

Detekce kolizí mezi svaly během simulace

KIV/PPR - ANALÝZA PROBLÉMU

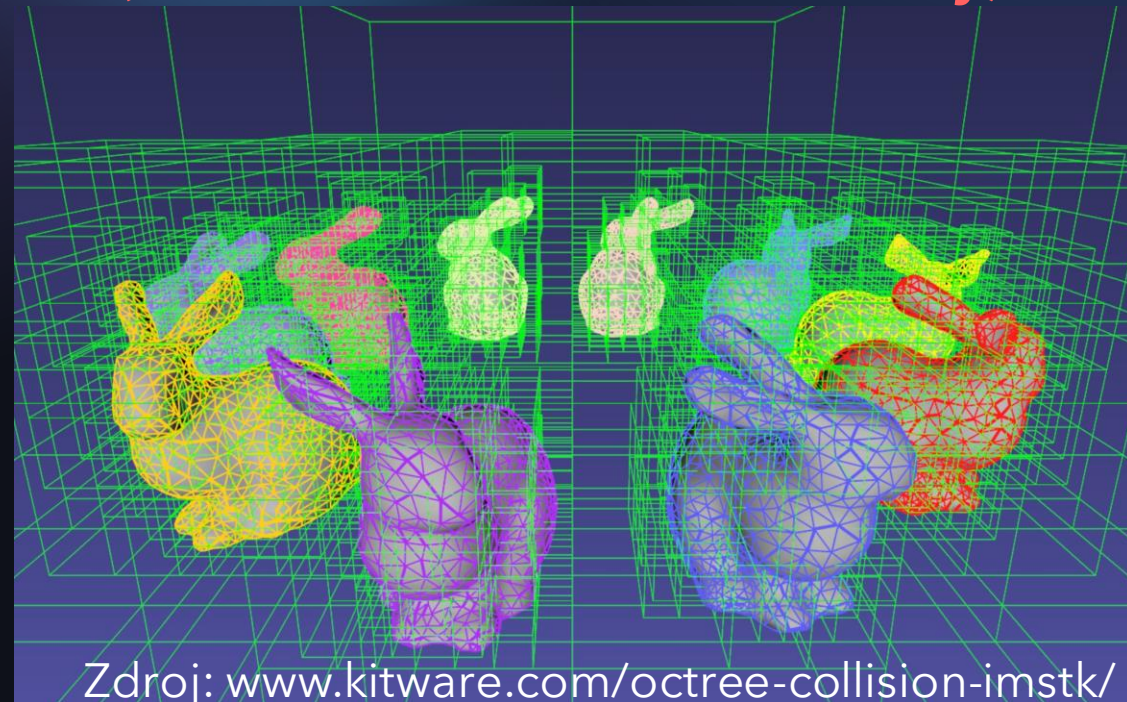
ONDŘEJ HAVLÍČEK

2023/2024

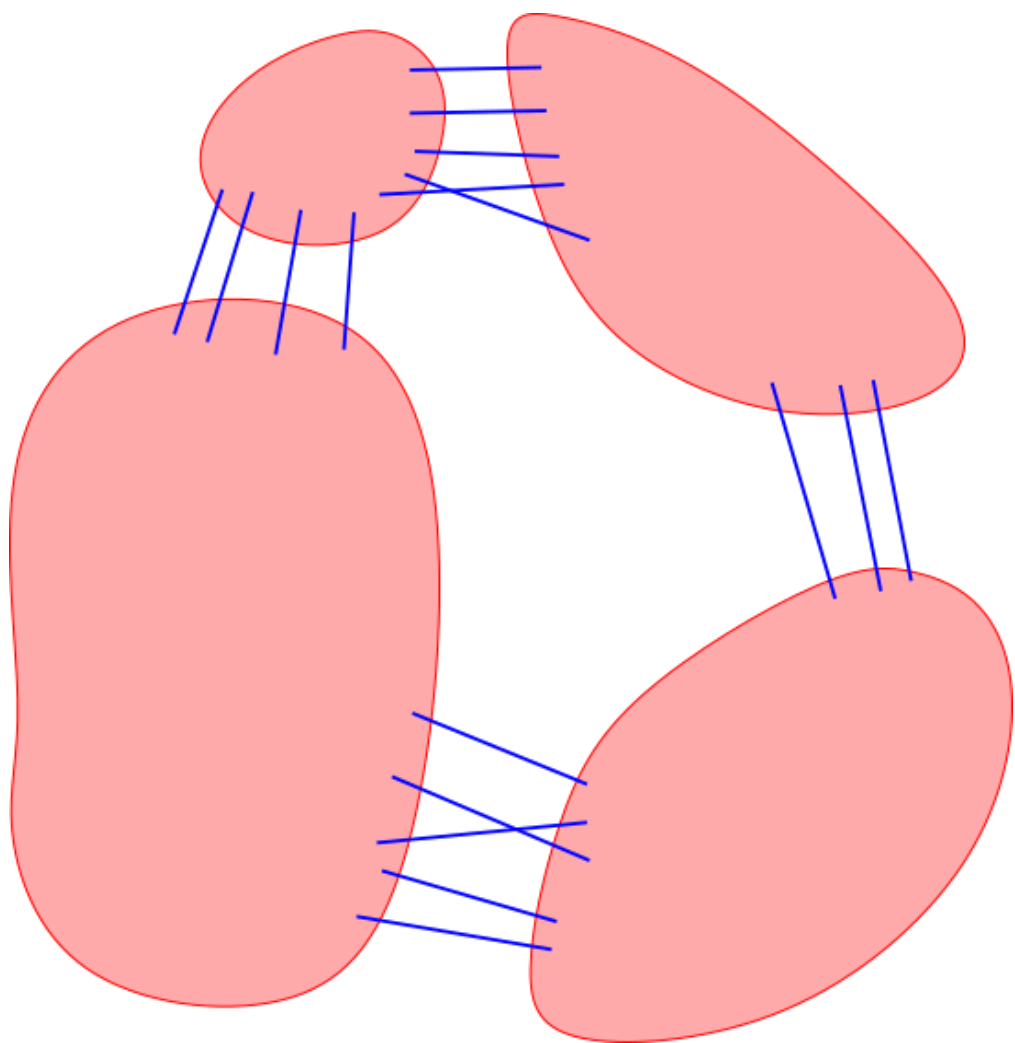
Problém

- 3D simulace muskuloskeletálního systému v reálném čase
 - Rigidní kosti
 - Elastické svaly
- Pohyb kostí
 - deformace svalů v okolí
 - kolize Δ sítí svalů

**Dělení prostoru je pomalé
(sestavení a aktualizace struktury)**

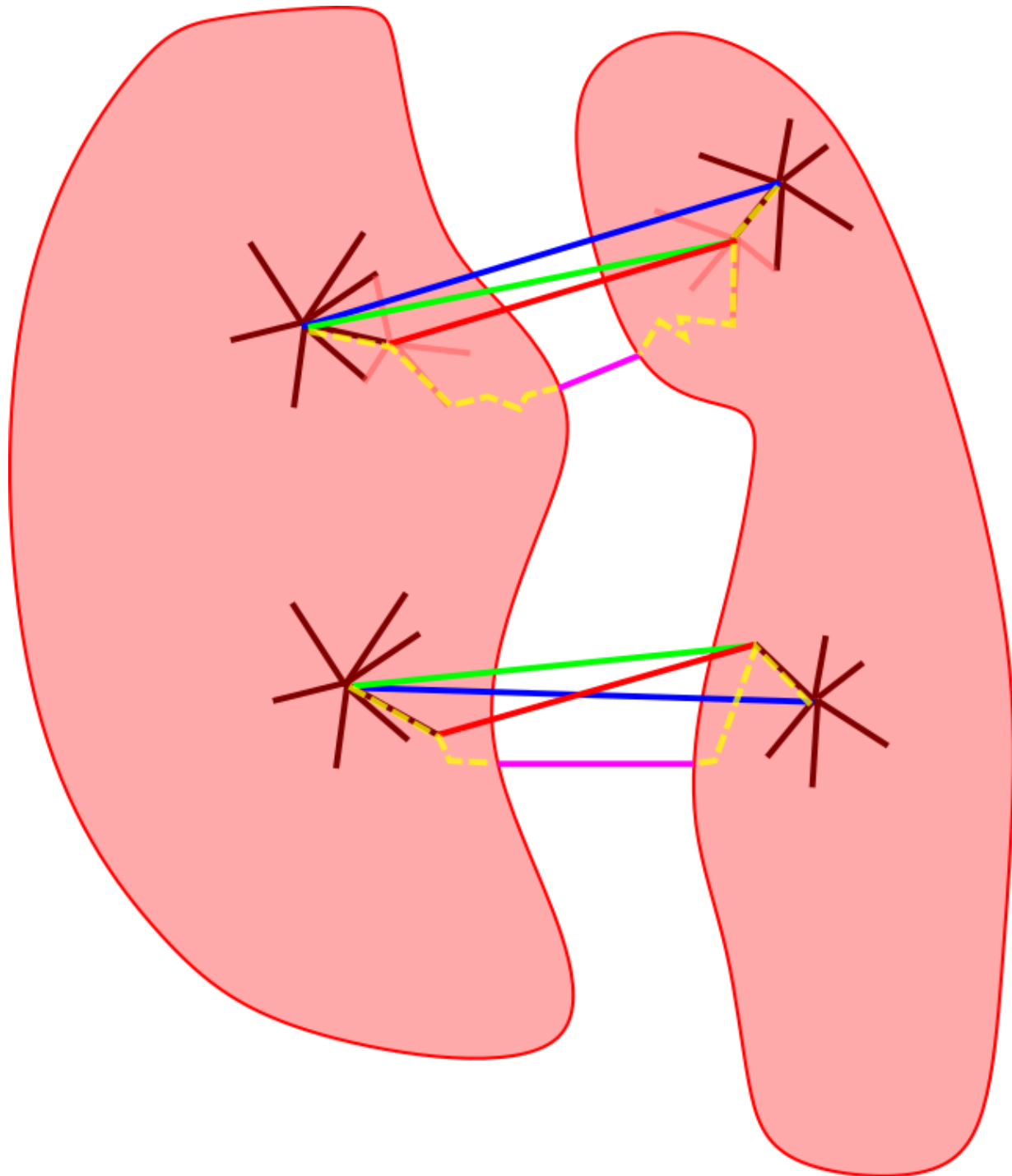


Zdroj: www.kitware.com/octree-collision-imstk/



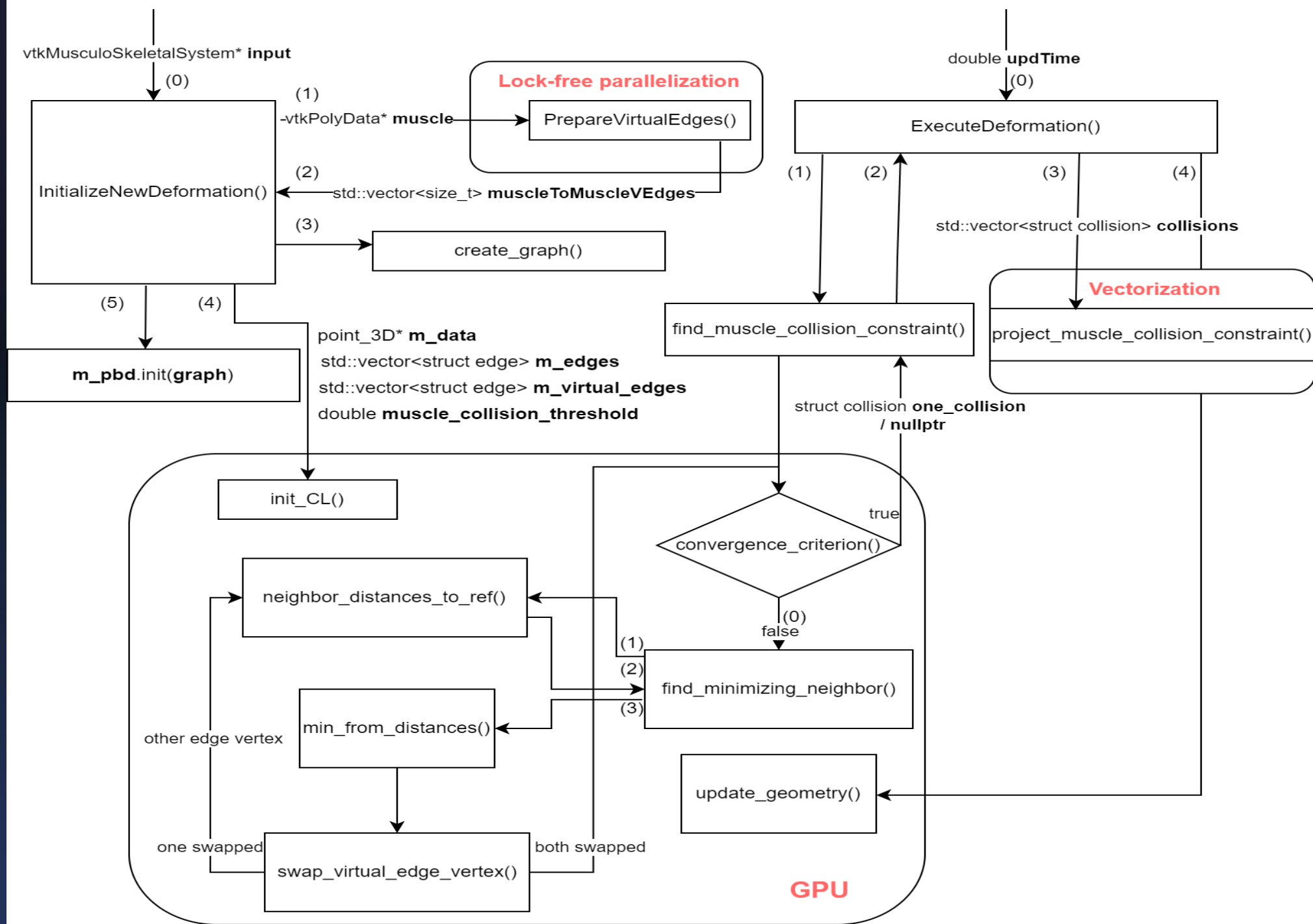
Řešení

- Virtuální hrany mezi svaly
- Minimalizace délky
- Iterační algoritmus
 - Změna vrcholu na souseda, pokud se zkrátí virtuální hrana
 - To samé pro druhou stranu (zig-zag)
- Konvergenční kritérium:
 - Dost malá vzdálenost (kolize)
 - Počet iterací
 - Nedochozí ke změně



Řešení

- Virtuální hrany mezi svaly
- Minimalizace délky
- Iterační algoritmus
 - Změna vrcholu na souseda, pokud se zkrátí virtuální hrana
 - To samé pro druhou stranu (zig-zag)
- Konvergenční kritérium:
 - Dost malá vzdálenost (kolize)
 - Počet iterací
 - Nedochozí ke změně



Struktury

<https://vtk.org/doc/nightly/html/classvtkPolyData.html>

```
class point_3D{
public:
    ...
private:
    void init(size_t id);
    size_t m_id = 666;
    double m_velo[POINT_DIM] = {0};
    double m_force[POINT_DIM] = {0};
    double m_work[POINT_DIM] = {0};
    double m_accum[POINT_DIM] = {0};
    double m_laplacian[POINT_DIM] = {0};
    double m_inv_mass = 1;
    size_t m_fixed = std::numeric_limits<size_t>::max();
};
```

```
struct edge{
    double value;
    double stiffness;
    double abs_stiffness;
    size_t end;
    size_t first_triangle;
    size_t second_triangle;
};
```

```
struct collision{
    size_t part_dynamic;
    size_t part_static;
    size_t vec_dynamic;
    double x;
    double y;
    double z;
};
```

Diskuze

- Hodně logiky na GPU, možné přesunout na CPU
- Začínat vždy se stejnými virtuálními hranami či pokračovat se starými?