A Constraint Hierarchies Approach to Geometric Constraints on Sketches

MOSIM 2008 / SAC-GCR 2008

Christophe Jermann
LINA – Université de Nantes

Hiroshi Hosobe
National Institute of Informatics Tokyo
Motivation

- A geometric constraint system in 2D:
  - Entities: 3 points A, B, C, 3 lines D, E, F
  - Constraints: 3 pt-pt distances, 6 pt-ln incidences
Motivation

• A geometric constraint system in 2D:
  • Entities: 3 points A,B,C, 3 lines D,E,F
  • Constraints: 3 pt-pt distances, 6 pt-ln incidences
Motivation

- A geometric constraint system in 2D:
  - Entities: 3 points A, B, C, 3 lines D, E, F
  - Constraints: 3 pt-pt distances, 6 pt-ln incidences
Motivation

• A geometric constraint system in 2D:
  • Entities: 3 points A,B,C, 3 lines D,E,F
  • Constraints: 3 pt-pt distances, 6 pt-ln incidences

The sketch expresses the designer’s intents
→ Solvers should take the sketch into account!
Motivation

- A geometric constraint system in 3D:
  - Entities: 1 point P, 2 lines L₁, L₂
  - Constraints: Fixed(P), Fixed(L₁), Parallelism(L₁,L₂), Incidence(P,L₂), Ortho_distance(L₁,L₂,4)
Motivation

• A geometric constraint system in 3D:
  • Entities: 1 point P, 2 lines L₁, L₂
  • Constraints: Fixed(P), Fixed(L₁), Parallelism(L₁,L₂), Incidence(P,L₂), Ortho_distance(L₁,L₂,4)

No solution!!!!!
Motivation

- A geometric constraint system in 3D:
  - Entities: 1 point $P$, 2 lines $L_1$, $L_2$
  - Constraints: Fixed($P$), Fixed($L_1$), Parallelism($L_1$, $L_2$), Incidence($P$, $L_2$), Ortho-distance($L_1$, $L_2$, 4)
Motivation

• A geometric constraint system in 3D:
  • Entities: 1 point P, 2 lines L₁, L₂
  • Constraints: Fixed(P), Fixed(L₁), Parallelism(L₁,L₂), Incidence(P,L₂), Ortho_distance(L₁,L₂,4)
Motivation

• A geometric constraint system in 3D:
  • Entities: 1 point P, 2 lines L₁, L₂
  • Constraints: Fixed(P), Fixed(L₁), Parallelism(L₁,L₂), Incidence(P,L₂), Ortho_distance(L₁,L₂,4)
Motivation

• A geometric constraint system in 3D:
  • Entities: 1 point P, 2 lines L₁, L₂
  • Constraints: Fixed(P), Fixed(L₁), Parallelism(L₁,L₂), Incidence(P,L₂), Ortho_distance(L₁,L₂,4)

Debugging a geometric constraint system is tedious
→ Solvers should relax constraints as needed!
Outline

Background
1. Geometric Constraints Systems
   - Definitions
   - Considered Method
2. Constraint Hierarchies
   - Definitions
   - Considered Method

Contribution
3. Geometric Constraint Hierarchies
   - Definitions, advantages
   - Proposed Method
4. Conclusions
   - Discussion
   - Future work
1- Geometric Constraints Systems

- Geometric constraint systems (GCSs) = entities + constraints (w.r.t. a geometric universe)
- Solution = configuration of the entities that satisfies all the constraints
- Considered solving method: flow-based decomposition-recombination planner (+ numerical evaluation / IBB) 

\[ Hoffmann \text{ et al. } 97-00, \text{ Jermann et al. } 03 \]

Example:

- Entities:
  Point P, Lines \( L_1, L_2 \)
- Constraints:
  \( c1: \text{Fixed}(P) \)
  \( c2: \text{Fixed}(L_1) \)
  \( c3: \text{Parallelism}(L_1, L_2) \)
  \( c4: \text{Incidence}(P, L_2) \)
  \( c5: \text{Ortho\_distance}(L_1, L_2, 4) \)

Failure : no solution returned!
2- Constraint Hierarchies

- Constraint Hierarchies = variables + constraints + strengths + solution criterion
- Solution = an assignment that is not dominated by any other w.r.t. the solution criterion
- locally-predicate-better (LPB) = satisfied constraints set inclusion in each hierarchy level considered in decreasing strength order

Example:
- Variables:
  \[ x_1, x_2, x_3 \in \{0,1,2,3\} \]
- Constraints:
  \begin{align*}
  \text{required} & : c_1: x_1 = x_2 \\
  \text{strong} & : c_2: x_2 + 1 = x_3 \\
  \text{weak} & : c_3: x_1 = 0 \\
  \text{weak} & : c_4: x_3 = 3 \\
  \end{align*}
- Assignments:
  \begin{align*}
  (x_1,x_2,x_3) & \rightarrow (\text{sat. const.}) \\
  a_1: (0,2,3) & \rightarrow (\emptyset,\{c_2\},\{c_3,c_4\}) \\
  a_2: (0,0,3) & \rightarrow (\{c_1\},\emptyset,\{c_3,c_4\}) \\
  a_3: (2,2,3) & \rightarrow (\{c_1\},\{c_2\},\{c_4\}) \\
  a_4: (0,0,1) & \rightarrow (\{c_1\},\{c_2\},\{c_3\}) \\
  \end{align*}
- LPB-comparison:
  \begin{align*}
  a_1 & \text{ violates } c_1 \\
  & \rightarrow \text{ not a solution} \\
  a_2 & \leq_{\text{LPB}} a_3 \\
  & \rightarrow \text{ not an LPB-sol.} \\
  a_3 & \geq_{\text{LPB}} a_4, a_4 \geq_{\text{LPB}} a_3 \\
  & \rightarrow \text{ both LPB-sol.} \\
  \end{align*}
2- Constraint Hierarchies

• Considered solving method: maximum matching based identification of a maximal set of satisfiable constraints per hierarchy level (LPB-maximal set of constraints) [Gangnet and Rosenberg, 93]

Example:

• Variables:
  \[ x_1, x_2, x_3 \in \{0,1,2,3\} \]

• Constraints:
  - required \( c_1: x_1 = x_2 \)
  - strong \( c_2: x_2 + 1 = x_3 \)
  - weak \( c_3: x_1 = 0 \)
  - weak \( c_4: x_3 = 3 \)
**3- Geometric Constraint Hierarchies**

- **Idea:** Introduce preferences in GCSs in order to
  - Handle the user’s sketch as a set of very weak constraints (positions, orientations, topology, …)
  - any GCS becomes over-constrained
  - Handle over-constrained GCSs by relaxing constraints automatically according to user’s preferences
  - users achieve the desired solution by playing with preferences instead of debugging his constraints

- **Problem:**
  - Hoffmann et al. method does not handle preferences
  - Gangnet and Rosenberg method does not handle DOFs
3- Geometric Constraint Hierarchies

**Proposed method:**
- A mix of both = prioritized flow-based algorithm
- In each iteration:
  - the introduced constraint $c$ is one of the strongest ones;
  - $c$ is distributed;
  - if it cannot be saturated, it is relaxed.

**Example:**
- **Entities:**
  Point $P$, Lines $L_1$, $L_2$
- **Constraints:**
  - **required** $c_1$: Fixed($P$)
  - **required** $c_2$: Fixed($L_1$)
  - **strong** $c_3$: Parallelism($L_1, L_2$)
  - **weak** $c_4$: Incidence($P, L_2$)
  - **weak** $c_5$: Ortho_distance($L_1, L_2, 4$)

Success: $c_5$ relaxed!
4- Conclusions

- The proposed method:
  - achieves the LPB criterion for GCSs with preferences;
  - has the same complexity as that of Hoffmann et al.;
  - is incremental as that of Gangnet and Rosenberg;

- But:
  - it cannot include as is the “extension step” of Hoffmann et al.’s method
    → could be run in two steps: first LPB-maximal constraint set extraction, second DR-planning
  - it can return an under-constrained LPB-maximal constraint set (due to non-unary DOFs)
    → the relaxed constraints could be taken into account as an optimization criterion (satisfied as much as possible)
4- Conclusions

- How to handle the sketch then?
  - Principle:
    - User’s explicit constraints = Strong
    - Sketch = (very) Weak
      - Fixed positions
      - Fixed orientations
      - Fixed relative positions
    \[\rightarrow\] Fixes remaining DOFs when the GCS is under-constrained
    and/or selects the “best” solution (optimization)

  => Every GCS with a sketch becomes over-constrained

- Achieving the desired solution:
  - user indicates less/more important constraints
    \[\rightarrow\] their priority is de/increased and the solution is updated consequently
4- Conclusions

• Future work:
  • Theoretical and practical comparison to state-of-the-art geometric constraint solvers and constraint hierarchy solvers
  • Application to other fields where constraints and entities hold multiple DOFs (Computer-aided drawing, User interfaces, …)
  • Devise user-friendly interaction schemes that render transparent the use of preferences in order to achieve the desired solution