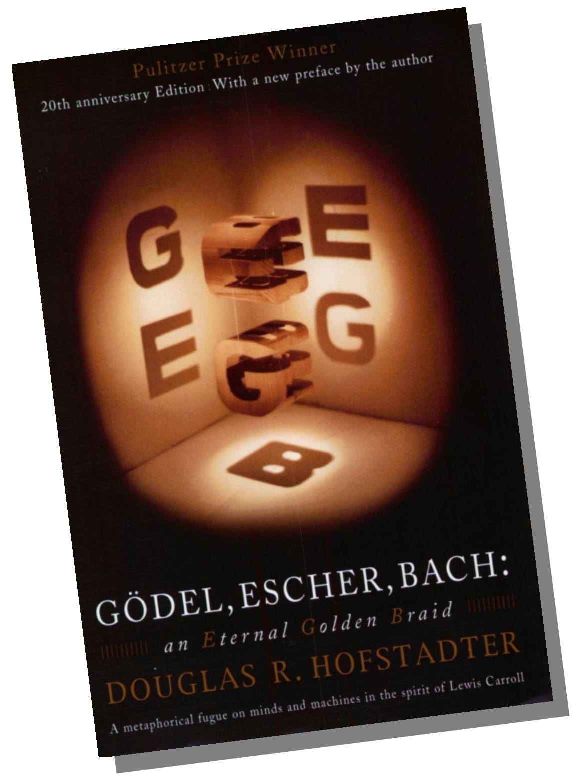
Constructability of Trip-lets

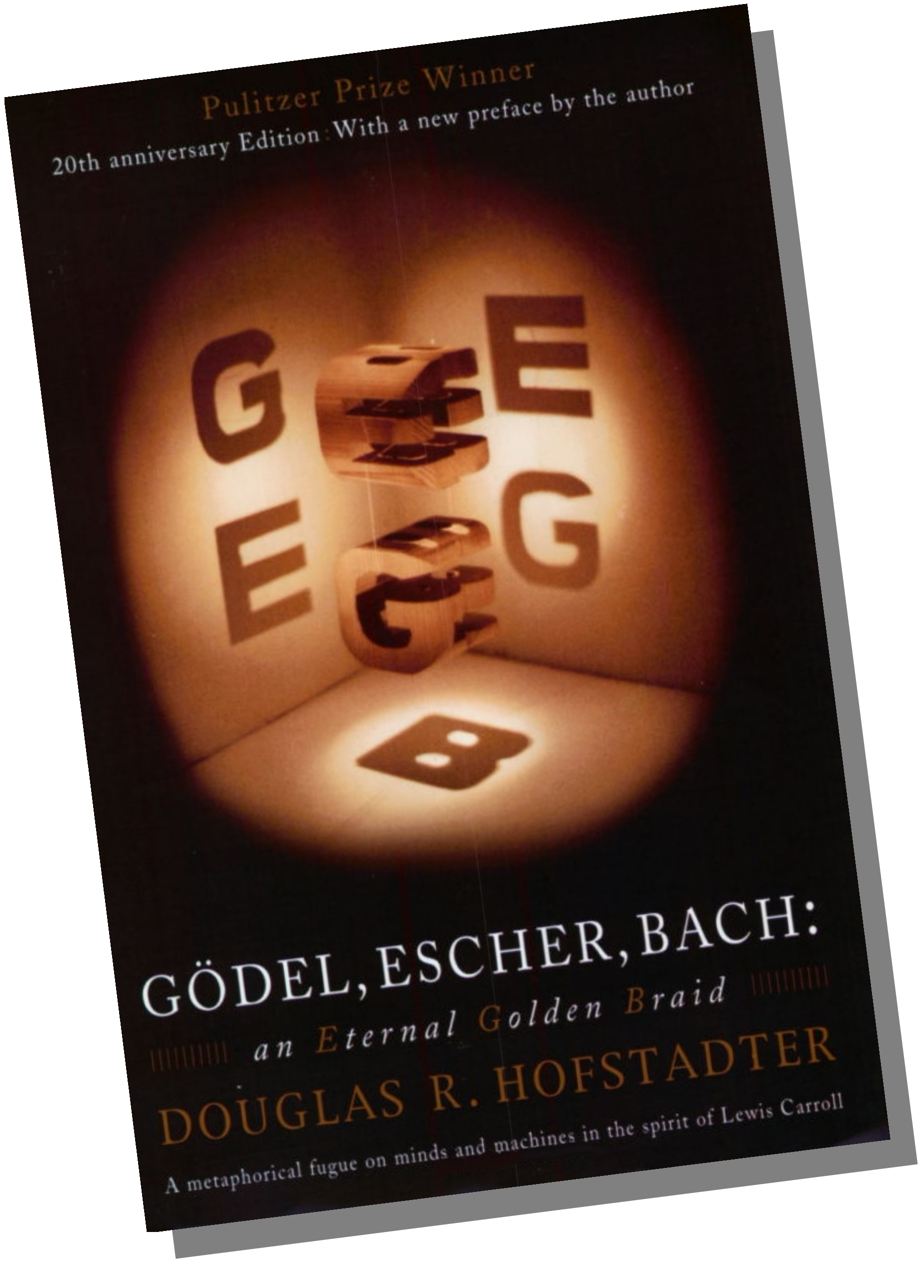


Jeroen Keiren   Freek van Walderveen*   Alexander Wolff

Eindhoven University of Technology, the Netherlands

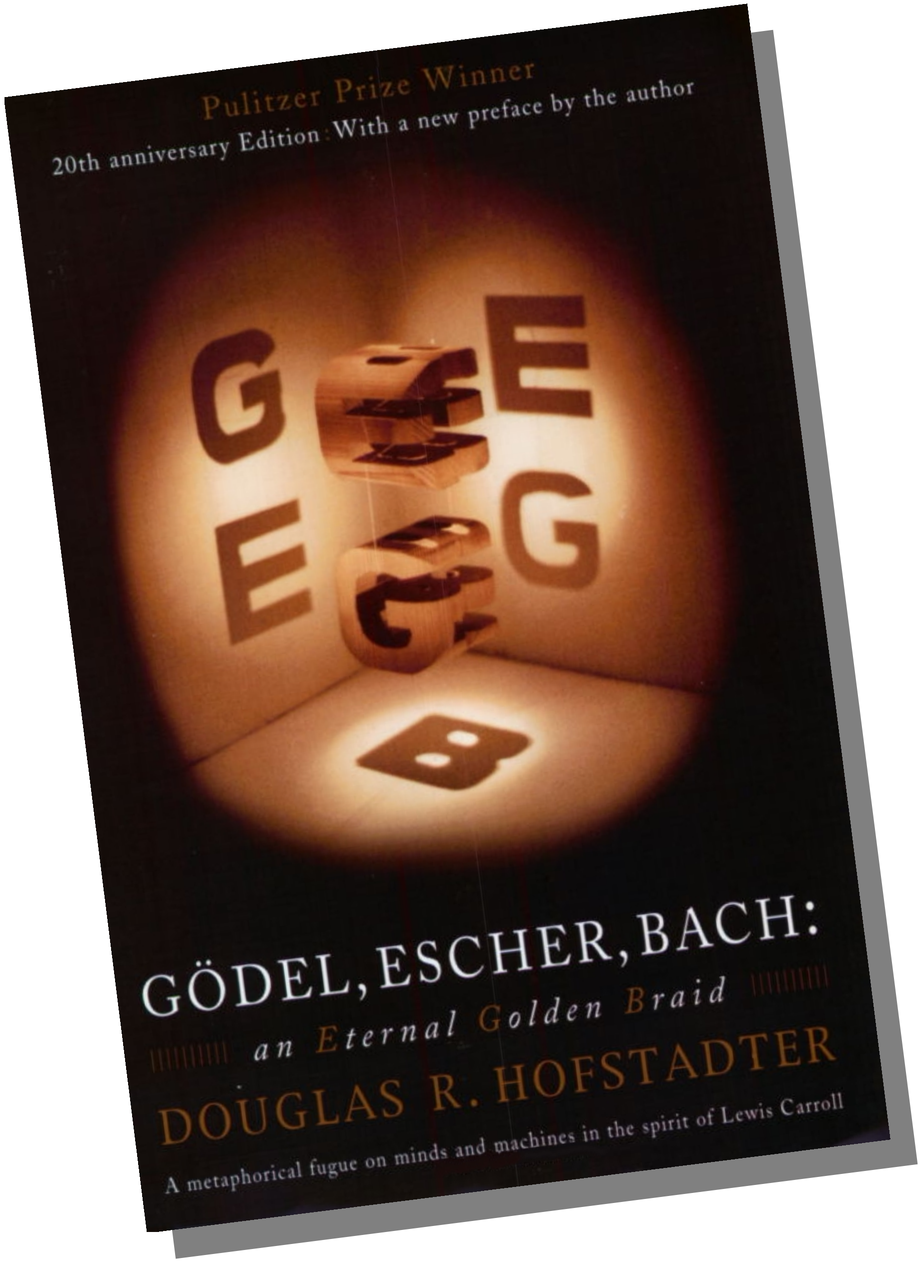
*Now at **MADALGO, University of Aarhus, Denmark**

Trip-lets



> ‘**Trip-let**’ is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters.

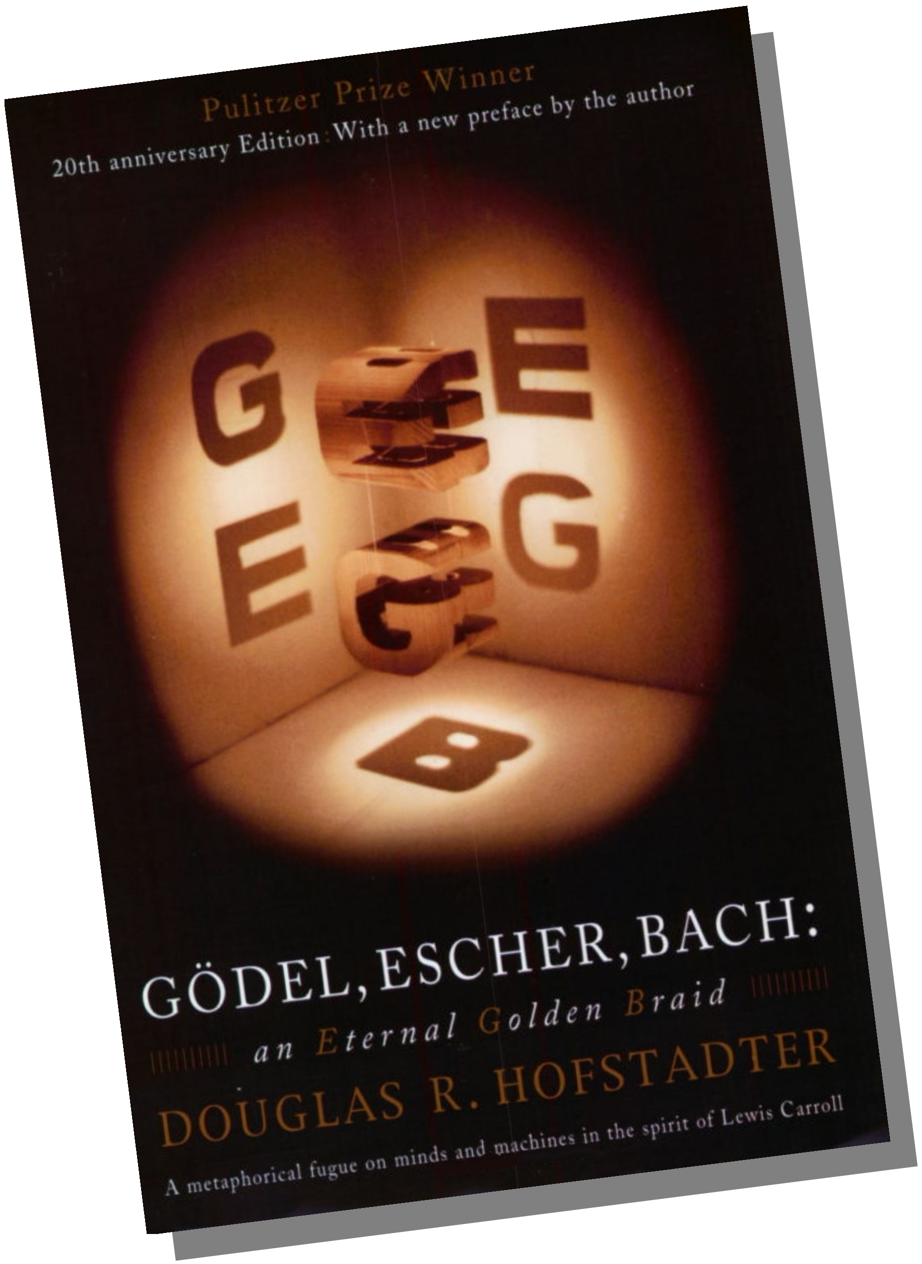
Trip-lets



> “ ‘**Trip-let**’ is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters. ”

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows’ negatives.

Trip-lets



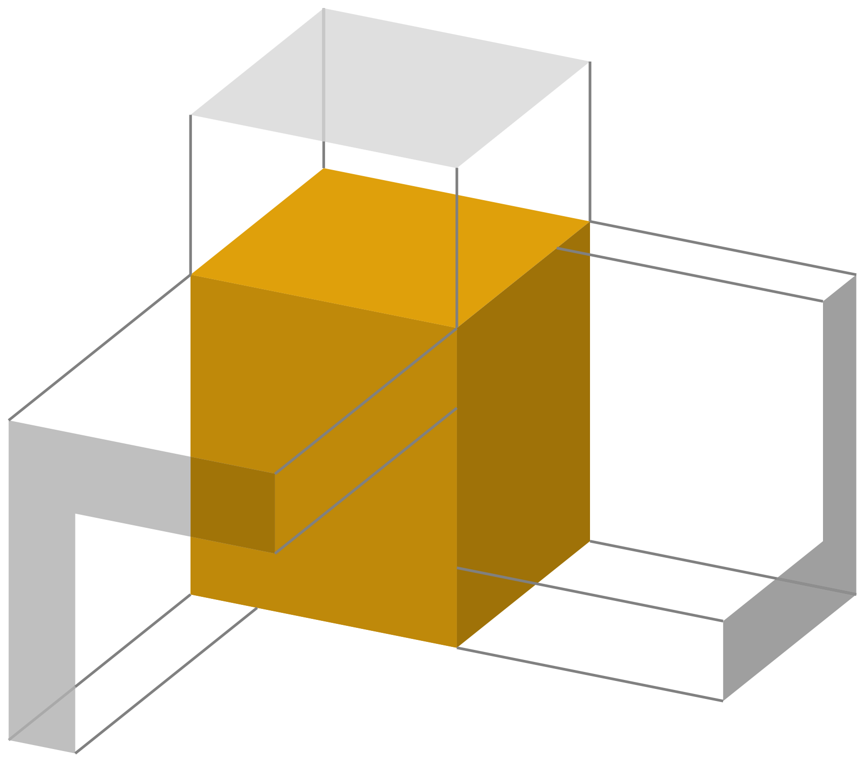
> "'**Trip-let**' is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters."

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows' negatives.

Is this always possible?

Or are there combinations of letters for which it does not work out?

Trip-lets



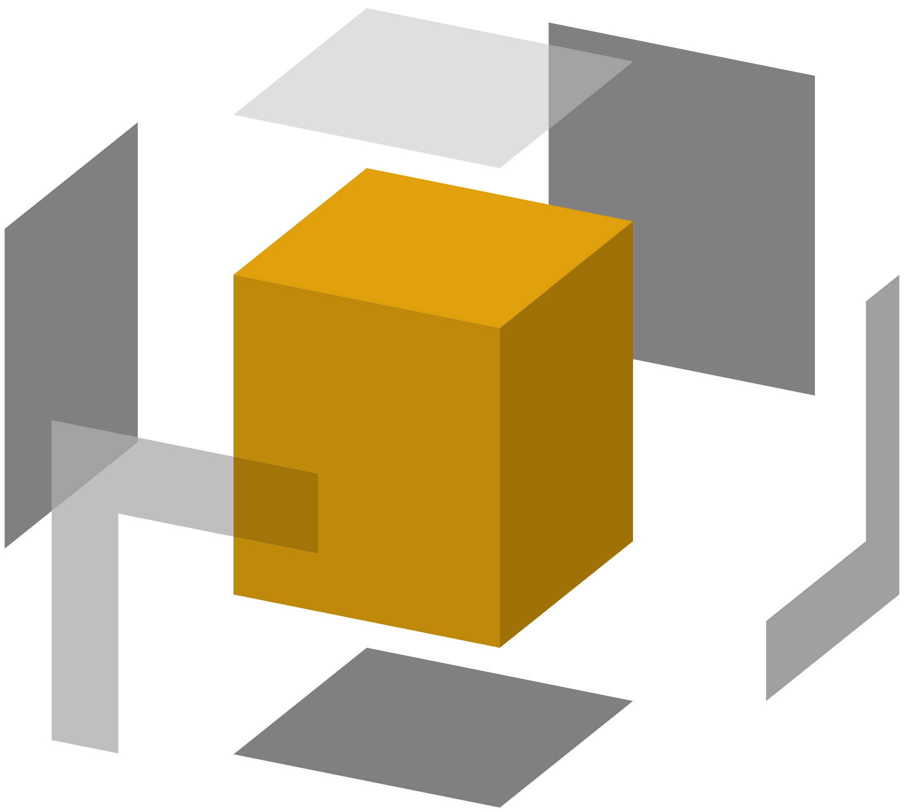
> “‘**Trip-let**’ is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters.”

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows’ negatives.

Is this always possible?

Or are there combinations of letters for which it does not work out?

Trip-lets



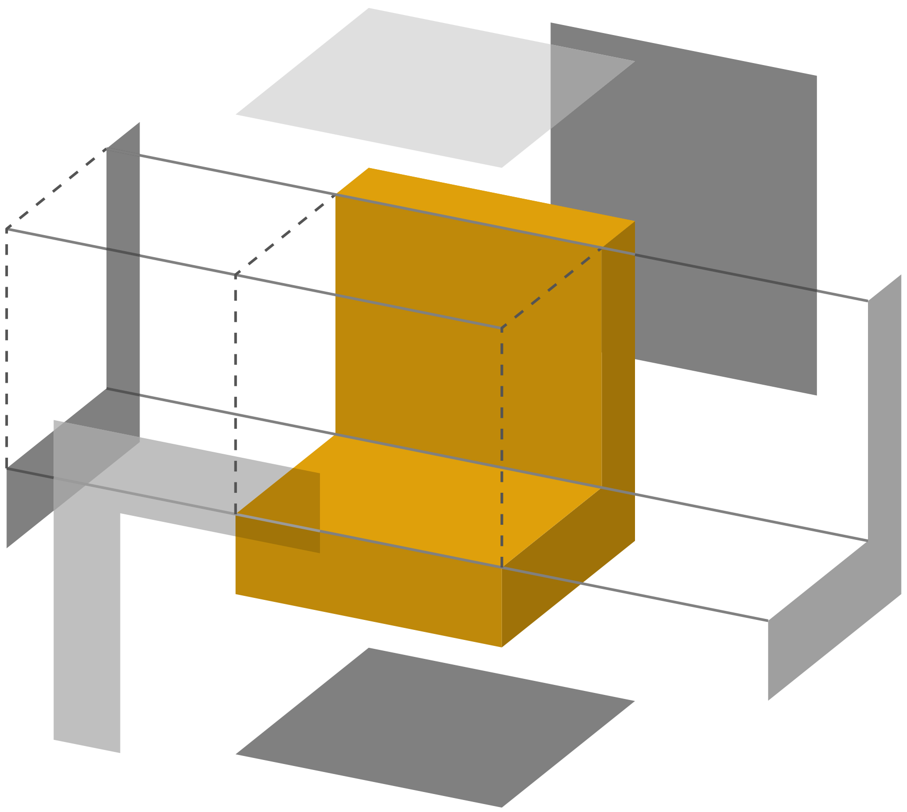
> "'**Trip-let**' is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters."

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows' negatives.

Is this always possible?

Or are there combinations of letters for which it does not work out?

Trip-lets



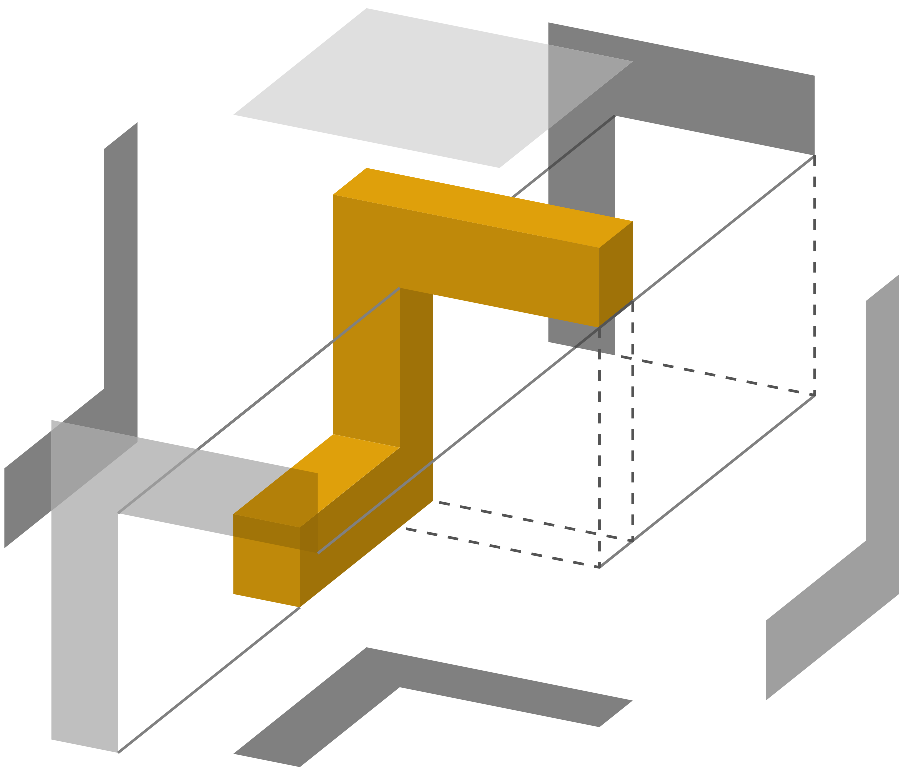
> "'**Trip-let**' is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters."

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows' negatives.

Is this always possible?

Or are there combinations of letters for which it does not work out?

Trip-lets



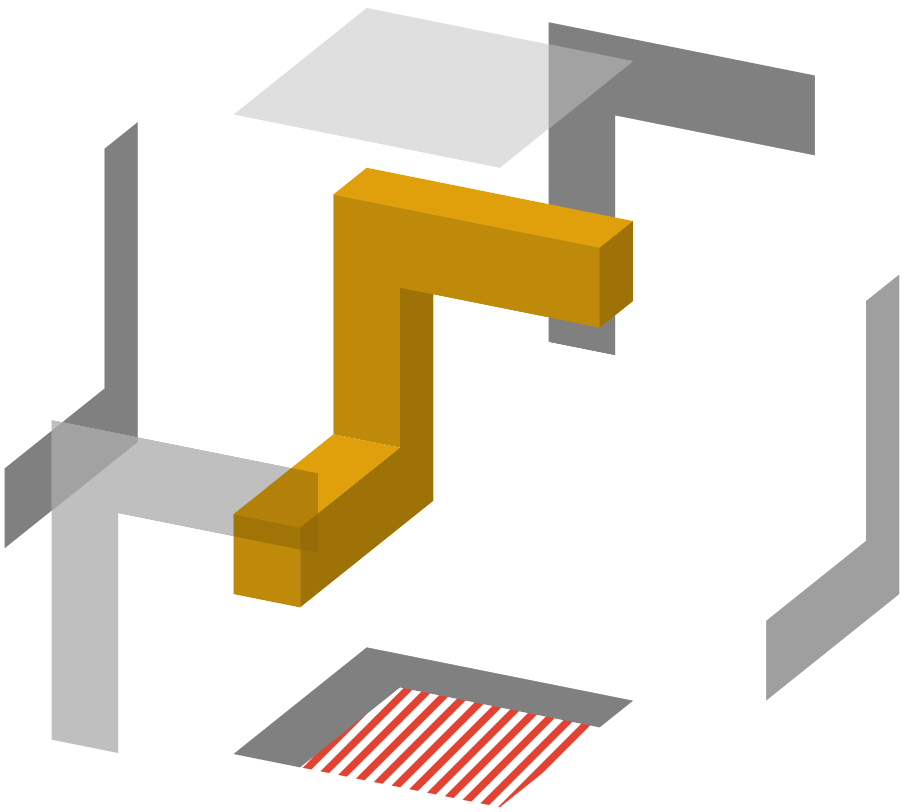
“ ‘**Trip-let**’ is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters. ”

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows’ negatives.

Is this always possible?

Or are there combinations of letters for which it does not work out?

Trip-lets



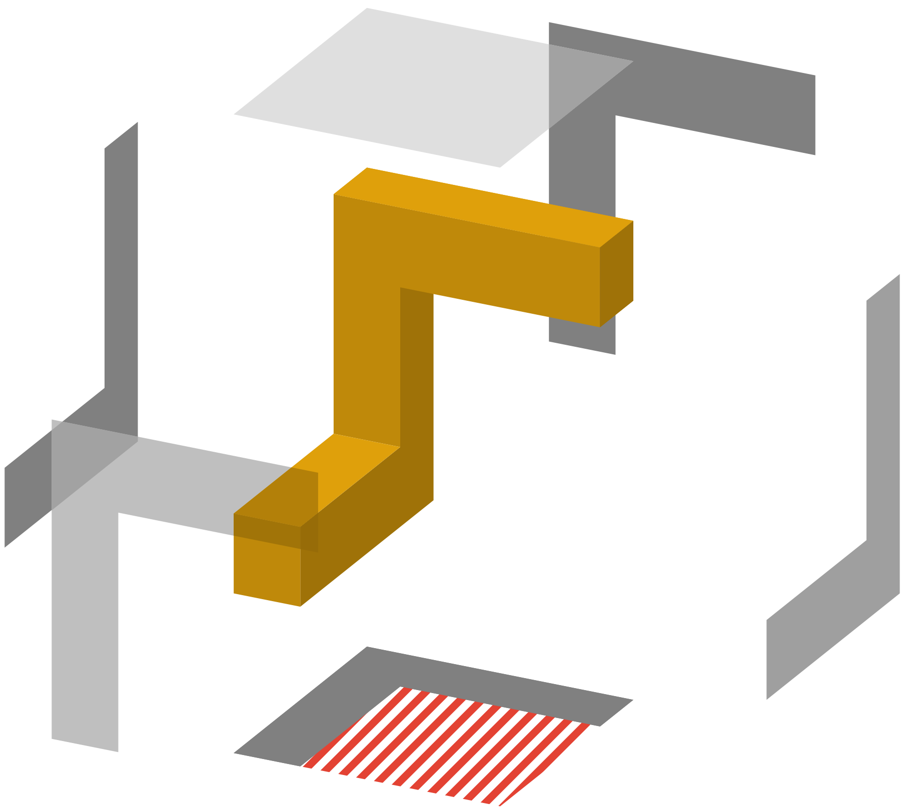
“ ‘**Trip-let**’ is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters. ”

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows’ negatives.

Is this always possible? **No!**

Or are there combinations of letters for which it does not work out?

Trip-lets



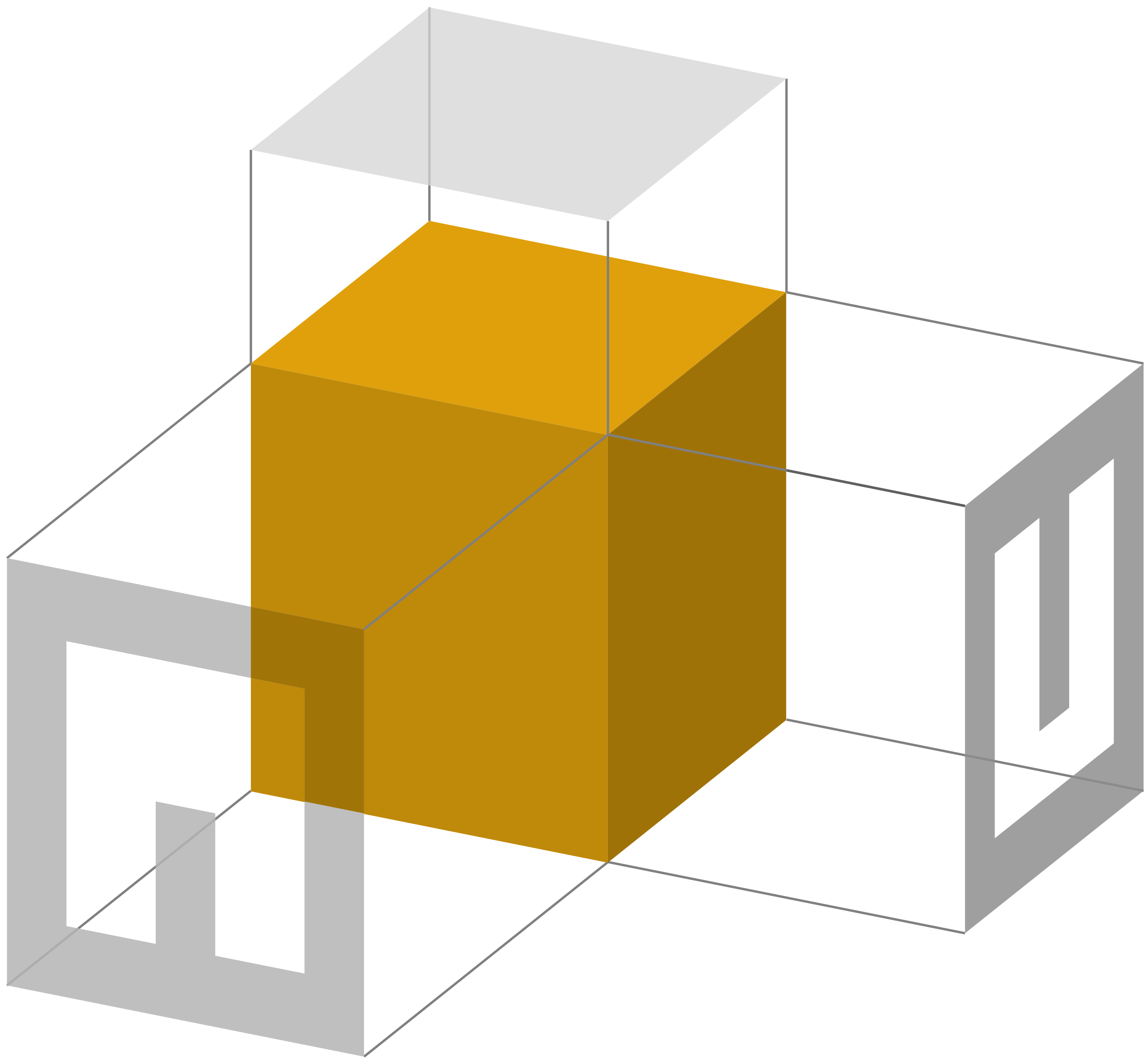
“ ‘**Trip-let**’ is the name which I have given to blocks shaped in such a way that their **shadows** in three orthogonal dimensions are three different letters. ”

Conversely:
construction for three given letters (**shapes**) by removing from a cube all material obstructing the shadows’ negatives.

Question 1:
For which shapes is this possible?

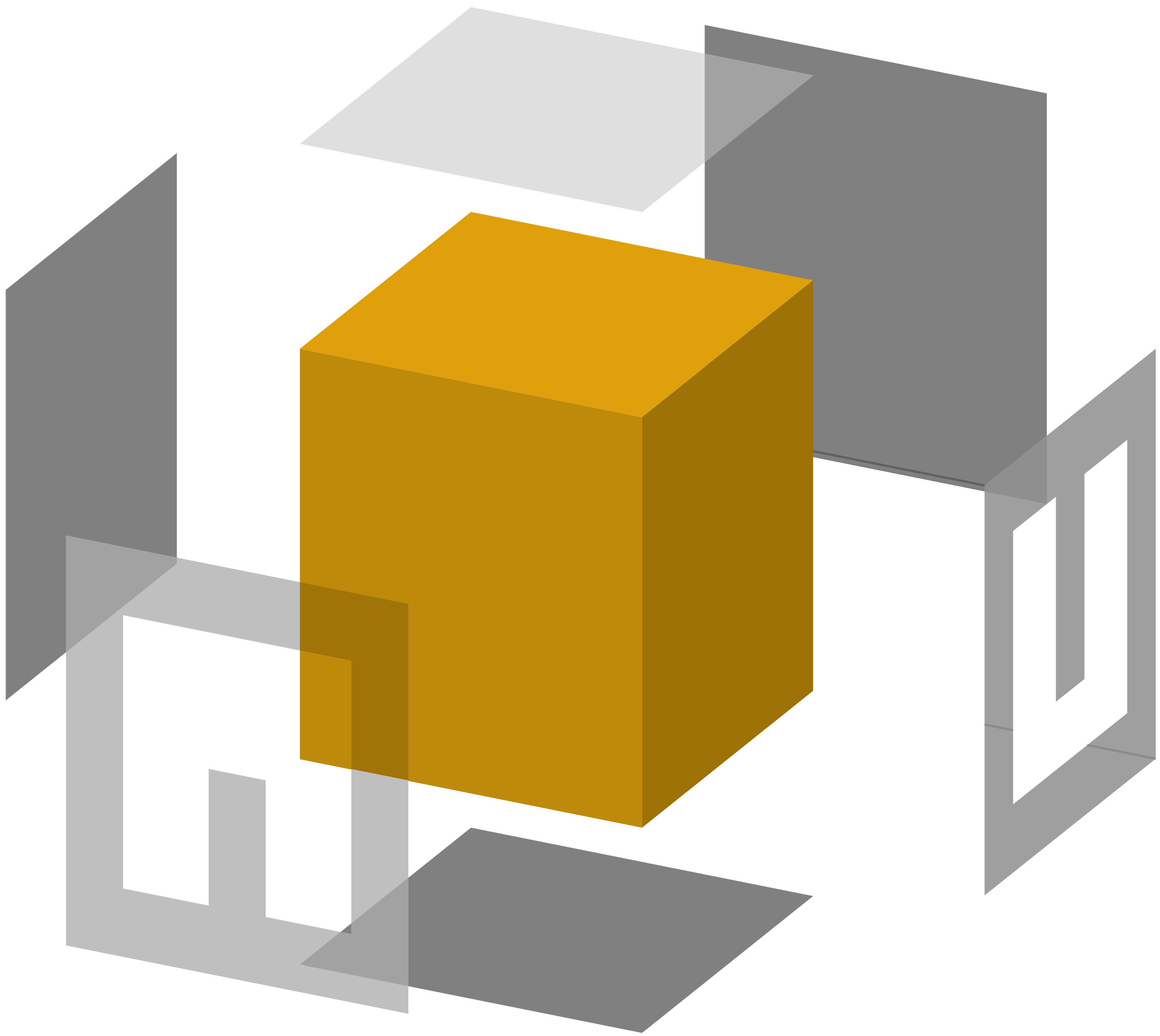
Trip-lets

a second example



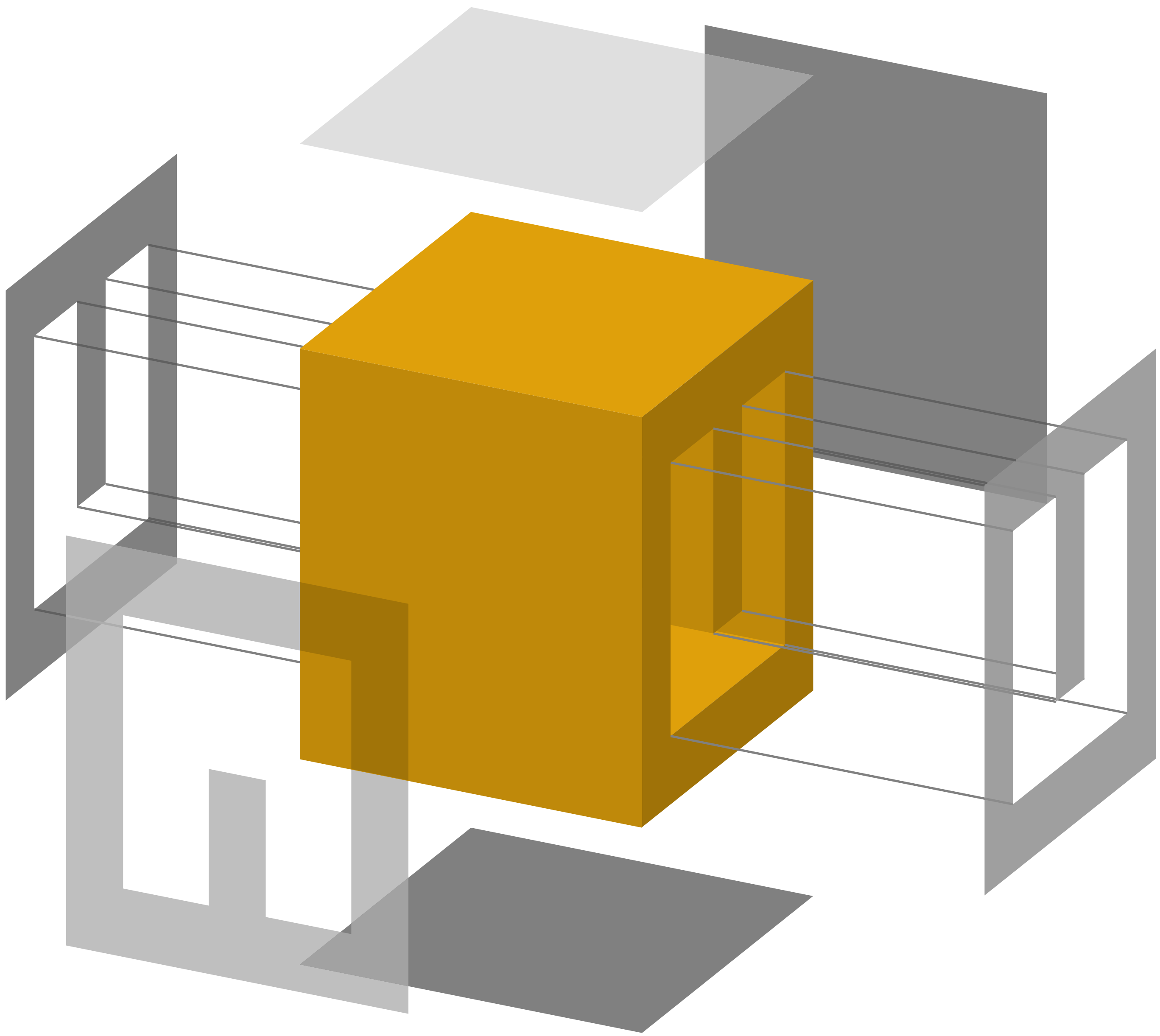
Trip-lets

a second example



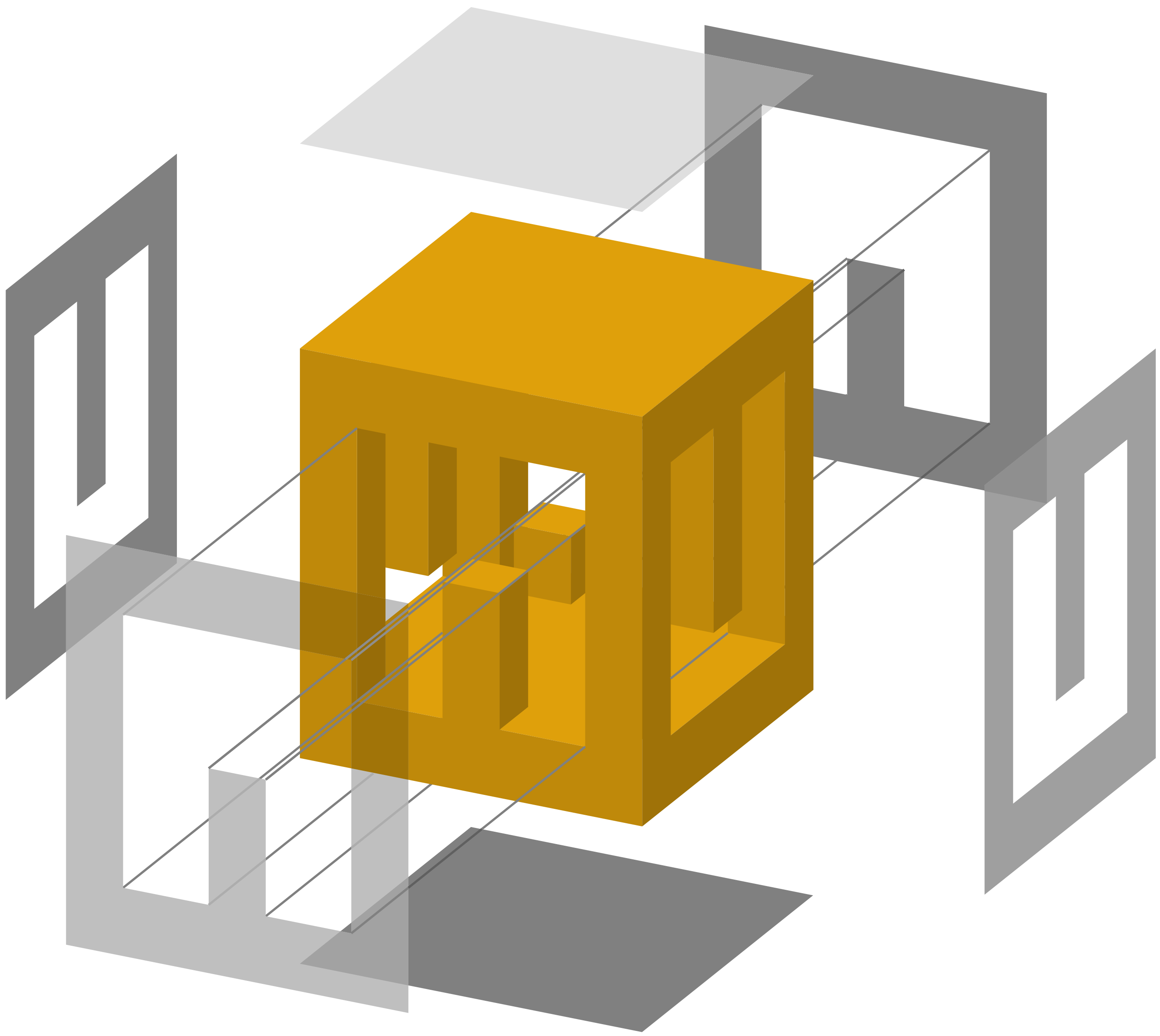
Trip-lets

a second example



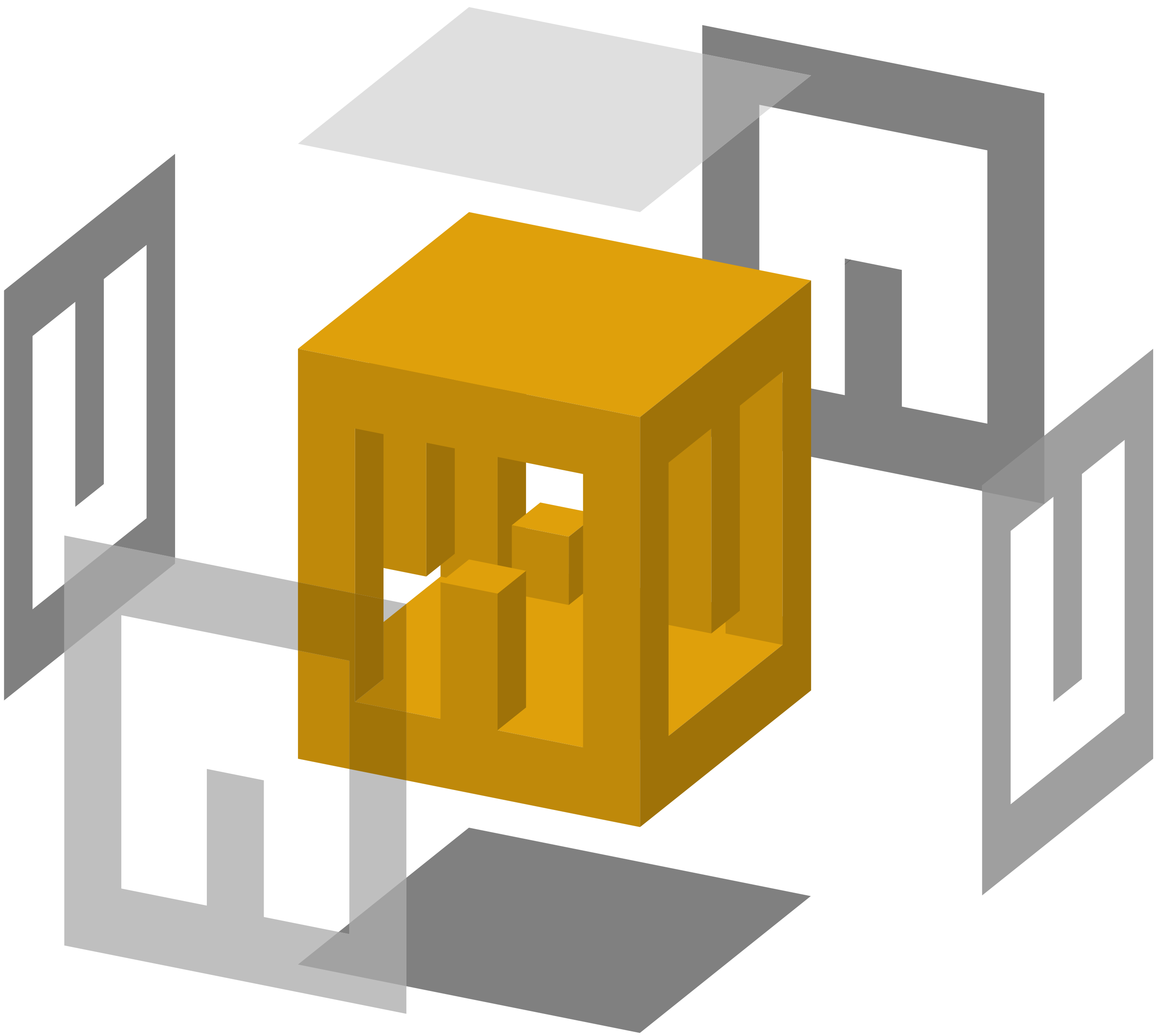
Trip-lets

a second example



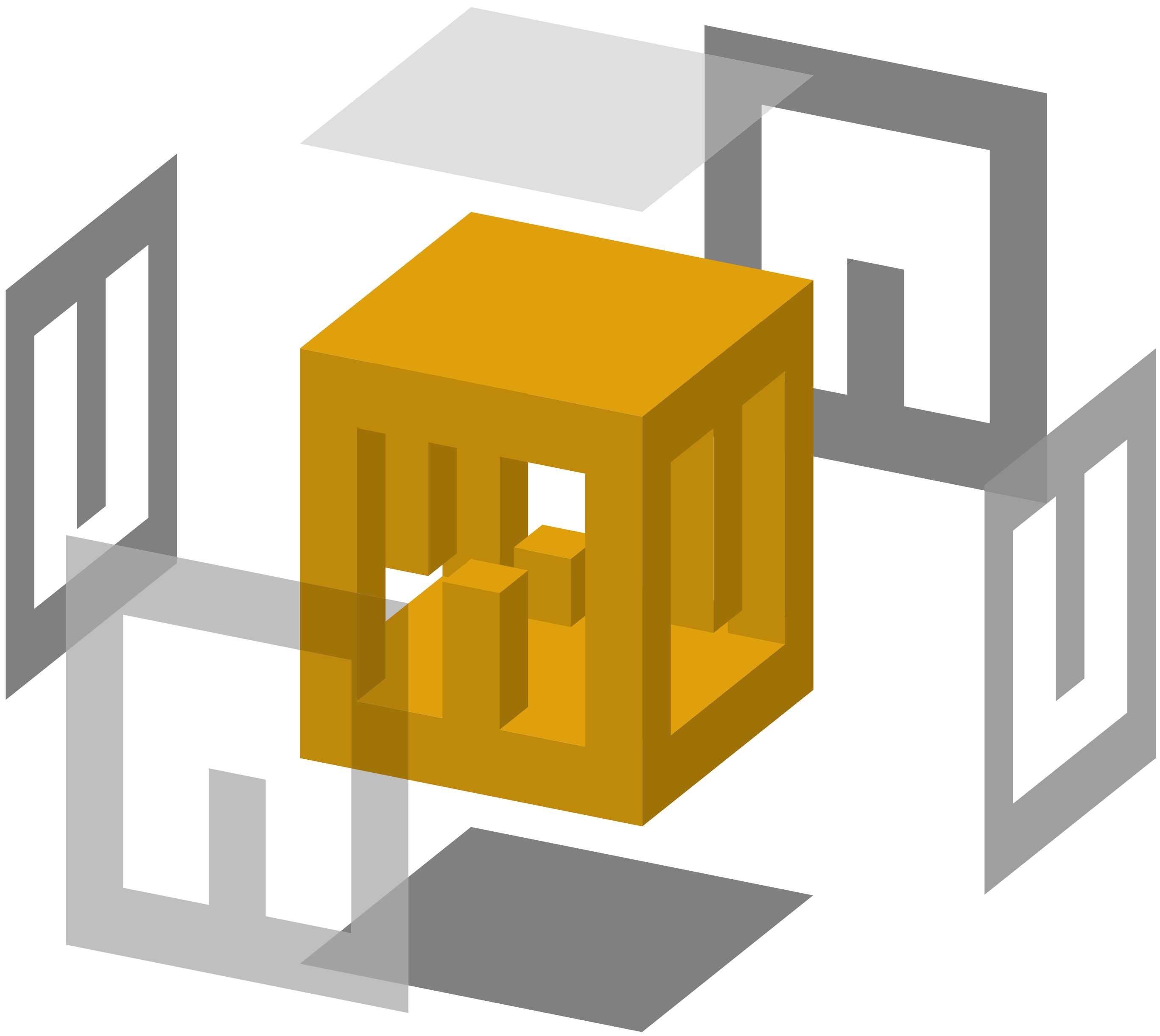
Trip-lets

a second example



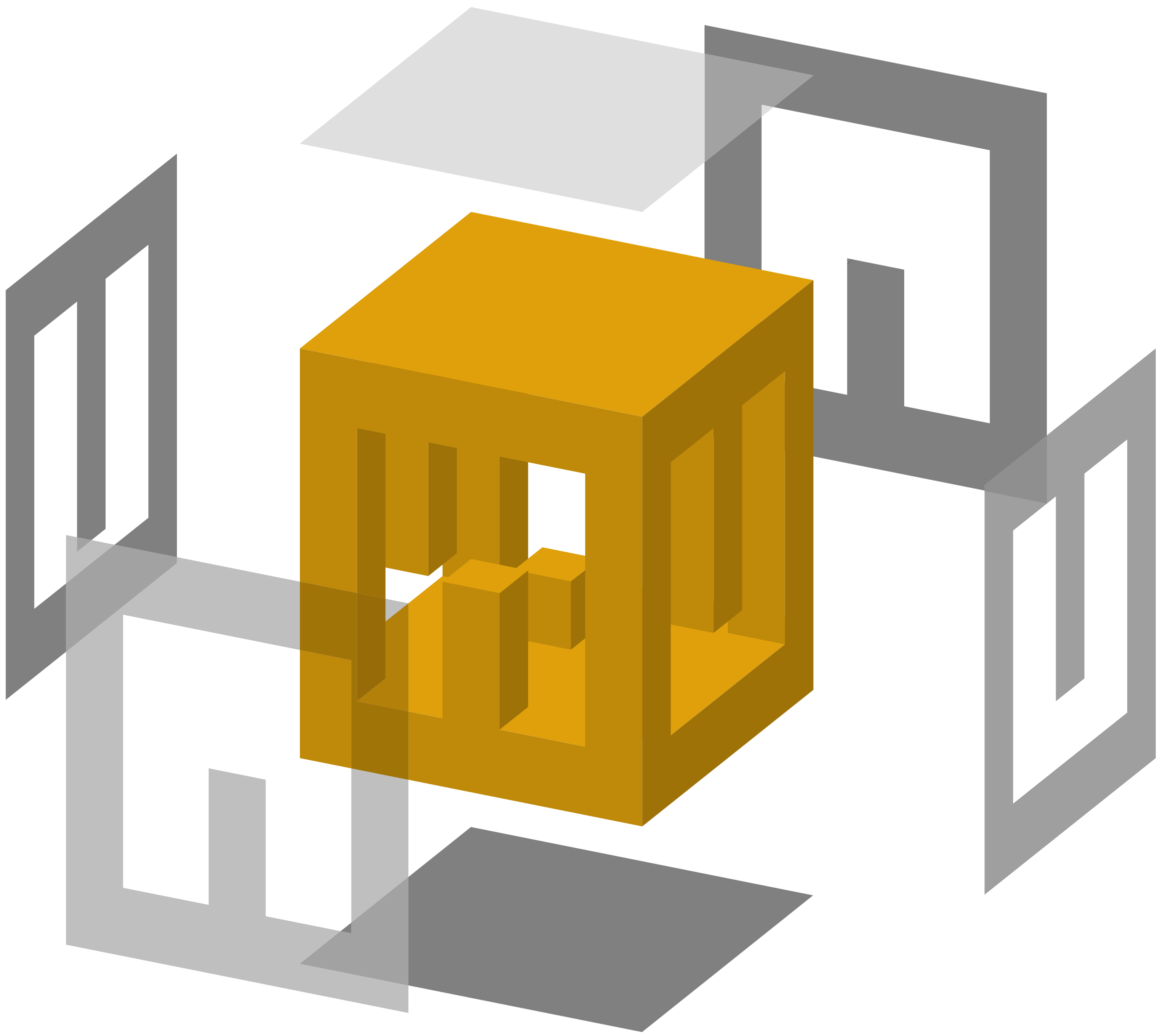
Trip-lets

a second example



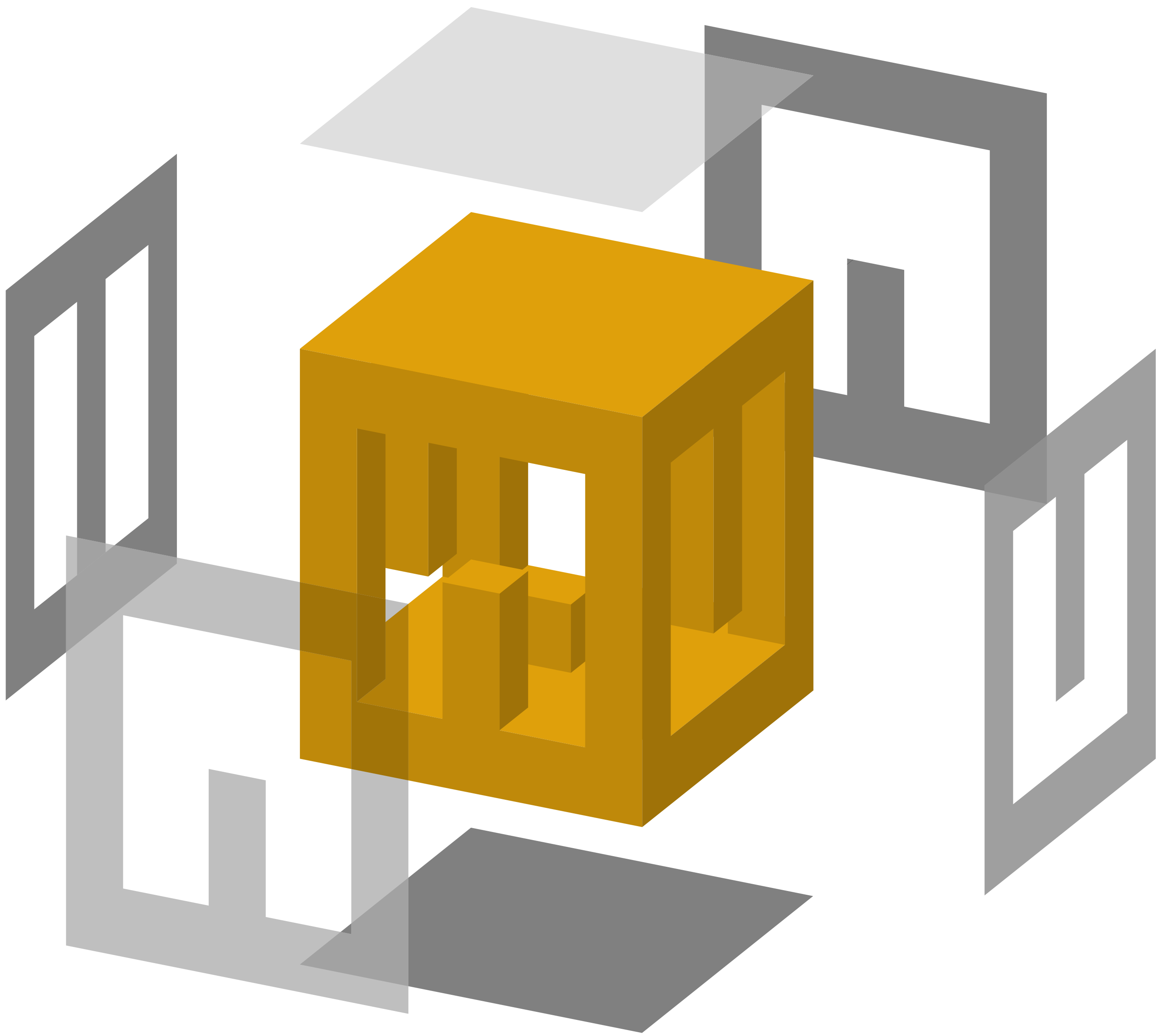
Trip-lets

a second example



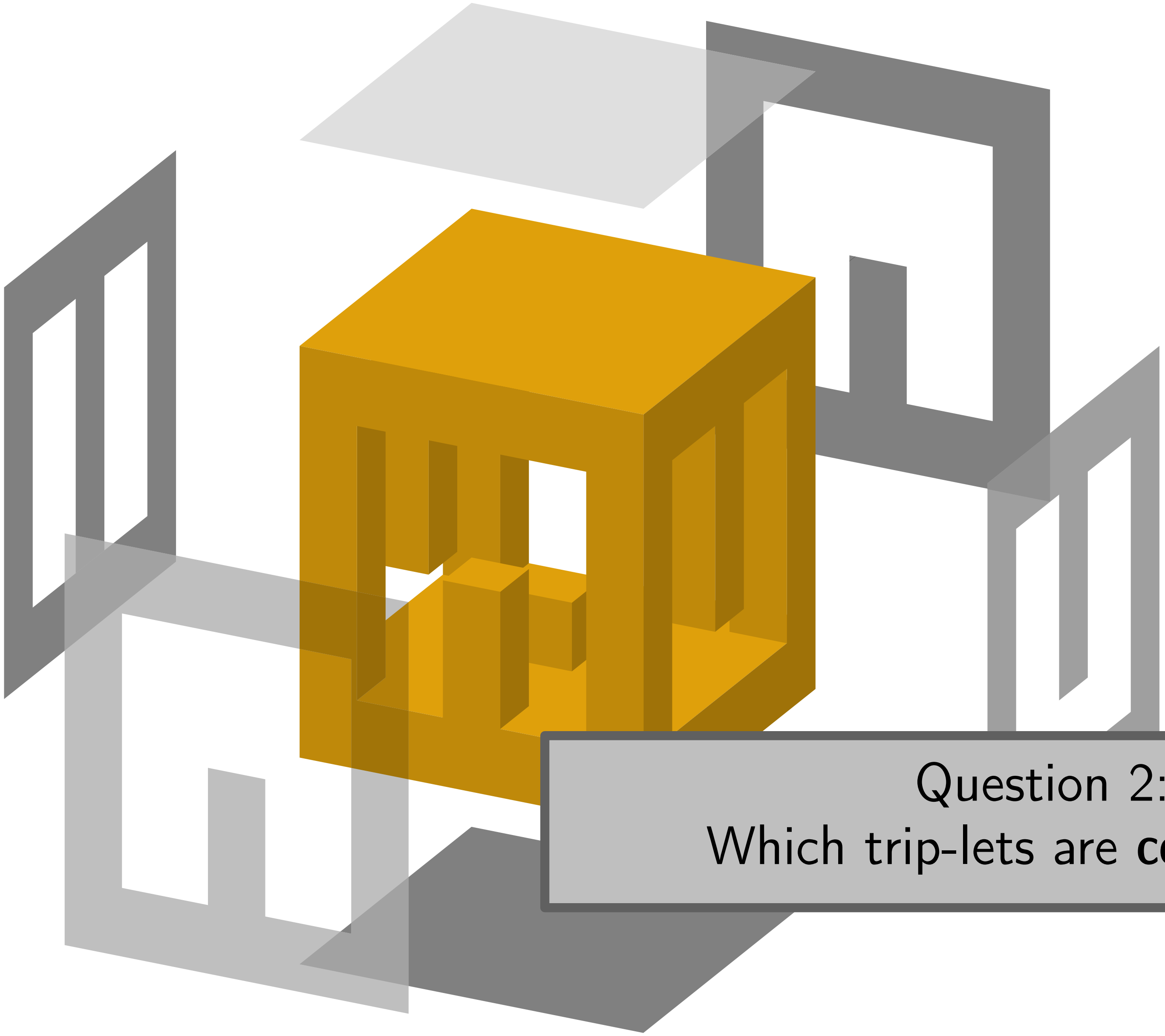
Trip-lets

a second example



Trip-lets

a second example



Question 2:
Which trip-lets are **connected**?

Overview

Given three polygonal shapes with n vertices in total.

- How many vertices can a trip-let have?
- How many vertices can its shadows have?
- How fast can we find out whether a trip-let can be made? (Q1)
- How fast can we find out whether the trip-let is connected? (Q2)

Overview

Given three polygonal shapes with n vertices in total.

- How many vertices can a trip-let have?
- How many vertices can its shadows have?
- How fast can we find out whether a trip-let can be made? (Q1)
- How fast can we find out whether the trip-let is connected? (Q2)

Earlier work

- Q1 and Q2 posed by O'Rourke at SCG'98

Overview

Given three polygonal shapes with n vertices in total.

- How many vertices can a trip-let have?
- How many vertices can its shadows have?
- How fast can we find out whether a trip-let can be made? (Q1)
- How fast can we find out whether the trip-let is connected? (Q2)

Earlier work

- Q1 and Q2 posed by O'Rourke at SCG'98
- Related work: e.g. finding possible viewpoints in the plane for given shadows [Bottino and Laurentini 2003], [Ohgami and Sugihara 2008 (EuroCG)]

Overview

Given three polygonal shapes with n vertices in total.

- How many vertices can a trip-let have?
- How many vertices can its shadows have?
- How fast can we find out whether a trip-let can be made? (Q1)
- How fast can we find out whether the trip-let is connected? (Q2)

Earlier work

- Q1 and Q2 posed by O'Rourke at SCG'98
- Related work: e.g. finding possible viewpoints in the plane for given shadows [Bottino and Laurentini 2003], [Ohgami and Sugihara 2008 (EuroCG)]
- Intersecting 3D Nef polyhedra [Hachenberger et al. 2007]: trip-let construction in $O(n^4 \log n)$ time, expected

Trip-let complexity

Given three polygonal shapes with n vertices in total, how many vertices can a corresponding trip-let have?

Trip-let complexity

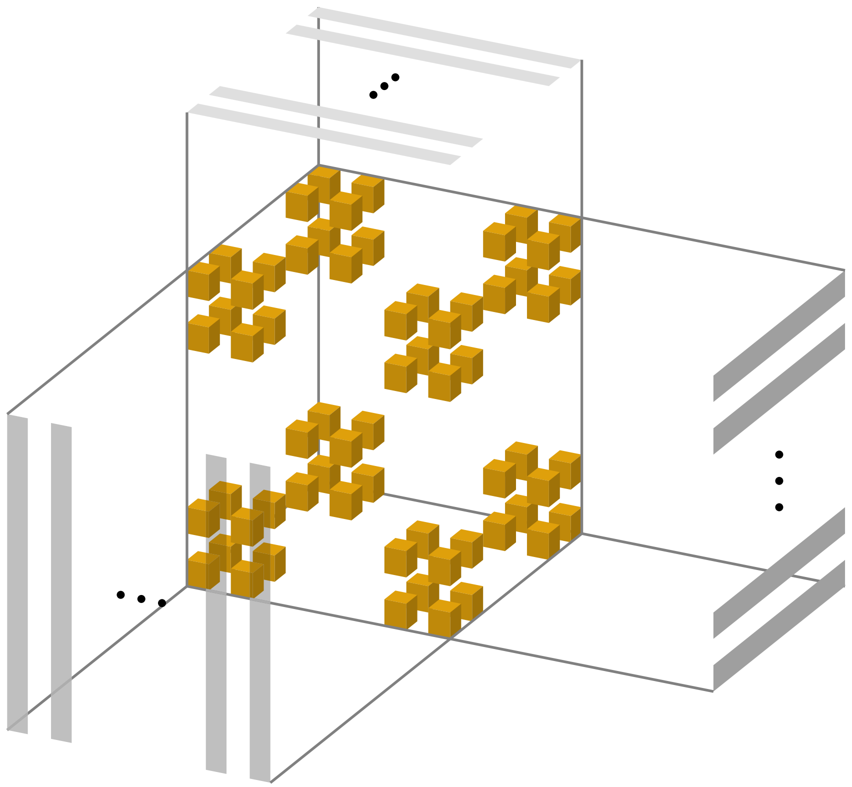
Given three polygonal shapes with n vertices in total, how many vertices can a corresponding trip-let have?

- $\Rightarrow$ lower bound on running time for constructive algorithms

Trip-let complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a corresponding trip-let have?

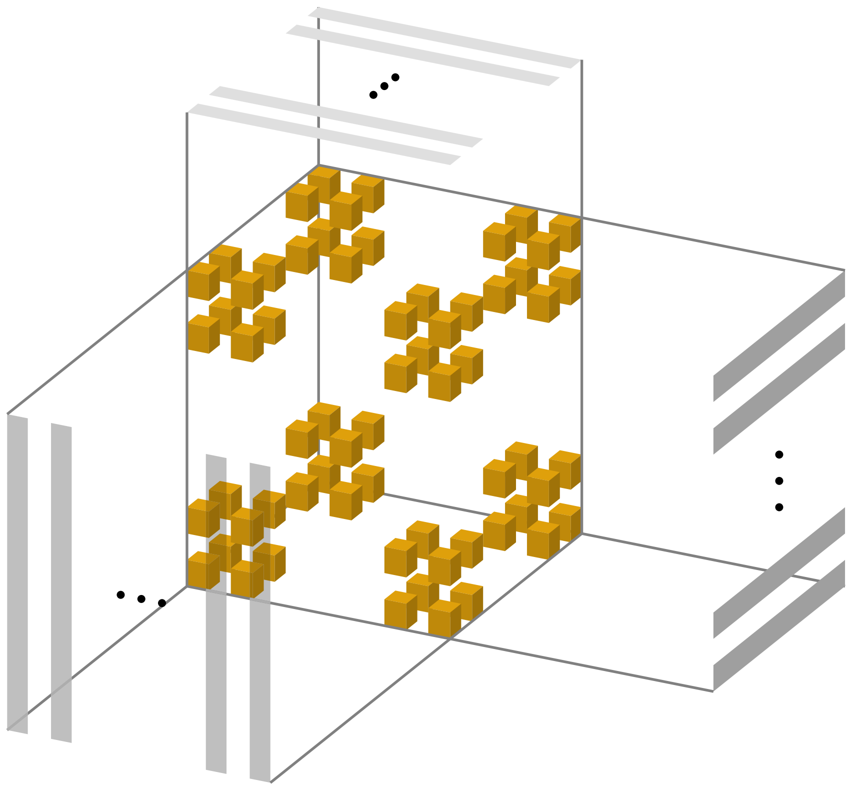
- $\Rightarrow$ lower bound on running time for constructive algorithms
- $\Theta(n^3)$ in the worst case



Trip-let complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a corresponding trip-let have?

- $\Rightarrow$ lower bound on running time for constructive algorithms
- $\Theta(n^3)$ in the worst case



What about connected trip-lets?

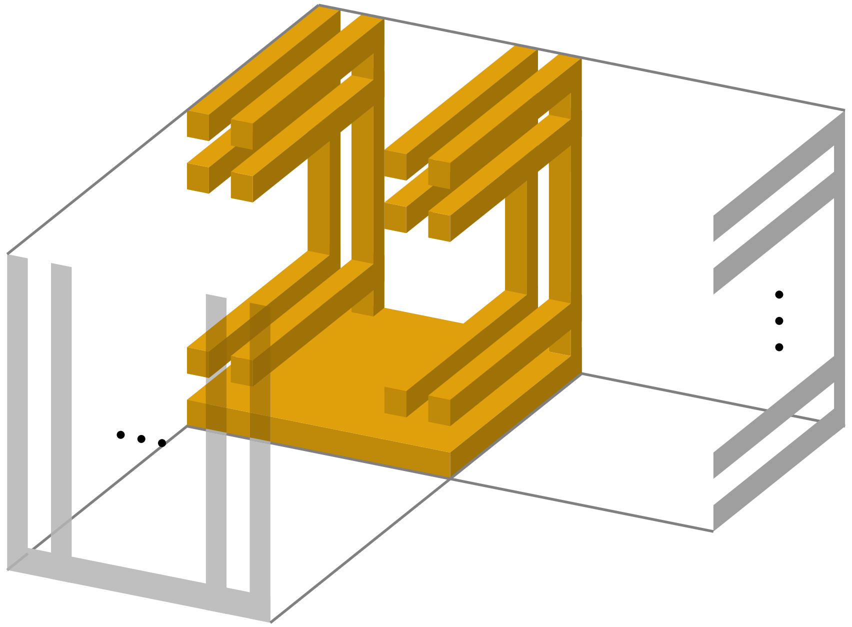
Trip-let complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a corresponding trip-let have?

- $\Rightarrow$ lower bound on running time for constructive algorithms
- $\Theta(n^3)$ in the worst case

What about connected trip-lets?

- $\Omega(n^2)$ is possible;



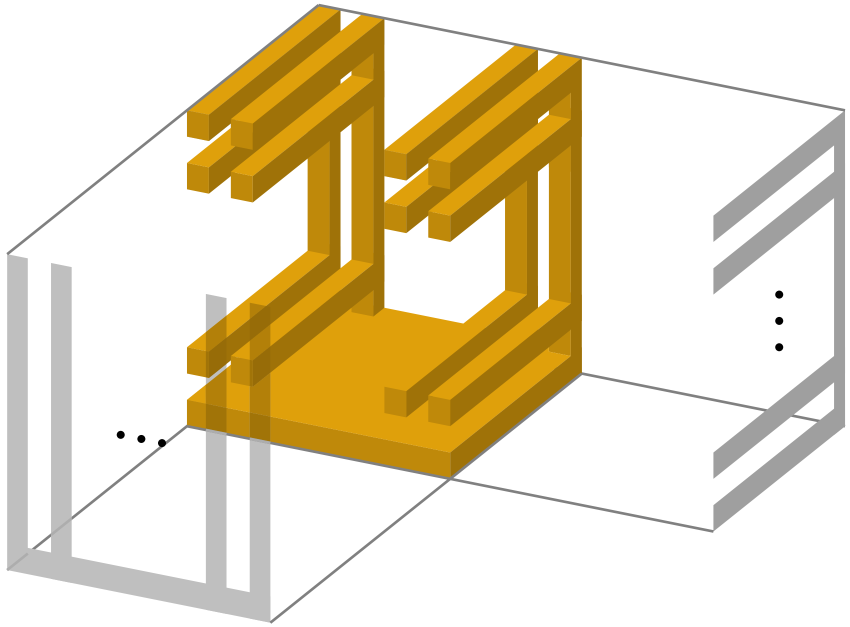
Trip-let complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a corresponding trip-let have?

- $\Rightarrow$ lower bound on running time for constructive algorithms
- $\Theta(n^3)$ in the worst case

What about connected trip-lets?

- $\Omega(n^2)$ is possible;
- but can it get any worse?
  (we don't think so)



Shadow complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a trip-let's shadows have?

Shadow complexity

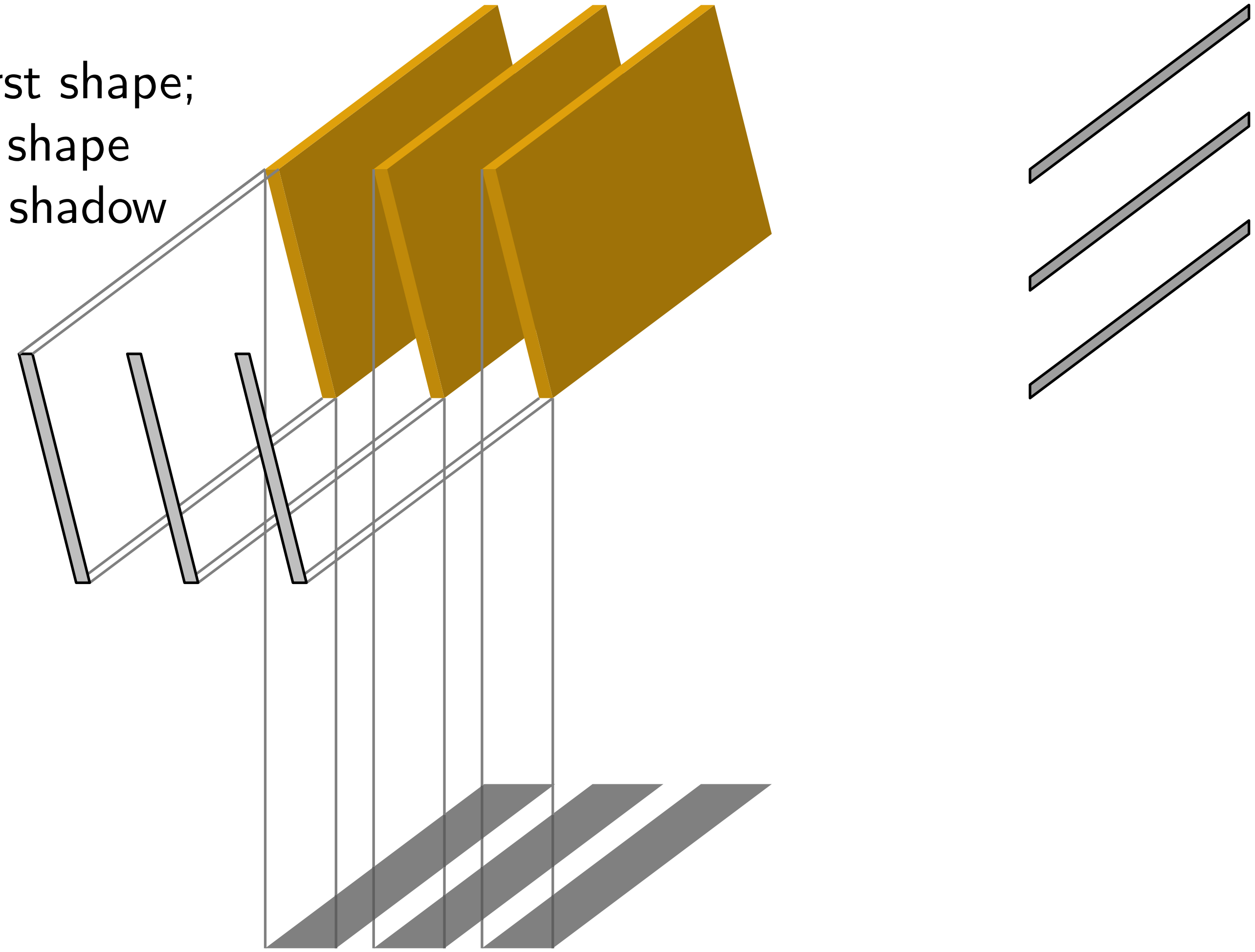
Given three polygonal shapes with n vertices in total,
how many vertices can a trip-let's shadows have?

- $\Rightarrow$ lower bound on running time for constructive algorithms verifying shadows

Shadow complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a trip-let's shadows have?

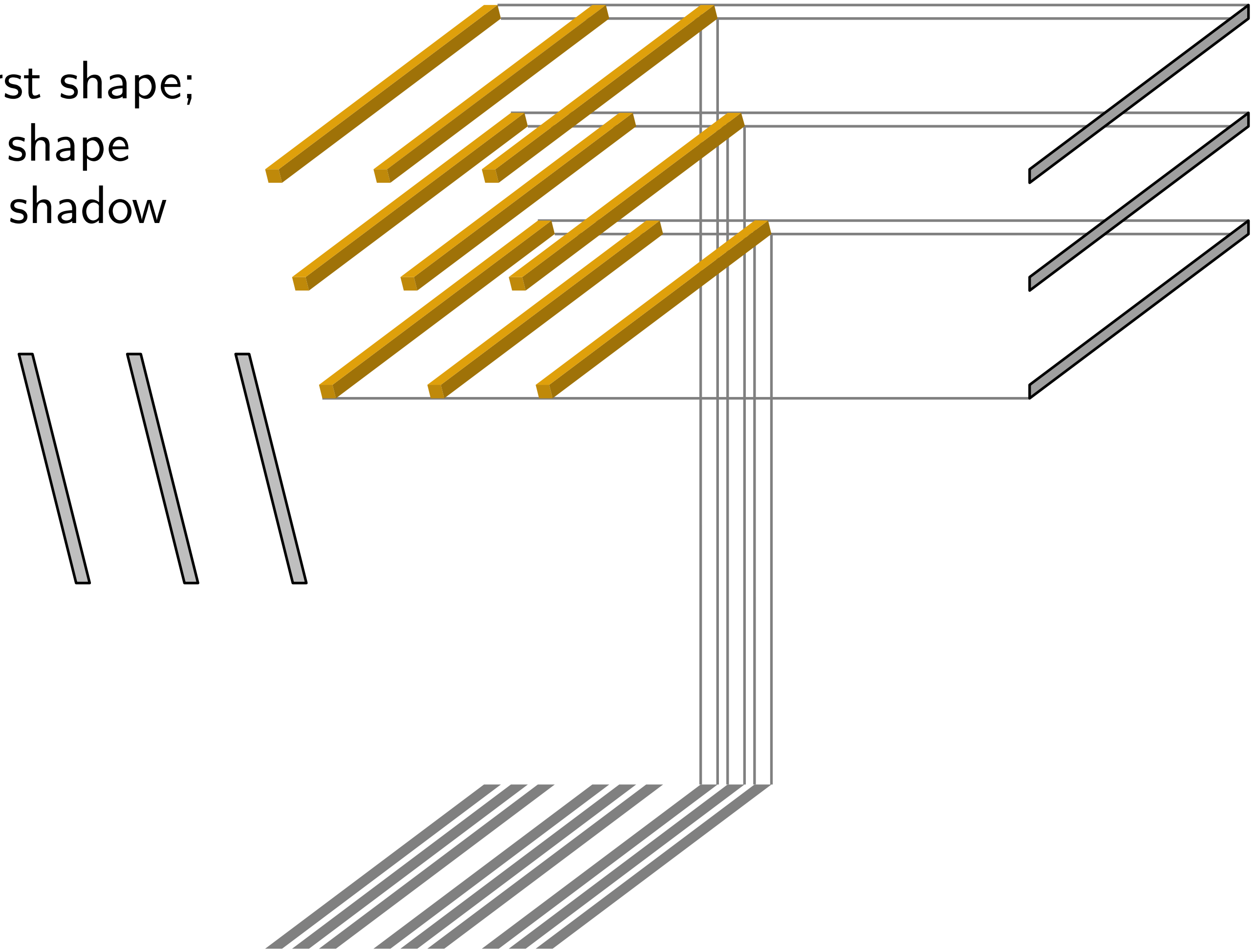
- $\Rightarrow$ lower bound on running time for constructive algorithms verifying shadows
- slanted vertical strips in first shape;
  horizontal strips in second shape
  $\Rightarrow \Theta(n^2)$ parallel strips in shadow



Shadow complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a trip-let's shadows have?

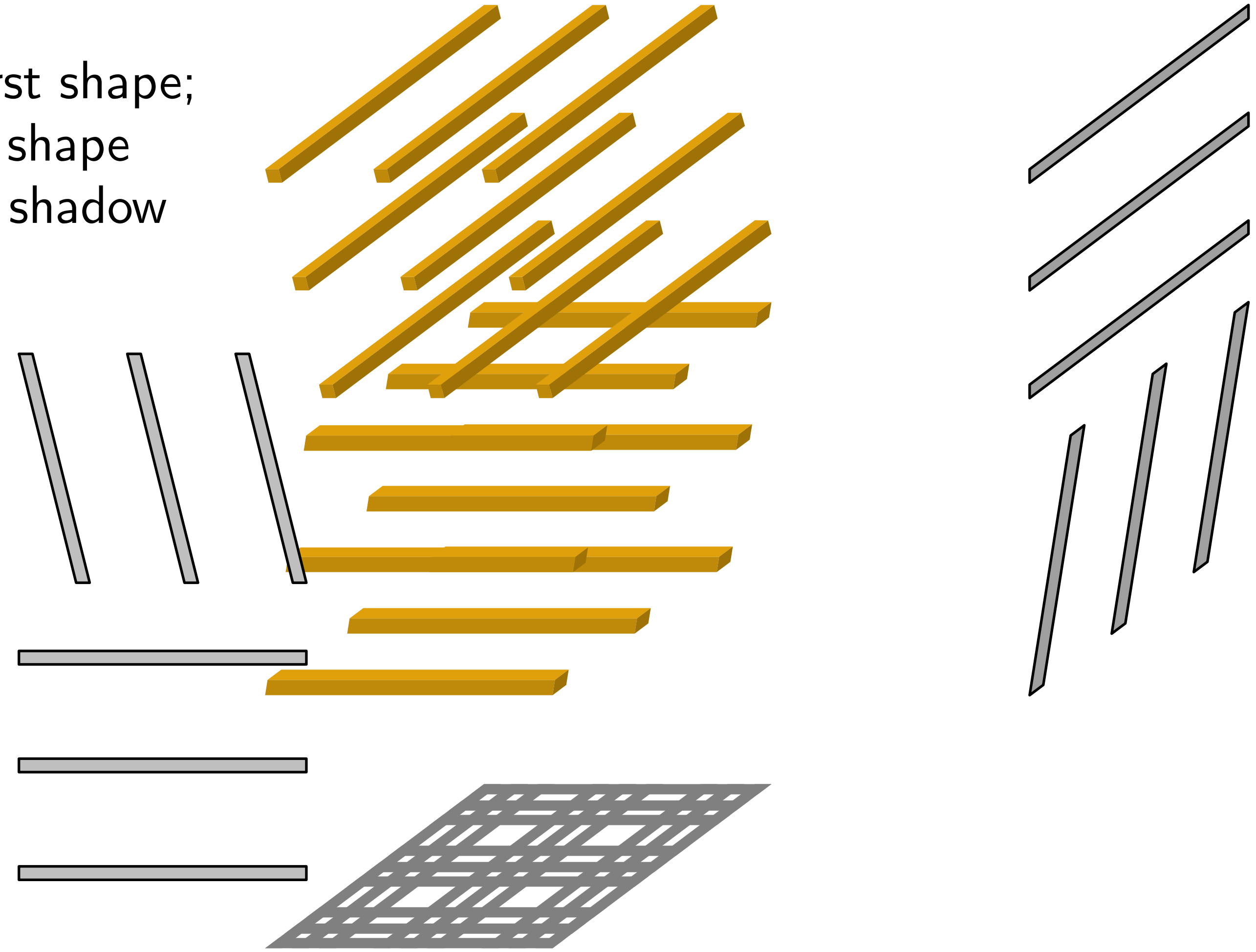
- $\Rightarrow$ lower bound on running time for constructive algorithms verifying shadows
- slanted vertical strips in first shape;
  horizontal strips in second shape
  $\Rightarrow \Theta(n^2)$ parallel strips in shadow



Shadow complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a trip-let's shadows have?

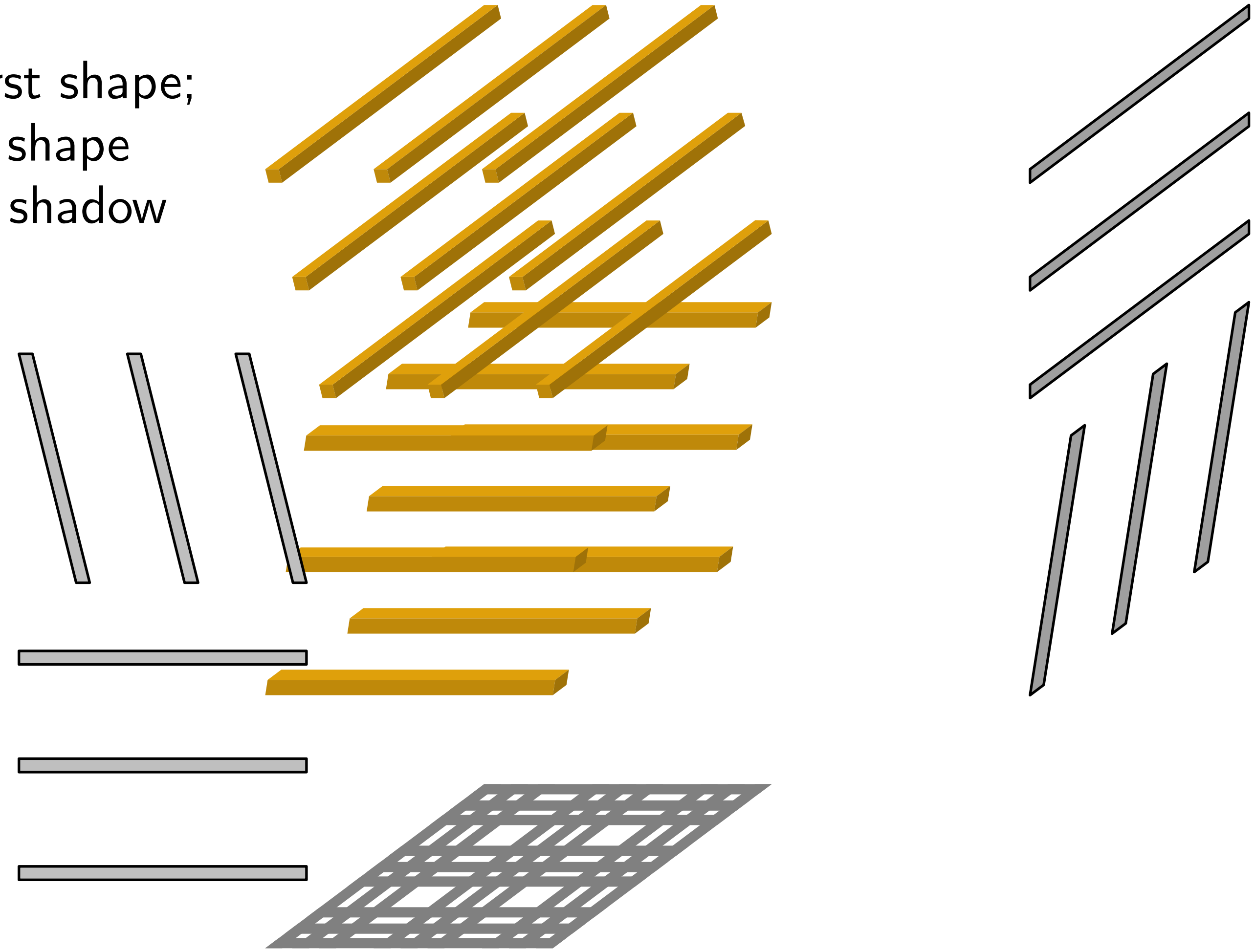
- $\Rightarrow$ lower bound on running time for constructive algorithms verifying shadows
- slanted vertical strips in first shape;
  horizontal strips in second shape
  $\Rightarrow \Theta(n^2)$ parallel strips in shadow
- copy & rotate
  $\Rightarrow$ grid with $\Theta(n^4)$ holes



Shadow complexity

Given three polygonal shapes with n vertices in total,
how many vertices can a trip-let's shadows have?

- $\Rightarrow$ lower bound on running time for constructive algorithms verifying shadows
- slanted vertical strips in first shape;
  horizontal strips in second shape
  $\Rightarrow \Theta(n^2)$ parallel strips in shadow
- copy & rotate
  $\Rightarrow$ grid with $\Theta(n^4)$ holes



- For rectilinear shapes, complexity is only $\Theta(n^2)$ in the worst case.

Determining validity (Question 1)

Given three polygonal shapes with n vertices in total, does a trip-let exist for these shapes?

Determining validity (Question 1)

Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

Determining validity (Question 1)

Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

Determining validity (Question 1)

Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.

Determining validity (Question 1)

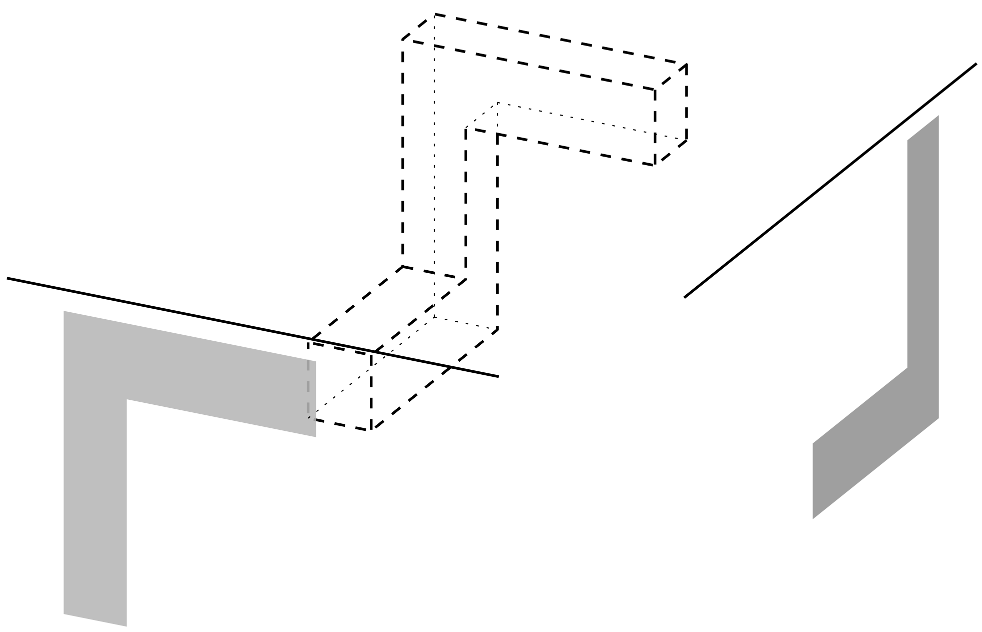
Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.



Determining validity (Question 1)

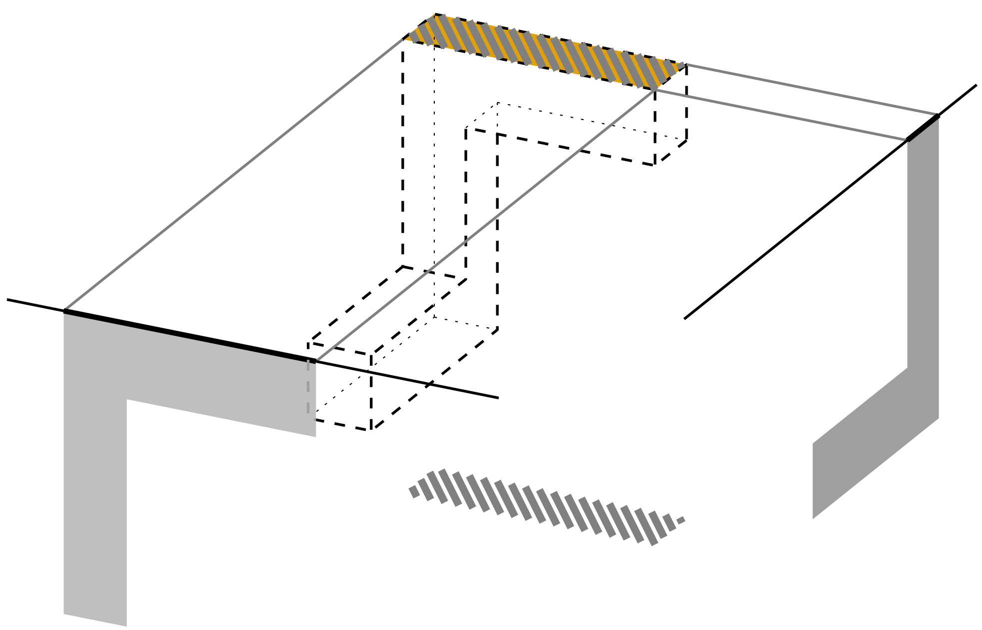
Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.



Determining validity (Question 1)

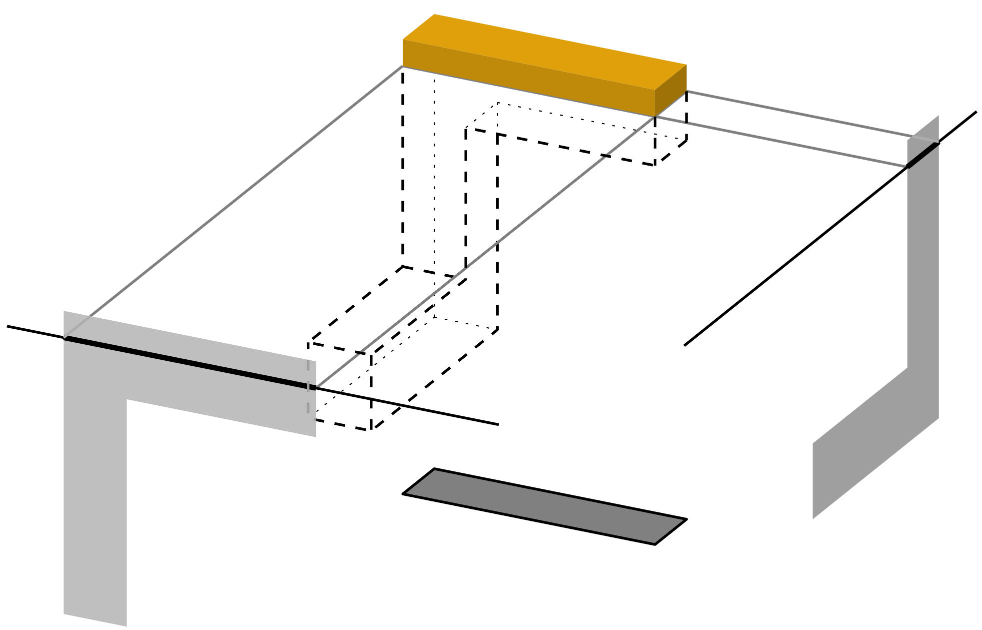
Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.



Determining validity (Question 1)

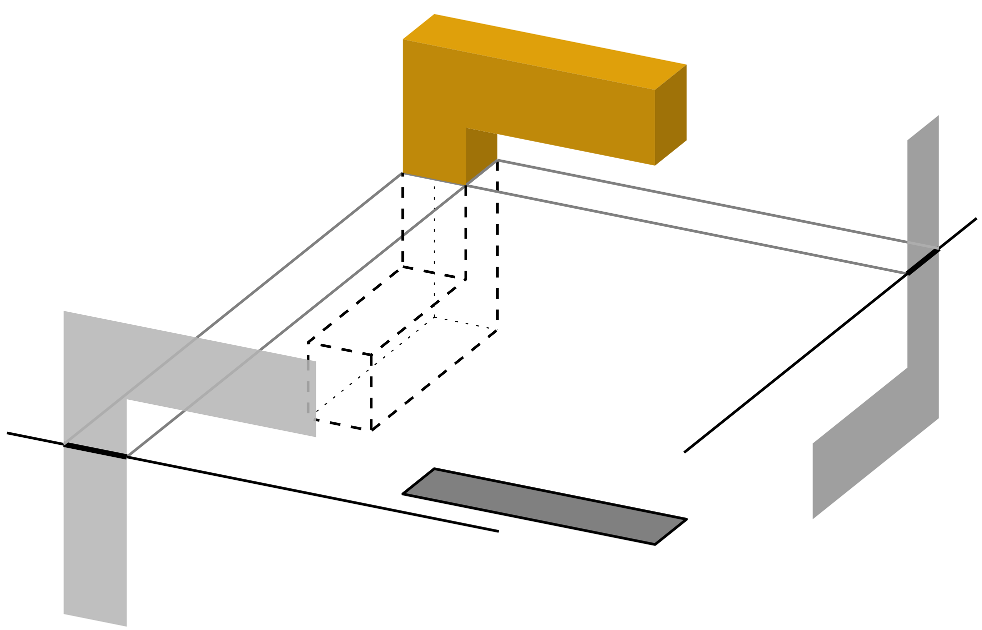
Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.



Determining validity (Question 1)

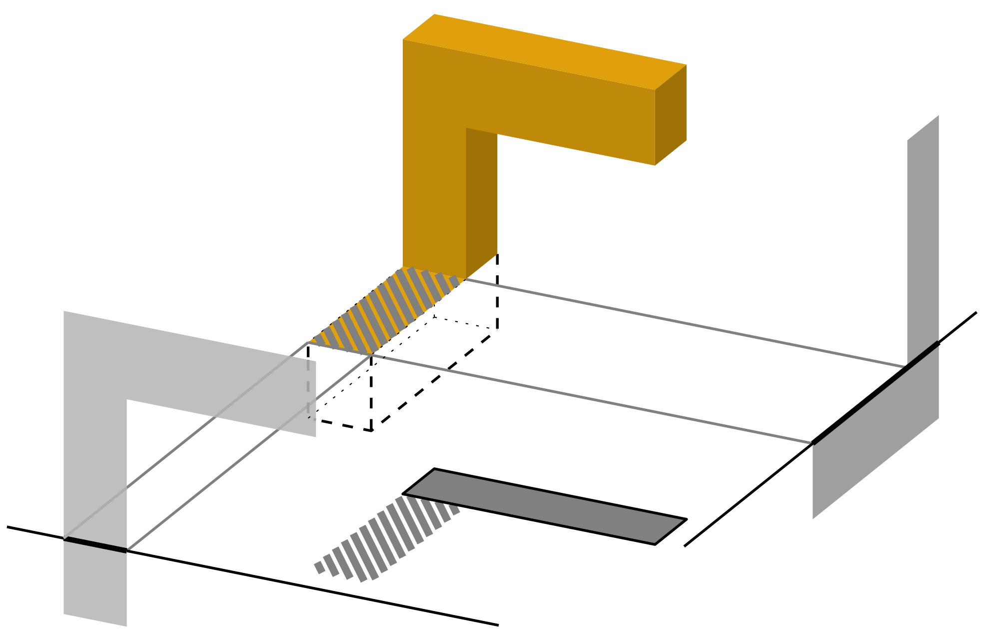
Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.



Determining validity (Question 1)

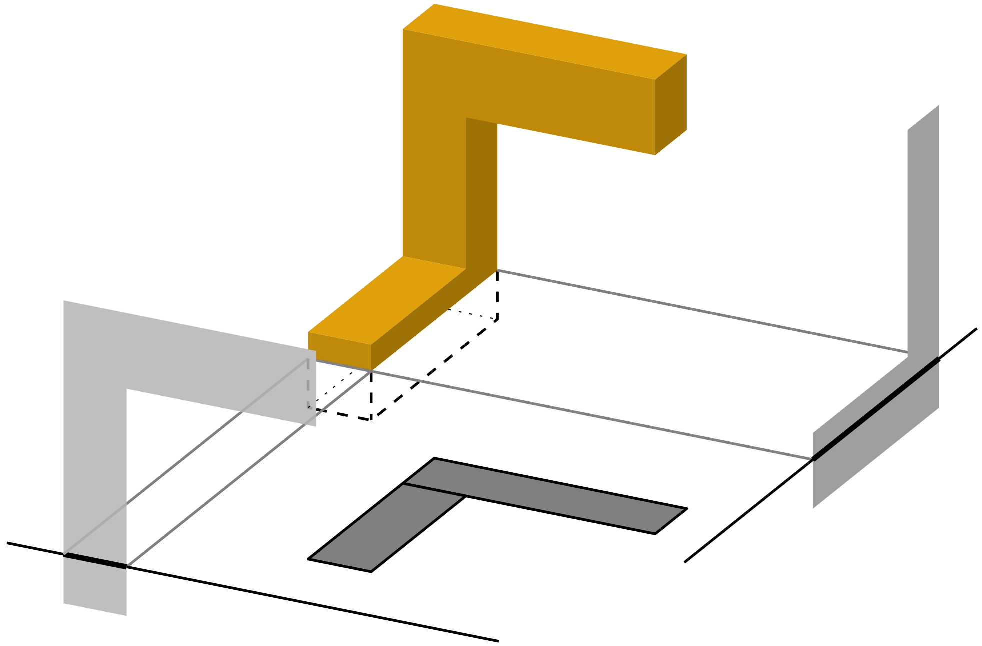
Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.



Determining validity (Question 1)

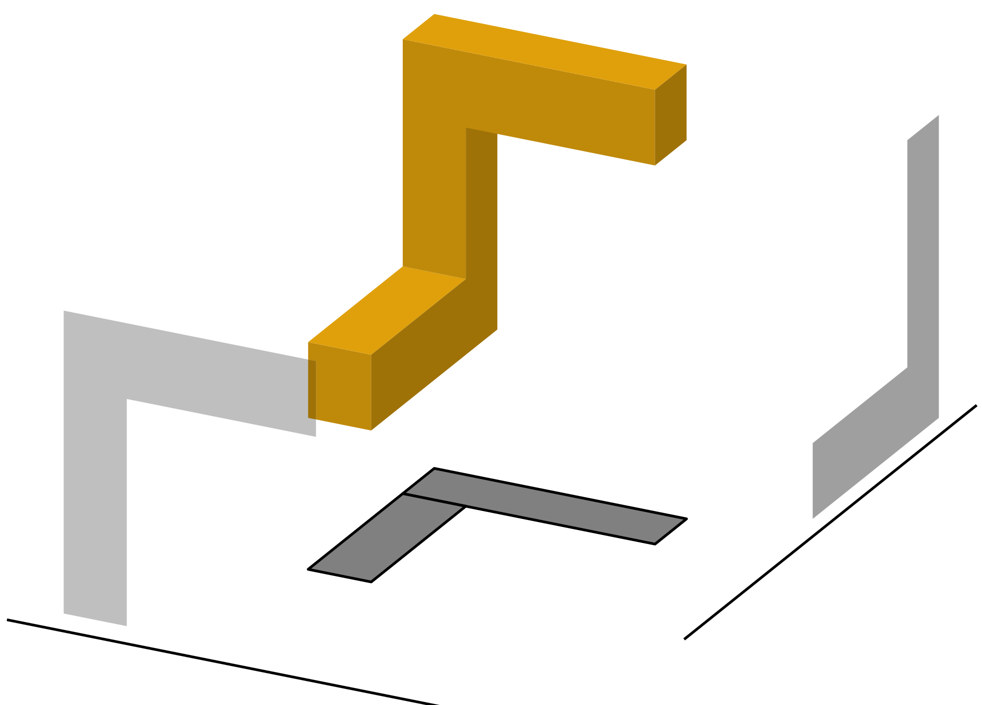
Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.



Determining validity (Question 1)

Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

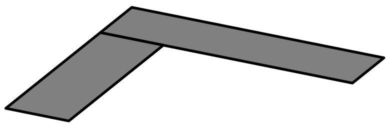
  We only consider rectilinear polygons.

- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.

- Take union of rectangle set (size: $O(n^2)$), and compare shadow to shape.

  How to do this efficiently?



Determining validity (Question 1)

Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

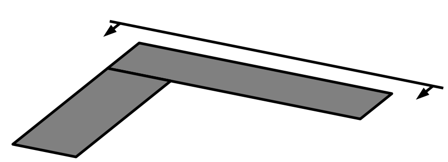
- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.

- Take union of rectangle set (size: $O(n^2)$), and compare shadow to shape.

  How to do this efficiently?

  - Sweep-line algorithm.
  - Extended segment-tree as status structure.



Determining validity (Question 1)

Given three polygonal shapes with n vertices in total,
does a trip-let exist for these shapes?

- Constructive algorithm: general polygons $\Rightarrow$ running time $\Omega(n^4)$.

  We only consider rectilinear polygons.

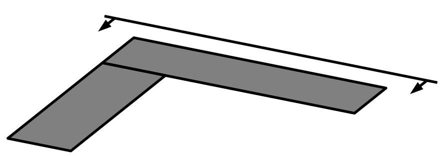
- Check each of the three shadows separately.

- Use a sweep-plane algorithm to get a set of rectangles covering a shape.

- Take union of rectangle set (size: $O(n^2)$), and compare shadow to shape.

  How to do this efficiently?

  - Sweep-line algorithm.
  - Extended segment-tree as status structure.



- Algorithm runs in $O(n^2 \log n)$ time.

Determining connectedness (Question 2)

Given three polygonal shapes with n vertices in total,
can a corresponding trip-let be connected?

Determining connectedness (Question 2)

Given three polygonal shapes with n vertices in total,
can a corresponding trip-let be connected?

- Goal: running time $\approx O(k)$, with k the number of vertices of the trip-let.

Determining connectedness (Question 2)

Given three polygonal shapes with n vertices in total,
can a corresponding trip-let be connected?

- Goal: running time $\approx O(k)$, with k the number of vertices of the trip-let.

- Sweep-plane algorithm keeping track of connected components in object.

Determining connectedness (Question 2)

Given three polygonal shapes with n vertices in total,
can a corresponding trip-let be connected?

- Goal: running time $\approx O(k)$, with k the number of vertices of the trip-let.
- Sweep-plane algorithm keeping track of connected components in object.
- Running time: $O((n^2 + k) \log n)$

Open problems

- Our algorithms only work for rectilinear shapes;
  3D Nef algorithm of Hachenberger et al. works for general polygons;
  what about "smooth" closed curves as used in characters?

Open problems

- Our algorithms only work for rectilinear shapes;
  3D Nef algorithm of Hachenberger et al. works for general polygons;
  what about "smooth" closed curves as used in characters?

- We know connected trip-lets have worst-case complexity $\Omega(n^2)$ and $O(n^3)$.
  Does there exist any connected trip-let with $\omega(n^2)$ vertices?

Open problems

- Our algorithms only work for rectilinear shapes;

  3D Nef algorithm of Hachenberger et al. works for general polygons;

  what about “smooth” closed curves as used in characters?

- We know connected trip-lets have worst-case complexity $\Omega(n^2)$ and $O(n^3)$.

  Does there exist any connected trip-let with $\omega(n^2)$ vertices?

- Our algorithms are almost worst-case optimal for actually constructing the objects and their shadows;

  What about non-constructive algorithms?

  That is, can an algorithm be faster than $O(n^2)$ in the worst-case?