

Online plan adaptation of head and neck IMPT treatments based on cone beam CT imaging and GPU Monte Carlo simulations

P. Botas^{1,2}, J. Kim³, B. Winey¹, H. Paganetti¹

¹Massachusetts General Hospital & Harvard Medical School, Boston, MA

²Heidelberg University, Heidelberg, Germany

³Yonsei University, Seoul, South Korea

PTCOG57, Cincinnati



Motivation

Problem:

- Proton therapy is **sensitive to geometry**
- Robust optimization cannot account for all scenarios
- **Smaller margins:** better plans

Potential solution:

- **Adaptive therapy would correct inter-fractional geometry changes**, allowing margin reduction
- **Head and neck cases** are candidates to benefit from the technique

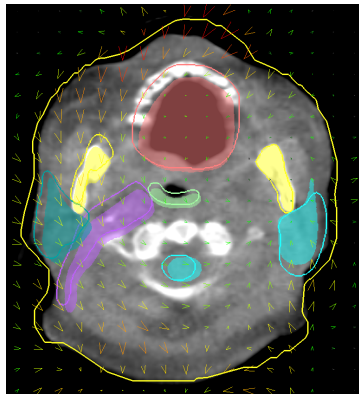


Fig: Head and neck patient geometry changes. The arrows represent a vector field.

The need for adaptive proton therapy

10 head & neck patients planned **without CTV margins**, evaluated at 60 weeks:

- Reduced margins → sensitive to errors
- Coverage deteriorates:

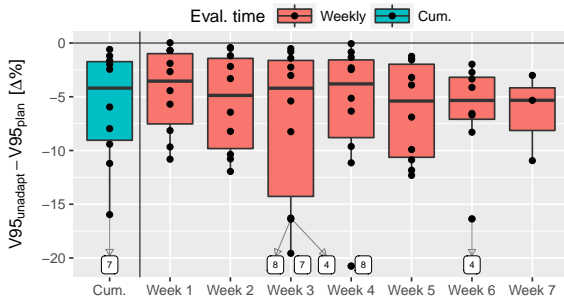


Fig: V95 in CTV decreases

The need for adaptive proton therapy

10 head & neck patients planned **without CTV margins**, evaluated at 60 weeks:

- Reduced margins → sensitive to errors
- Coverage deteriorates:

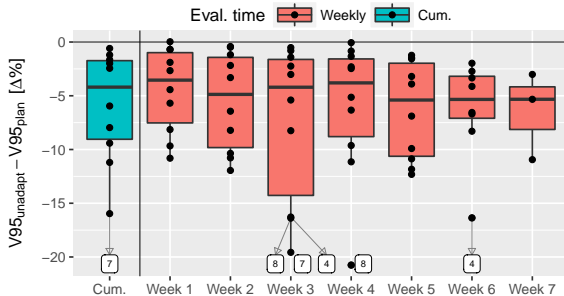


Fig: V95 in CTV decreases

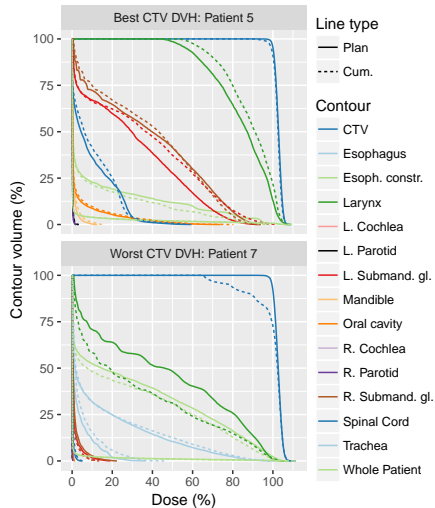


Fig: DVHs after full treatment

Adaptive proton therapy ingredients: the framework

Cone Beam CT (CBCT)

A priori CT-based scatter correction WEPL error $< 2\%$ in head cases.

Park et al., Med Phys. 2015;42(8), Kim et al., Phys Med Bio. 2017;62(1)

Image Registration: Plastimatch

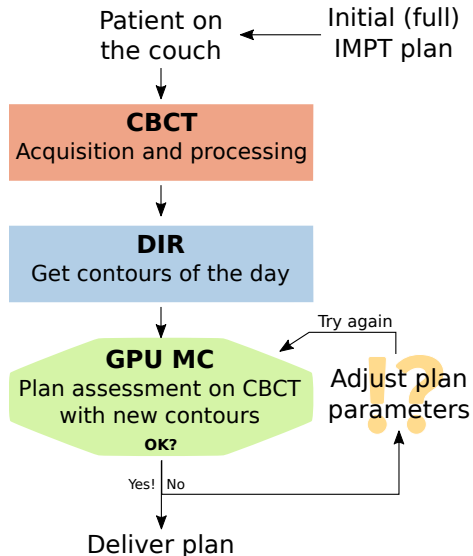
Rigid and deformable (DIR), GPU B-spline

Shackleford et al., Phys Med Biol. 2010;55(21)

Fast GPU MC: gPMC

Accurate calculation engine developed with UT Southwestern.

Qin et al., Phys Med Biol. 2016;61(20)



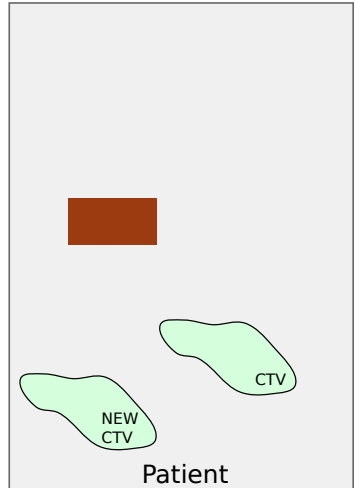
Adaptation method

Consists of 2 steps:

- ① **Geometrical adaptation:** Move individual spots following a deformation vector field and correct energies
- ② **Weight tuning:** Adjust the weight of the spots, if necessary

Geometrical adaptation

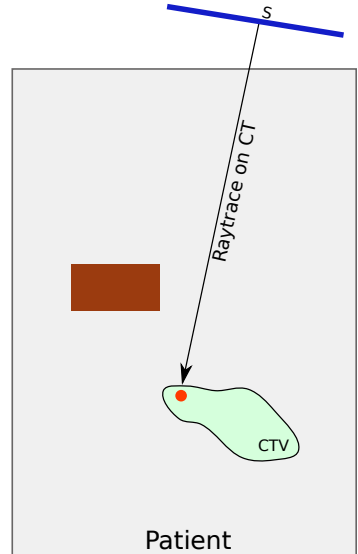
Per spot $s_i = (x_0, y_0, E_0)$:



Geometrical adaptation

Per spot $s_i = (x_0, y_0, E_0)$:

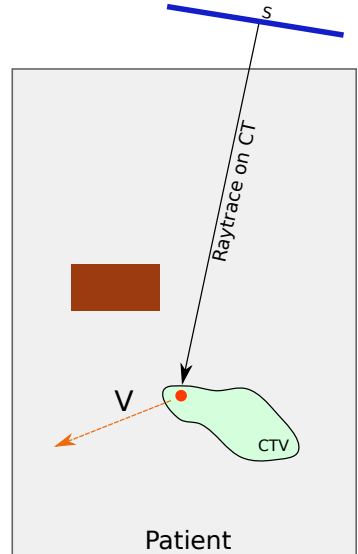
1: **Raytrace** s_i in CT (r_i)



Geometrical adaptation

Per spot $s_i = (x_0, y_0, E_0)$:

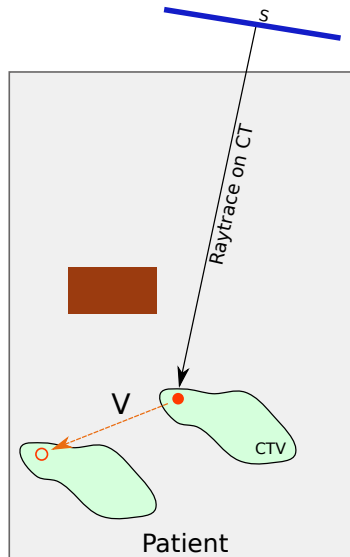
- 1: **Raytrace** s_i in CT (r_i)
- 2: **Probe** VF at r_i coords: v_i



Geometrical adaptation

Per spot $s_i = (x_0, y_0, E_0)$:

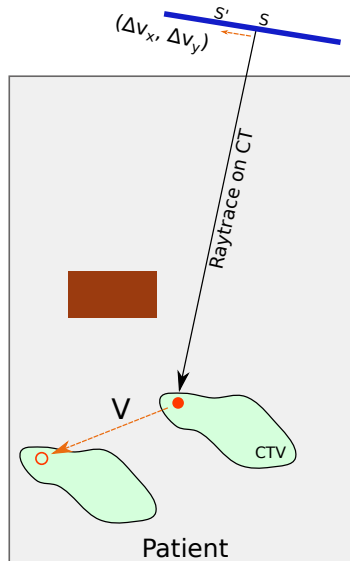
- 1: **Raytrace** s_i in CT (r_i)
- 2: **Probe** VF at r_i coords: v_i
- 3: **Apply** v_i **to** r_i coords: position where the r_i should be in the CBCT



Geometrical adaptation

Per spot $s_i = (x_0, y_0, E_0)$:

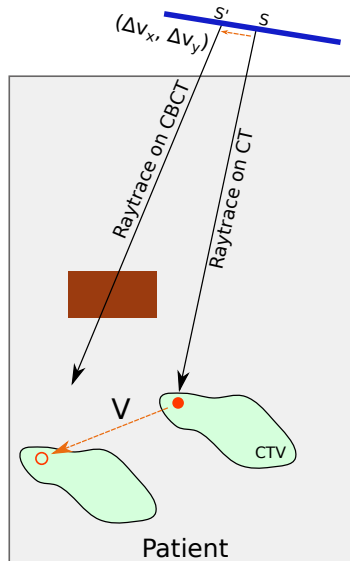
- 1: **Raytrace** s_i in CT (r_i)
- 2: **Probe** VF at r_i coords: v_i
- 3: **Apply** v_i **to** r_i coords: position where the r_i should be in the CBCT
- 4: **Apply** v_i **to** $s \rightarrow$
 $s'_i = (x_0 + \Delta v_x, y_0 + \Delta v_y, E_0)_i$



Geometrical adaptation

Per spot $s_i = (x_0, y_0, E_0)$:

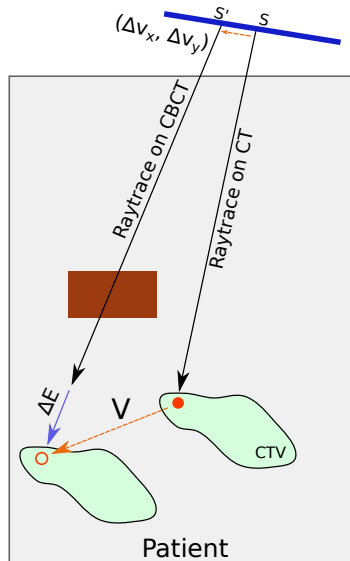
- 1: **Raytrace** s_i in CT (r_i)
- 2: **Probe** VF at r_i coords: v_i
- 3: **Apply** v_i **to** r_i coords: position where the r_i should be in the CBCT
- 4: **Apply** v_i **to** $s \rightarrow$
 $s'_i = (x_0 + \Delta v_x, y_0 + \Delta v_y, E_0)_i$
- 5: **Raytrace** s'_i in CBCT



Geometrical adaptation

Per spot $s_i = (x_0, y_0, E_0)$:

- 1: **Raytrace** s_i in CT (r_i)
- 2: **Probe** VF at r_i coords: v_i
- 3: **Apply** v_i **to** r_i coords: position where the r_i should be in the CBCT
- 4: **Apply** v_i **to** $s \rightarrow$
 $s'_i = (x_0 + \Delta v_x, y_0 + \Delta v_y, E_0)_i$
- 5: **Raytrace** s'_i in CBCT
- 6: **Get** ΔE_i

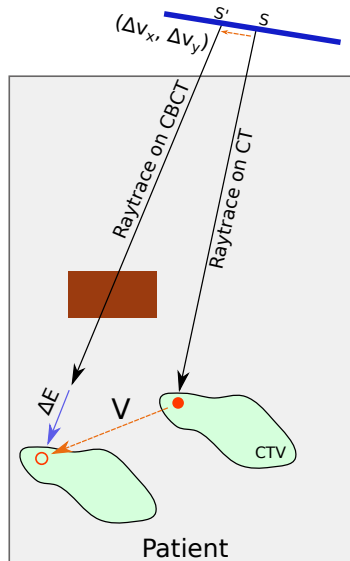


Geometrical adaptation

Per spot $s_i = (x_0, y_0, E_0)$:

- 1: **Raytrace** s_i in CT (r_i)
- 2: **Probe** VF at r_i coords: v_i
- 3: **Apply** v_i **to** r_i coords: position where the r_i should be in the CBCT
- 4: **Apply** v_i **to** $s \rightarrow$
 $s'_i = (x_0 + \Delta v_x, y_0 + \Delta v_y, E_0)_i$
- 5: **Raytrace** s'_i in CBCT
- 6: **Get** ΔE_i

Spot adaptation: $(\Delta v_x, \Delta v_y, \Delta E)_i$



Geometrical adaptation

Energy and position layers organization is distorted.

Geometrical adaptation

Energy and position layers organization is distorted.

Four strategies constraining the geometrical adaptation:

- **Free:** No constrains shifts
- **Isocenter shift:** Average VF in CTV
- **Range shifter:** Average energy shift
- **Iso. + range:** Average VF and energy shifts

Weight tuning:

How to create dose-influence matrices **fast** with GPU MC (gPMC)?

Weight tuning:

How to create dose-influence matrices **fast** with GPU MC (gPMC)?

Observation from initial plans: 8.2% of the spots deliver 50% of the protons

Weight tuning:

How to create dose-influence matrices **fast** with GPU MC (gPMC)?

Observation from initial plans: 8.2% of the spots deliver 50% of the protons

Weight tuning steps:

- ① Simulate **geometrical adaptation** with gPMC
- ② Score the **dose per spot** in region of interest

Weight tuning:

How to create dose-influence matrices **fast** with GPU MC (gPMC)?

Observation from initial plans: 8.2% of the spots deliver 50% of the protons

Weight tuning steps:

- 1 Simulate **geometrical adaptation** with gPMC
- 2 Score the **dose per spot** in region of interest
- 3 **Select set** of spots giving 50% of the dose (at least 10% of the total spots)
- 4 Accumulate adapted dose without the set
- 5 Calculate **remaining dose** for coverage in target

Weight tuning:

How to create dose-influence matrices **fast** with GPU MC (gPMC)?

Observation from initial plans: 8.2% of the spots deliver 50% of the protons

Weight tuning steps:

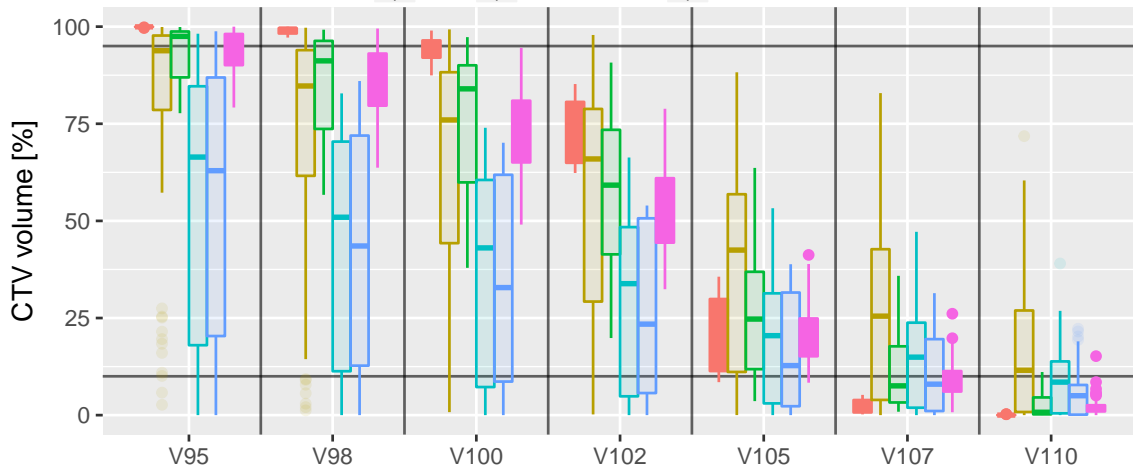
- 1 Simulate **geometrical adaptation** with gPMC
- 2 Score the **dose per spot** in region of interest
- 3 **Select set** of spots giving 50% of the dose (at least 10% of the total spots)
- 4 Accumulate adapted dose without the set
- 5 Calculate **remaining dose** for coverage in target
- 6 **Tune set weights** to fill the remaining dose and spare OARs

Results: all geometrical adaptations (no weight tuning)

Results: all geometrical adaptations (no weight tuning)

Method

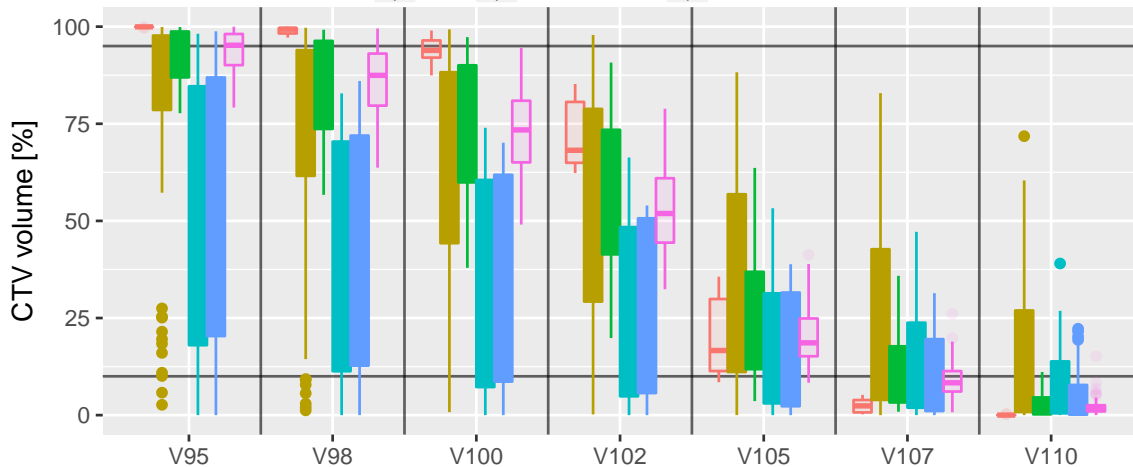
| | | |
|------|---------------|---------------------------|
| Plan | Isocenter | Isocenter – Range shifter |
| Free | Range shifter | None |



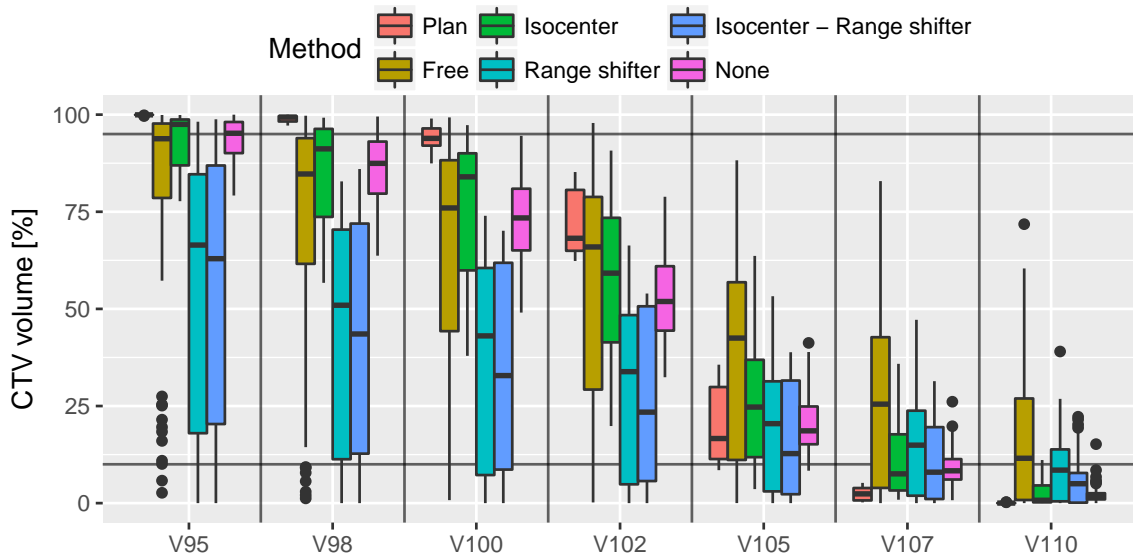
Results: all geometrical adaptations (no weight tuning)

Method

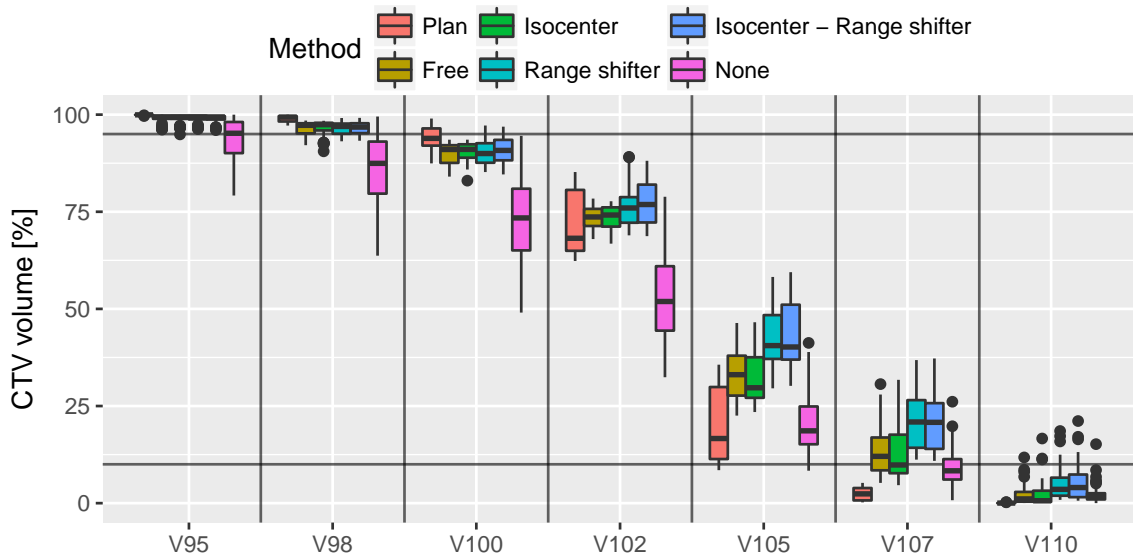
| | | |
|------|---------------|---------------------------|
| Plan | Isocenter | Isocenter – Range shifter |
| Free | Range shifter | None |



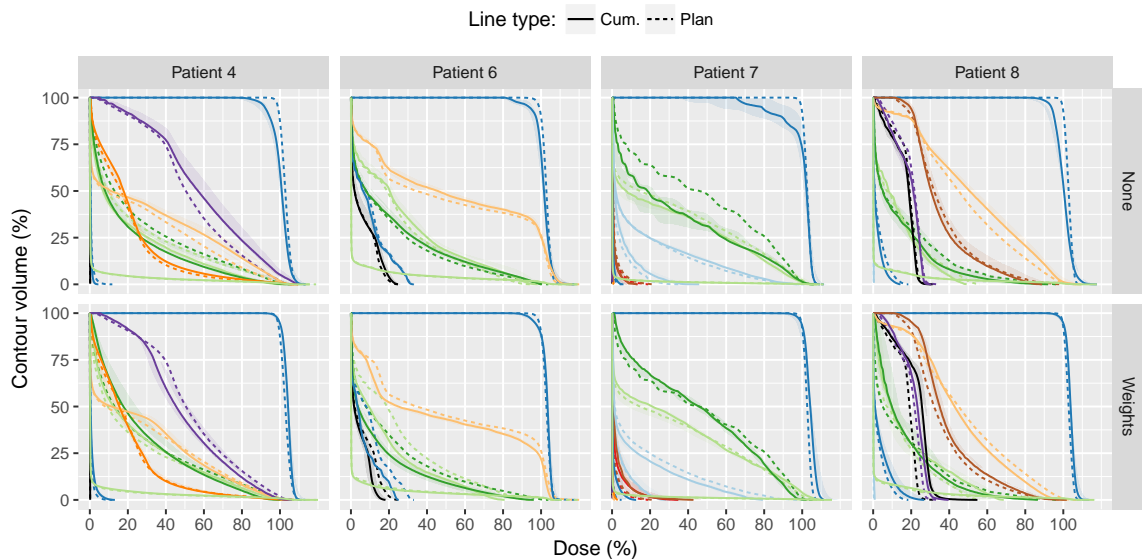
Results: all geometrical adaptations (no weight tuning)



Results: all geometrical adaptations + weight tuning



Results with free geometrical adaptation + weight tuning



Timing and conclusions

Timing, timing, timing!!

| <i>(seconds)</i> | Minimum | Average | Maximum | Expected |
|--------------------|---------|--------------|---------|-------------------|
| Geometrical adapt. | 11.7 | 16.9 | 26.57 | ~ 1 – 5 |
| gPMC validation | 115.6 | 261.9 | 419.2 | ~ 30 |
| Weight tuning | 12.0 | 44.8 | 198.0 | ~ 5 – 120 |
| Total | - | 322.7 | - | ~ 60 – 120 |

Table: Current and expected times (s)

Timing and conclusions

Timing, timing, timing!!

| <i>(seconds)</i> | Minimum | Average | Maximum | Expected |
|--------------------|---------|--------------|---------|-------------------|
| Geometrical adapt. | 11.7 | 16.9 | 26.57 | ~ 1 – 5 |
| gPMC validation | 115.6 | 261.9 | 419.2 | ~ 30 |
| Weight tuning | 12.0 | 44.8 | 198.0 | ~ 5 – 120 |
| Total | - | 322.7 | - | ~ 60 – 120 |

Table: Current and expected times (s)

Conclusions:

- If adaptation is needed, weight tuning is generally necessary
- Tuning the **weight of a subset of spots** might be enough
- The algorithm has the potential to **be applicable online, pending hardware and parallelization**
- The algorithm might **allow further margin reduction**



Visit <https://gray.mgh.harvard.edu/>!