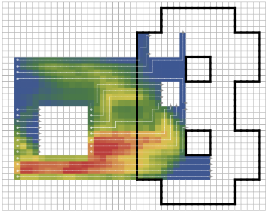
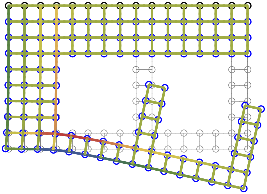
Digital Twins for Programmable Matter

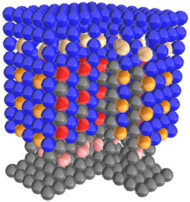
Jakub Lengiewicz

team: P. Hołobut, P. Chodkiewicz, A. Górzyńska-Lengiewicz, A. Macios (IPPT PAN)

in cooperation with: J. Bourgeois, B. Piranda (FEMTO-ST)   S. Bordas (UniLu)







Partially supported by Polish National Science Centre (NCN)
"Micromechanics of Programmable Matter", 2012-2017
Grant No. 2011/03/D/ST8/04089

MSCA IF
"MOrPhEM"
2019-2021

Outline

Motivation

a movie...

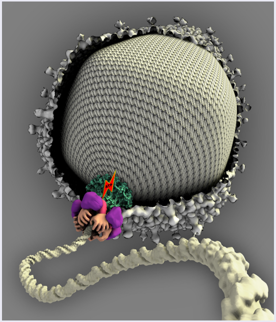
Motivation

- mechanical tasks
- digital & physical twin

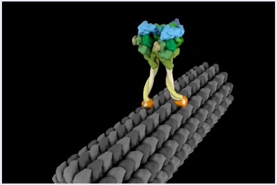
Motivation: molecular length scale

Length-scale: $\sim 10^{-9}$m – $\sim 10^{-6}$m

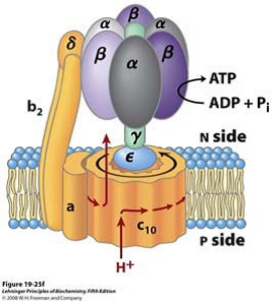
- molecular machines, bio-molecular complexes, ...
- highly specialized, controlled by their structure and the basic laws of physics



Propulsion of flagella



Dynein "walking" along microtubule

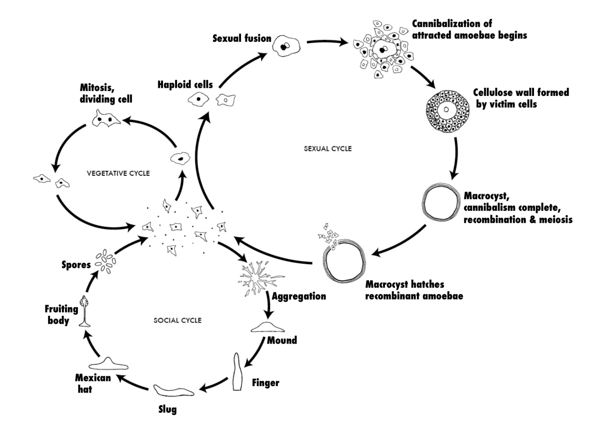


ATP synthase complex in mitochondria

Motivation: single-cell length scale

Length-scale: $\sim 10^{-6}$m – $\sim 10^{-3}$m

- cells, microorganisms ...
- can move, co-operate, react to external stimuli, ...
- example – *slime molds*:



Motivation: complex-organism length scale

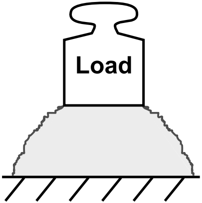
Length-scale: $\sim 10^{-3}$m – $\sim 10^{0}$m

- organisms
- can move, co-operate, react to external stimuli, ...
- example – *fire ants*:

Motivation: synthetic systems → modular robotics

Programmable Matter (PM) – definition

A class of future meta-materials which have the ability to freely and actively change their shape or other physical properties in a programmable and controllable manner.

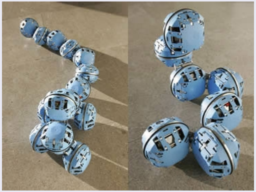


Hardware – provides the ability to *perform physical actions* and *process information* at a very *fine length-scale*.
Software – defines the *actual behavior* of the system, and *can be modified* if needed.

Modular robotic approach

Miche (MIT)

PolyBot (Xerox PARC)



ATRON (Univ. South. Denmark)

and many more

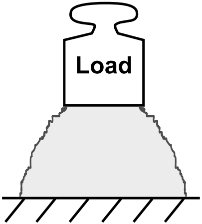
see [Yim et al., 2007]

Motivation: synthetic systems → modular robotics

Programmable Matter (PM) – definition

A class of future meta-materials which have the ability to freely and actively change their shape or other physical properties in a programmable and controllable manner.

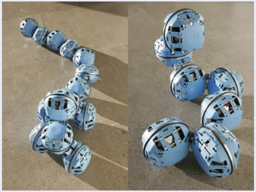
Hardware – provides the ability to *perform physical actions* and *process information* at a very *fine length-scale*.
Software – defines the *actual behavior* of the system, and *can be modified* if needed.



Modular robotic approach

Miche (MIT)

PolyBot (Xerox PARC)



ATRON (Univ. South. Denmark)

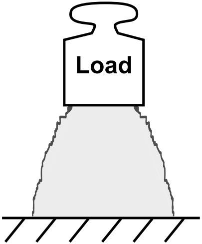
and many more

see [Yim et al., 2007]

Motivation: synthetic systems → modular robotics

Programmable Matter (PM) – definition

A class of future meta-materials which have the ability to freely and actively change their shape or other physical properties in a programmable and controllable manner.

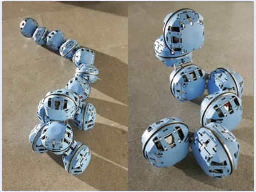


Hardware – provides the ability to *perform physical actions* and *process information* at a very *fine length-scale*.
Software – defines the *actual behavior* of the system, and *can be modified* if needed.

Modular robotic approach

Miche (MIT)

PolyBot (Xerox PARC)

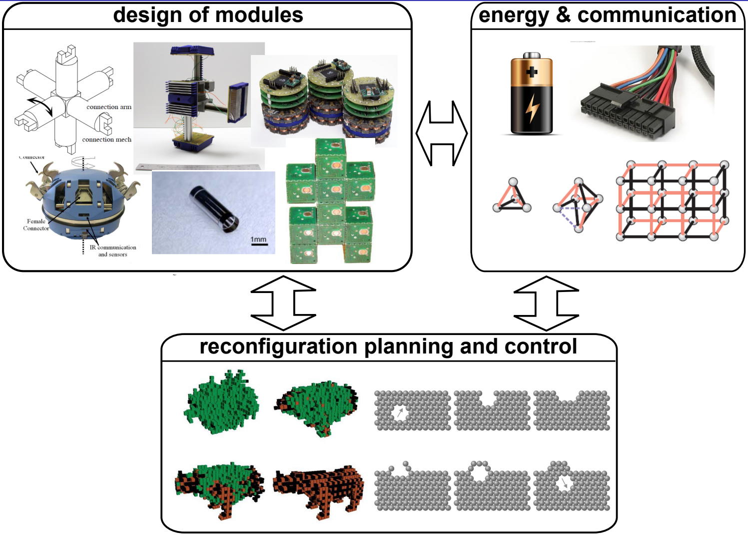


ATRON (Univ. South. Denmark)

and many more

see [Yim et al., 2007]

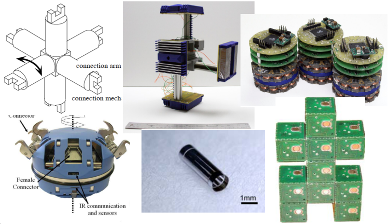
Modular robotic approach: main issues



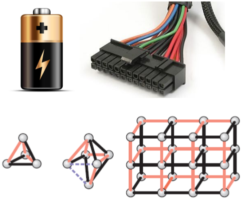
- **miniaturisation** $\Longrightarrow$ co-operation of millions of sub-millimeter-size modules
- interdisciplinary problem $\Longrightarrow$ robotics, mechanics, informatics, physics(?), ...

Modular robotic approach: main issues

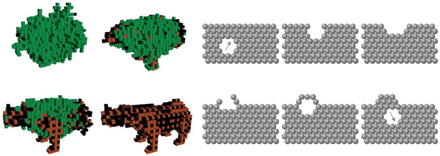
design of modules



energy & communication



reconfiguration planning and control



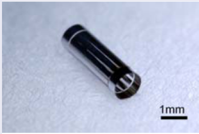
- **miniaturisation** $\Longrightarrow$ co-operation of millions of sub-millimeter-size modules
- interdisciplinary problem $\Longrightarrow$ robotics, mechanics, informatics, physics(?), ...

Towards miniaturization – suitable hardware designs

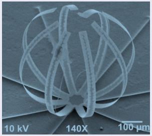
Spherical/cylindrical designs with no moving parts (catom)

[Kirby et al., IROS 2007], [Karagozler et al., IROS 2009], [Reid et al., NSTI Nanotech, 2008]

magnetic catom $r \simeq 30mm$



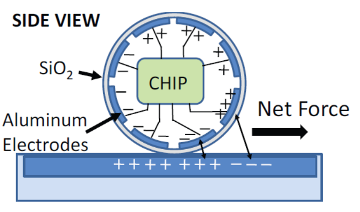
electrostatic catom



spherical catom

Electrostatic actuation mechanism

[M.E. Karagozler et al., proc. ISCAS, 2011]

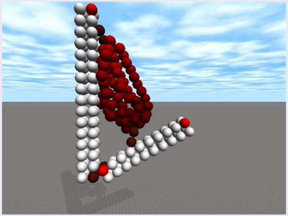


Predictions for the spherical catom:

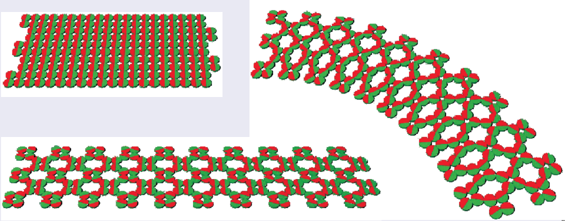
- radius $r = 65\mu m$
- max. lift $f_{\max} \approx 1000$ catoms
- mass $m = 0.69$ microgram
- max torque $\tau_{\max} = 16\,\mathrm{pNm}$

What can we do with a large amount of modules?

Collective actuation

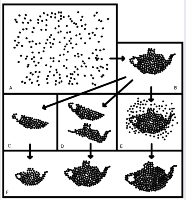


[Christensen et al., Neur.Comp.Appl., 2010]

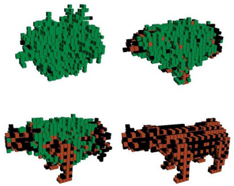


[J. Campbell and P. Pillai, IJRR, 2008]

Pure shape-change problems $\Longrightarrow$ transforming sculptures



[Rubenstein et al., IROS, 2009],



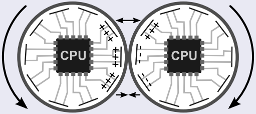
[Stoy, Springer, 2006]

Module design assumptions: assumed physical properties

Our idea: weak&active vs. strong&static connections

weak & active

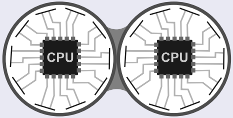
- is relatively weak
- can move w.r.t. neighbors
- can attach/detach
- existing prototypes: electromagnetic/electrostatic cylinders/spheres



weak & active

strong & static

- is much stronger
- is static (immobile)
- can attach/detach
- question how to realize this (e.g. self-soldering connections, Neubert et al., 2014)



strong & static

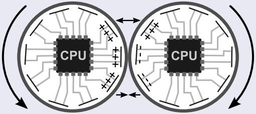
→ Special evolving modular structures need to be designed

Module design assumptions: assumed physical properties

Our idea: weak&active vs. strong&static connections

weak & active

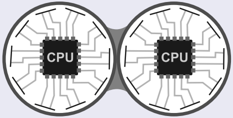
- is relatively weak
- can move w.r.t. neighbors
- can attach/detach
- existing prototypes: electromagnetic/electrostatic cylinders/spheres



weak & active

strong & static

- is much stronger
- is static (immobile)
- can attach/detach
- question how to realize this (e.g. self-soldering connections, Neubert et al., 2014)

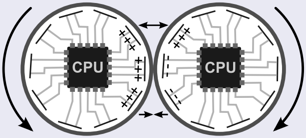


strong & static

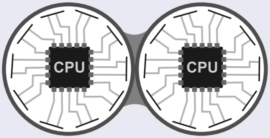
→ Special evolving modular structures need to be designed

Module design assumptions: assumed physical properties

Our idea: weak&active vs. strong&static connections

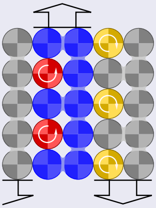


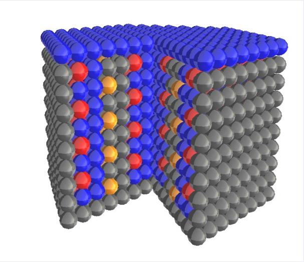
weak (propelling) connection [red & yellow]



strong (fixed) connection [gray & blue]

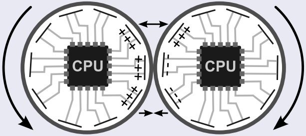
Example: actuator microstructures



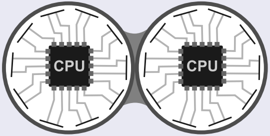


Module design assumptions: assumed physical properties

Our idea: weak&active vs. strong&static connections

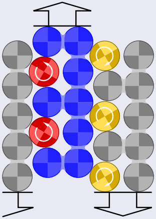


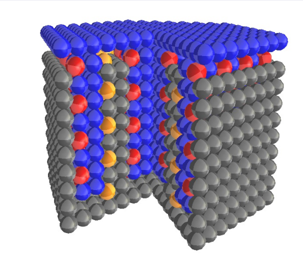
weak (propelling) connection [red & yellow]



strong (fixed) connection [gray & blue]

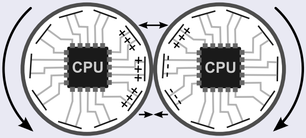
Example: actuator microstructures



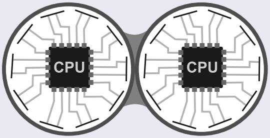


Module design assumptions: assumed physical properties

Our idea: weak&active vs. strong&static connections

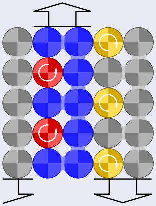


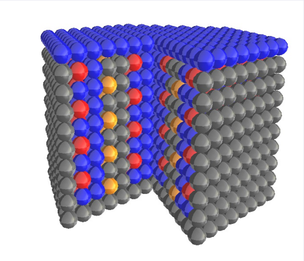
weak (propelling) connection [red & yellow]



strong (fixed) connection [gray & blue]

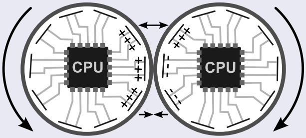
Example: actuator microstructures



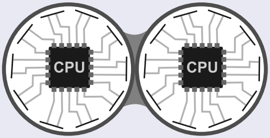


Module design assumptions: assumed physical properties

Our idea: weak&active vs. strong&static connections

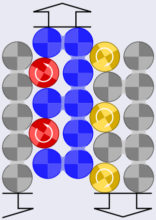


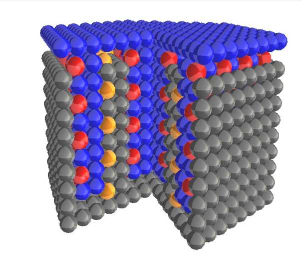
weak (propelling) connection [red & yellow]



strong (fixed) connection [gray & blue]

Example: actuator microstructures

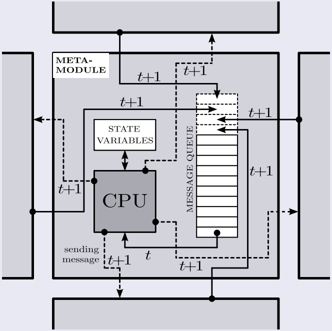




Module design assumptions: computation and data flow

Organization of information exchange and processing

- modular robot is an asynchronous distributed computing architecture – MIMD (multiple instruction, multiple data)
- independent CPU and memory is located in every module
- communication with direct neighbors
- possibility to reconfigure

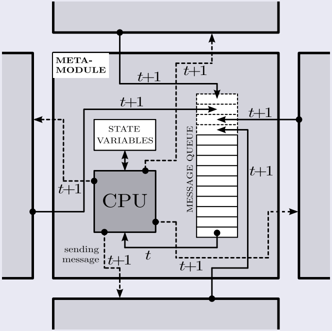


→ special distributed algorithms need to be developed

Module design assumptions: computation and data flow

Organization of information exchange and processing

- modular robot is an asynchronous distributed computing architecture – MIMD (multiple instruction, multiple data)
- independent CPU and memory is located in every module
- communication with direct neighbors
- possibility to reconfigure

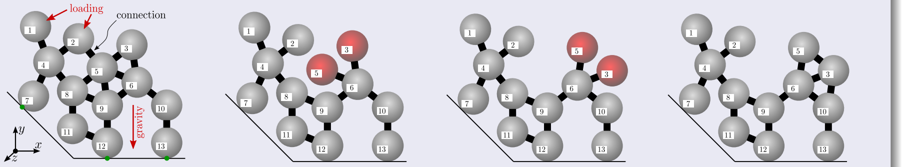


→ special distributed algorithms need to be developed

Outline

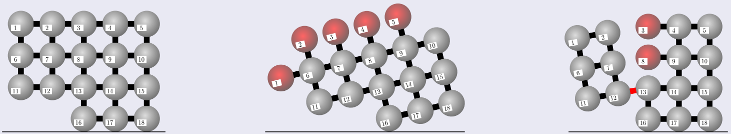
Possible reconfiguration failures – problem statement

Safe reconfiguration scenario



initial step 1 step 2 step 3

Mechanical failures during reconfiguration



initial scenario 1 scenario 2

[P. Hołobut and J. Lengiewicz, Distributed computation of forces in modular-robotic ensembles as part of reconfigur. planning, *Proceedings of the IEEE International Conference on Robotics and Automation*, ***2017***]

Mechanical problem to be solved

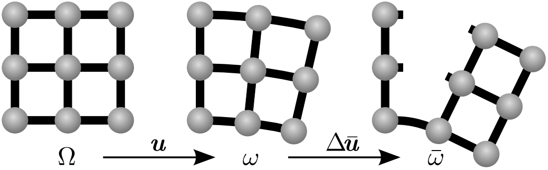


Figure: Initial Ω, loaded ω and perturbed $\bar{\omega}$ configurations

- modular robot $\longrightarrow$ simplified frame model of modular structure

$$\boldsymbol{K}^{11}_{pq} = \frac{E}{L^3}\begin{pmatrix} AL^2 & 0 & 0 \\ 0 & 12I & 6IL \\ 0 & 6IL & 4IL^2 \end{pmatrix}, \qquad \boldsymbol{K}^{12}_{pq} = \frac{E}{L^3}\begin{pmatrix} -AL^2 & 0 & 0 \\ 0 & -12I & 6IL \\ 0 & -6IL & 2IL^2 \end{pmatrix},$$

- assembled system of equations for deformed configuration ω

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F},$$

- assembled system of eqs. for perturbed (to be predicted) configuration $\bar{\omega}$

$$\bar{\boldsymbol{K}}(\boldsymbol{u} + \Delta\bar{\boldsymbol{u}}) = \bar{\boldsymbol{F}},$$

Distributed solution

Weighted Jacobi iterative scheme

$$\bar{\boldsymbol{K}}(\boldsymbol{u} + \Delta\bar{\boldsymbol{u}}) = \bar{\boldsymbol{F}}$$

- calculations performed by the distributed system itself
- weighted Jacobi's (iterative) distributed solution scheme to predict deformation and forces for a desired reconfiguration step
- iteration $i+1$ for the module p (with its neighbors q):

$$\Delta\bar{\boldsymbol{u}}_p^{i+1} = \frac{2}{3}\bar{\boldsymbol{D}}_p^{-1}\left(\bar{\boldsymbol{F}}_p - \bar{\boldsymbol{f}}_p - \bar{\boldsymbol{R}}_p\Delta\bar{\boldsymbol{u}}_p^i - \sum_q \bar{\boldsymbol{K}}_{pq}^{12}\Delta\bar{\boldsymbol{u}}_q^i\right) + \frac{1}{3}\Delta\bar{\boldsymbol{u}}_p^i,$$

  where diagonal and remainder parts of $\bar{\boldsymbol{K}}$ read

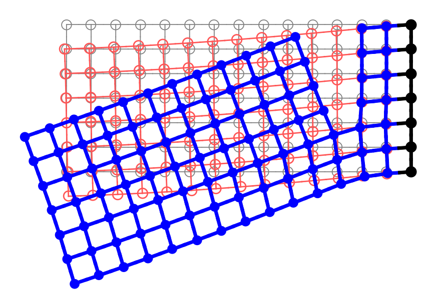
$$\bar{\boldsymbol{D}}_p = \mathrm{diag}\left(\sum_q \bar{\boldsymbol{K}}_{pq}^{11}\right) \quad \text{and} \quad \bar{\boldsymbol{R}}_p = \left(\sum_q \bar{\boldsymbol{K}}_{pq}^{11}\right) - \bar{\boldsymbol{D}}_p$$

- the scheme only relies on local information from neighbours at each iteration

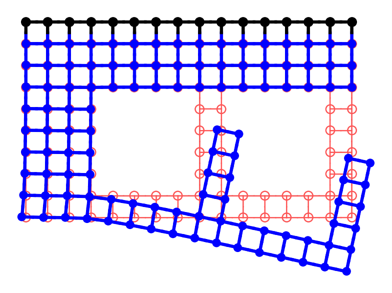
Two basic examples

Examples

- four connections are planned to be released
- three configurations: undeformed, deformed and predicted



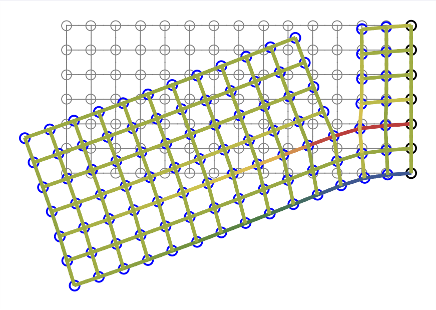
Example 1



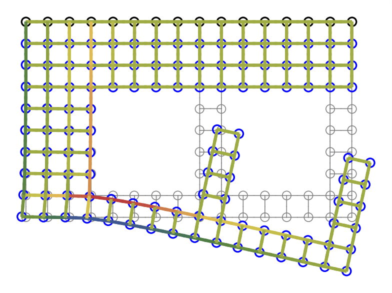
Example 2

Results

- axial/shear/bending stresses can be predicted



Example 1 (axial forces)

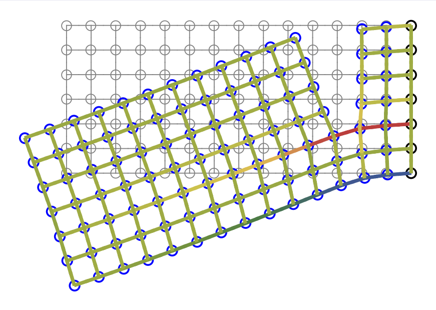


Example 2 (axial forces)

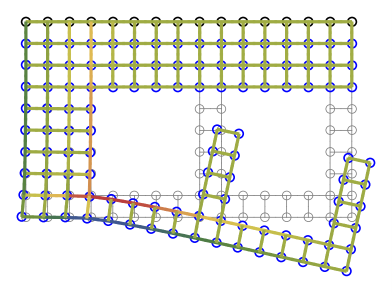
- reconfiguration planner now can decide if the predicted forces are acceptable
- Success!

Results

- axial/shear/bending stresses can be predicted



Example 1 (axial forces)

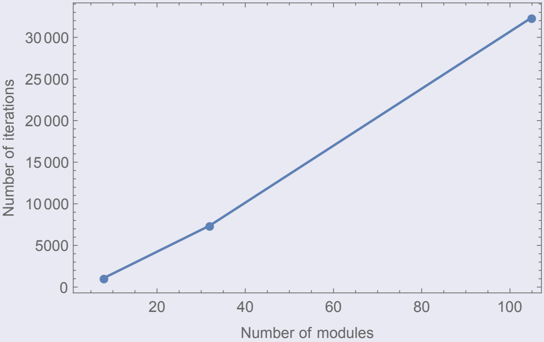


Example 2 (axial forces)

- reconfiguration planner now can decide if the predicted forces are acceptable
- Success!???

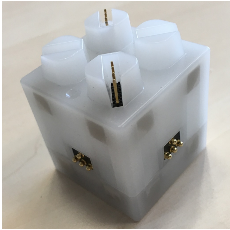
Problems...

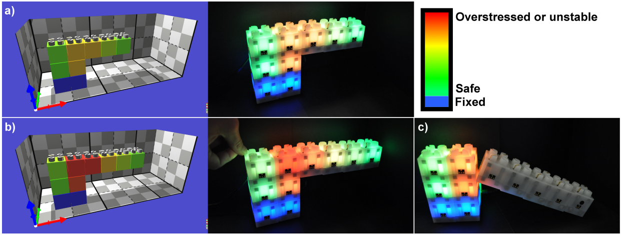
Time complexity



- CPU time scales poorly with increasing system size – improvements needed
- weighted-Jacobi scheme is only conditionally convergent
- distributed solution schemes of computational mechanics can be adapted
  - → multigrid methods
  - → multi-scale modeling (coarsening)

Implementation on the *Blinky Blocks* modular robot



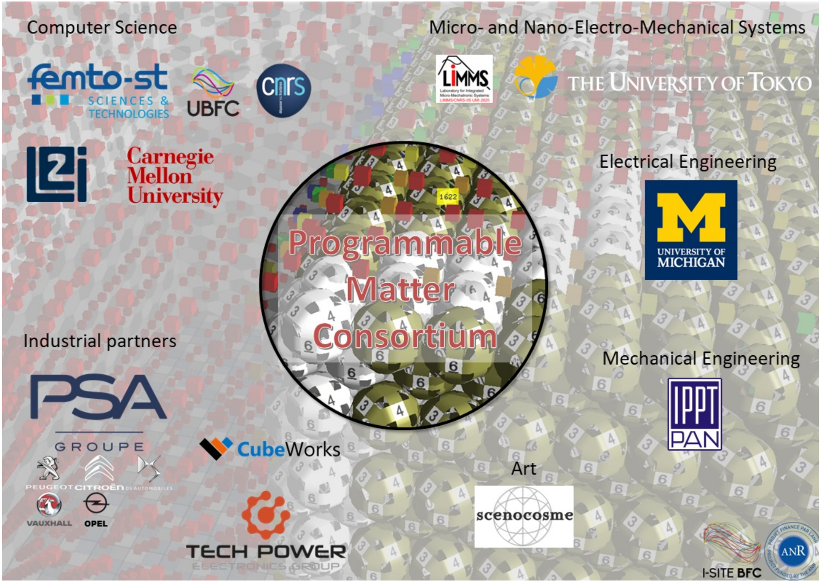


proof-of-concept

- cooperation with the group of Prof. J. Bourgeois, femto-st/University of Franche-Comté (France)
- cubic topology, magnetic connection, message passing, visualization with colors
- individual connection strength identified experimentally
- our algorithm is run on a robot + experimental validation of results

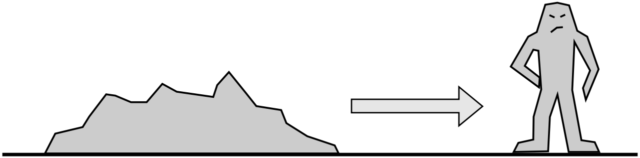
[7] B. Piranda, P. Chodkiewicz, P. Hołobut, S. Bordas, J. Bourgeois and J. Lengiewicz, Distributed autonomous detection of mechanically unsafe reconfiguration scenarios, *work in progress*.

Informal consortium: IPPT PAN – Uni.Lu – femto-st et Al.



Outline

Problem of pure shape change & locomotion



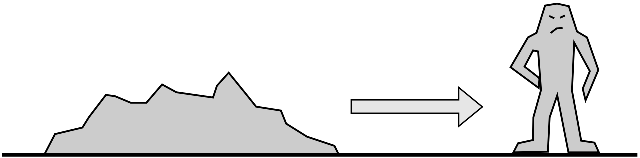
assumptions:

- system under external load (e.g. gravity)
- mobile (active) connections are weak (may be locked beneath the surface)
- CPU time << communication time << module physical motion time

aim:

- perform reconfiguration from initial to a desired shape
- minimize overall reconfiguration time

Possible approaches to reconfiguration



usual approach:

- translation of the surface modules only (e.g., Bourgeois et al., 2016)
- drawback: relatively high overall reconfiguration time

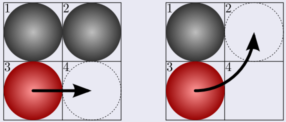
our approach:

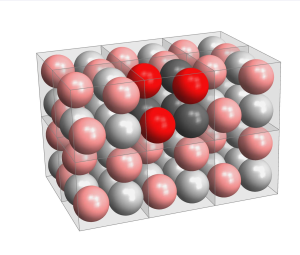
- maintain a porous frame structure and engage a **volume** of modules to move
- expected improved overall reconfiguration time

[6] J. Lengiewicz and P. Hołobut, Efficient Collective Shape Shifting and Locomotion of Massively-Modular Robotic Structures, *Autonomous Robots*, DOI: 10.1007/s10514-018-9709-6.

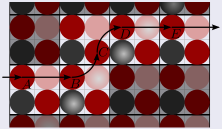
Flow in porous frame structure (1)

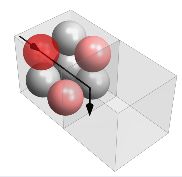
Elementary moves and porous structure made of meta-modules

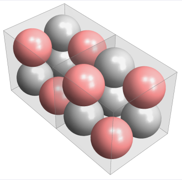




volumetric flow and outflow/inflow through the surface



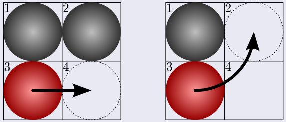


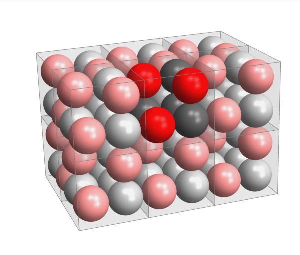


- flow along non-intersecting streamlines

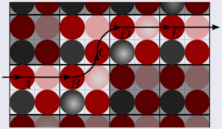
Flow in porous frame structure (1)

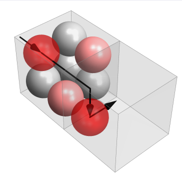
Elementary moves and porous structure made of meta-modules

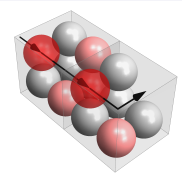




volumetric flow and outflow/inflow through the surface



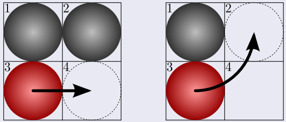


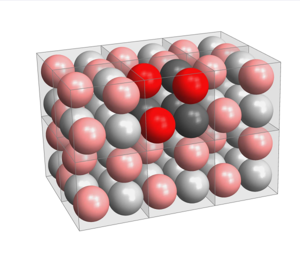


- flow along non-intersecting streamlines

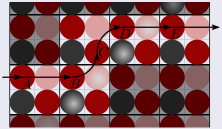
Flow in porous frame structure (1)

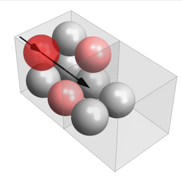
Elementary moves and porous structure made of meta-modules

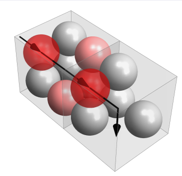




volumetric flow and outflow/inflow through the surface



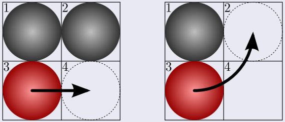


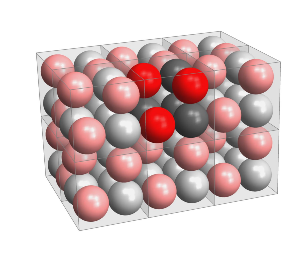


- flow along non-intersecting streamlines

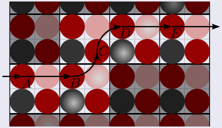
Flow in porous frame structure (1)

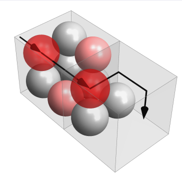
Elementary moves and porous structure made of meta-modules

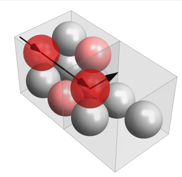




volumetric flow and outflow/inflow through the surface



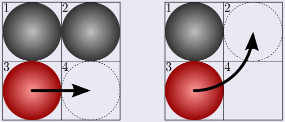


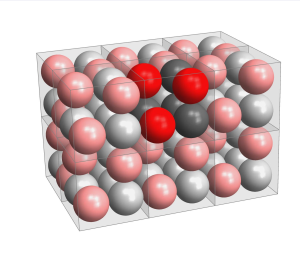


- flow along non-intersecting streamlines

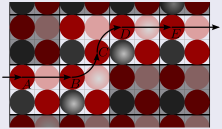
Flow in porous frame structure (1)

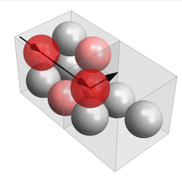
Elementary moves and porous structure made of meta-modules

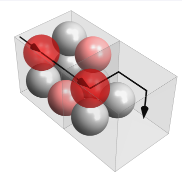




volumetric flow and outflow/inflow through the surface



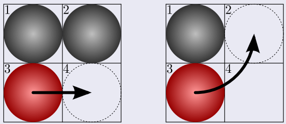


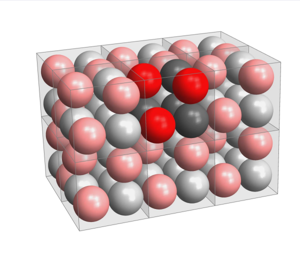


- flow along non-intersecting streamlines

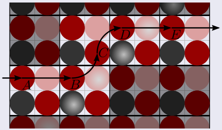
Flow in porous frame structure (1)

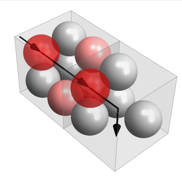
Elementary moves and porous structure made of meta-modules

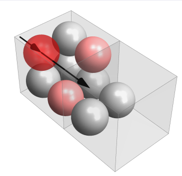




volumetric flow and outflow/inflow through the surface



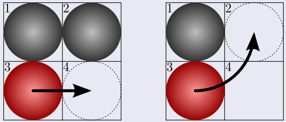


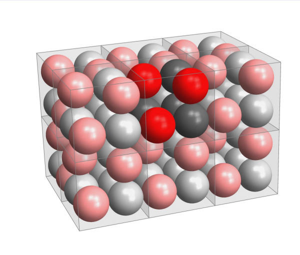


- flow along non-intersecting streamlines

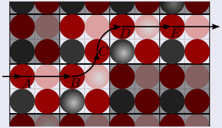
Flow in porous frame structure (1)

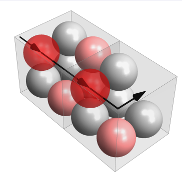
Elementary moves and porous structure made of meta-modules

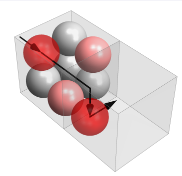




volumetric flow and outflow/inflow through the surface



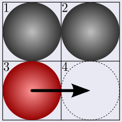


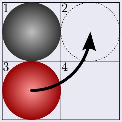


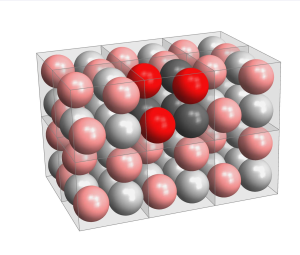
- flow along non-intersecting streamlines

Flow in porous frame structure (1)

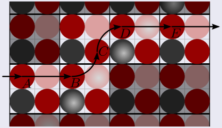
Elementary moves and porous structure made of meta-modules

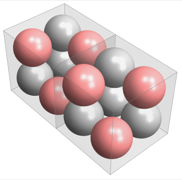


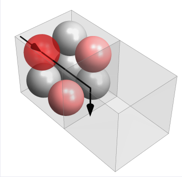




volumetric flow and outflow/inflow through the surface



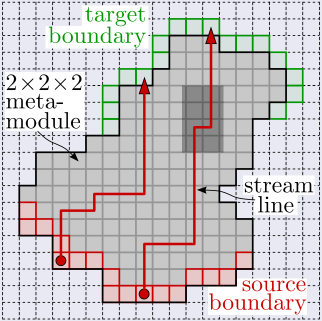




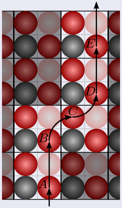
- flow along non-intersecting streamlines

Flow in porous frame structure (2)

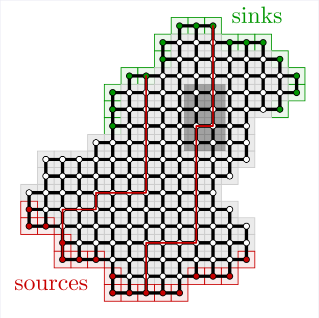
Flow as a graph (meta-module as a graph vertex)



Robot built of meta-modules



Flow of modules



Graph corresponding to the robot

- reconfiguration step as a flow through a porous structure
- problem to find a maximum set of disjoint streamlines

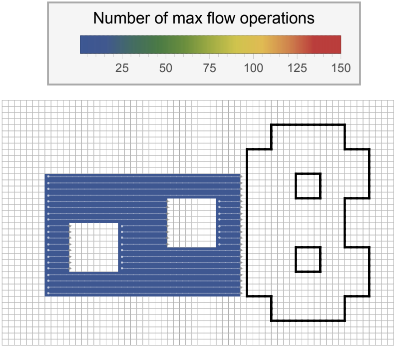
→ integral max-flow problem with unit vertex capacity to be solved

→ distributed algorithm run by the modular robot itself needs to be used

Shape transformation example

- multi-step procedure
- time lapse 4x

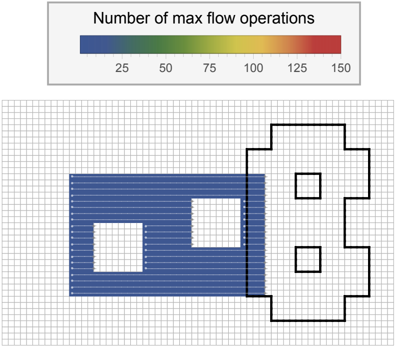
step 04:



Shape transformation example

- multi-step procedure
- time lapse 4x

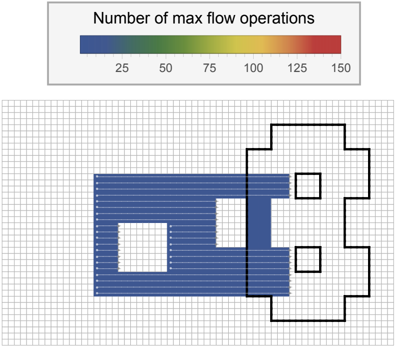
step 08:



Shape transformation example

- multi-step procedure
- time lapse 4x

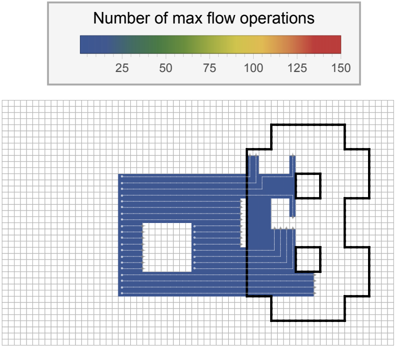
step 12:



Shape transformation example

- multi-step procedure
- time lapse 4x

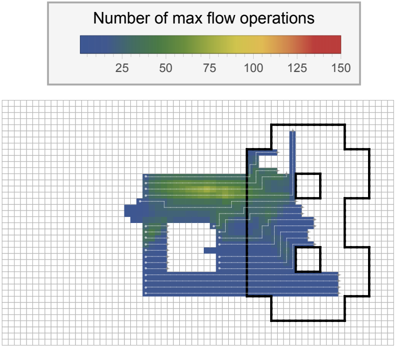
step 16:



Shape transformation example

- multi-step procedure
- time lapse 4x

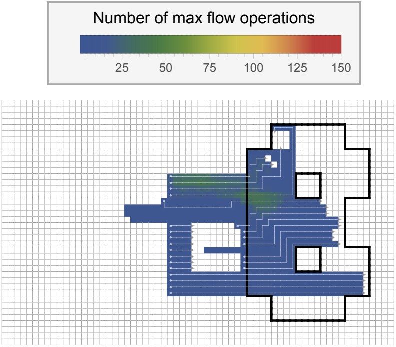
step 20:



Shape transformation example

- multi-step procedure
- time lapse 4x

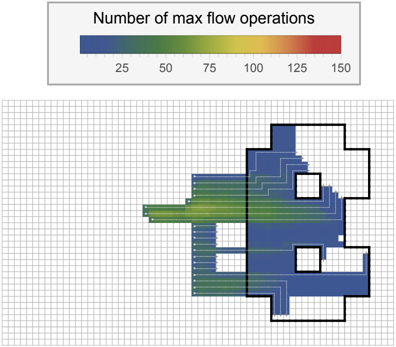
step 24:



Shape transformation example

- multi-step procedure
- time lapse 4x

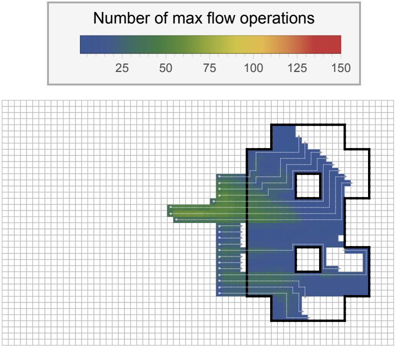
step 28:



Shape transformation example

- multi-step procedure
- time lapse 4x

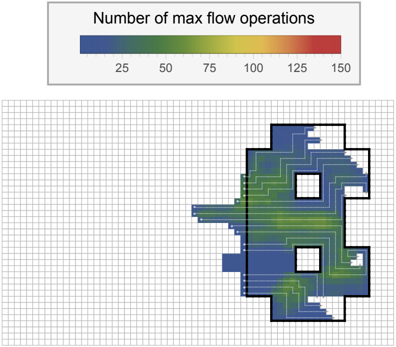
step 32:



Shape transformation example

- multi-step procedure
- time lapse 4x

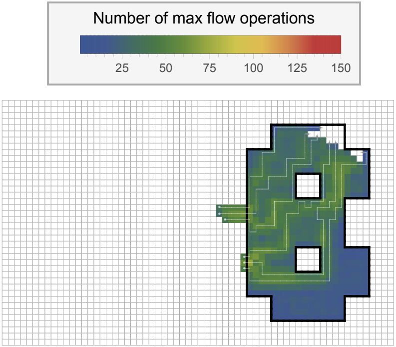
step 36:



Shape transformation example

- multi-step procedure
- time lapse 4x

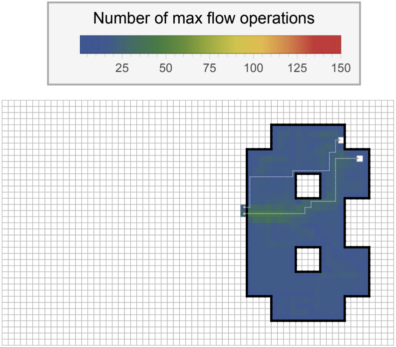
step 40:



Shape transformation example

- multi-step procedure
- time lapse 4x

step 44:



CPU/time/memory complexity assessments

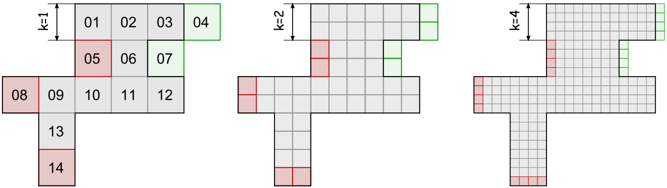


Figure: Three different resolutions k of the same geometry

Table: Assessed complexities for the increasing resolution k in p dimensions.

	pure algorithm	algorithm with heuristics
N.of steps	$\sim k$	$\sim k$
Avg. iterations/step	$\sim k^p$	$\sim k^{p-1}$
Avg. CPU/step	$\sim k^{2p-1}$	$\sim k^{2p-2}$
Avg. CPU/step/module	$\sim k^{p-1}$	$\sim k^{p-2}$
Max. memory/module	$\sim k^{p-1}$	$\sim k^{p-1}$

- N.of steps scales favorably but CPU time may be a bottleneck.

CPU/time/memory complexity assessments

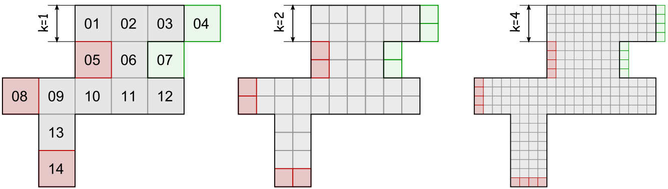


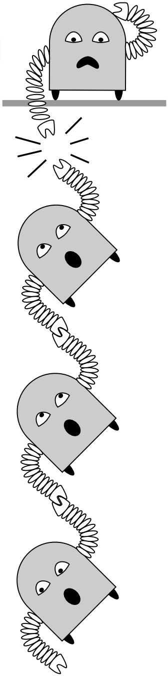
Figure: Three different resolutions k of the same geometry

Table: Assessed complexities for the increasing resolution k in p dimensions.

	pure algorithm	algorithm with heuristics
N.of steps	$\sim k$	$\sim k$
Avg. iterations/step	$\sim k^p$	$\sim k^{p-1}$
Avg. CPU/step	$\sim k^{2p-1}$	$\sim k^{2p-2}$
Avg. CPU/step/module	$\sim k^{p-1}$	$\sim k^{p-2}$
Max. memory/module	$\sim k^{p-1}$	$\sim k^{p-1}$

- N.of steps scales favorably but CPU time may be a bottleneck.

Thank you!



[1] P. Hołobut, M. Kursa, and J. Lengiewicz, A class of microstructures for scalable collective actuation of Programmable Matter, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* ***2014***.

[2] P. Hołobut, M. Kursa, and J. Lengiewicz, Efficient modular-robotic structures to increase force-to-weight ratio of scalable collective actuators, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* ***2015***.

[3] P. Hołobut, P. Chodkiewicz, A. Macios, and J. Lengiewicz, Internal localization algorithm based on relative positions for cubic-lattice modular-robotic ensembles, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* ***2016***.

[4] P. Hołobut and J. Lengiewicz, Distributed computation of forces in modular-robotic ensembles as part of reconfiguration planning, *Proceedings of the IEEE International Conference on Robotics and Automation* ***2017***

[5] J. Lengiewicz, M. Kursa, and P. Hołobut, Modular-Robotic Structures for Scalable Collective Actuation, *Robotica*, **2017**.

[6] J. Lengiewicz and P. Hołobut, Efficient Collective Shape Shifting and Locomotion of Massively-Modular Robotic Structures, *Autonomous Robots*, **2019**.

[7] B. Piranda, P. Chodkiewicz, P. Hołobut, S. Bordas, J. Bourgeois and J. Lengiewicz, Distributed autonomous detection of mechanically unsafe reconfiguration scenarios, *work in progress*.