Quadrics arrangement in classifying rigid motions of a 3D digital image

K. Pluta[1,2], G. Moroz[3], Y. Kenmochi[1], P. Romon[2]

[1] University Paris-Est, LIGM, France
[2] University Paris-Est, LAMA, France
[3] INRIA Nancy-Grand-Est, Project Vegas, France

23.09.2016

AGENDA

Introduction to digitized rigid motions

Image patch and its alterations

Problem as arrangement of hypersurfaces

Dimension reduction

Computing sample points

Implementation and experiments

Introduction to digitized rigid motions

“How do I rotate in 3D, George?”

“That is easy Lenny, take a 3×3 skew-symmetric matrix and apply Cayley Transform.”

“Skew-, what, George?”

RIGID MOTIONS IN $\mathbb{R}^3$

$$\left| \begin{array}{ll} \mathcal{U}: \mathbb{R}^3 & \to \mathbb{R}^3 \\ \mathbf{x} & \mapsto \mathbf{R}\mathbf{x} + \mathbf{t} \end{array} \right.$$

$\mathbf{x}$ – point

$\mathbf{R}$ – rotation matrix

$\mathbf{t}$ – translation vector

Properties:

- distance and angle preserving maps
- bijective
- inverse: $\mathcal{T} = \mathcal{U}^{-1}$ is also a rigid motion

Rigid motions in $\mathbb{R}^3$

$$\left| \begin{array}{ll} \mathcal{U} : \mathbb{R}^3 & \to \mathbb{R}^3 \\ \mathbf{x} & \mapsto \mathbf{R}\mathbf{x} + \mathbf{t} \end{array} \right.$$

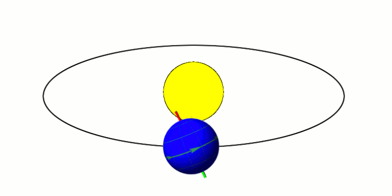
$\mathbf{x}$ – point

$\mathbf{R}$ – rotation matrix

$\mathbf{t}$ – translation vector

Properties:

- distance and angle preserving maps
- bijective
- inverse: $\mathcal{T} = \mathcal{U}^{-1}$ is also a rigid motion



The orbit of Earth is nearly circular, its eccentricity is just about 0.0167.

Cayley Transform

$$\begin{aligned}\mathbf{R} &= (\mathbf{I}-\mathbf{A})(\mathbf{I}+\mathbf{A})^{-1}\\ &= \frac{1}{k}\begin{bmatrix}1+a^2-b^2-c^2 & 2(ab-c) & 2(b+ac)\\ 2(ab+c) & 1-a^2+b^2-c^2 & 2(bc-a)\\ 2(ac-b) & 2(a+bc) & 1-a^2-b^2+c^2\end{bmatrix},\end{aligned}$$

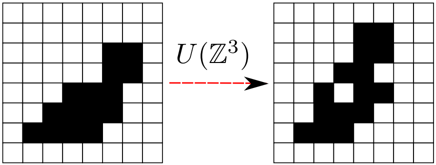
where $k = 1 + a^2 + b^2 + c^2$,

$$\mathbf{A} = \begin{bmatrix} 0 & c & -b\\ -c & 0 & a\\ b & -a & 0\end{bmatrix}$$

Rigid motions in $\mathbb{Z}^3$

$$U = \mathcal{D} \circ \mathcal{U}_{|\mathbb{Z}^3}$$

where $\mathcal{D}$ is a digitization e.g. rounding function



Properties:

- in general, does not preserve distances and angles
- not always injective
- not always surjective $U(\mathbb{Z}^3) \subsetneq \mathbb{Z}^3$

MOTIVATIONS

IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 23, NO. 2, FEBRUARY 2014 885

Topology-Preserving Rigid Transformation of 2D Digital Images

Phuc Ngo, Nicolas Passat, Yukiko Kenmochi, and Hugues Talbot

***Abstract*— We provide conditions under which 2D digital images preserve their topological properties under rigid transformations. We consider the two most common digital topology models, namely dual adjacency and well-composedness. This paper leads to the proposal of optimal preprocessing strategies that ensure the topological invariance of images under arbitrary rigid transformations. These results and methods are proved to be valid for various kinds of images (binary, gray-level, label), thus providing generic and efficient tools, which can be used in particular in the context of image registration and warping.**

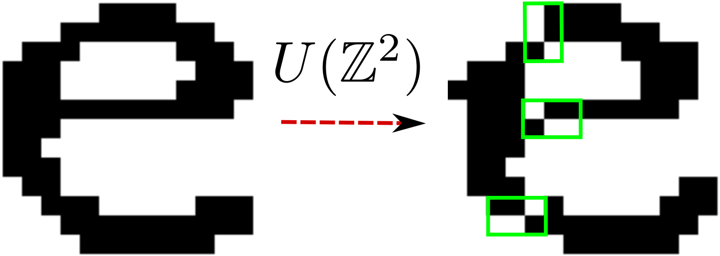
***Index Terms*— Digital images, rigid transformation, digital topology, image preprocessing, registration, warping.**

I. INTRODUCTION

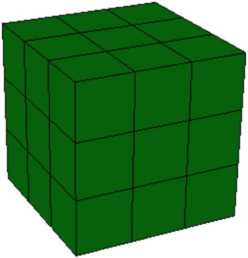
IN IMAGE computing, the preservation of topological prop- the sampling operation, which is mandatory to guarantee the stability of the transformations inside $\mathbb{Z}^2$.

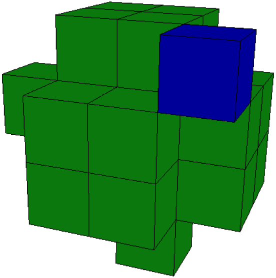
In this article, that is an extended and improved version of the conference paper [21], we deal with these topological issues (Sec. III), and we provide some methods for topological analysis and preprocessing of images before rigid transformations. We first give some conditions under which a 2D digital image preserves its topological properties under arbitrary rigid transformations (Sec. IV). Then, we propose methods for analysing and preprocessing digital images before rigid transformation, in order to preserve their topological properties (Sec. V). This paper is generic on two sides: (i) the main two digital topology models are considered, namely the dual adjacency, and the well-composedness ones;

Motivations

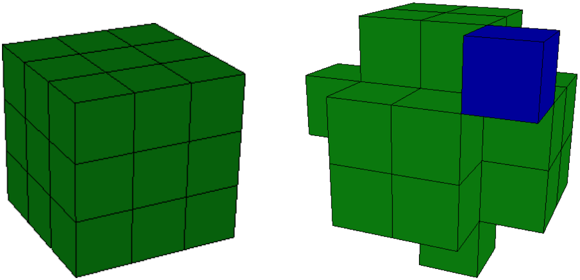


Motivations





MOTIVATIONS



Our goal

Generate all the variations of the $3 \times 3 \times 3$ cube under digitized rigid motions.

Image patch and its alterations

"George, look that when I rotate a photo of my house by 30° then there are holes in windows!"

"Since you use piecewise constant transformation you have to live with that, Lenny."

"If you are that smart then figure out how to explain this to clients who are going to see the house."

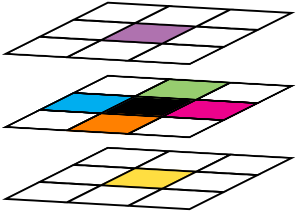
IMAGE PATCH

Definition

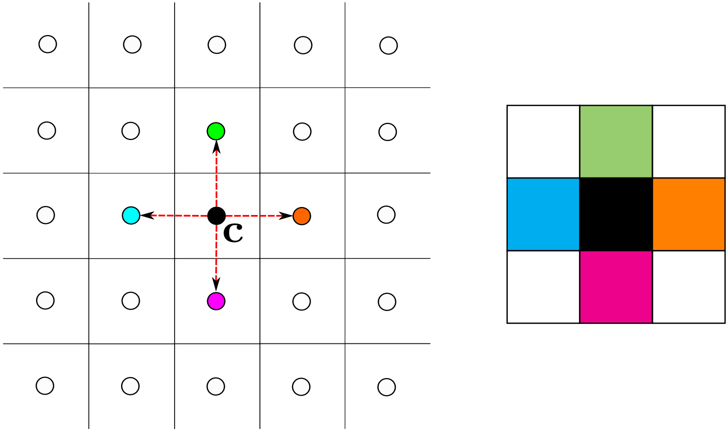
In general, we consider a finite set $\mathcal{N} \subset \mathbb{Z}^3$, called an *image patch* whose center $\mathbf{c}$ and radius r of $\mathcal{N}$ are given by $\mathbf{c} = \frac{1}{|\mathcal{N}|}\sum_{\mathbf{v}\in\mathcal{N}} \mathbf{v}$ and $r = \max_{\mathbf{v}\in\mathcal{N}} \|\mathbf{v} - \mathbf{c}\|$, respectively.

IMAGE PATCH

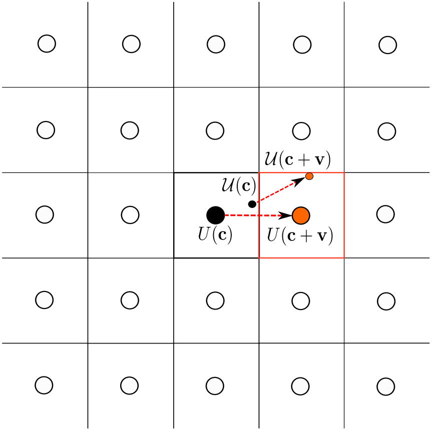
In particular, we consider the image patch
$\mathcal{N} = \{(1,0,0),(0,1,0),(0,0,1),(0,0,0),(-1,0,0),(0,-1,0),(0,0,-1)\}$
where the center is $(0,0,0)$.

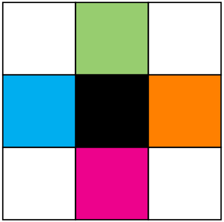


ALTERATIONS STEP-BY-STEP

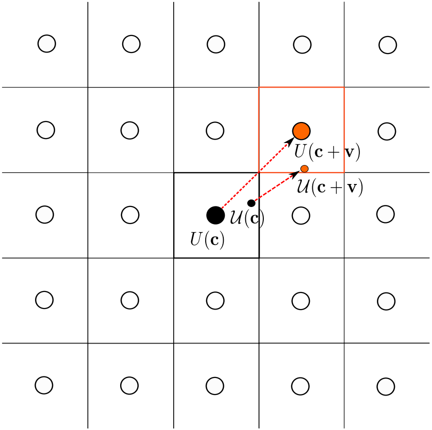


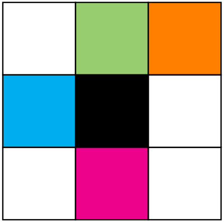
ALTERATIONS STEP-BY-STEP

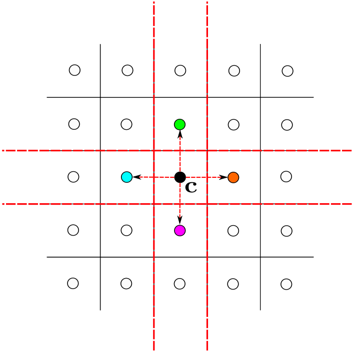




Alterations step-by-step





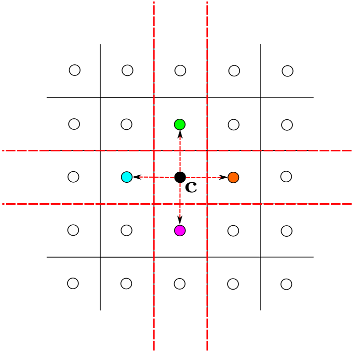


Formulation

$$\mathbf{R}_i\mathbf{v} + t_i = k_i - \frac{1}{2}$$

where $\mathbf{v} \in \mathcal{N} \subset \mathbb{Z}^3, k_i \in H(\mathcal{N}) = \mathbb{Z} \cap [-r', r']$ and $r' = r + \sqrt{3}$ – it the maximal radius of the transformed image patch.

Critical rigid motions



Formulation

$$\mathbf{R}_i \mathbf{v} + t_i = k_i - \frac{1}{2}$$

where $\mathbf{v} \in \mathcal{N} \subset \mathbb{Z}^3, k_i \in H(\mathcal{N}) = \mathbb{Z} \cap [-r', r']$ and $r' = r + \sqrt{3}$ – it the maximal radius of the transformed image patch.

The parameter space

$\Omega = \left\{(a, b, c, t_1, t_2, t_3) \in \mathbb{R}^6 \mid a, b, c \geq 0, -\frac{1}{2} < t_i < \frac{1}{2} \text{ for } i = 1, 2, 3\right\}$

Problem as arrangement of hypersurfaces

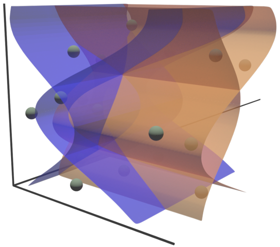
“Wow, George what are these hypersurface equations the blackboard about? You’re working on this for a week or so.”

“I’ve been thinking how to solve problems of my recent life, Lenny.”

“George, life is something to experience not a problem to solve. Just let it happens.”

Problem as arrangement of hypersurfaces

Warning: Explanation base on 2D rigid motions



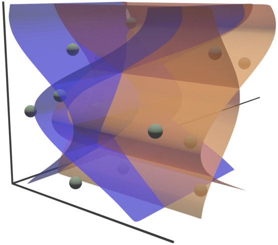
Visualization of 3D parameter space of 2D digitized rigid motions

Surfaces are given by

$$\mathbf{R}_i\mathbf{v} + t_i = k_i - \frac{1}{2}$$

PROBLEM AS ARRANGEMENT OF HYPERSURFACES

Warning: Explanation base on 2D rigid motions



Visualization of 3D parameter space
of 2D digitized rigid motions

In particular, surfaces are given by

$$\frac{1-a^2}{1+a^2}v_1 - \frac{2a}{1+a^2}v_2 + t_1 = k_1 - \frac{1}{2}$$
$$\frac{2a}{1+a^2}v_1 + \frac{1-a^2}{1+a^2}v_2 + t_2 = k_2 - \frac{1}{2}$$

Problem as arrangement of hypersurfaces

Complexity of the arrangement

Overall number of hypersurfaces is $\mathcal{O}(r^4)$ and the complexity of the arrangement is bounded by number of hypersurfaces to the power of dimensionality of the space; $\mathcal{O}(r^{24})$.

Contributions

- Formulation of the problem as an arrangement of surfaces in two 3D spaces
- Algorithm to compute sample points of 3-dimensional connected components in an arrangement of quadrics
- Implementation of proposed algorithm

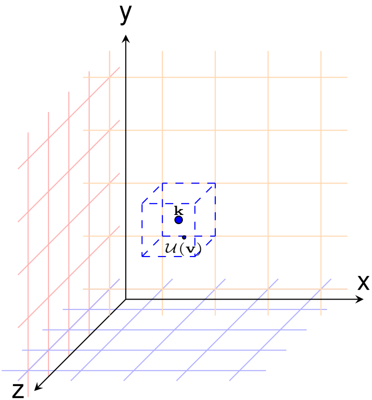
Dimension reduction

“What is going on Lenny?”

“Do you remember this hard problem in 6D which I was trying to solve for a long while? I’ve uncoupled variables and is so simple, right now.”

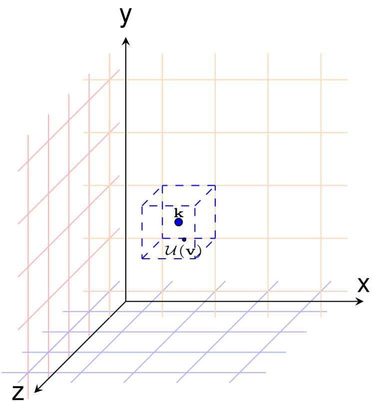
“Uncoupling, right...”

Uncoupling the parameters



$$k_i - \frac{1}{2} < \mathbf{R}_i \mathbf{v} + t_i < k_i + \frac{1}{2}$$

UNCOUPLING THE PARAMETERS



$$k_i - \frac{1}{2} - \mathbf{R}_i\mathbf{v} < t_i < k_i + \frac{1}{2} - \mathbf{R}_i\mathbf{v}$$

Uncoupling the parameters

For simplification let's us consider $\mathbf{v} = (1, 0, 0) \in \mathcal{N}$ and $i = 1$ – x-axis.

$$k_1 - \frac{1}{2} - \frac{1 + a^2 - b^2 - c^2}{1 + a^2 + b^2 + c^2} < t_1 < k_1 + \frac{1}{2} - \frac{1 + a^2 - b^2 - c^2}{1 + a^2 + b^2 + c^2}$$

UNCOUPLING THE PARAMETERS

For simplification let's us consider $\mathbf{v} = (1, 0, 0) \in \mathcal{N}$ and $i = 1$ – x-axis.

$$k_1 - \frac{1}{2} - \frac{1 + a^2 - b^2 - c^2}{1 + a^2 + b^2 + c^2} < t_1 < k_1 + \frac{1}{2} - \frac{1 + a^2 - b^2 - c^2}{1 + a^2 + b^2 + c^2}$$

$$k_1 + \frac{1}{2} - \frac{1 + a^2 - b^2 - c^2}{1 + a^2 + b^2 + c^2} - k_1 + \frac{1}{2} + \frac{1 + a^2 - b^2 - c^2}{1 + a^2 + b^2 + c^2}$$

Uncoupling the parameters

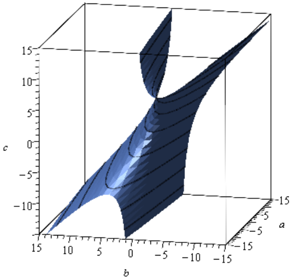
Thanks to the rational expressions in Cayley transform we obtain polynomials of degree 2:

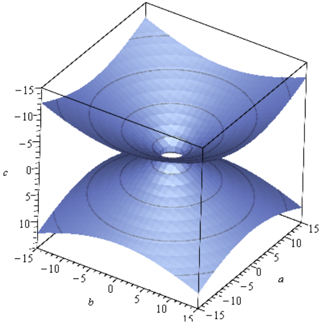
$$q_i[\mathbf{v}, k_i](a, b, c) = (1 + a^2 + b^2 + c^2)(2k_i - 1 - 2\mathbf{R}_i \mathbf{v}),$$

UNCOUPLING THE PARAMETERS

$$Q_i[\mathbf{v}, \mathbf{v}', k_i, k_i'](a,b,c) = q_i[\mathbf{v}, k_i](a,b,c) + 2(1+a^2+b^2+c^2) - q_i[\mathbf{v}', k_i'](a,b,c),$$

for $i = 1, 2, 3$.





Reduced set of quadrics

Discarding quadrics

For the image patch $\mathcal{N} = \{(1,0,0),(0,1,0),(0,0,1),(0,0,0),(-1,0,0),$ $(0,-1,0),(0,0,-1)\}$ we obtain directly 441 quadrics reduced to 81.

What we want to compute?

At least one sample point for each 3-dimensional connected component of the set

$$Q_i[\mathbf{v}, \mathbf{v}', k_i, k_i'](a, b, c) \neq 0.$$

Computing arrangement of quadrics in 3D

“Wow George, what is going on?! Your flat is so clean and arranged in a different way! Finally, nice.”

“Well, it is you who said do not solve, just let things happen. So, yeah, my girlfriend moved in a week ago.”

“Alright...”

Relations with previous studies

B. Mourrain, J. P. Tecourt, and M. Teillaud. On the computation of an arrangement of quadrics in 3D

What is new?

- Use of non-generic directions
- Support for asymptotic critical values
  - K. Kurdyka, P. Orro, S. Simon, et al. Semialgebraic Sard theorem for generalized critical values
  - Z. Jelonek and K. Kurdyka. Quantitative generalized Bertini-Sard theorem for smooth affine varieties
- We store only 3D sample points

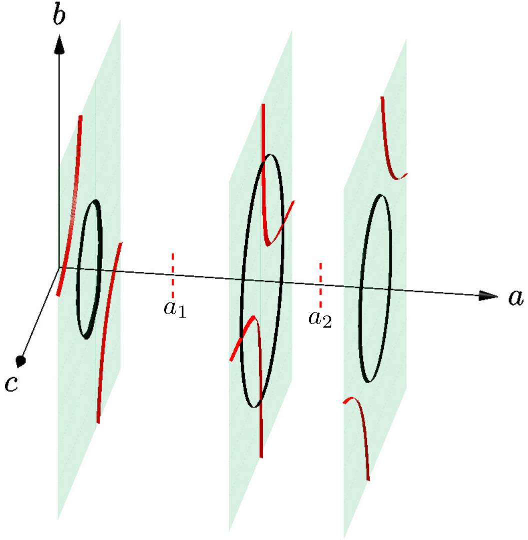
OUR ALGORITHM - THE GLOBAL IDEA

- Along non-generic direction, detect and sort all the events in which topology of an arrangement of quadrics changes
- In between two consecutive events, place a plane, intersect it with quadrics, and compute one point in each connected component bounded by obtained conics

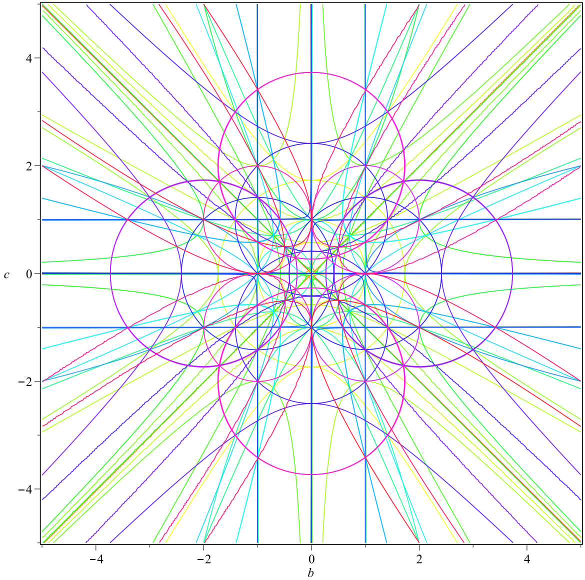
Our algorithm - the global idea

- Along non-generic direction, detect and sort all the events in which topology of an arrangement of quadrics changes
- In between two consecutive events, place a plane, intersect it with quadrics, and compute one point in each connected component bounded by obtained conics

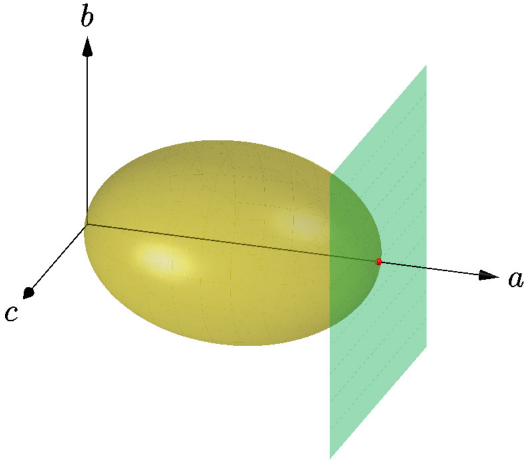
OUR ALGORITHM - THE GLOBAL IDEA



DETECTION OF EVENTS

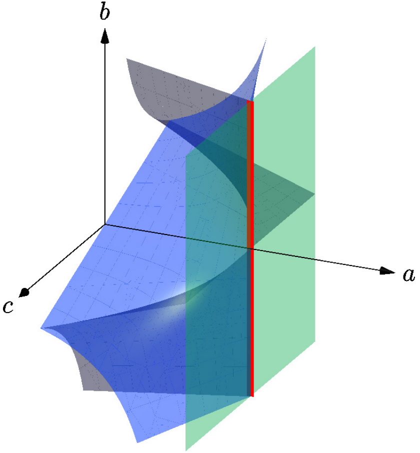


Detection of events



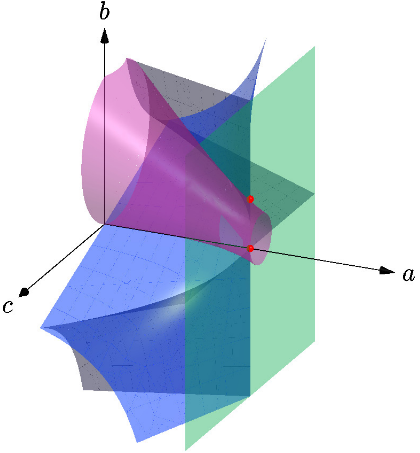
$$Q_i(s, b, c) = \partial_b Q_i(s, b, c) = \partial_c Q_i(s, b, c) = 0$$

DETECTION OF EVENTS



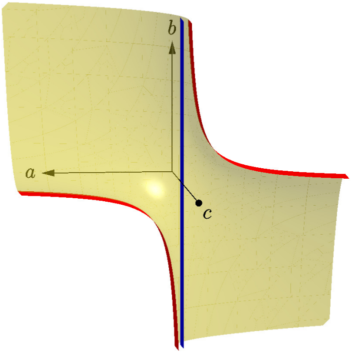
$$Q_i(s,b,c) = Q_j(s,b,c) = (\nabla Q_i \times \nabla Q_j)_1(s,b,c) = 0$$

Detection of events



$$Q_i(s,b,c) = Q_j(s,b,c) = Q_k(s,b,c) = 0$$

Detection of events



Types of asymptotic critical values:

- A_∞ – asymptote lives in a quadric
- B_∞ – asymptote given by an intersection of two quadrics

Recovering the translational part

“Hey Lenny have you seen this cool video about gyroscopic precession where a guy lifts up a 20 kg turning gear above his head like a feather?”

“Sure, is not science cool?”

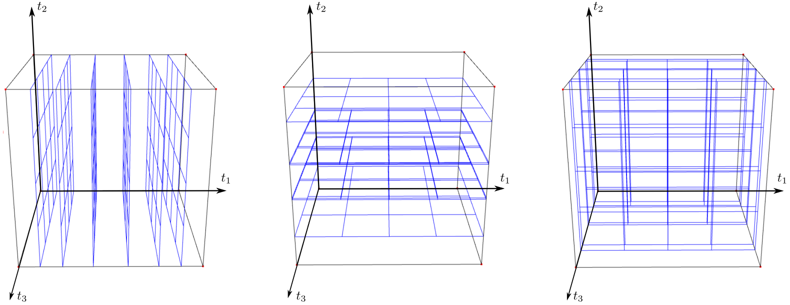
“Yeah, I’m working on a video where I going to run with such a gear turning.”

“George, I advise against...”

Recovering the translational part

Sample points of the translational part can be computed from the sample points of the previous step from

$$k_i - \frac{1}{2} - \mathbf{R}_i \mathbf{v} < t_i < k_i + \frac{1}{2} - \mathbf{R}_i \mathbf{v}$$



Implementation and experiments

“Hey guys! What are you doing in the kitchen?”

“Hey Kate! We are trying to find the optimal set of parameters to cook an egg in the microwave before it explodes.”

“What?! What are you doing in my kitchen?! George!!!”

<Booom!!!>

“Oh boy...”

Implementation

- Maple 2015
- Grid framework
- Raglib

Implementation

- Maple 2015
- Grid framework
- Raglib

Experiments

- Used machine: $2\times$ Intel(R) Xeon(R) E5-2680 v2 clocked at 2.8GHz, 251.717 GiB of RAM
- Computation of images of the image patch in around 40 minutes

Conclusion

- A new algorithm for computing arrangement of quadrics in 3D
- Use of non-generic directions
- Treatment of asymptotic critical values

Perspectives

- Optimization of our implementation
- Identification of image patches which break connectivity under digitized rigid motions in 3D

Implemented in Maple 2015 under Simplified BSD License (and Coffeeware license rev. 1)
github.com/copyme/RigidMotionsMapleTools

Thank you for your attention!