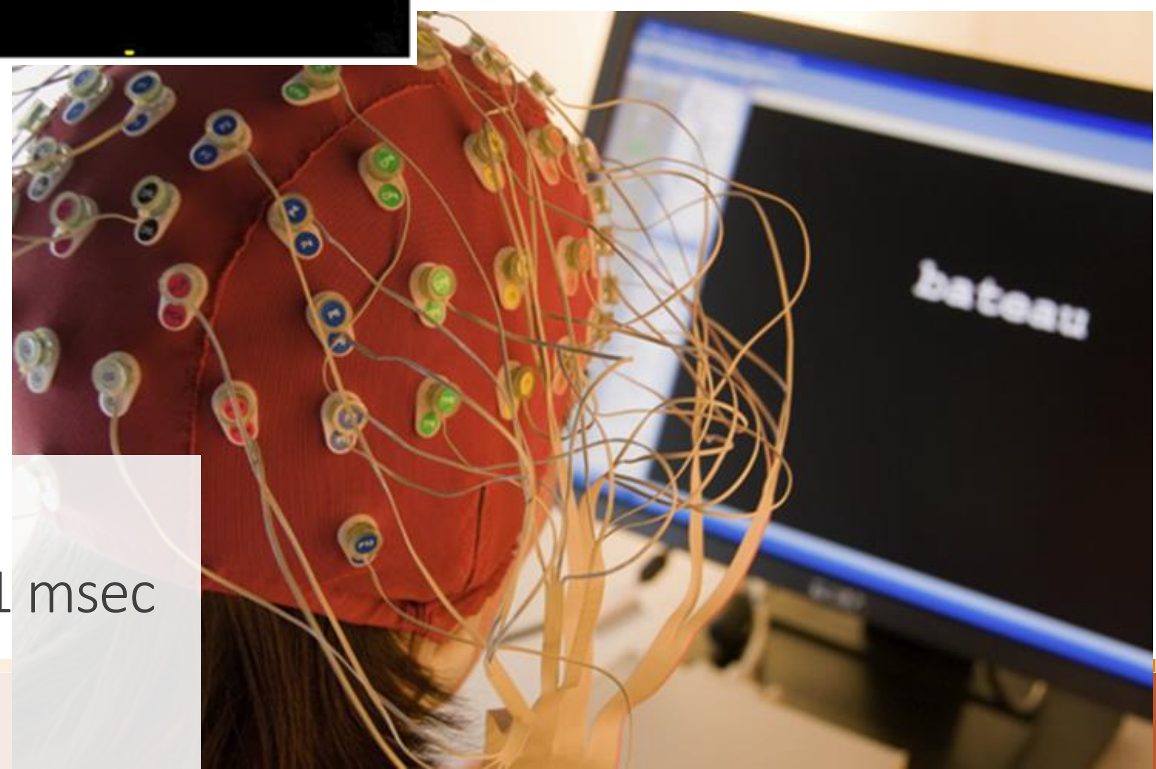
- epilepsy diagnosis and treatment,
- neurocognitive studies


EEG View
00:00
00:00:01
00:00:02
00:00:03
00:00:04
00:00:05
00:00:06
EEG FP2-REF1
EEG FT12-REF1
EEG AF8-REF1
EEG F10-REF1
EEG F8-REF1
EEG F4-REF1
EEG FT10-REF1
EEG FC6-REF1
EEG FC2-REF1
EEG T8-REF1
EEG C4-REF1
EEG TP10-REF1
EEG CP6-REF1
EEG CP2-REF1
EEG P8-REF1
EEG P4-REF1
EEG O2-REF1
EEG FZ-REF1
EEG CZ-REF1
EEG PZ-REF1
EEG FP1-REF1
EEG FT11-REF1
EEG AF7-REF1
EEG F9-REF1
EEG F7-REF1
EEG F3-REF1
EEG FT9-REF1
EEG FC5-REF1
EEG FC1-REF1
EEG T7-REF1
EEG C3-REF1
EEG TP9-REF1
EEG CP5-REF1
EEG CP1-REF1
EEG P7-REF1
EEG P3-REF1
EEG O1-REF1
EDF Annotations
1 second



Properties of fMRI

Poor temporal resolution > sec
Superb spatial resolution ~ 1 mm
Expensive



Properties of EEG

Superb temporal resolution < 1 msec
Poor spatial resolution > 2 cm
Inexpensive

Main goals

Use for imaging processing and visualization

Fast execution how fast?

High spatial resolution close to MRI/fMRI

Functional mapping classify activation source

Novel architecture dataflow/stream processing – high throughput

Fundamentals – Head model

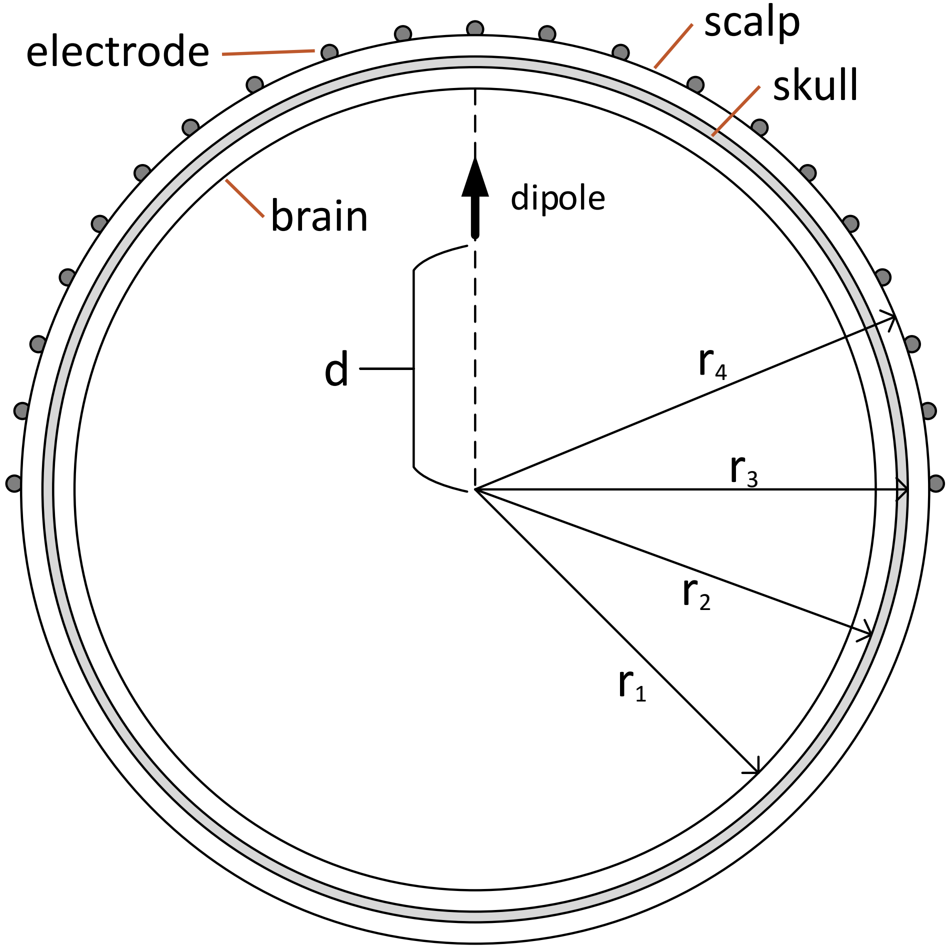
Simplified head geometry for computation

Various head models known

- Spherical, elliptical
- Realistic (segmented from MRI)
- 3, 4 or 5 layers (brain, skull, scalp)

Various forward solvers

- analytical (spherical model)
- BEM, FEM (realistic models)



EEG imaging

Generate maps of potential and derived values **on the scalp** or **on the cortex**

Image on scalp

- Volume conduction of skull blurs the source image
- Requires large number of electrodes
- **Laplacian** can increase spatial resolution (high-resolution EEG)

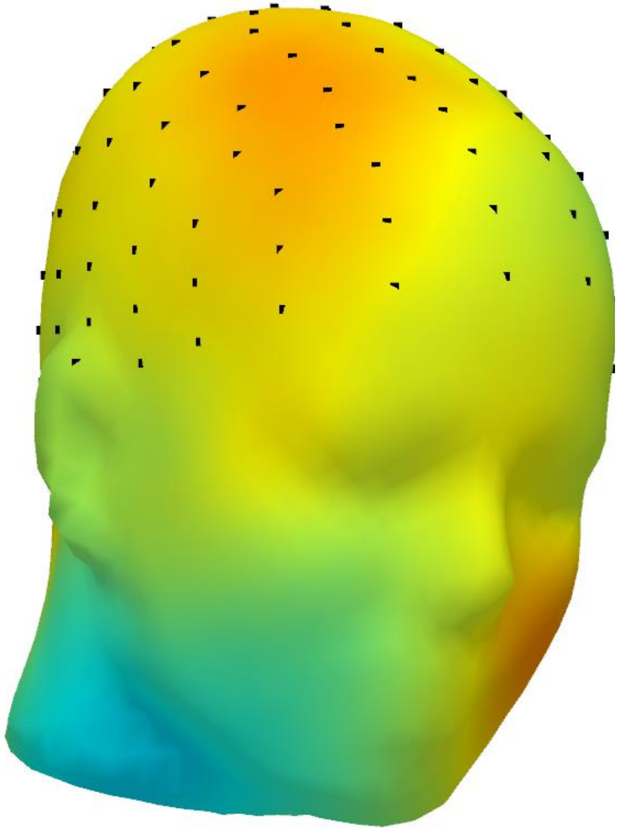
Image on the cortex

- Cortical imaging (back projection from scalp potential image)
- Intracranial electrodes (electrodes placed on cortex – invasive /ouch!/)
- Source localisation (determine sources from measured potential – **inverse problem**)

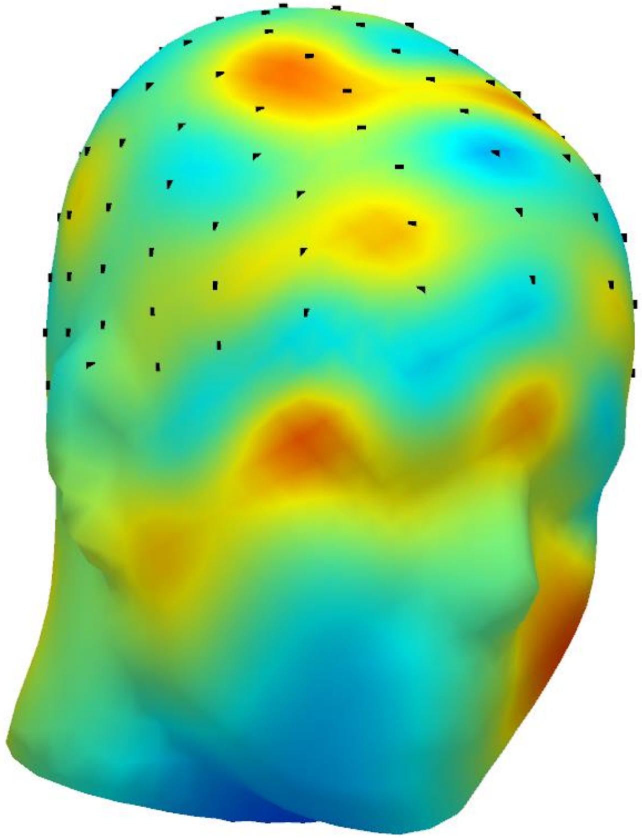
Surface Laplacian

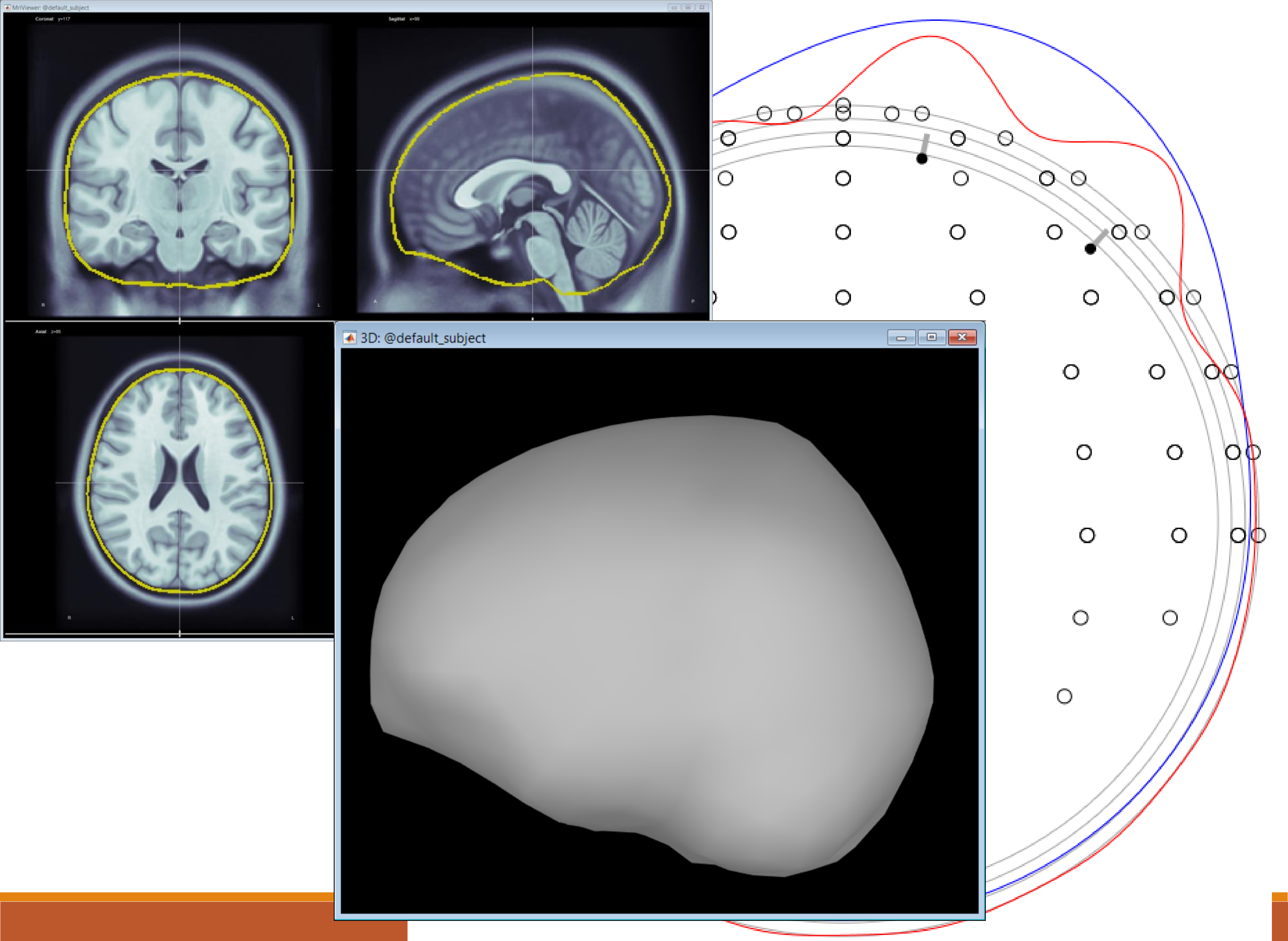
Normally computed at the electrode positions only

Scalp potential map



Scalp surface Laplacian – current source density

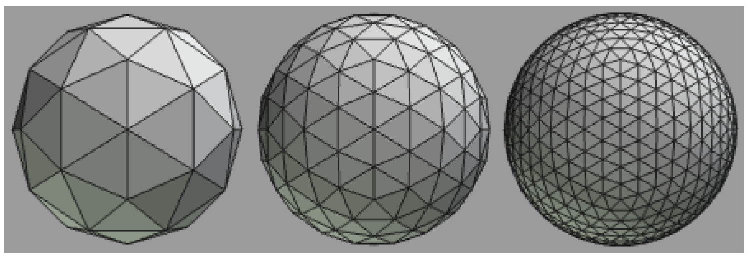



MriViewer: @default_subject
Coronal: y=117
Sagittal: x=99
Axial: z=95
3D: @default_subject

Cortical image spatial resolution

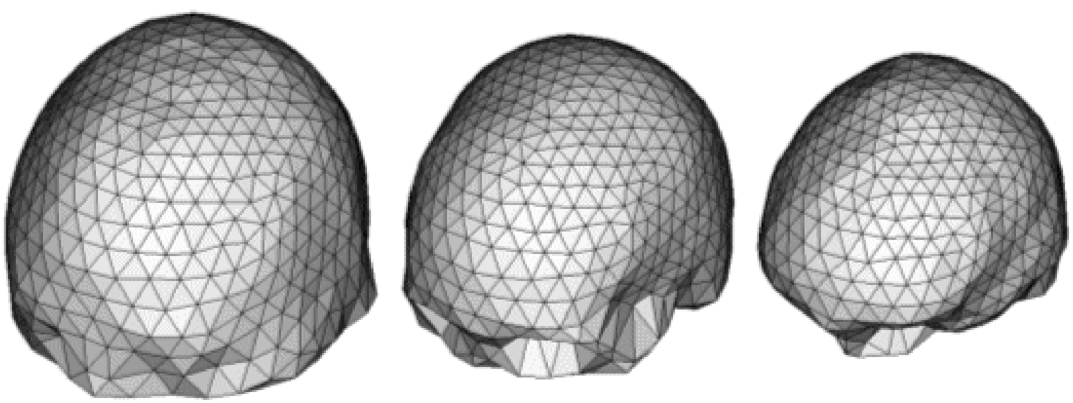
Polygon mesh: required vertex count vs spatial resolution

Icosphere mesh, r=80mm



ICOSPHERE LEVEL	VERTEX COUNT	AVG PATCH AREA [mm²]	APPROX PATCH DIAMETER [mm]
2	936	85.92	10.460
3	3816	21.08	5.180
4	15336	5.24	2.584
5	61416	1.31	1.291
6	245736	0.33	0.646

Realistic mesh



Scalp Outer skull Inner skull

VERTEX COUNT	AVG PATCH AREA [mm²]	APPROX PATCH DIAMETER [mm]
362	222.17	16.819
812	99.05	11.230
1082	74.33	9.728
1922	41.84	7.299
2432	33.07	6.489
3242	24.81	5.620
5762	13.96	4.216
7292	11.03	3.747
10242	7.85	3.162
12962	6.20	2.811

Spherical surface Laplacian pseudo code

compute A^{-1}

for each time sample K
    compute coefficients **C**
    for each surface point S
        for each electrode E_i
            $lap \leftarrow lap + c_i h\big(cos(S, E_i)\big)$
    ...

$LAP(S) = \sum_{i=1}^{n} c_i h\big(cos(S, E_i)\big)$

$\mathbf{GC} + \mathbf{T}c_0 = \mathbf{Z}$ $\mathbf{Z}$ = potentials

$\mathbf{T'C} = 0$

$$\begin{bmatrix} \boldsymbol{G} & \boldsymbol{T} \\ \boldsymbol{T'} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{C} \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{Z} \\ 0 \end{bmatrix} \rightarrow \quad Ax = b$$

$x = A^{-1}b$ gives $\boldsymbol{C}$ for each $\boldsymbol{Z}$

$$h(x) = -\frac{1}{4\pi}\sum_{n=1}^{\infty}\frac{(2n+1)}{n^{m-1}(n+1)^{m-1}}P_n(x)$$

$$g(x) = \frac{1}{4\pi}\sum_{n=1}^{\infty}\frac{(2n+1)}{n^{m}(n+1)^{m}}P_n(x)$$

Pseudo code

FORWARD STEP

```
for each time sample k in K do                  // N
    for each electrode e in E do                // 128
        for each dipole d in D do               // approx. 10K
            pot ← pot + compute potential at e generated by d
```

INVERSE STEP

find single dipole source **using optimisation** (location and moment)

or

find weights of a set of fixed dipoles **using linear MNE**

Baseline execution time results per sample

	Matlab 1-core	Java – CPU
Spherical Laplacian		
• 512 vertices	23.03 sec	77.22 msec
• 1024 vertices	23.15 sec	144.84 msec
• 5 120 vertices	24.40 sec	719.25 msec
• 32 768 vertices	-	4.56 sec
Forward solver (spherical)		
• 10 dipoles	0.95 sec	25.40 msec
• 100 dipoles	7.22 sec	28.60 msec
• 1 000 dipoles	70.03 sec	174.51 msec
• 10 000 dipoles	698.26 sec	1.42 sec

x 2048 for one second

CPU – intel Core i7-3820QM 2.7 GHz

CUDA implementation

Alternatives

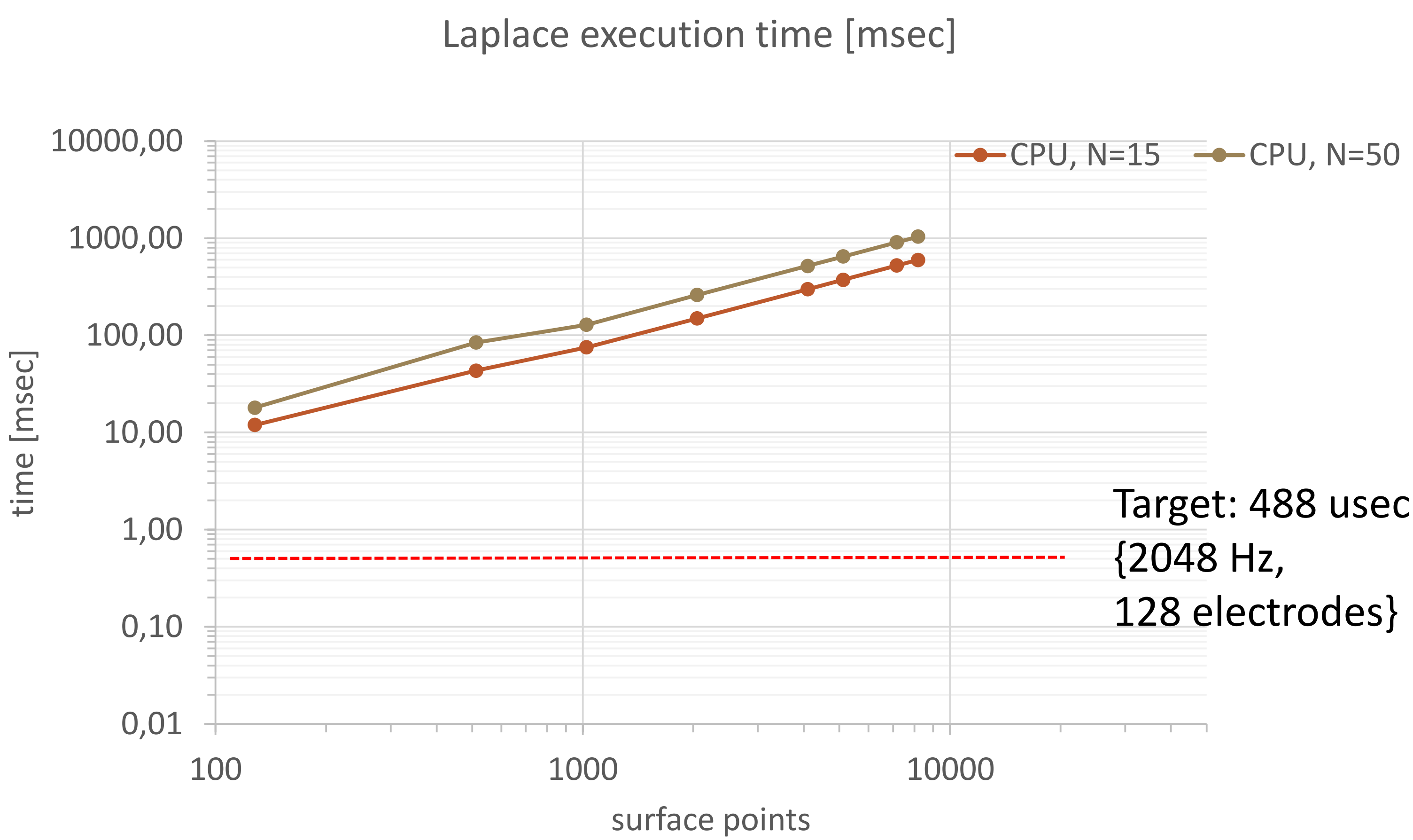
- OpenACC, OpenMP 4
- Parallel libraries
- OpenCL

CUDA – maximum freedom, ability to match architecture and algorithm, maximize performance, use advanced optimization tricks

C and various GPU algorithm variations,

C/GPU heterogeneous solution

CPU, Java, double precision, 1 core

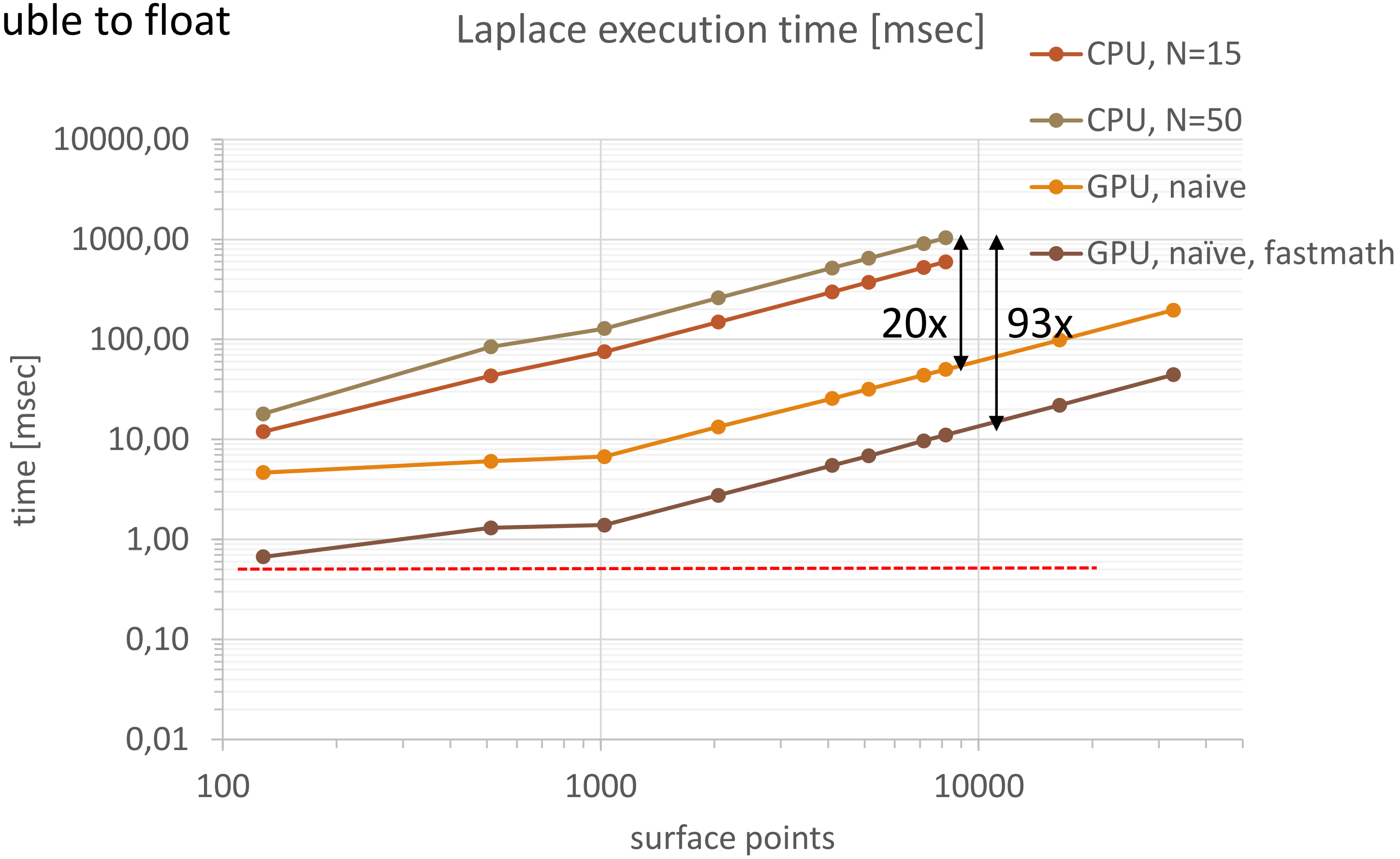


for each time sample K
  compute coefficients **C**
  for each surface point S
    for each electrode E_i
      $lap \leftarrow lap + c_i h\big(cos(S, E_i)\big)$

CUDA thread

$$h(x) = -\frac{1}{4\pi}\sum_{n=1}^{\infty}\frac{-(2n+1)}{n^{m-1}(n+1)^{m-1}}P_n(x)$$

From double to float



for each time sample K

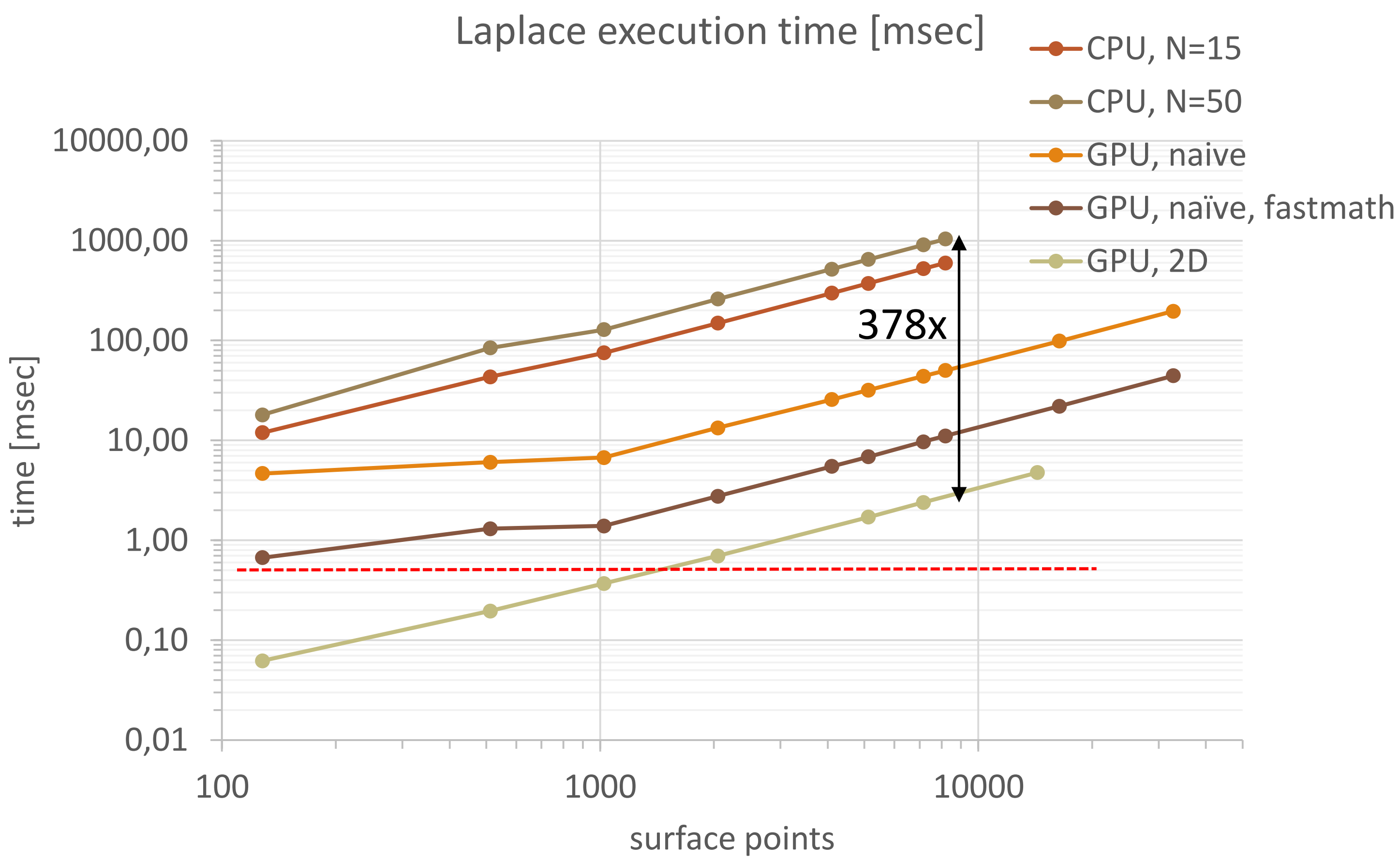
  compute coefficients **C**

  for each surface point S

    for each electrode E_i

      $lap \leftarrow lap + c_i h\left(cos(S, E_i)\right)$ CUDA thread

$$h(x) = -\frac{1}{4\pi}\sum_{n=1}^{\infty}\frac{-(2n+1)}{n^{m-1}(n+1)^{m-1}}P_n(x)$$



for each time sample K

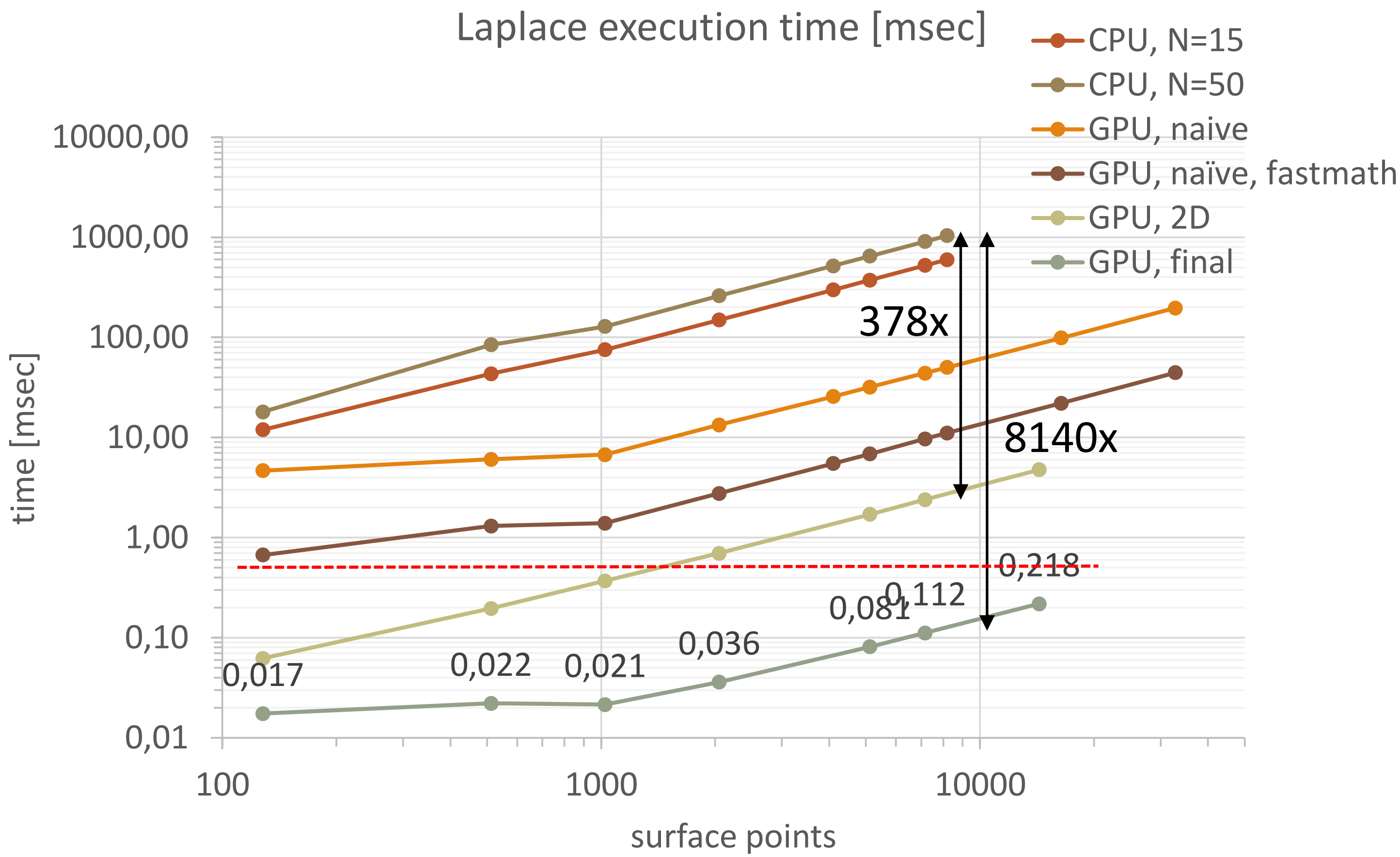
  compute coefficients **C**

  for each surface point S

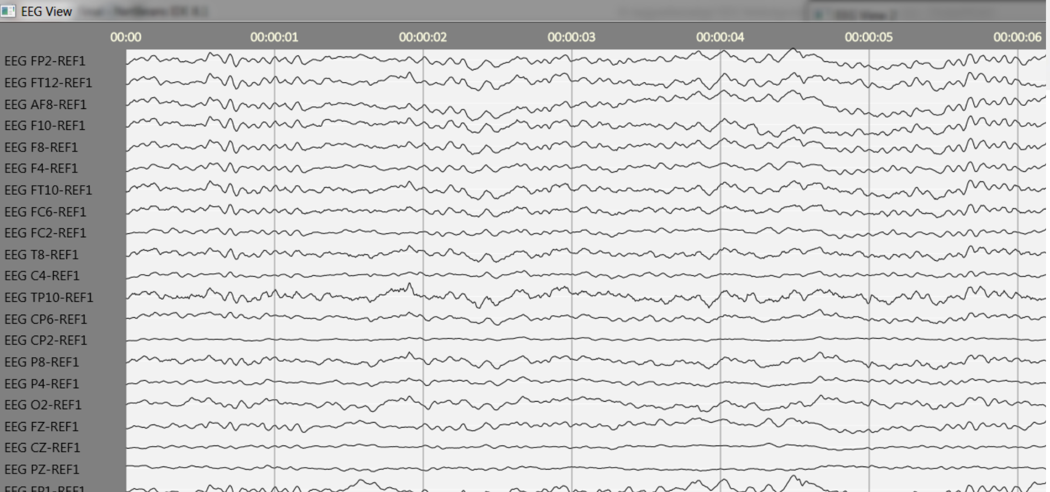
    for each electrode E_i

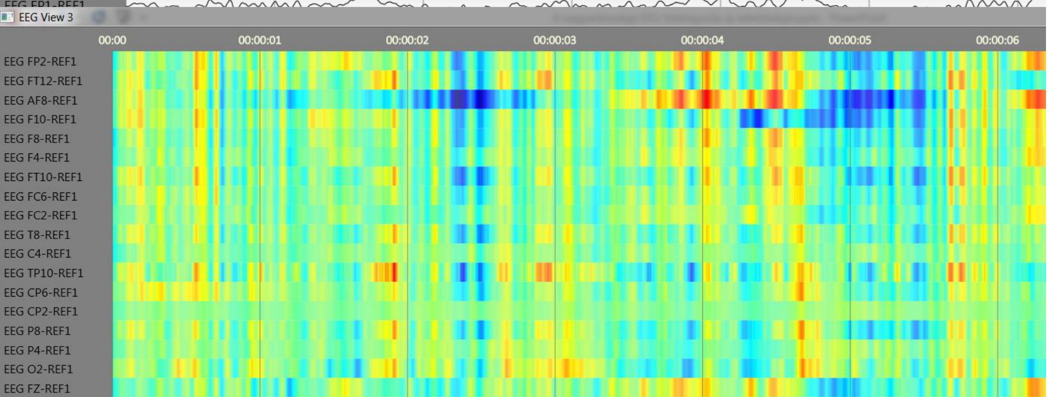
      $lap \leftarrow lap + c_i h(cos(S, E_i))$ CUDA thread

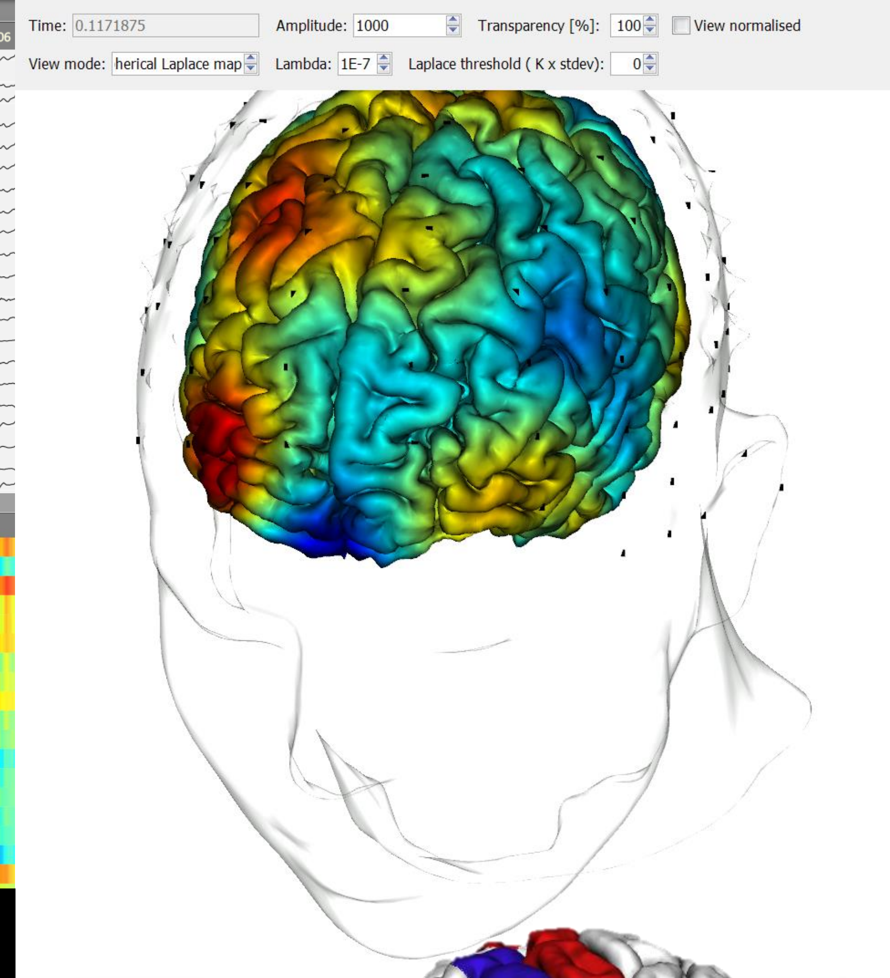
$$h(x) = -\frac{1}{4\pi}\sum_{n=1}^{\infty}\frac{-(2n+1)}{n^{m-1}(n+1)^{m-1}}P_n(x)$$

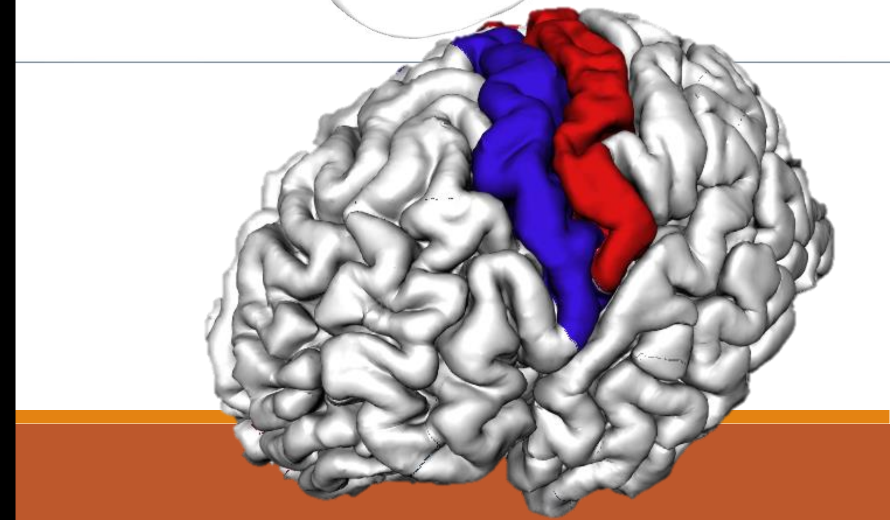


GPU - NVIDIA GTX 980 GPU card (2048 cores)


EEG View
00:00
00:00:01
00:00:02
00:00:03
00:00:04
00:00:05
00:00:06
EEG FP2-REF1
EEG FT12-REF1
EEG AF8-REF1
EEG F10-REF1
EEG F8-REF1
EEG F4-REF1
EEG FT10-REF1
EEG FC6-REF1
EEG FC2-REF1
EEG T8-REF1
EEG C4-REF1
EEG TP10-REF1
EEG CP6-REF1
EEG CP2-REF1
EEG P8-REF1
EEG P4-REF1
EEG O2-REF1
EEG FZ-REF1
EEG CZ-REF1
EEG PZ-REF1

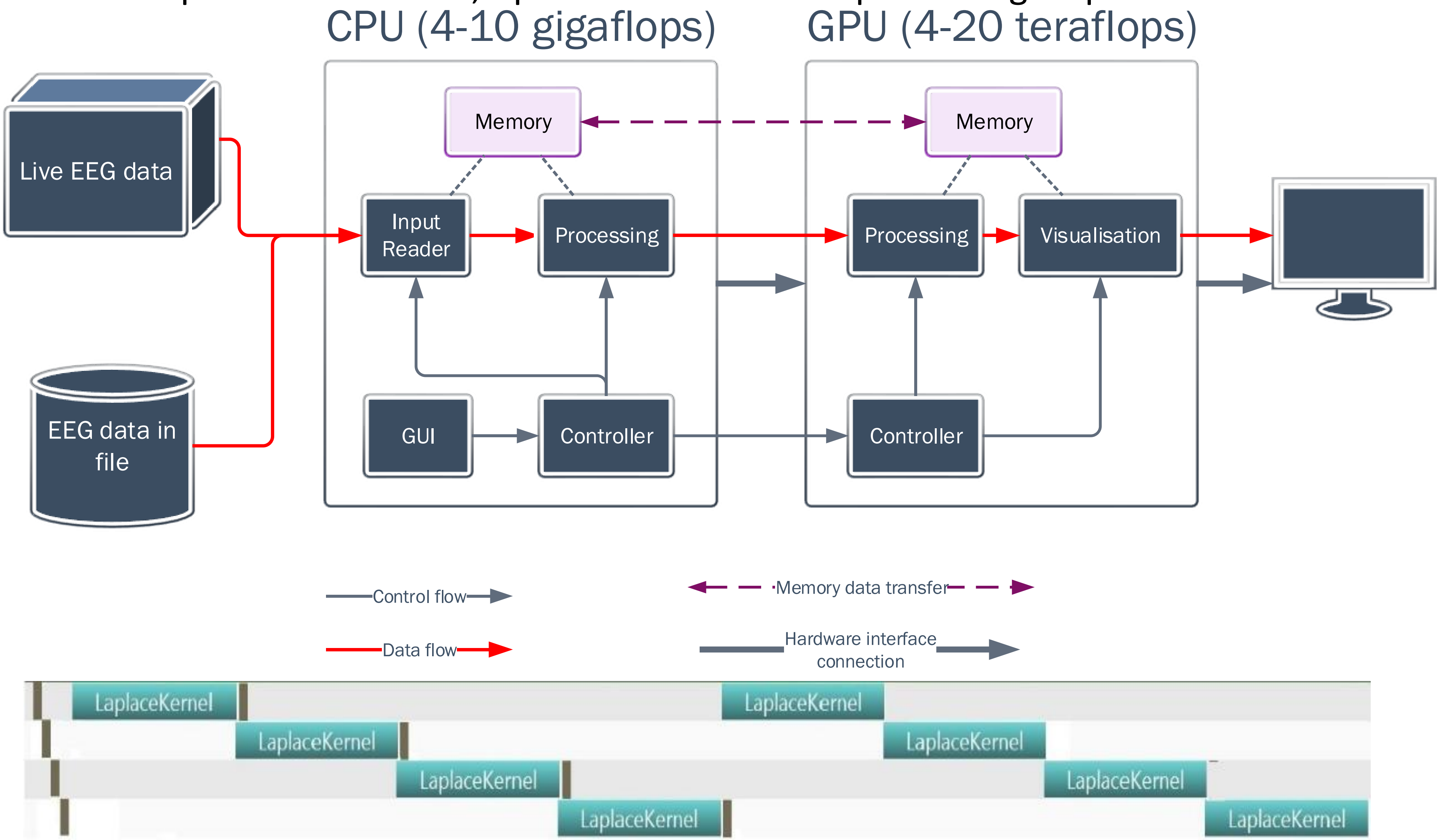

EEG View 3
00:00
00:00:01
00:00:02
00:00:03
00:00:04
00:00:05
00:00:06
EEG FP2-REF1
EEG FT12-REF1
EEG AF8-REF1
EEG F10-REF1
EEG F8-REF1
EEG F4-REF1
EEG FT10-REF1
EEG FC6-REF1
EEG FC2-REF1
EEG T8-REF1
EEG C4-REF1
EEG TP10-REF1
EEG CP6-REF1
EEG CP2-REF1
EEG P8-REF1
EEG P4-REF1
EEG O2-REF1
EEG FZ-REF1


Time: 0.1171875
Amplitude: 1000
Transparency [%]: 100
View normalised
View mode: herical Laplace map
Lambda: 1E-7
Laplace threshold (K x stdev): 0



Stream processing architecture: a high-level view

Compute and visualise; optimised end-to-end processing sequence



Summary

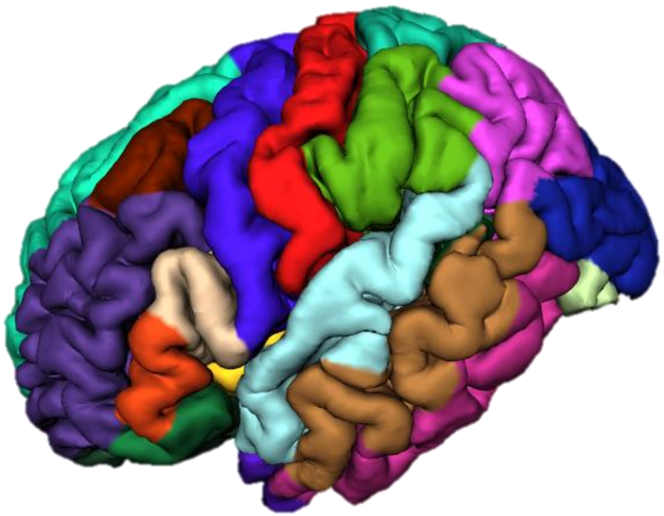
Real-time EEG imaging is possible **3 teraflops sustained performance**

Requires careful tuning and exploiting architecture features

Future work

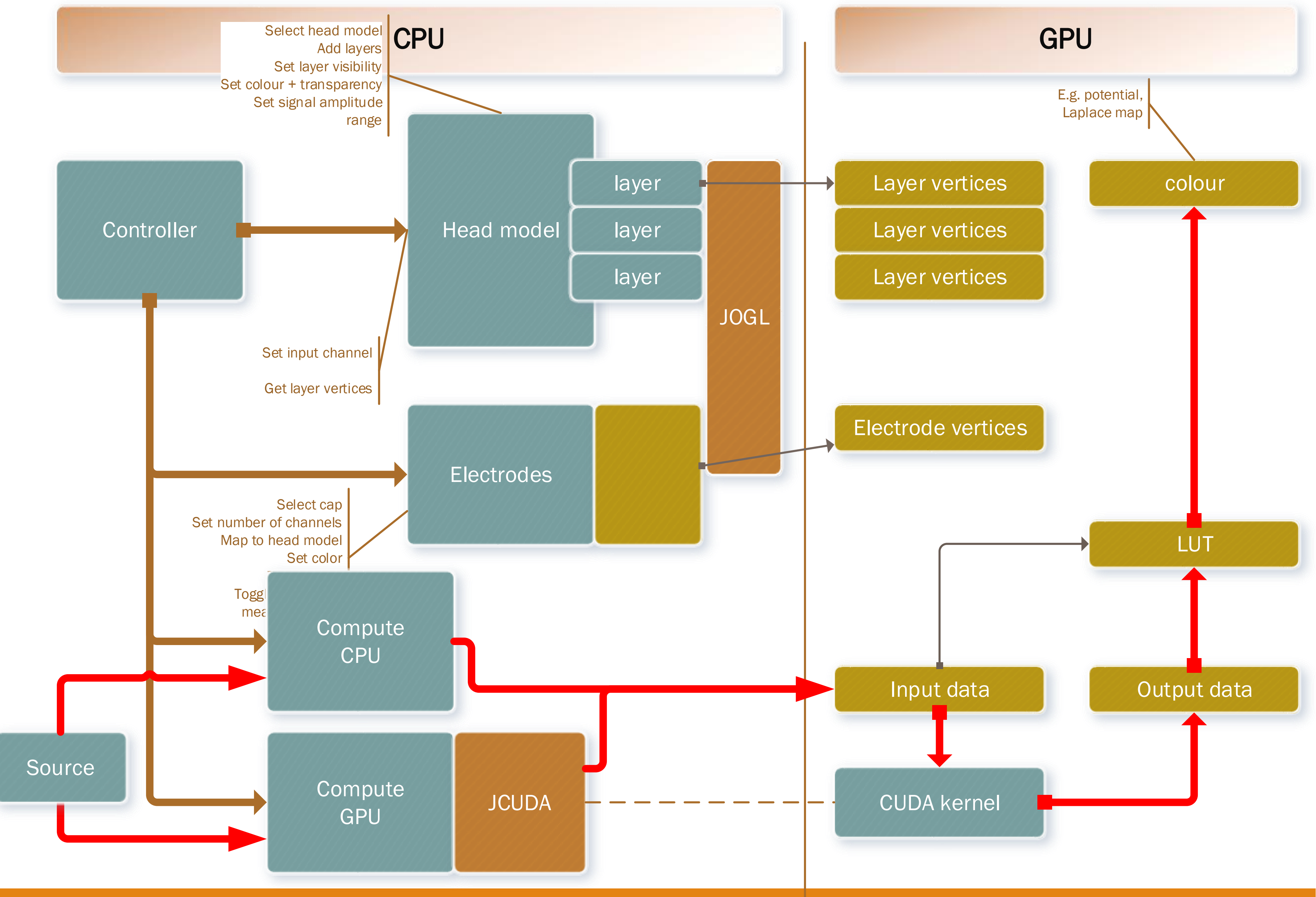
Implementing further algorithms

- Independent Component Analysis
- Short-time FFT, Wavelet transform
- Functional mapping, connectivity
- Statistics, machine learning



Multi-GPU environments

Application for arbitrary real-time data processing tasks (sensor data, audio, video, etc)



CPU
GPU
Select head model
Add layers
Set layer visibility
Set colour + transparency
Set signal amplitude range
E.g. potential, Laplace map
Controller
Head model
layer
layer
layer
JOGL
Layer vertices
Layer vertices
Layer vertices
colour
Set input channel
Get layer vertices
Electrodes
Electrode vertices
Select cap
Set number of channels
Map to head model
Set color
LUT
Compute CPU
Input data
Output data
Source
Compute GPU
JCUDA
CUDA kernel