

Diskrete Optimierung für GIS und Kunst

Michael Birsak



Introduction

Diskrete Optimierung für GIS und Kunst

Introduction

Diskrete Optimierung für GIS und Kunst

Introduction

Diskrete Optimierung für GIS und Kunst

$$\text{minimize } f_0(\boldsymbol{x})$$

Introduction

Diskrete Optimierung für GIS und Kunst

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq b_i \quad i = 1, \dots, m \end{array}$$

Introduction

Diskrete Optimierung für GIS und Kunst

$$\begin{array}{ll} \text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq b_i \quad i = 1, \dots, m \\ & \mathbf{x} \in \mathbb{R}^n \end{array}$$

Introduction

Diskrete Optimierung für GIS und Kunst

$$\begin{array}{ll}\text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq b_i \quad i = 1, \dots, m \\ & \mathbf{x} \in \mathbb{R}^n \\ & \mathbf{x} \in \mathbb{Z}^n\end{array}$$

Introduction

in general:

$$\begin{array}{ll}\text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq b_i \quad i = 1, \dots, m \\ & \mathbf{x} \in \mathbb{R}^n \\ & \mathbf{x} \in \mathbb{Z}^n\end{array}$$

Introduction

in general:

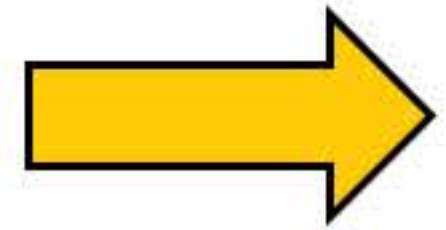
$$\begin{array}{ll}\text{minimize} & f_0(\mathbf{x}) \\ \text{subject to} & f_i(\mathbf{x}) \leq b_i \quad i = 1, \dots, m \\ & \mathbf{x} \in \mathbb{R}^n \\ & \mathbf{x} \in \mathbb{Z}^n\end{array}$$

in our work:

$$\begin{array}{l}f_0(\mathbf{x}) = \mathbf{c}^T \mathbf{x} \\ (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T = A\mathbf{x}\end{array}$$

Introduction

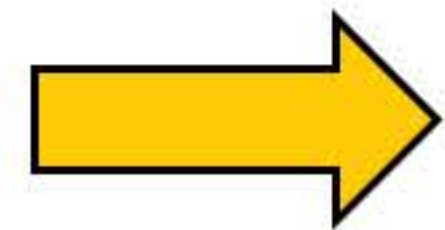
in our work: minimize $\mathbf{c}^T \mathbf{x}$
 subject to $A\mathbf{x} \leq \mathbf{b}$
 $\mathbf{x} \in \mathbb{Z}^n$



Integer Programming (IP or ILP)

Introduction

in our work: minimize $\mathbf{c}^T \mathbf{x}$
 subject to $A\mathbf{x} \leq \mathbf{b}$
 $\mathbf{x} \in \mathbb{Z}^n$

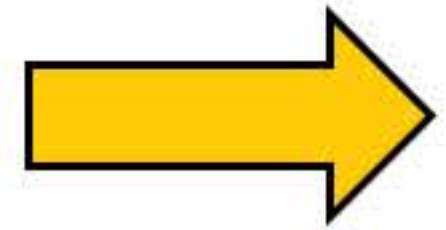


Integer Programming (IP or ILP)

if some $x_i \in \mathbb{R}$: Mixed Integer Programming (MIP or MILP)

Introduction

in our work: minimize $\mathbf{c}^T \mathbf{x}$
 subject to $A\mathbf{x} \leq \mathbf{b}$
 $\mathbf{x} \in \mathbb{Z}^n$



Integer Programming (IP or ILP)

if some $x_i \in \mathbb{R}$: Mixed Integer Programming (MIP or MILP)

if all $x_i \in \{0,1\}$: Zero-One or Binary Integer Programming (BIP)

Integer Programming

$$\begin{array}{ll}\text{minimize} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \mathbf{Ax} \leq \mathbf{b}\end{array}$$

$$\mathbf{x} \in \mathbb{Z}^n$$

- Widely used in practice
 - Production Planning
 - Scheduling

Integer Programming

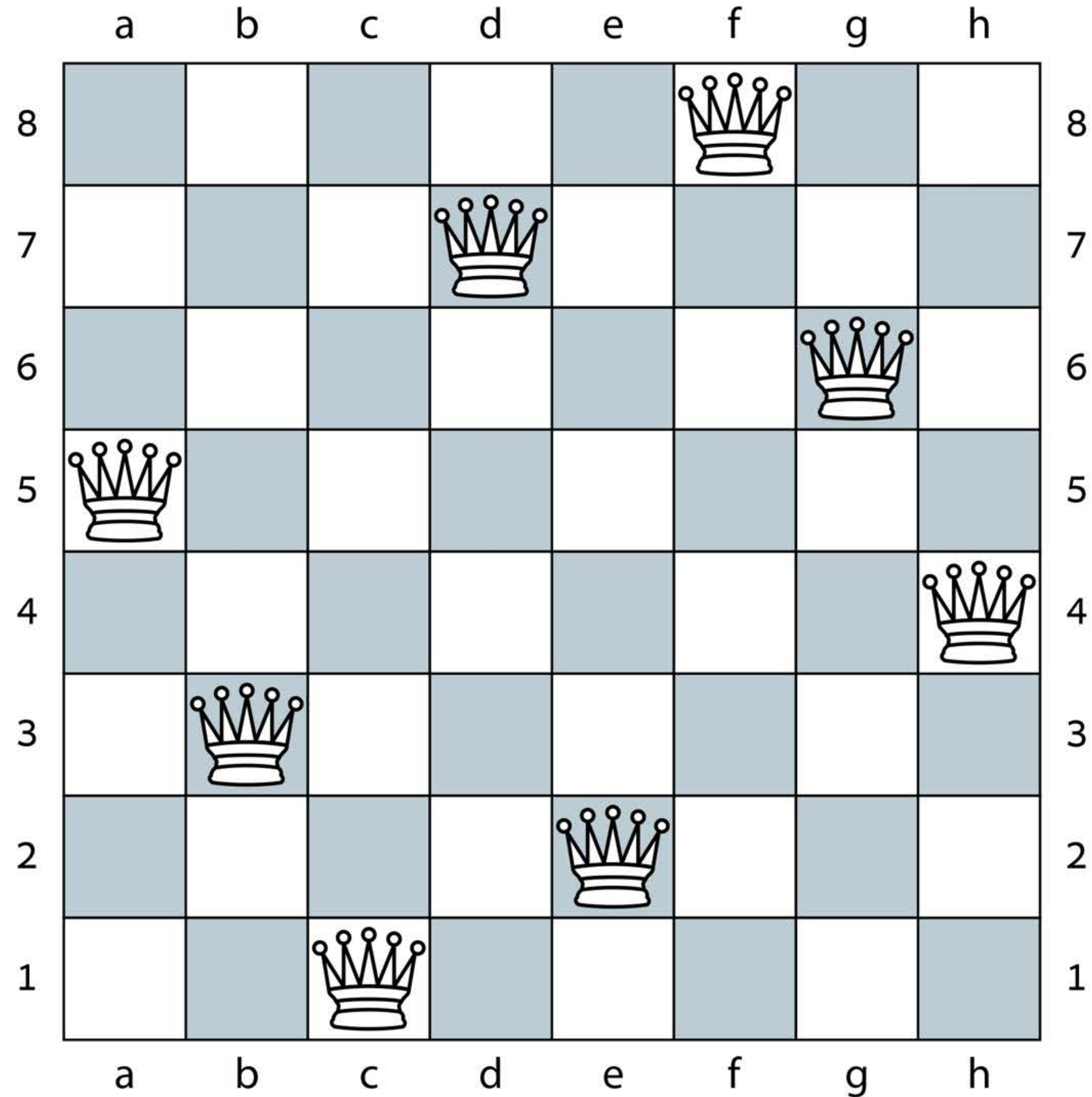
$$\begin{array}{ll}\text{minimize} & \mathbf{c}^T \mathbf{x} \\ \text{subject to} & \mathbf{Ax} \leq \mathbf{b}\end{array}$$

$$\mathbf{x} \in \mathbb{Z}^n$$

- Widely used in practice
 - Production Planning
 - Scheduling
- NP-complete

Integer Programming

Eight Queens Problem



Integer Programming

- Solution Strategies

Integer Programming

- Solution Strategies

Cutting Plane Algorithm

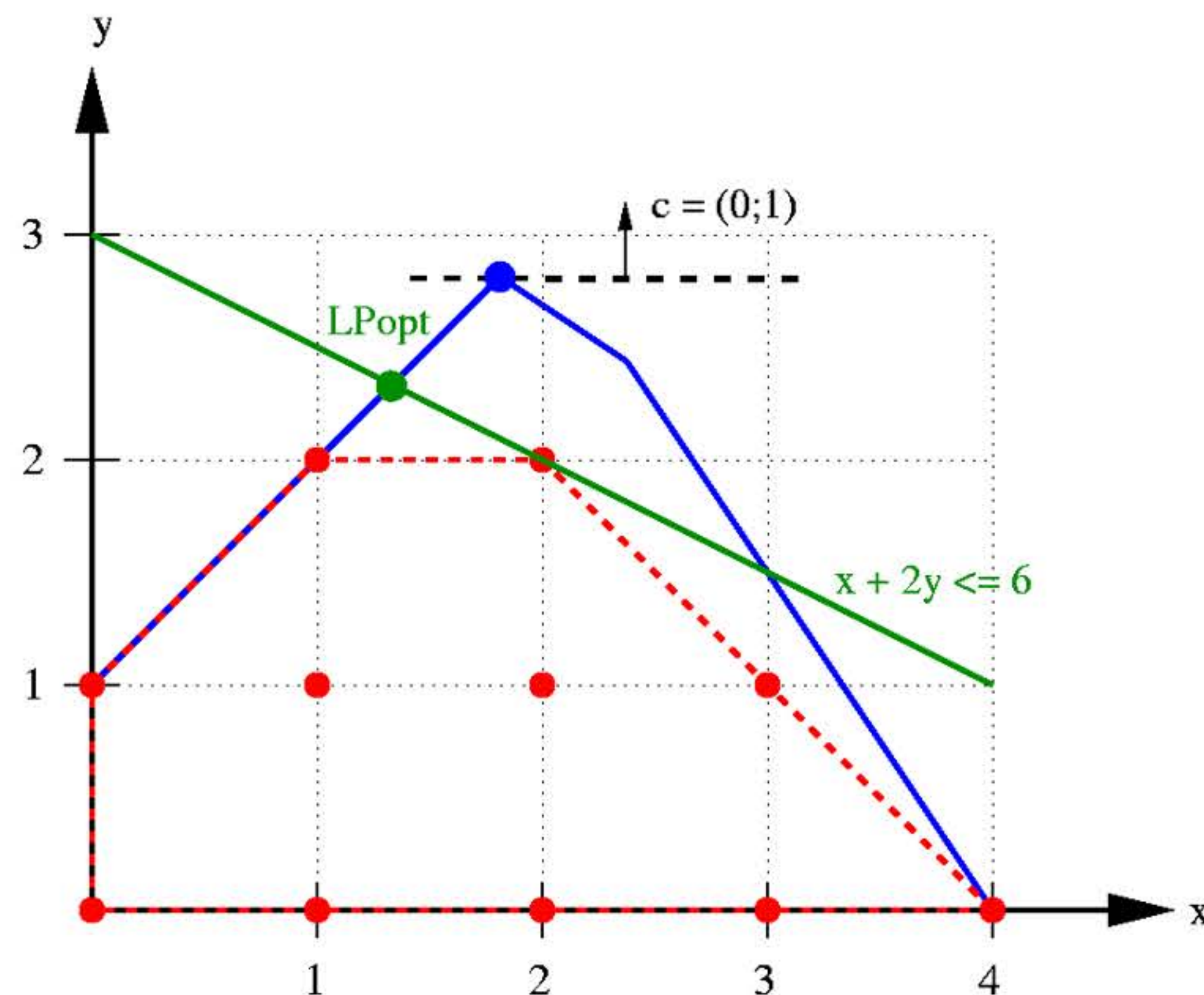
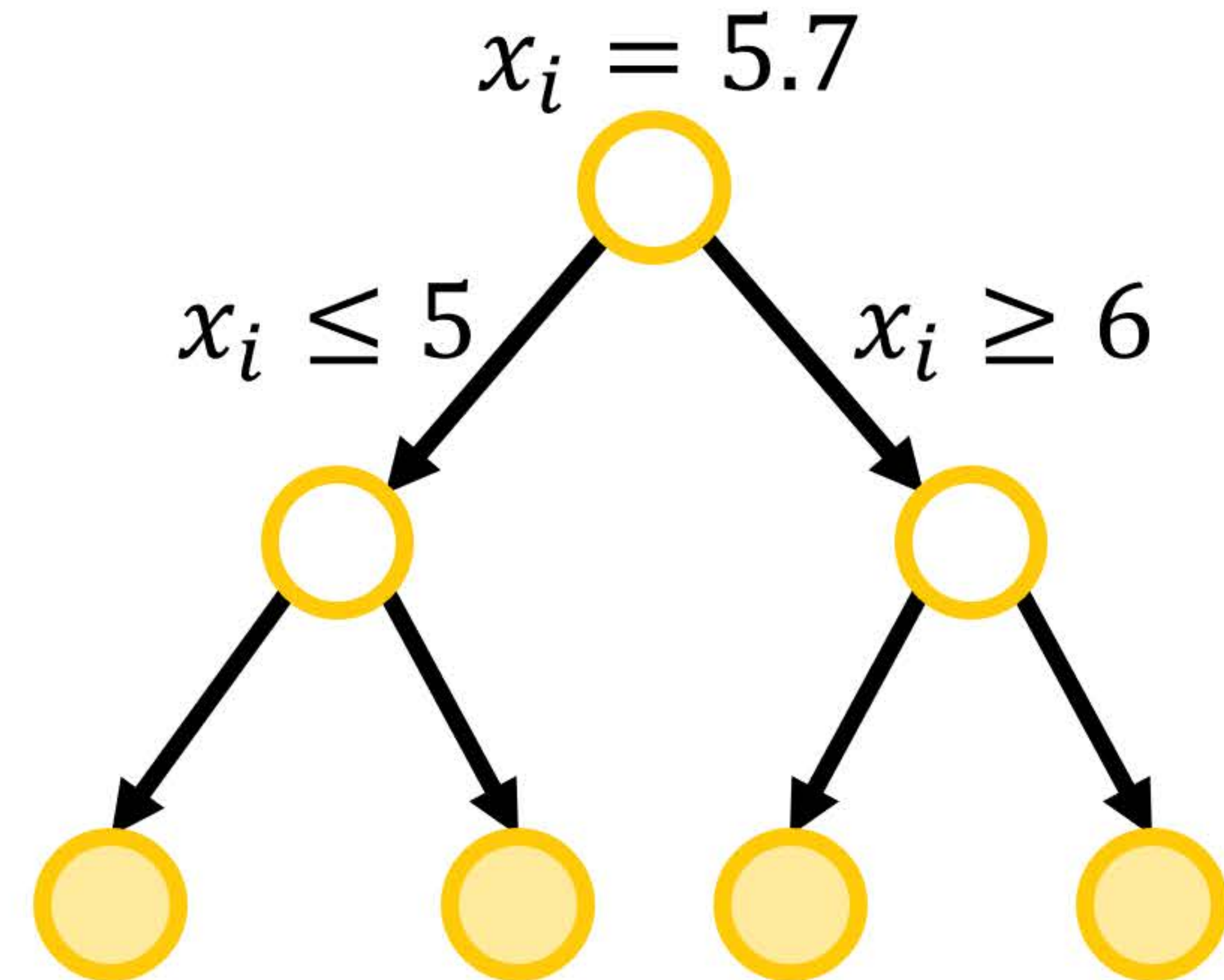


Image taken from [https://de.wikipedia.org/wiki/Ganzzahlige_lineare_Optimierung]

Branch-and-Bound



Integer Programming in Our Work



Path

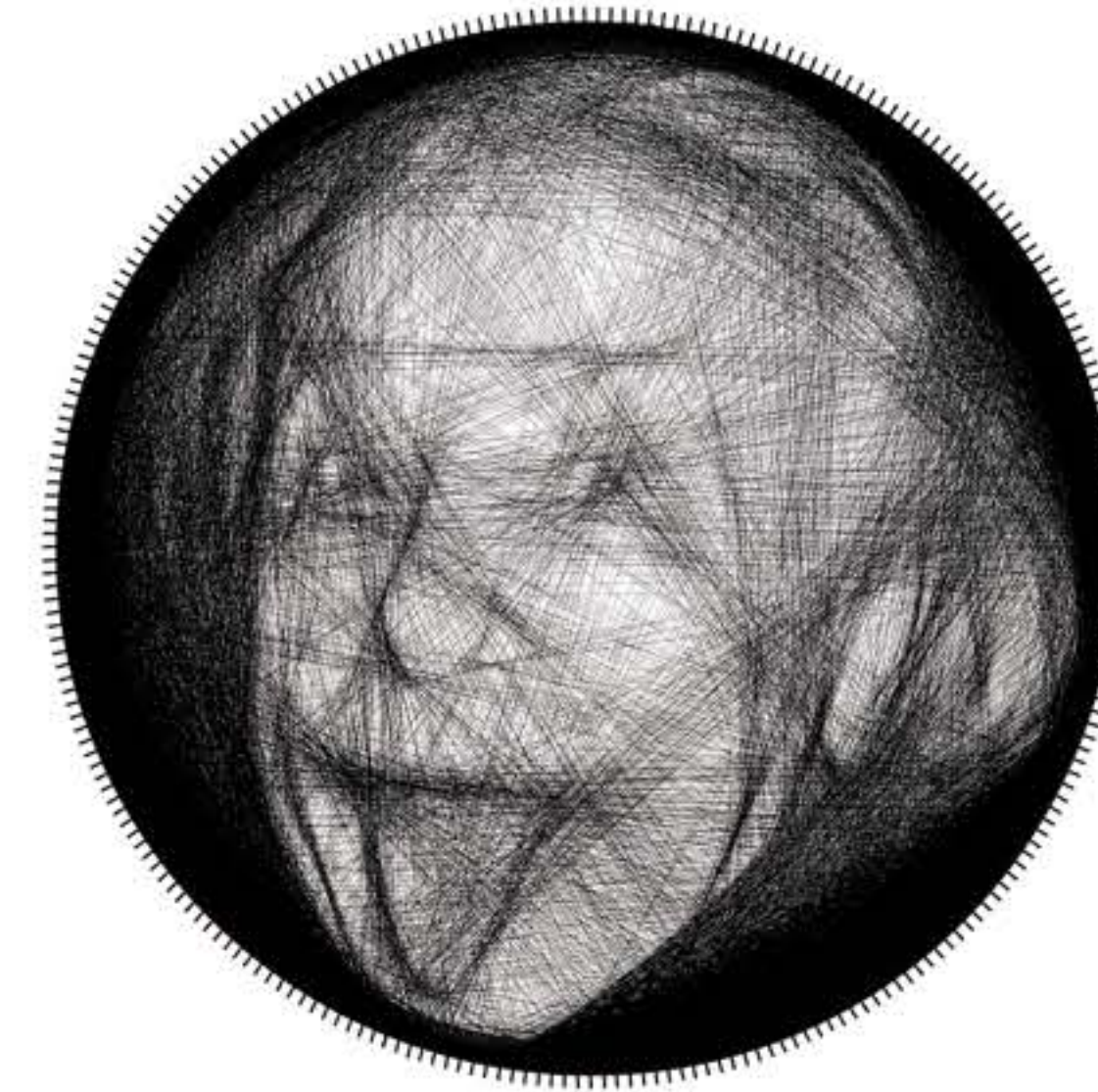
$$\max_{y_i} \sum_{i \in V_P} q(P_i) y_i$$

Layout

$$\max_{v_{\tau,i,x_c,y_c}} \sum_{\tau=1}^{\tau_{\max}} \sum_{i=1}^{n^*} \sum_{c \in C} q_{\tau,i,x_c,y_c} v_{\tau,i,x_c,y_c}$$



GUROBI
OPTIMIZATION



String Selection

$$\min_{x_j} \sum_{i=1}^m \delta_i$$

Not manageable
with Gurobi

➔ Own Greedy Solver

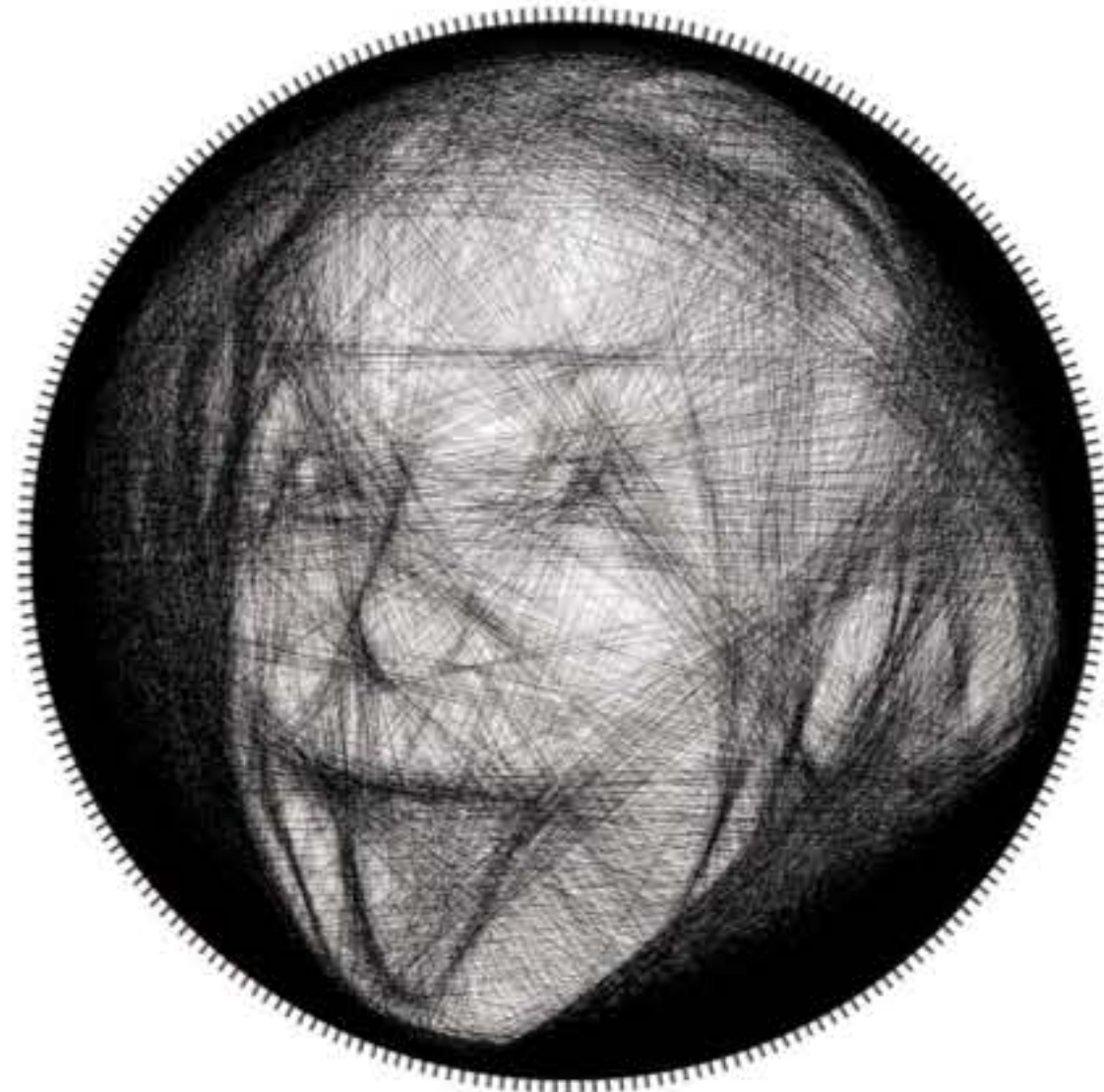
Overview



Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer

Dynamic Path Exploration on Mobile Devices

IEEE Transactions on Visualization and Computer Graphics. May 2018



Michael Birsak, Florian Rist, Peter Wonka, and Przemyslaw Musialski

String Art: Towards Computational Fabrication of String Images

Computer Graphics Forum (Proc. EUROGRAPHICS 2018). April 2018

Overview



Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer

Dynamic Path Exploration on Mobile Devices

IEEE Transactions on Visualization and Computer Graphics. May 2018



Michael Birsak, Florian Rist, Peter Wonka, and Przemyslaw Musialski

String Art: Towards Computational Fabrication of String Images

Computer Graphics Forum (Proc. EUROGRAPHICS 2018). April 2018

Motivation

- Mobile Devices (smartphones, tablets)

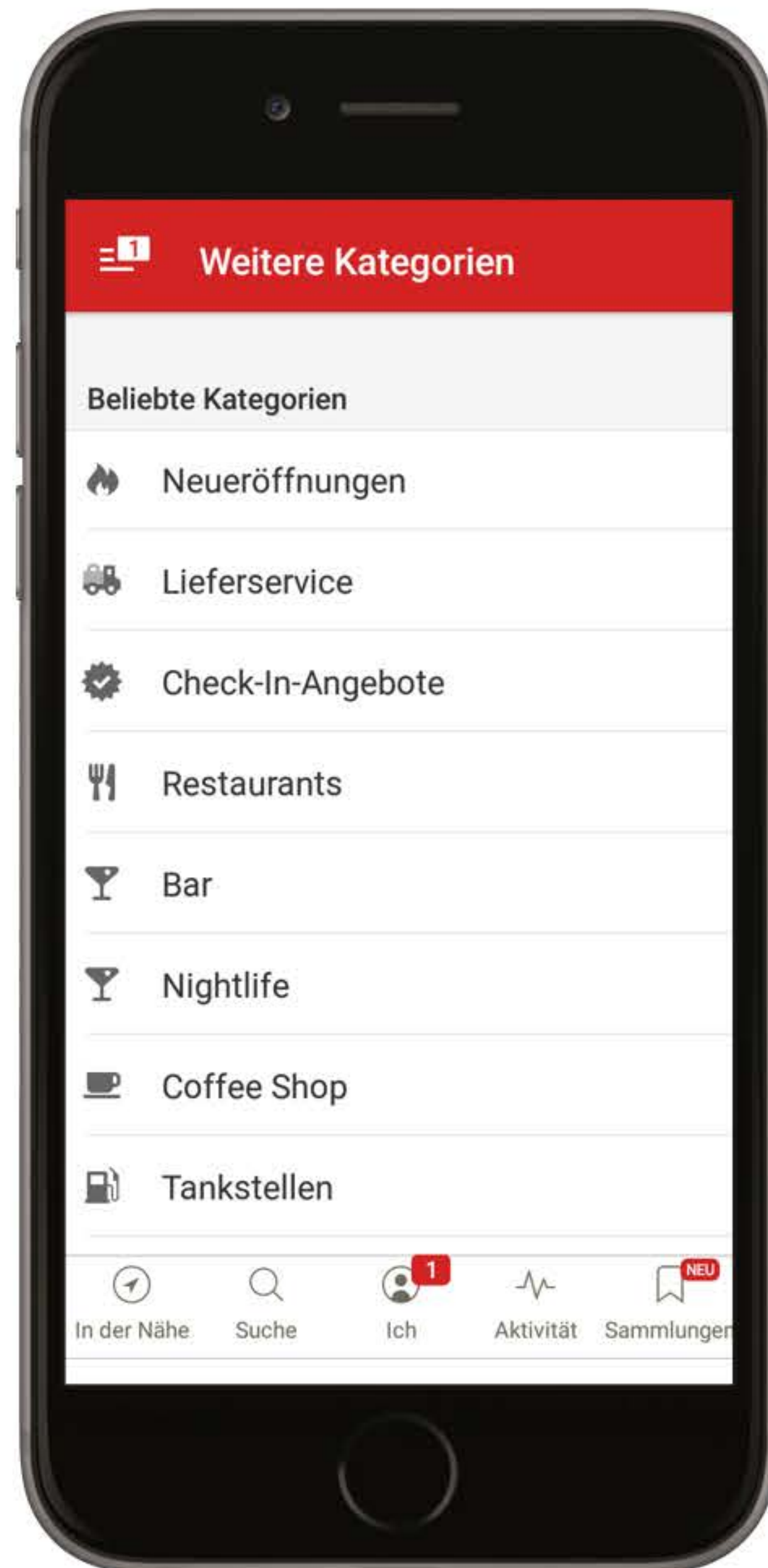


navigation

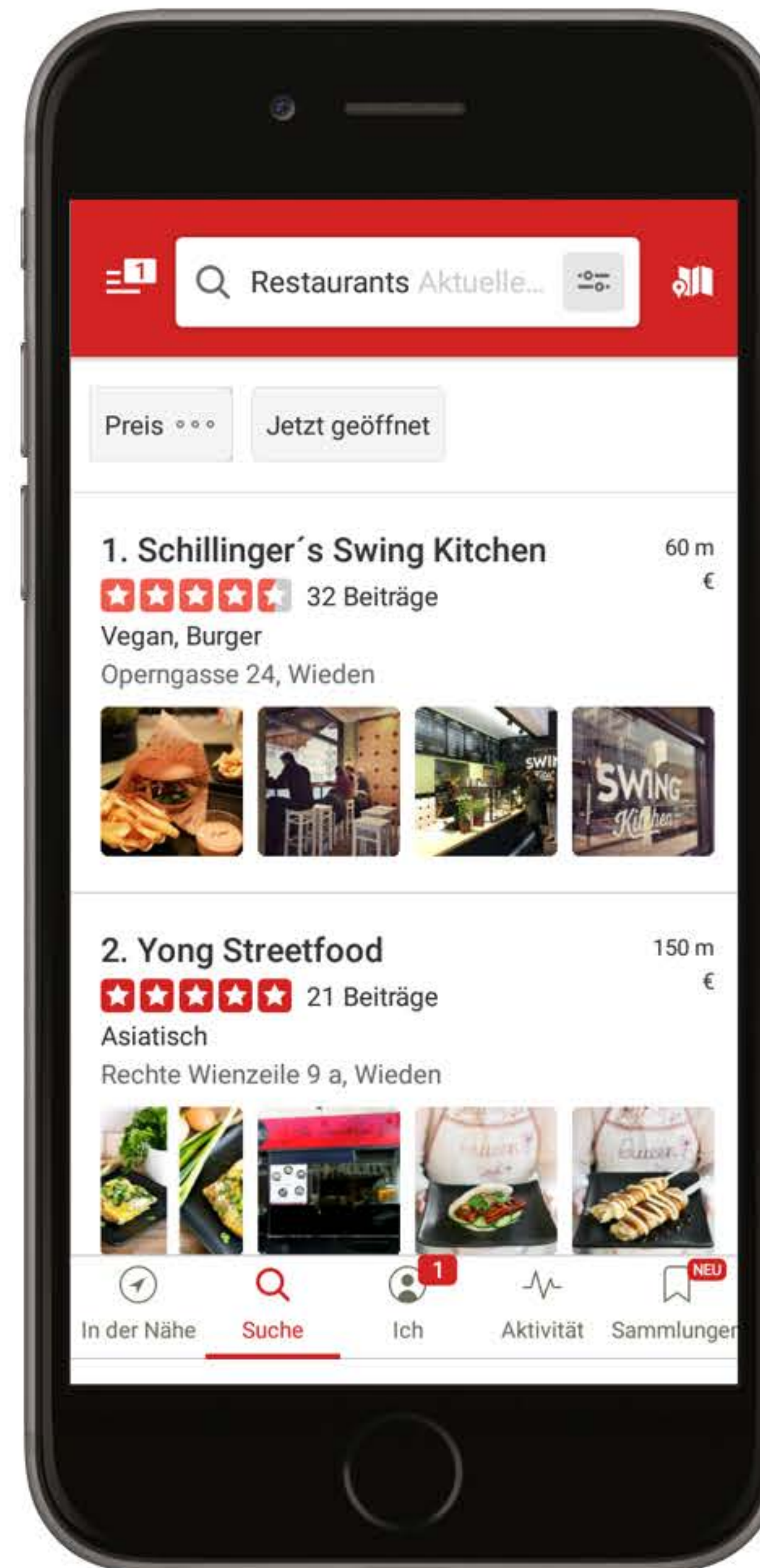
Motivation

- Common navigation approach using 

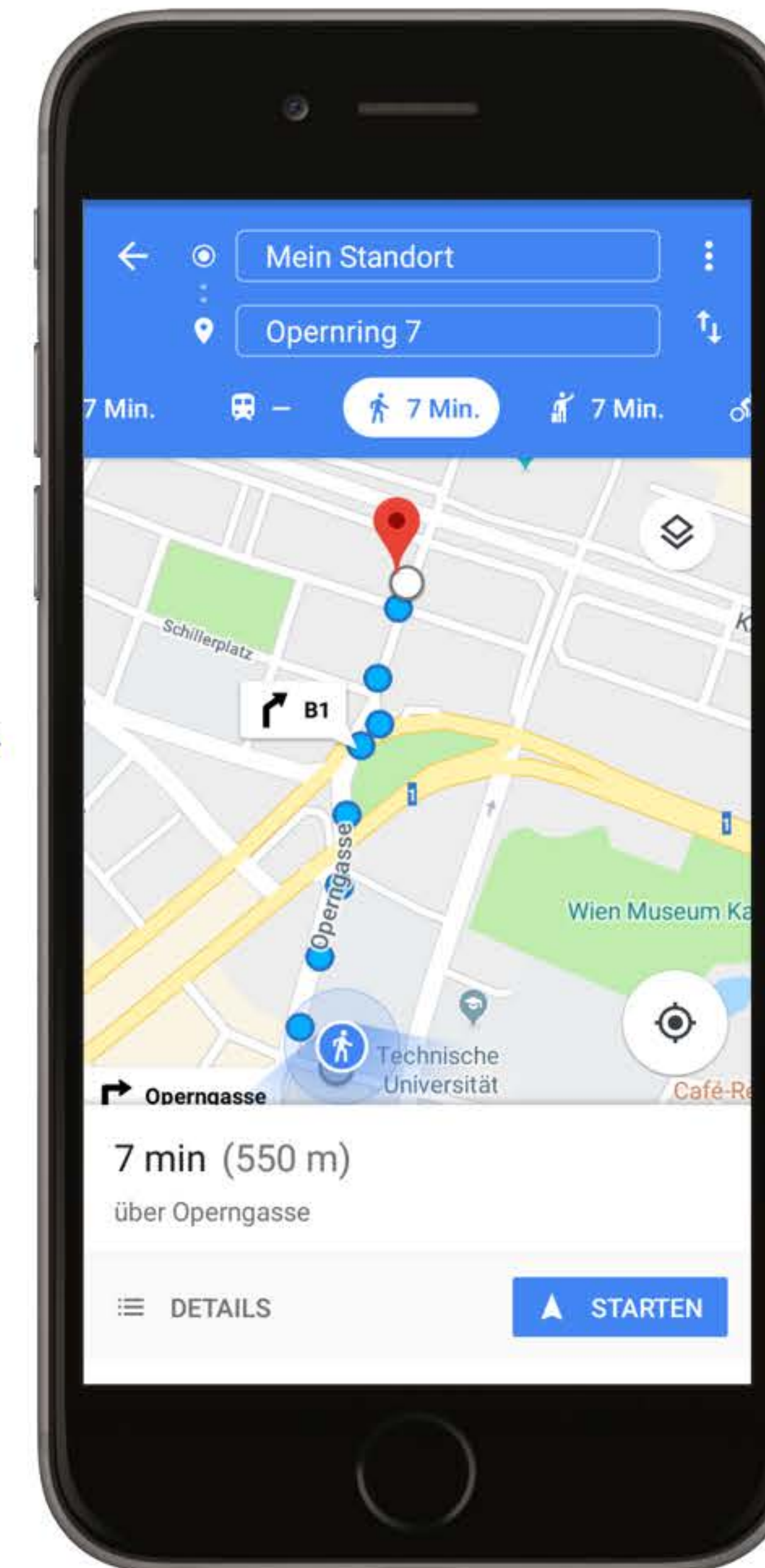
Choosing category



Choosing POI



Walking on shortest path



Motivation

- Problems of the common navigation scenario using yelp
 - 1. No guarantee that also the path is interesting
 - 2. Often user does not know where to go but is more interested in exploration

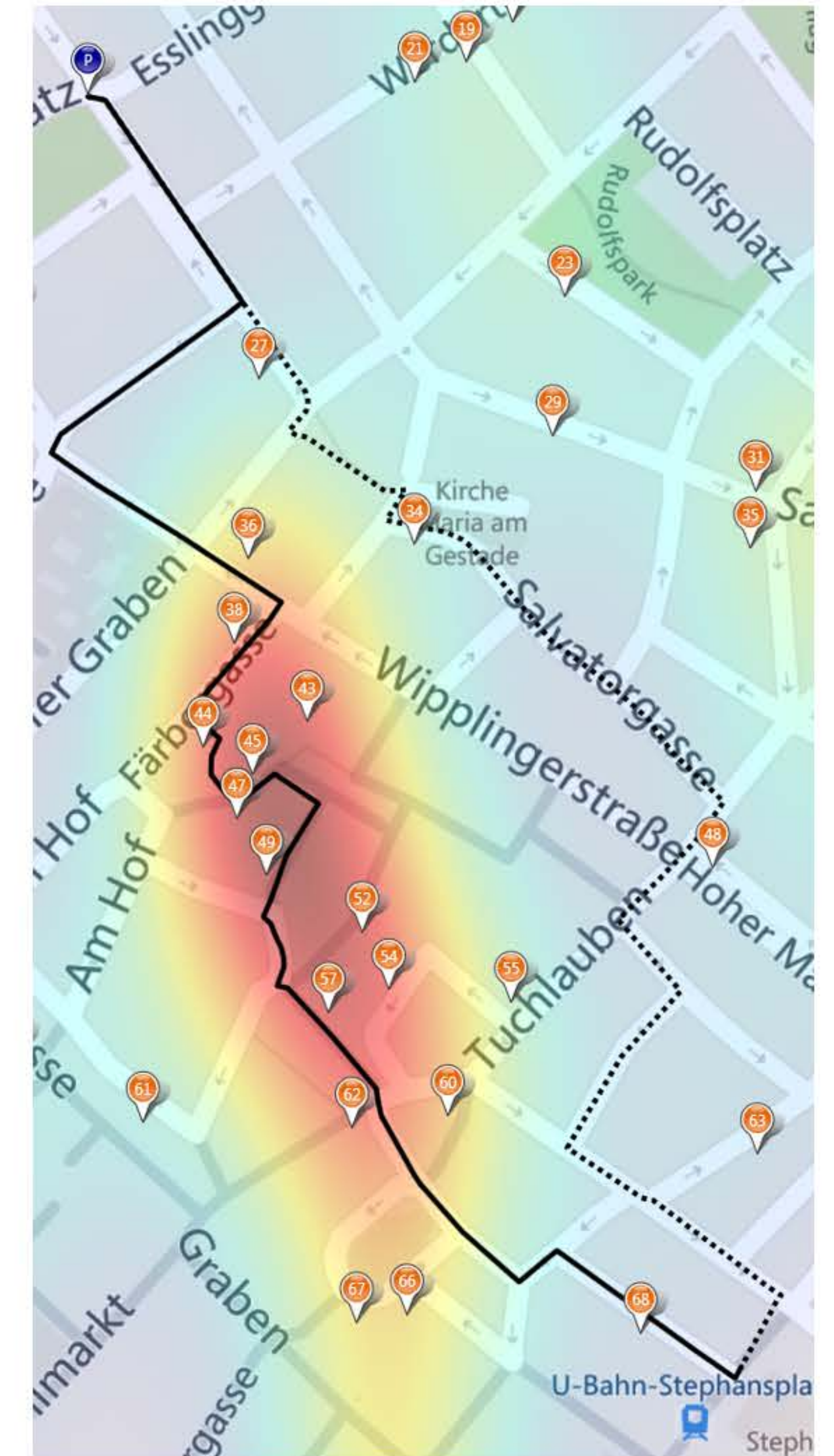
Contributions

Problems

1. No guarantee that also the path is interesting
2. Often user does not know where to go but is more interested in exploration

Solutions

1. Computation of interesting walking path
2. Focus on the environment, map cutout optimization and dynamic visualization



Contributions

Problems

1. No guarantee that also the path is interesting
2. Often user does not know where to go but is more interested in exploration

Solutions

1. Computation of interesting walking path
2. Focus on the environment, map cutout optimization and dynamic visualization



Contributions

Problems

1. No guarantee that also the path is interesting
2. Often user does not know where to go but is more interested in exploration

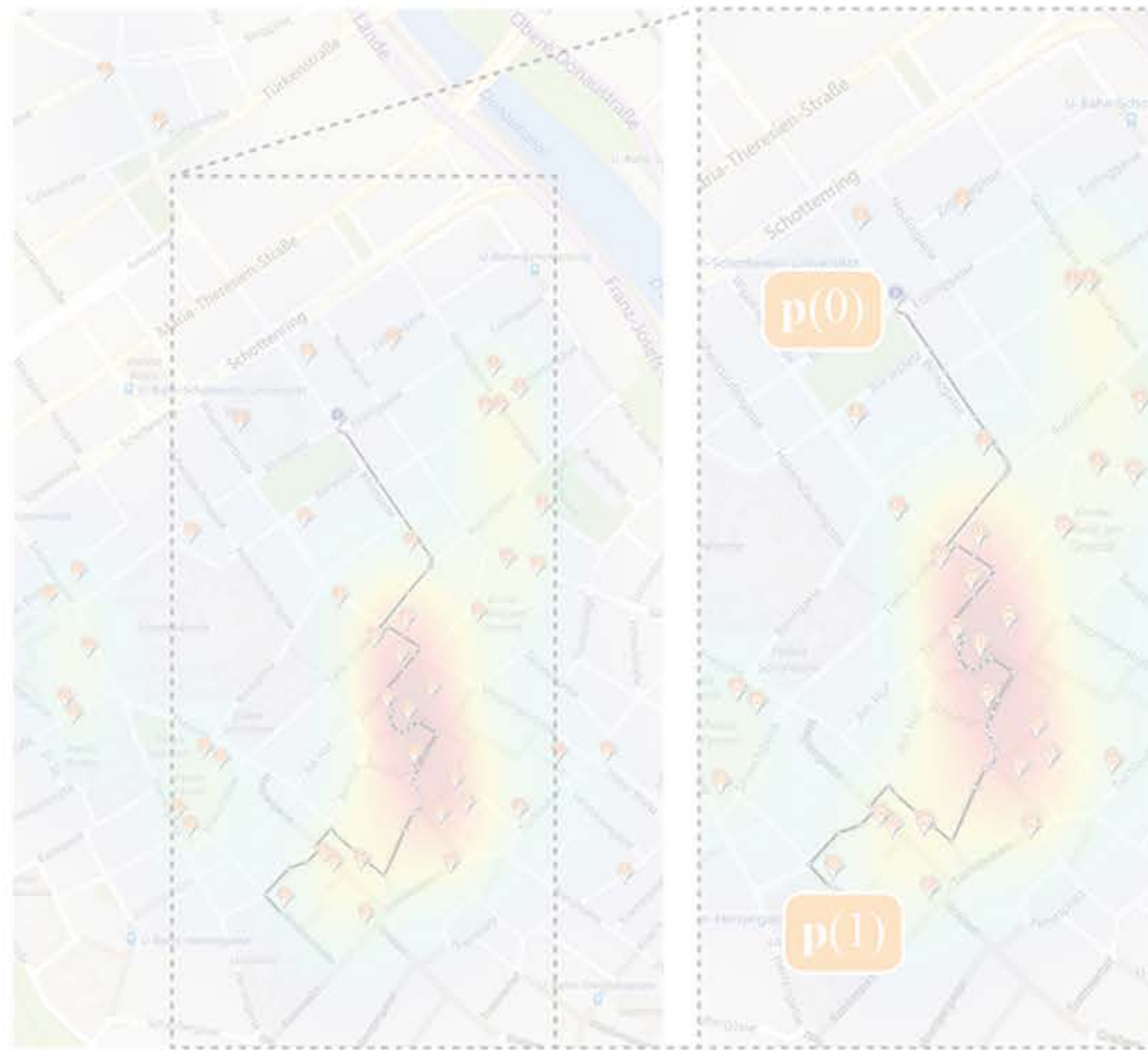
Solutions

1. Computation of interesting walking path
2. Focus on the environment, map cutout optimization and dynamic visualization

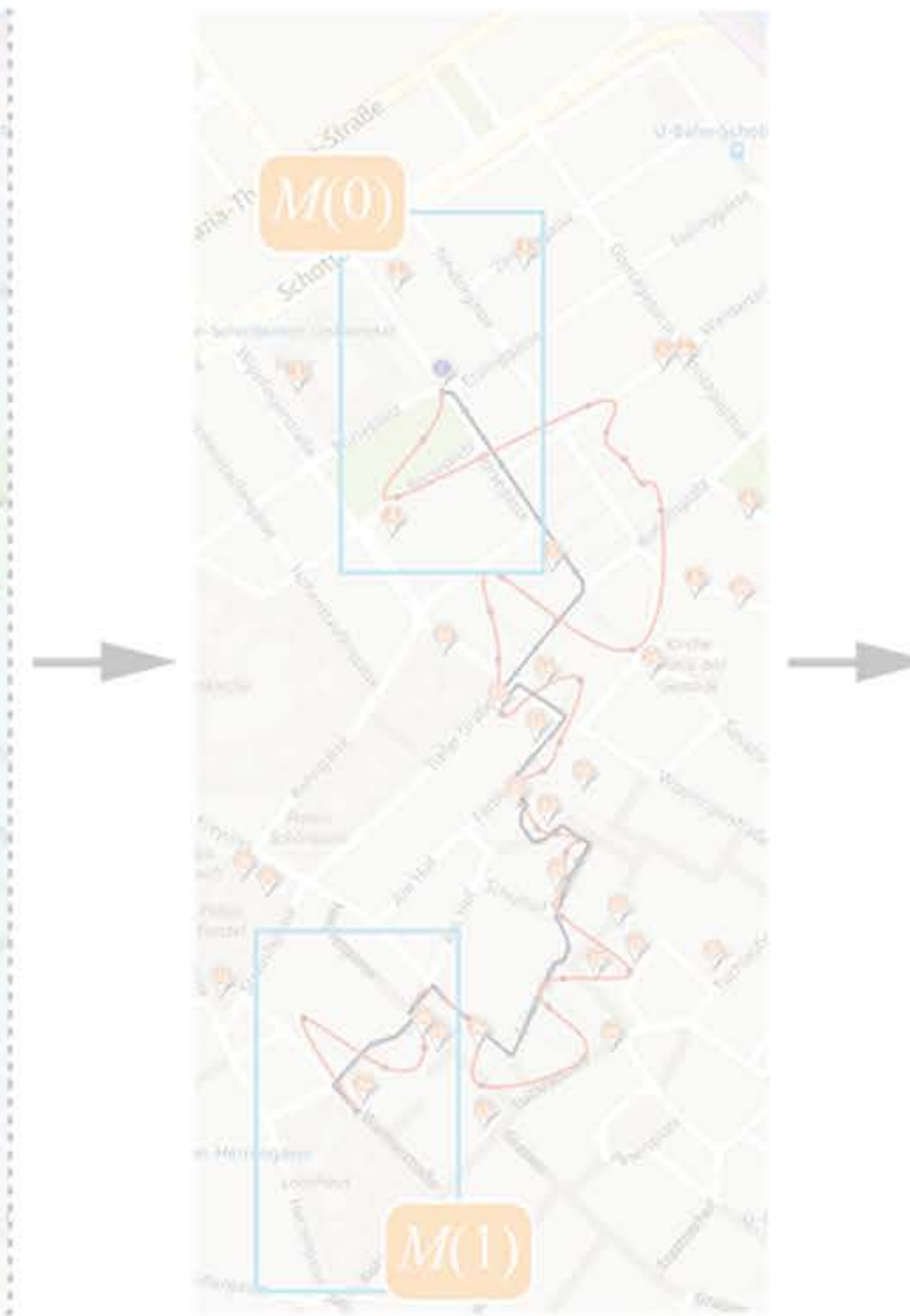


System Overview

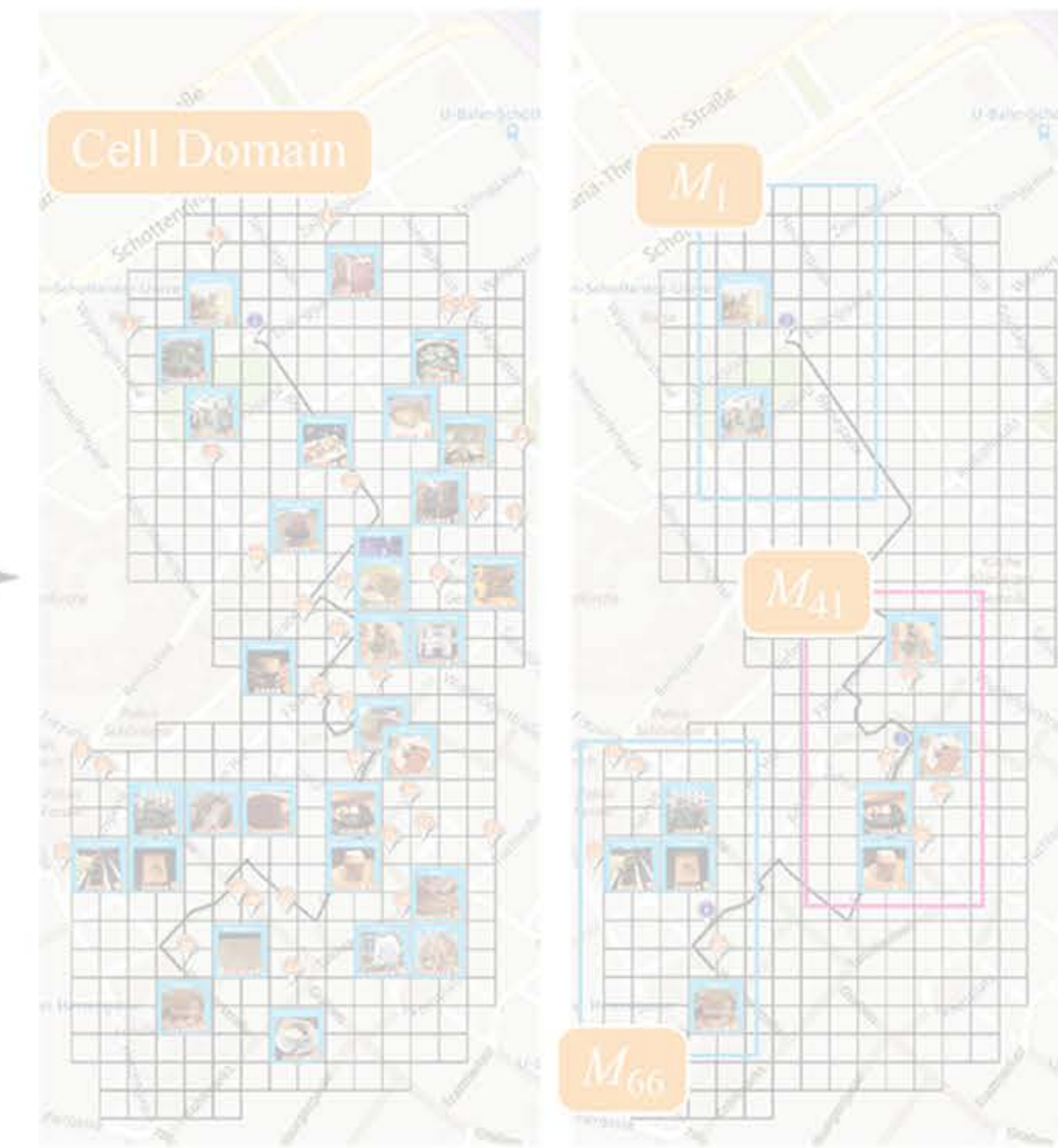
- Input: User position p , Points of Interest $P = \{P_1, P_2, \dots, P_n\}$



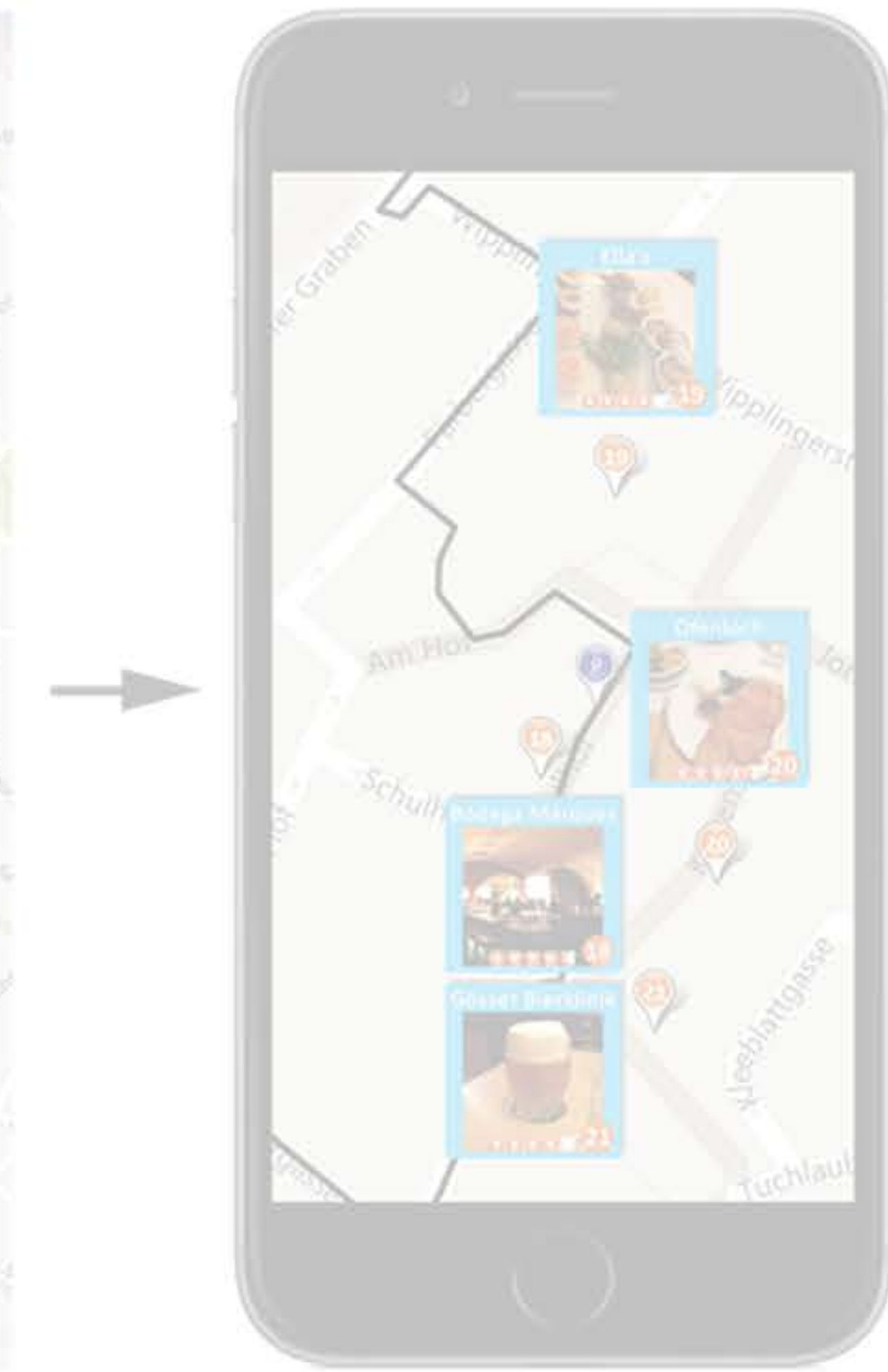
Path Computation



Map Cutout Optimization



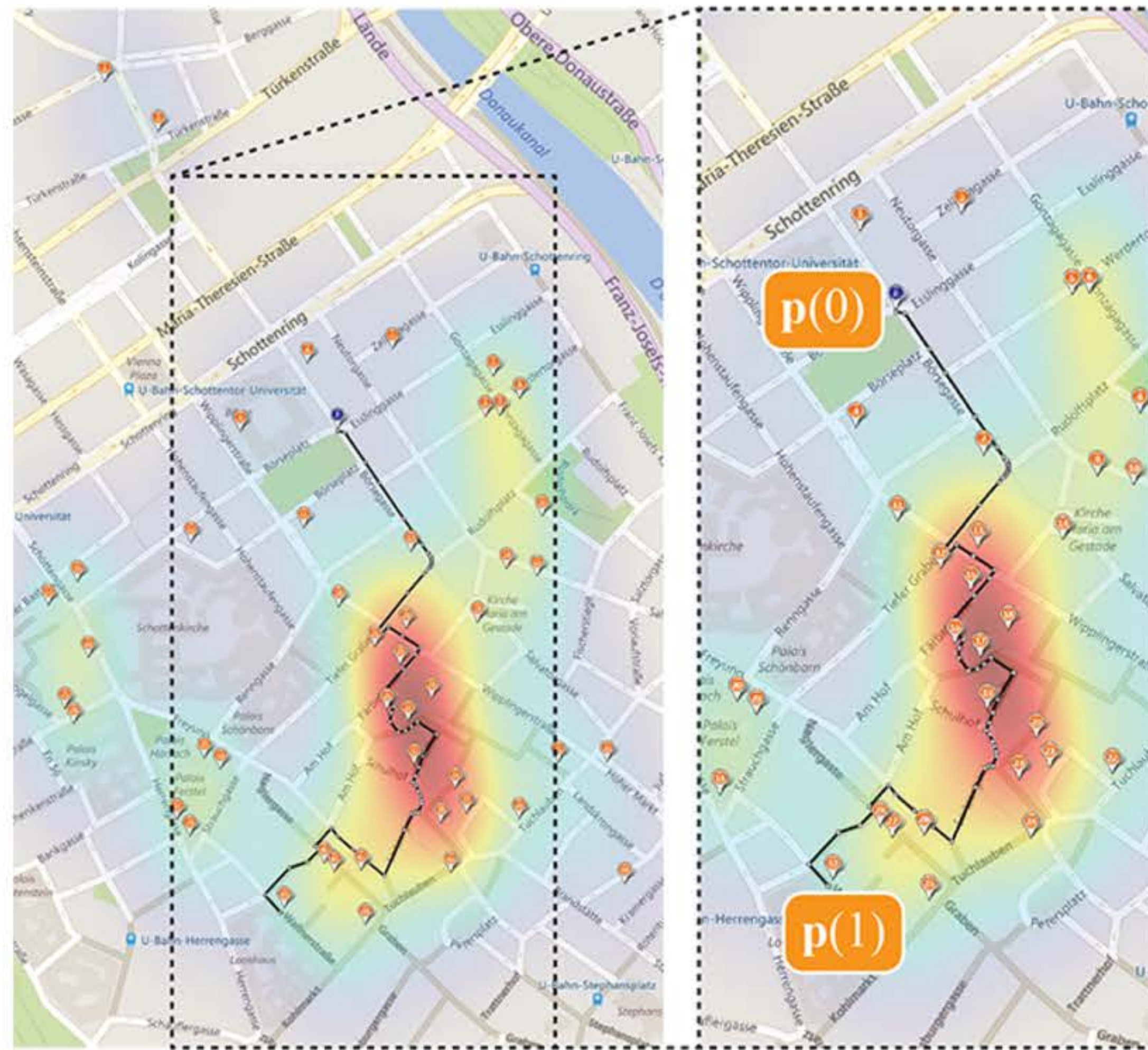
Detail Lens Placement



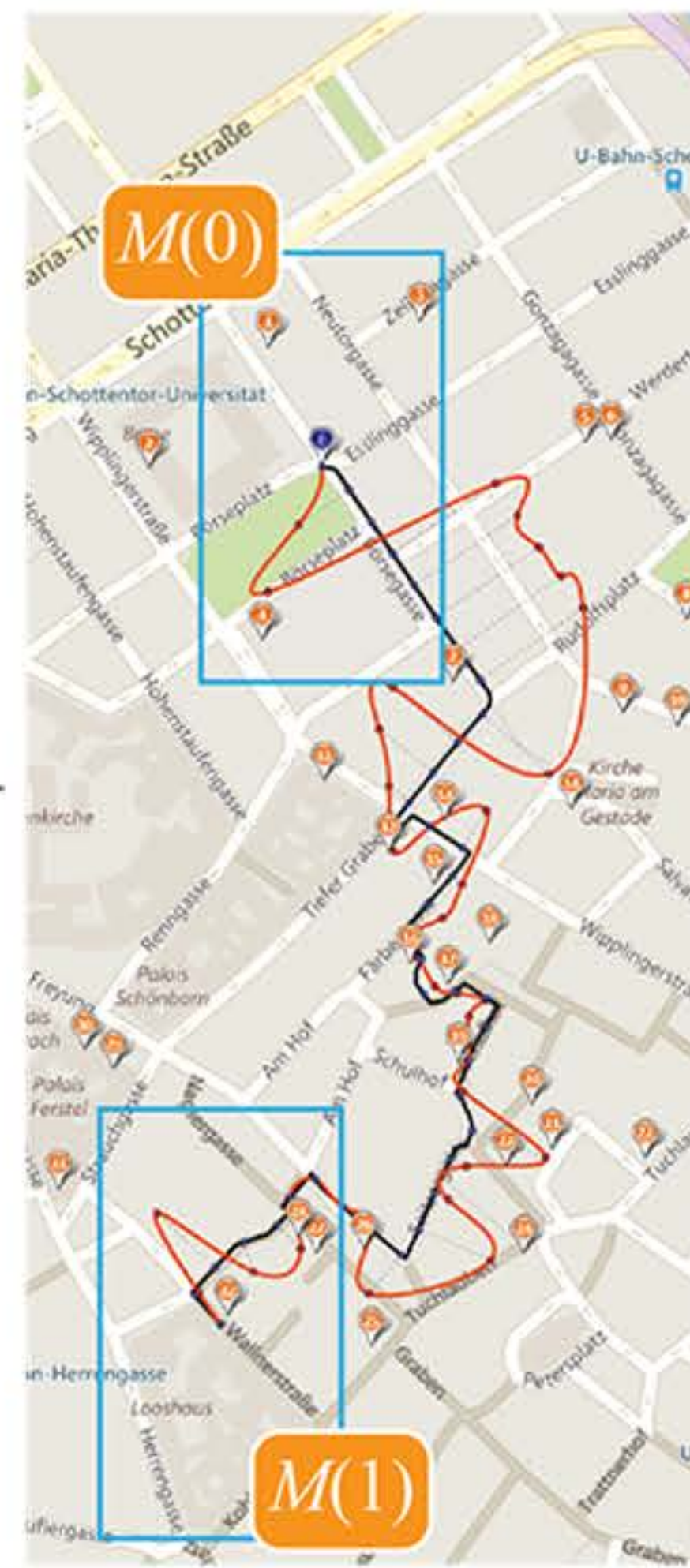
Visualization

System Overview

- Input: User position p , Points of Interest $P = \{P_1, P_2, \dots, P_n\}$



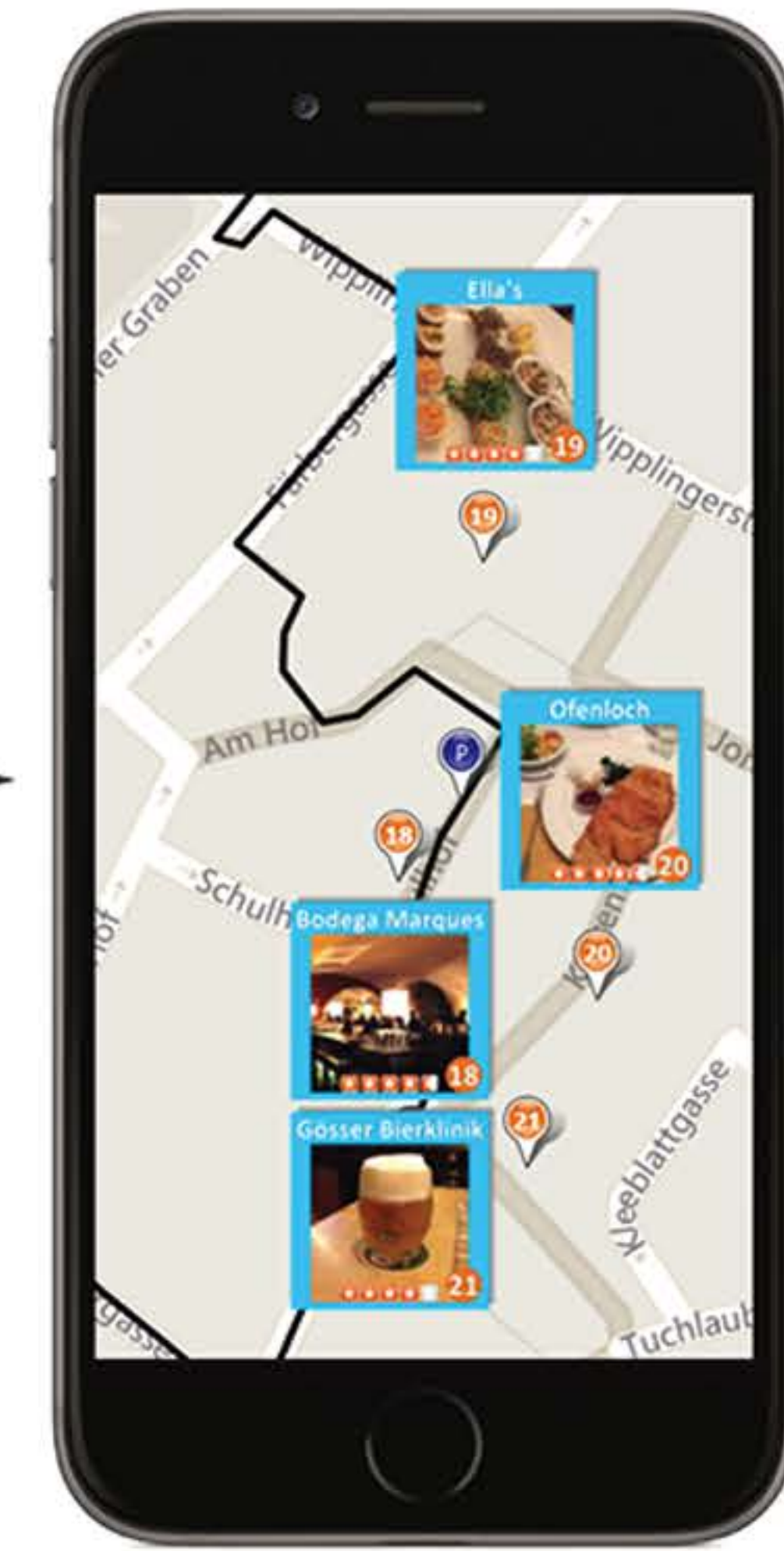
Path Computation



Map Cutout Optimization



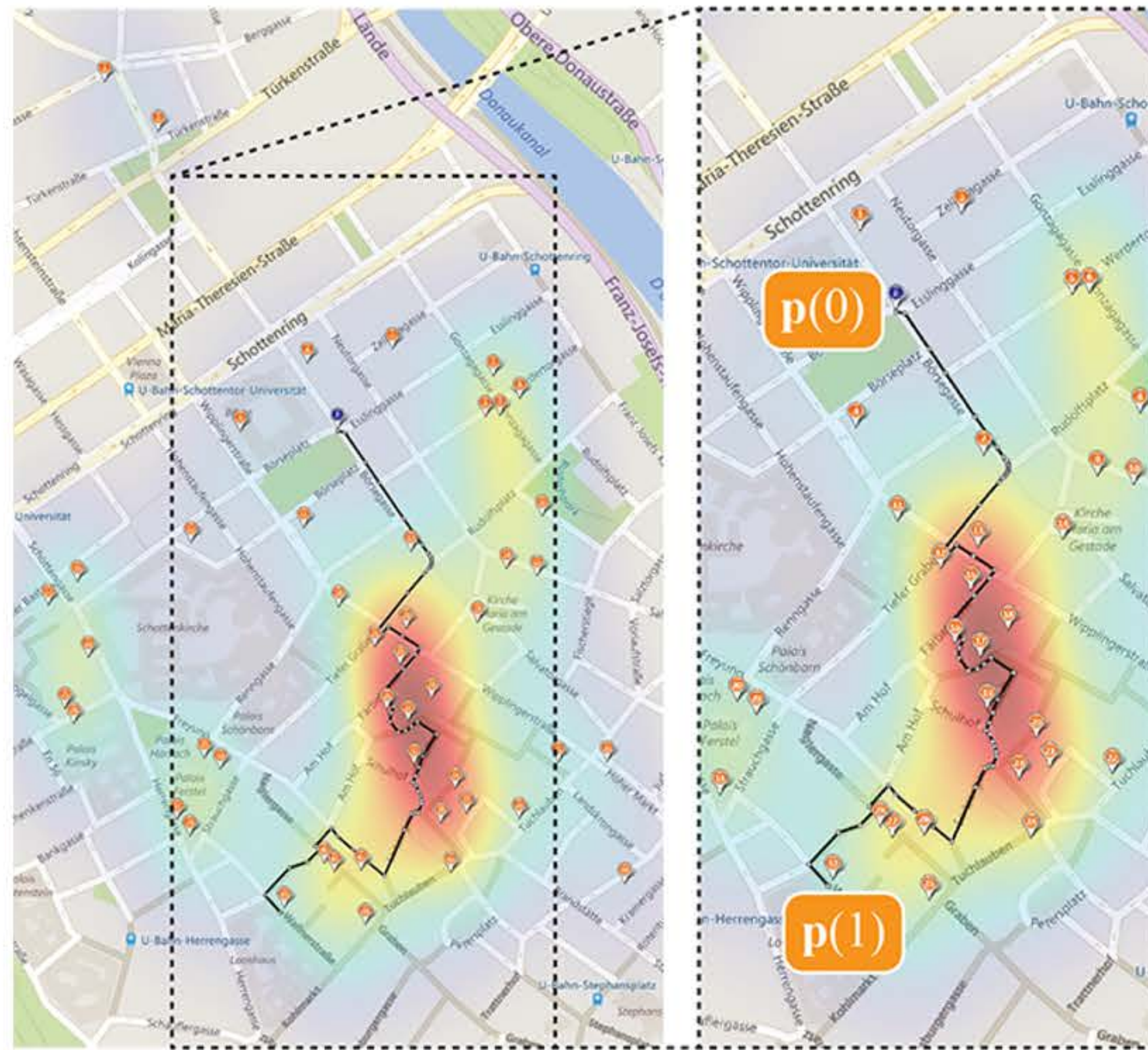
Detail Lens Placement



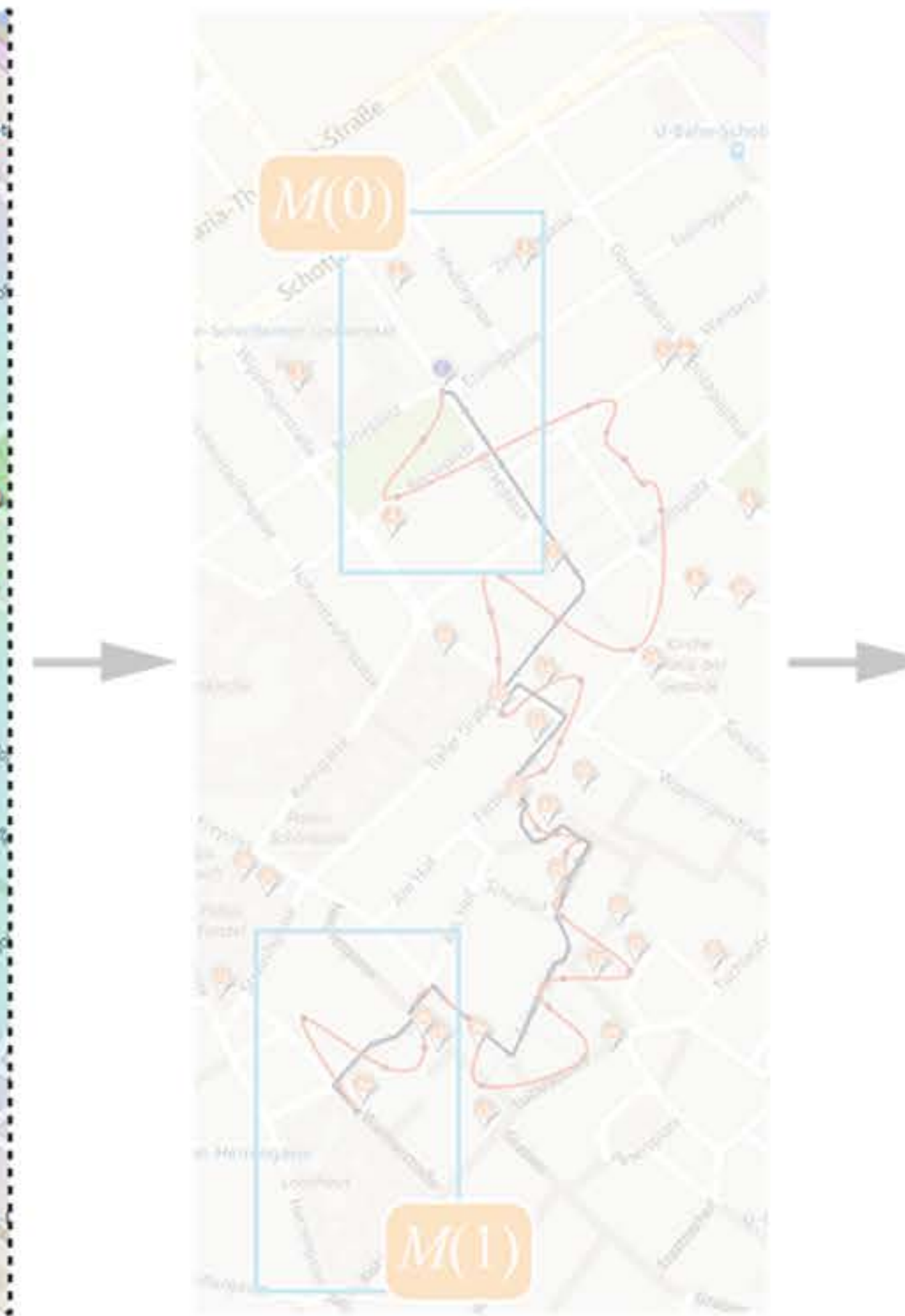
Visualization

System Overview

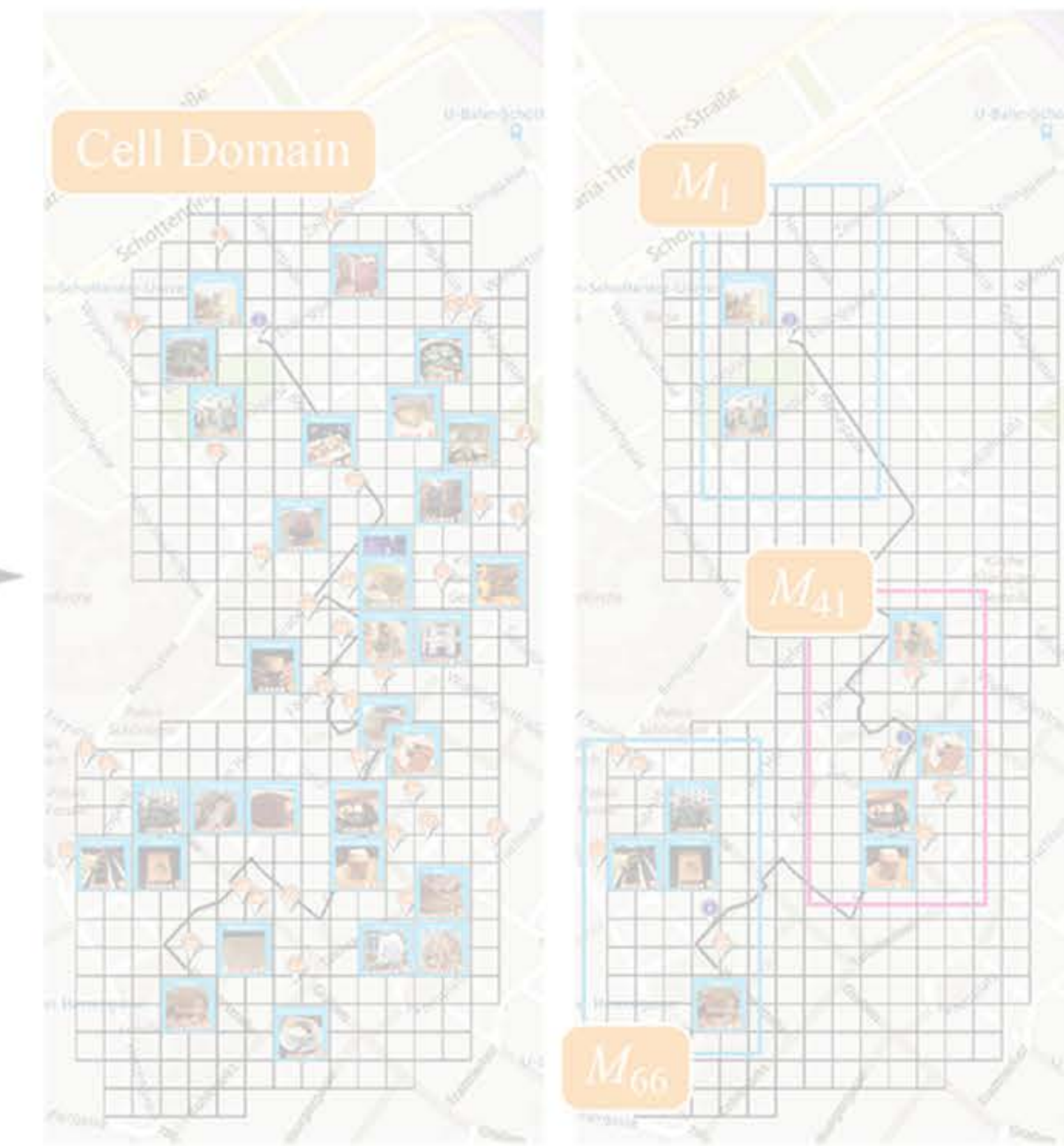
- Input: User position p , Points of Interest $P = \{P_1, P_2, \dots, P_n\}$



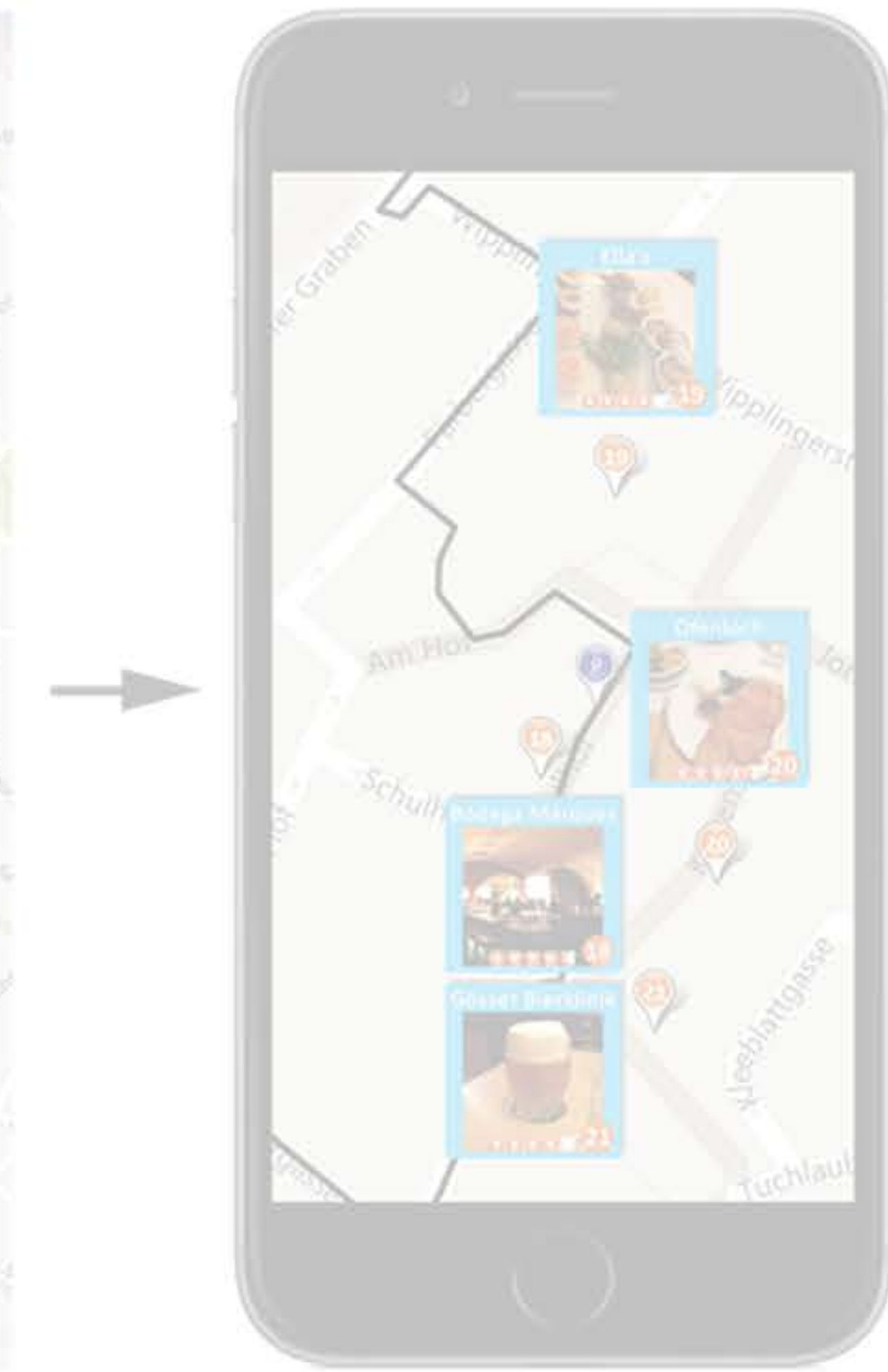
Path Computation



Map Cutout
Optimization



Detail Lens
Placement



Visualization

Path Computation

- Goals
 1. It should be interesting (modeled similar to Grabler et al. [2008])
 2. Walking distance should stay below threshold (e.g. $1.2 \cdot \delta_{min}$)
- Choice of destination d
 1. either by the user
 2. or by our system

Path Computation



Path

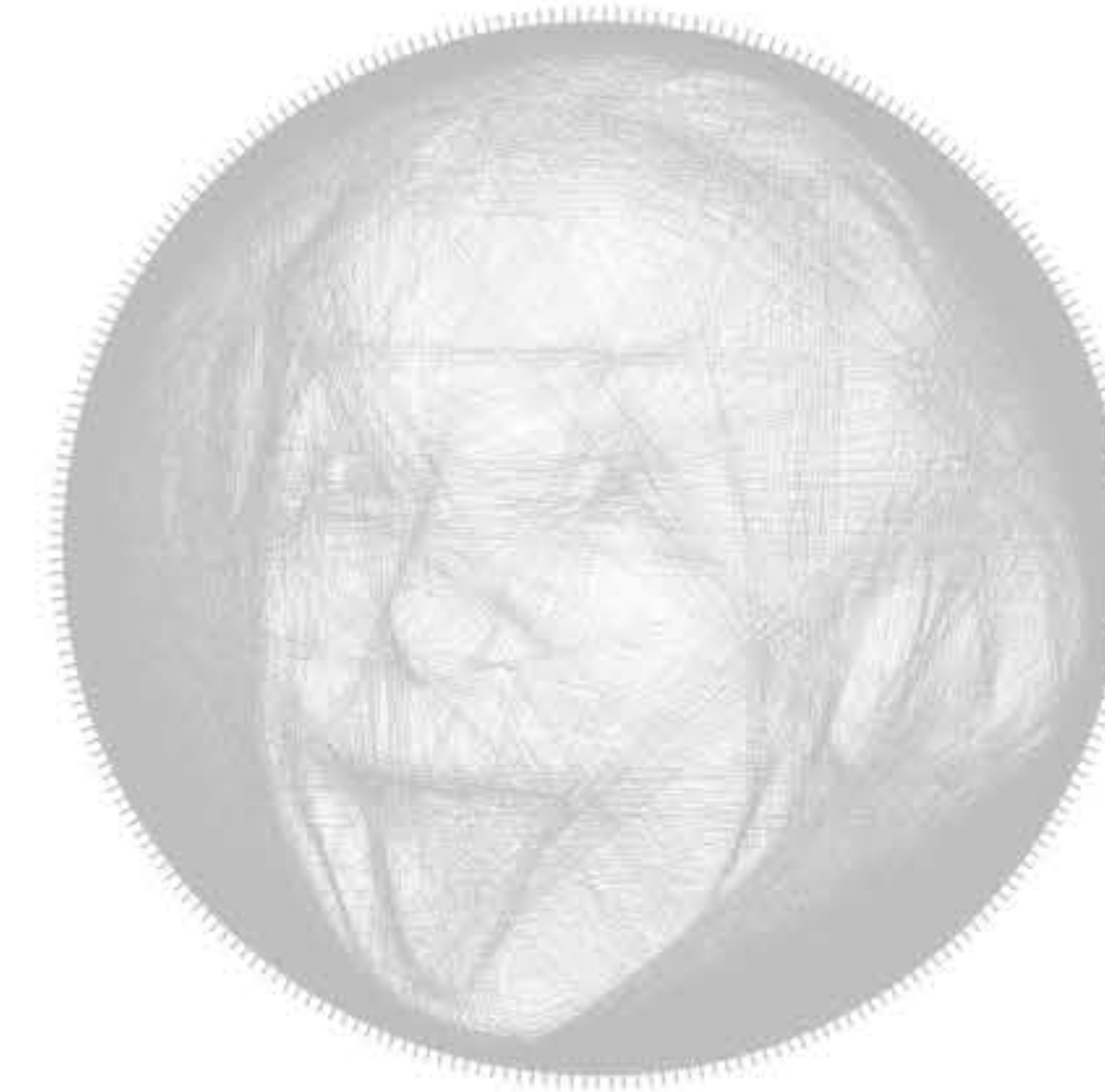
$$\max_{y_i} \sum_{i \in V_P} q(P_i) y_i$$

Layout

$$\max_{v_{\tau,i,x_c,y_c}} \sum_{\tau=1}^{\tau_{\max}} \sum_{i=1}^{n^*} \sum_{c \in C} q_{\tau,i,x_c,y_c} v_{\tau,i,x_c,y_c}$$



GUROBI
OPTIMIZATION



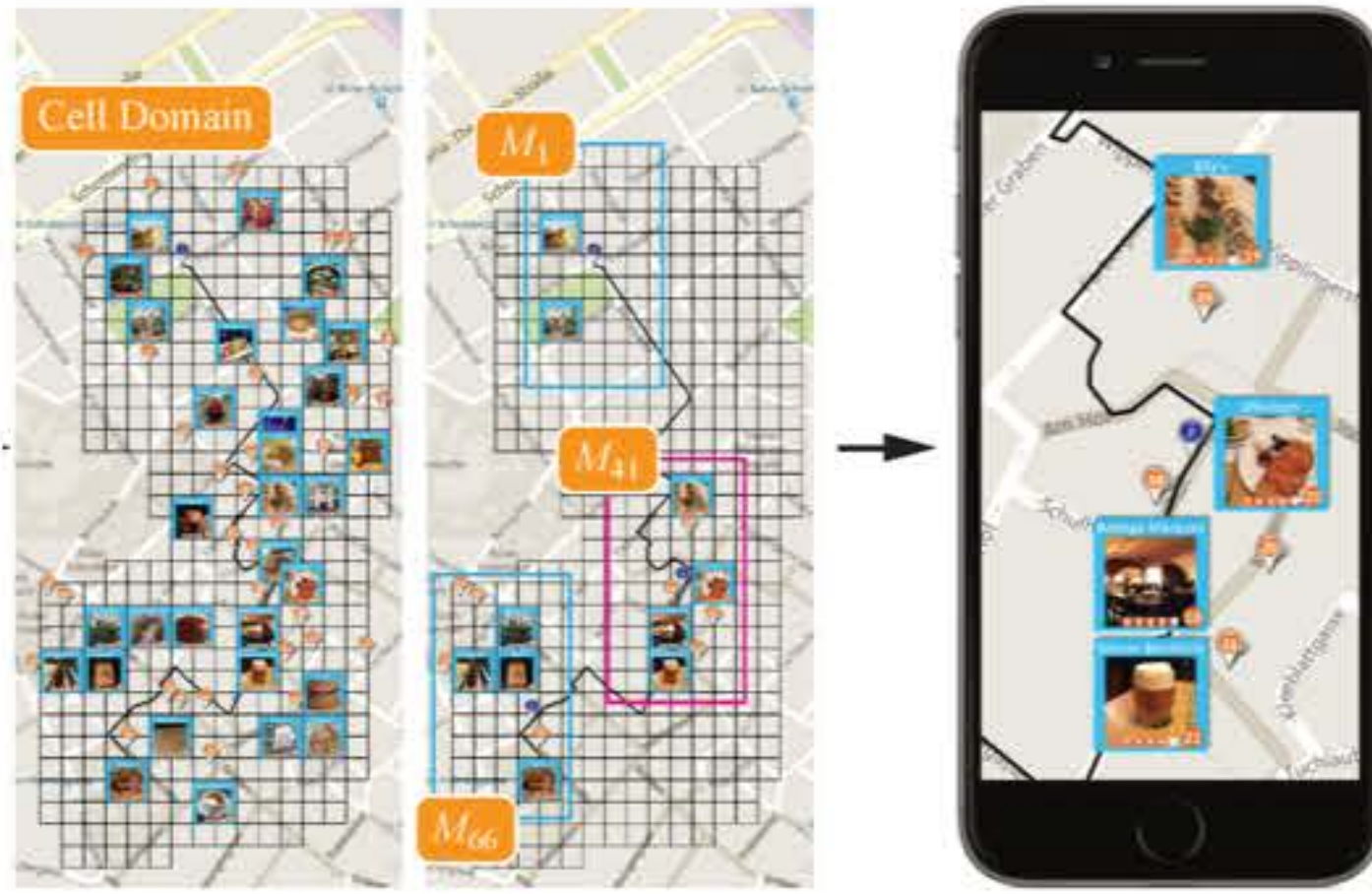
String Selection

$$\min_{x_j} \sum_{i=1}^m \delta_i$$

Not manageable
with Gurobi

➡ Own Greedy Solver

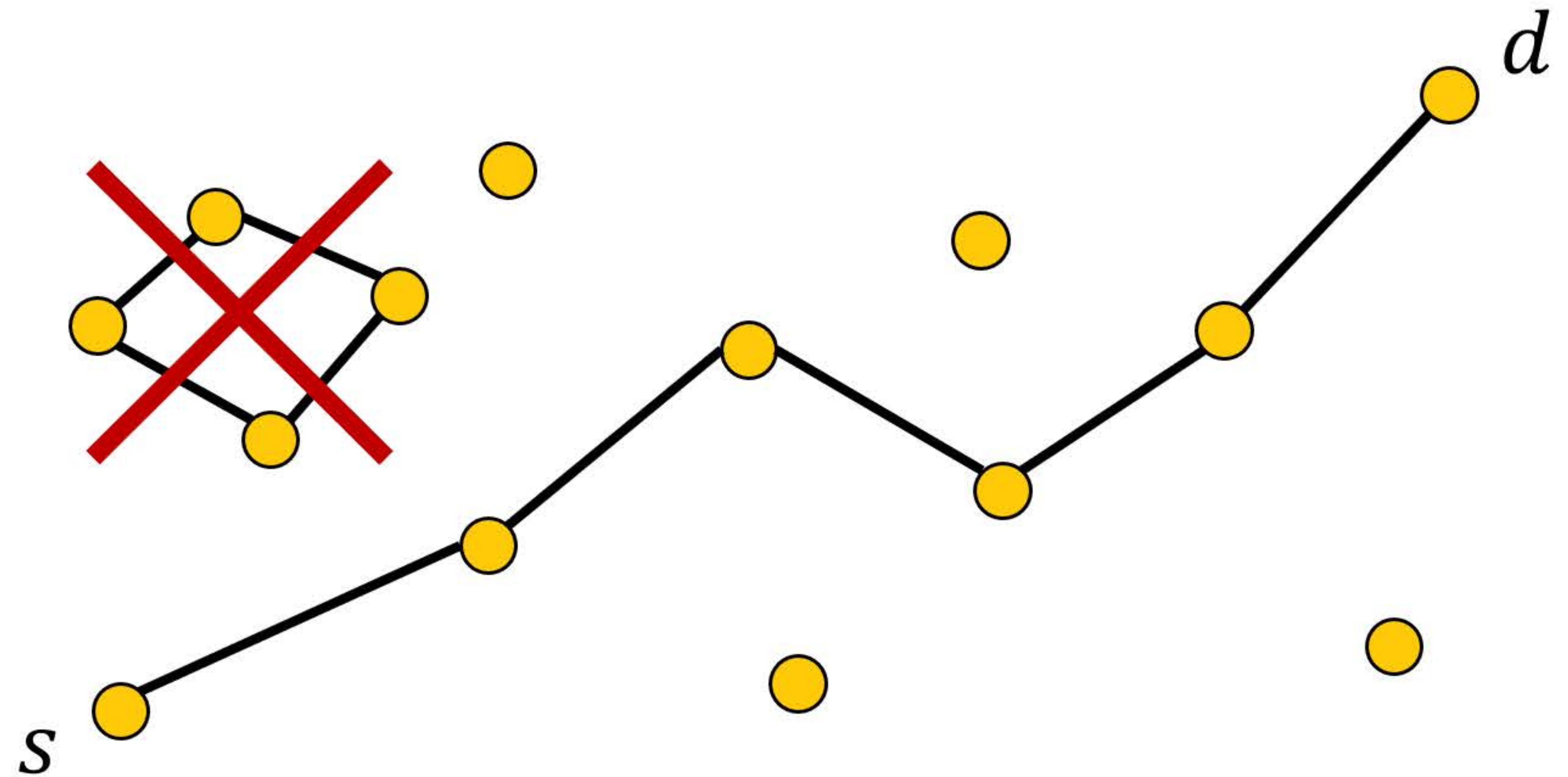
Path Computation



Path

$$\max_{y_i} \sum_{i \in V_P} q(P_i) y_i$$

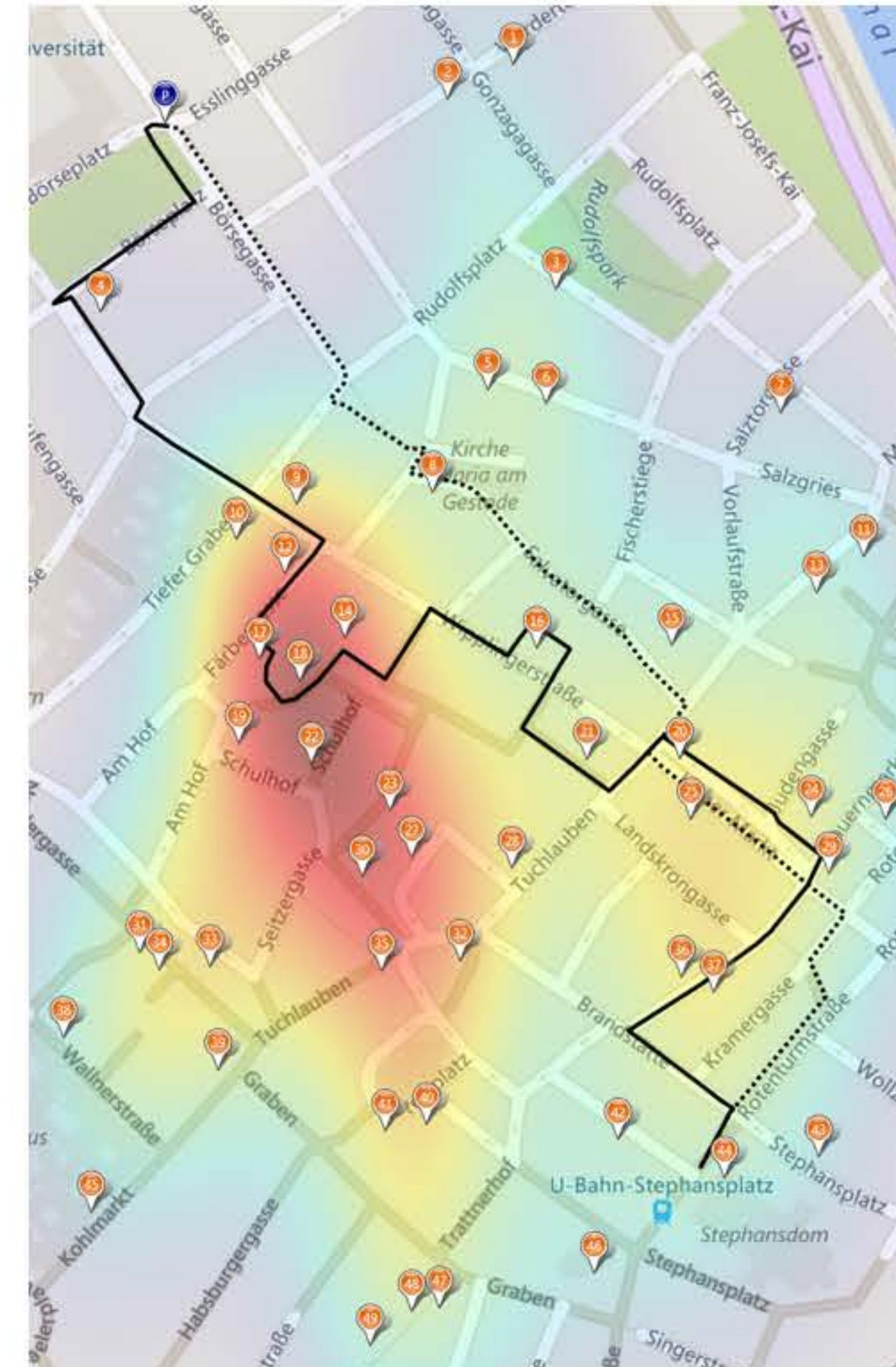
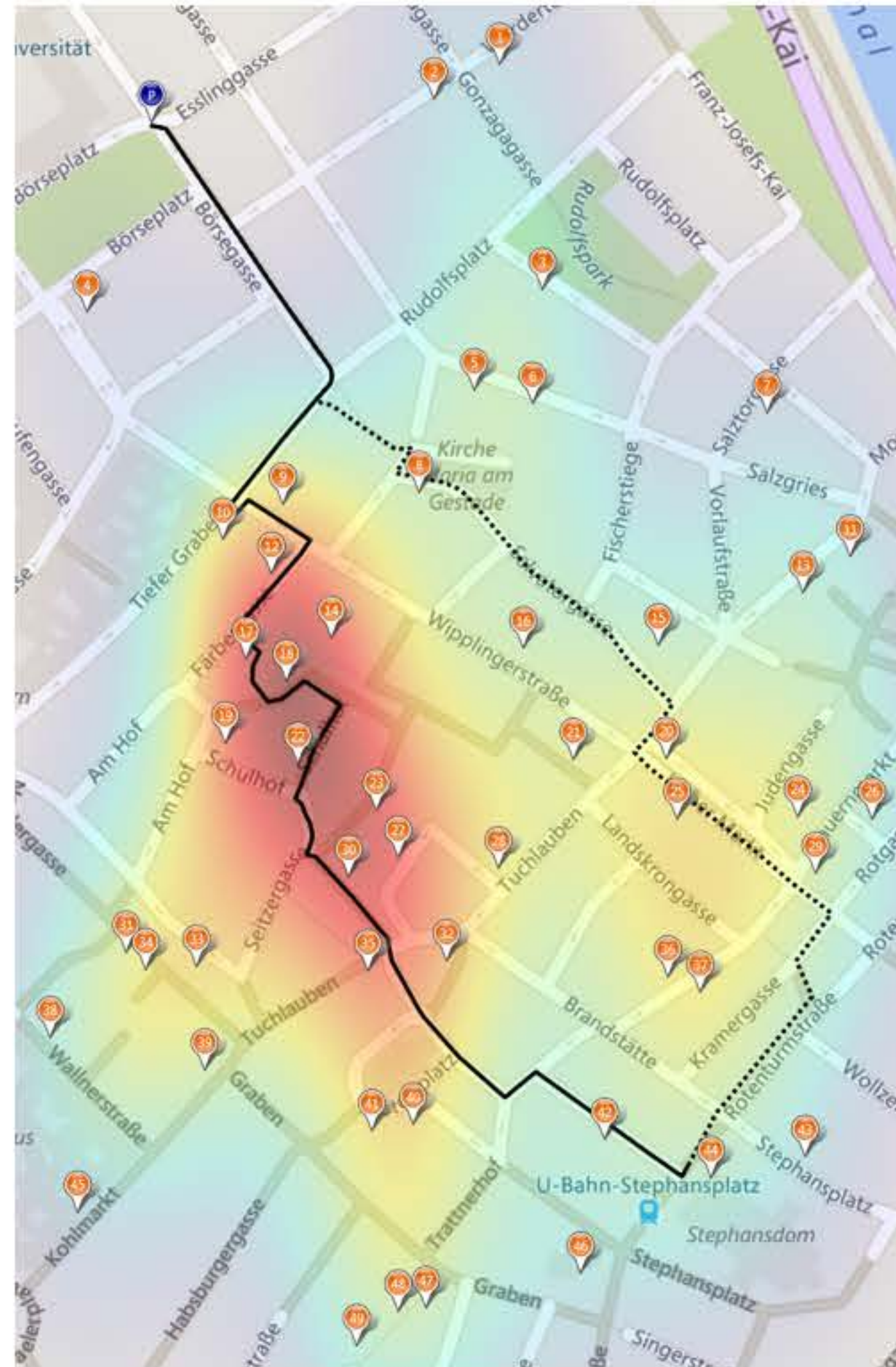
- $y_i = 1$ when vertex v_i is visited
- $q(P_i)$ is the ranking value (quality) of P_i
- subject to the following constraints:



GUROBI
OPTIMIZATION

Path Computation

- Influence of the walking distance threshold $\delta_{\max}$



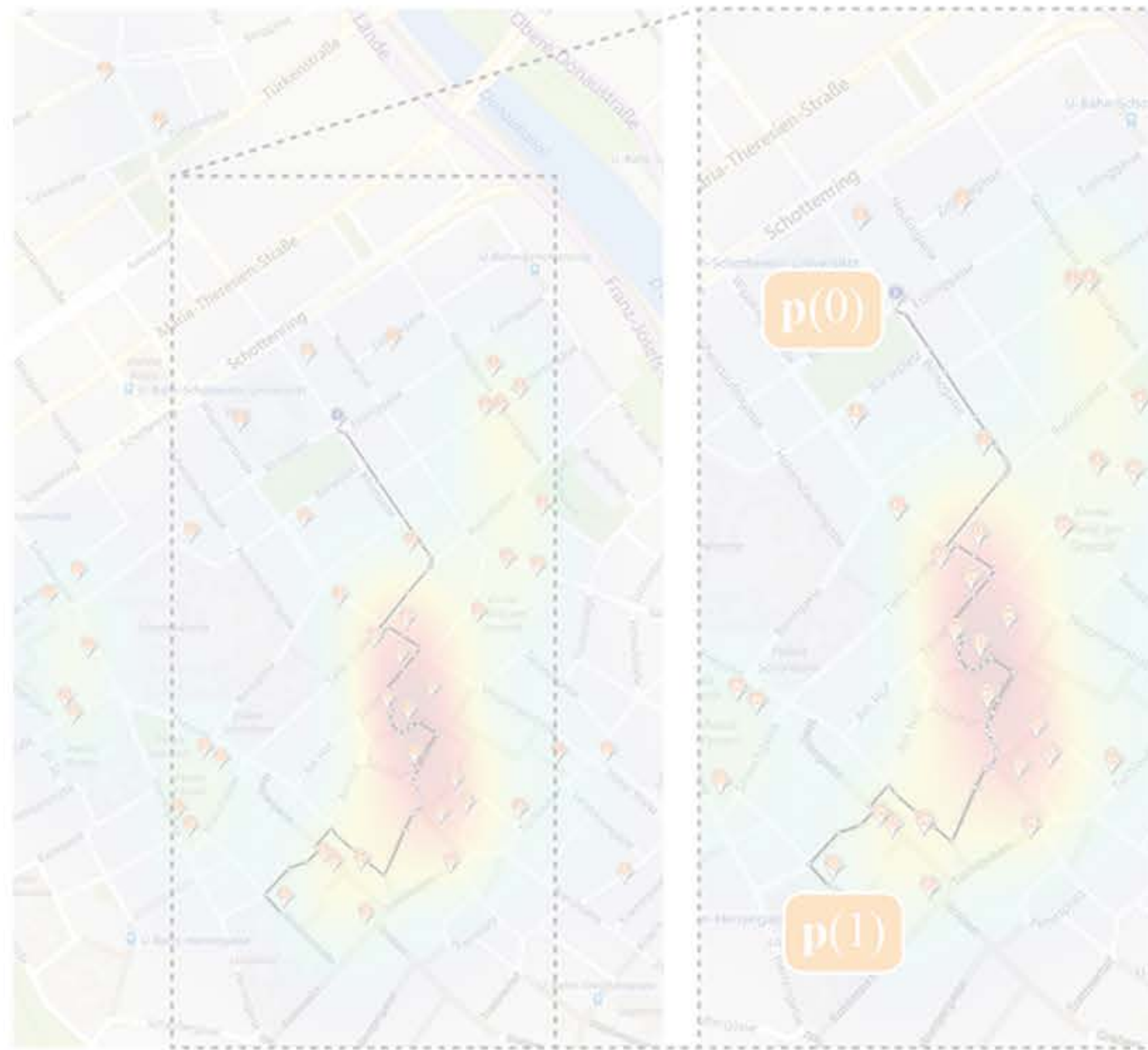
Path Quality =
 $\delta_{\max} = \delta_{\min} + \dots$

34
+10%

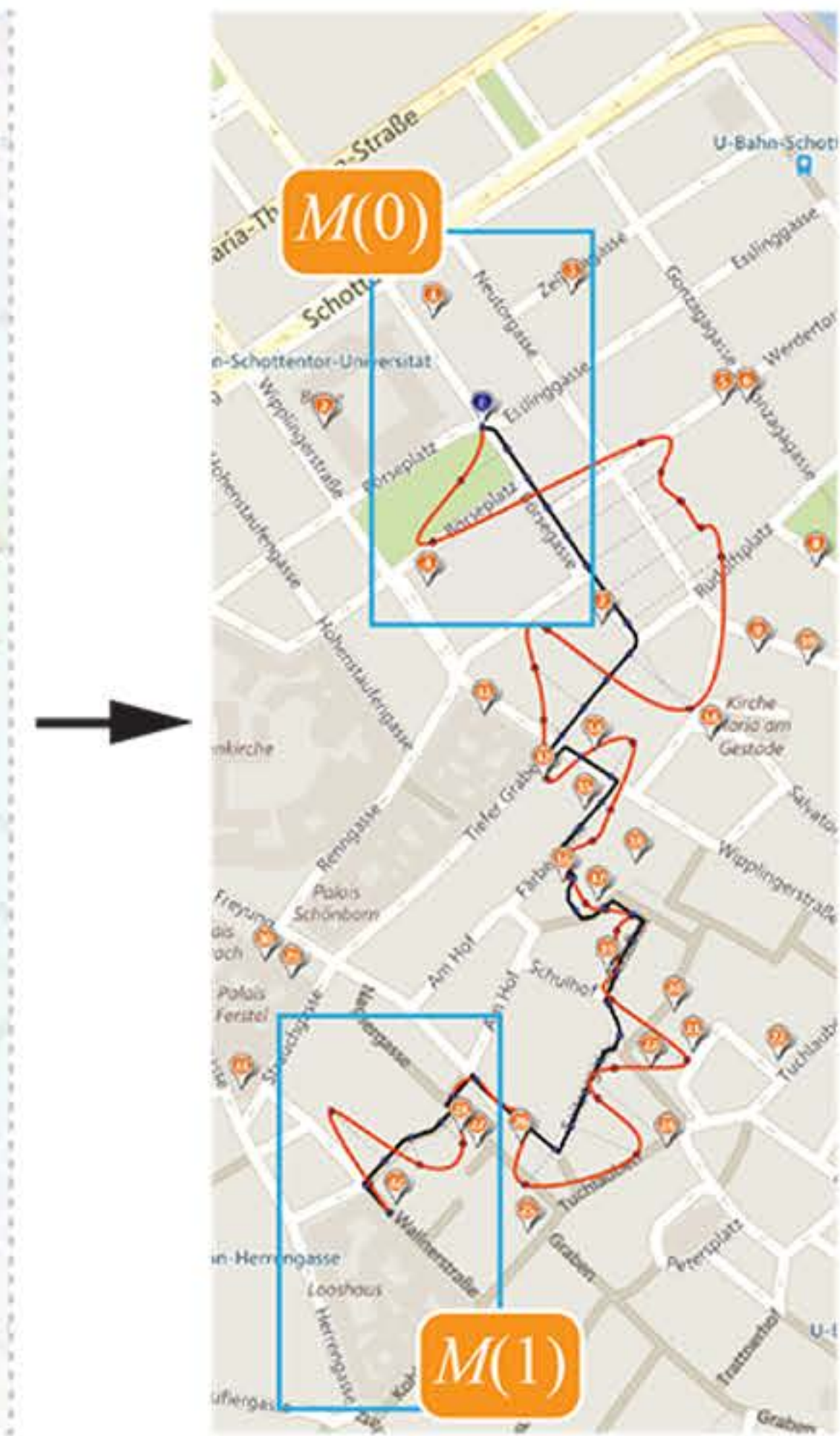
60.5
+50%

System Overview

- Input: User position p , Points of Interest $P = \{P_1, P_2, \dots, P_n\}$



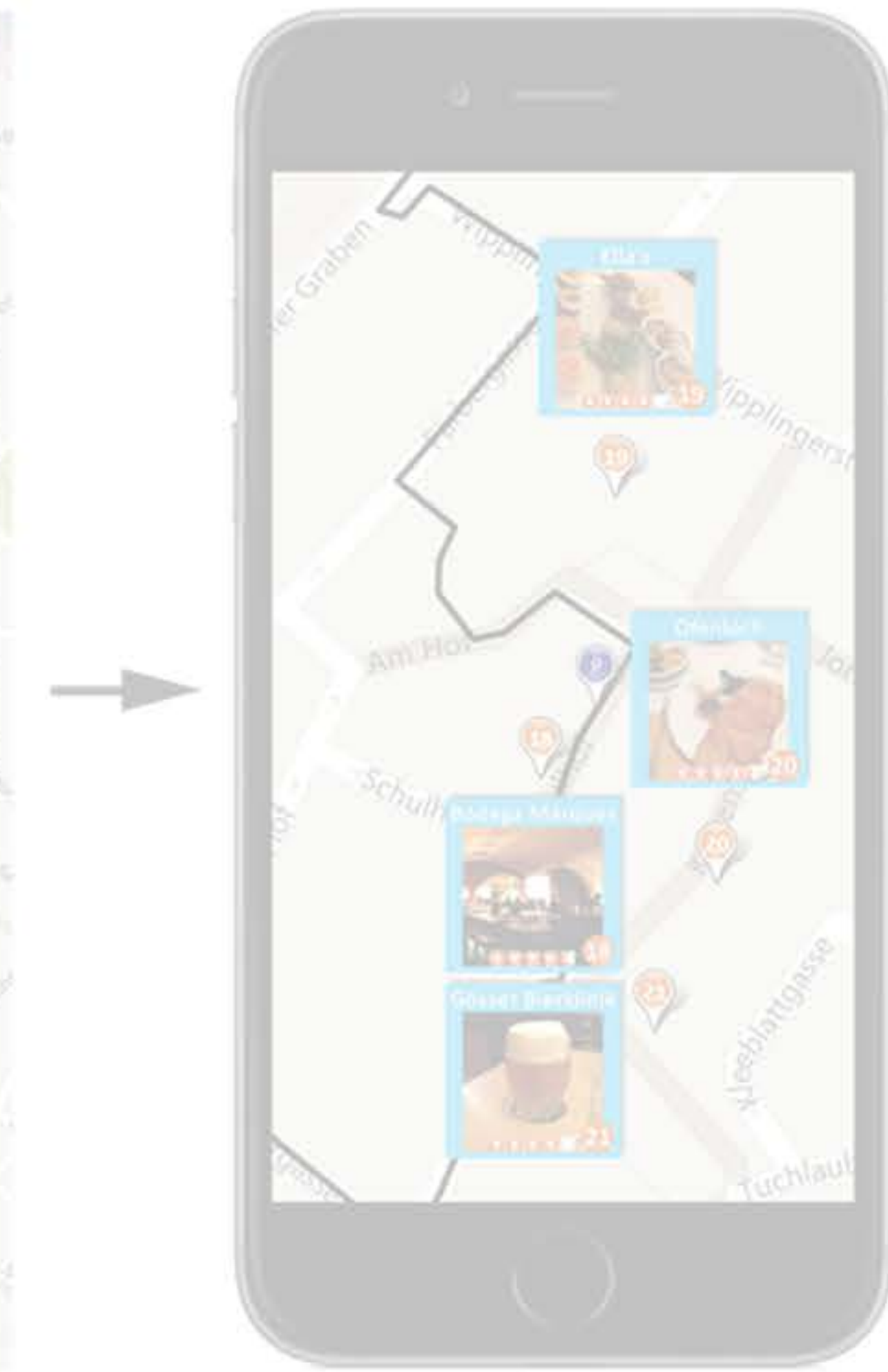
Path Computation



Map Cutout
Optimization



Detail Lens
Placement



Visualization

Map Cutout Optimization

- Motivation
 - Want to make better use of screen estate
- Approach
 - Computation of another path $\mathbf{m}(t)$ for map cutouts $M(t)$

Map Cutout Optimization

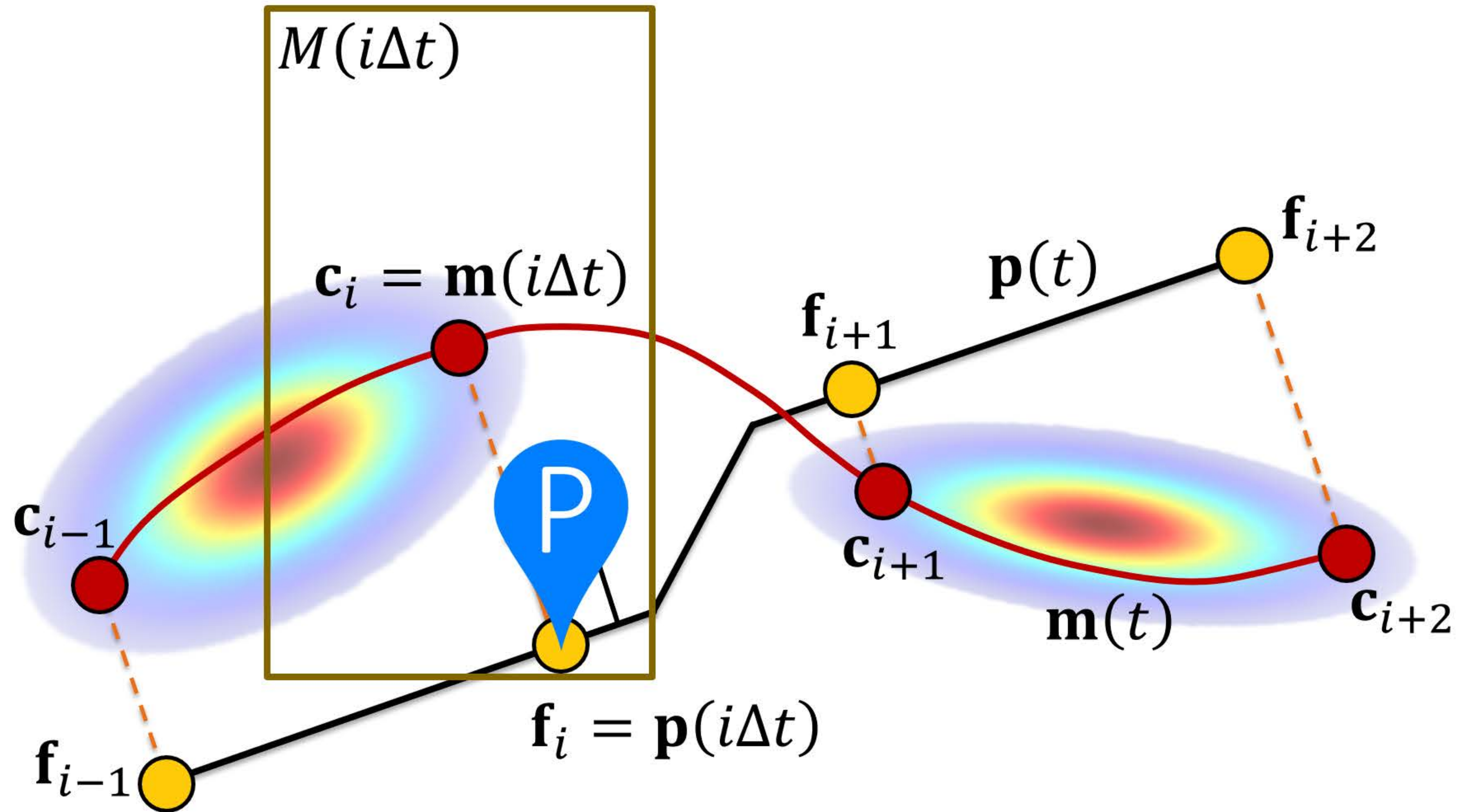
Requirements

1. It should be smooth
2. Should pass through *hot-spots*
3. Should follow the movement

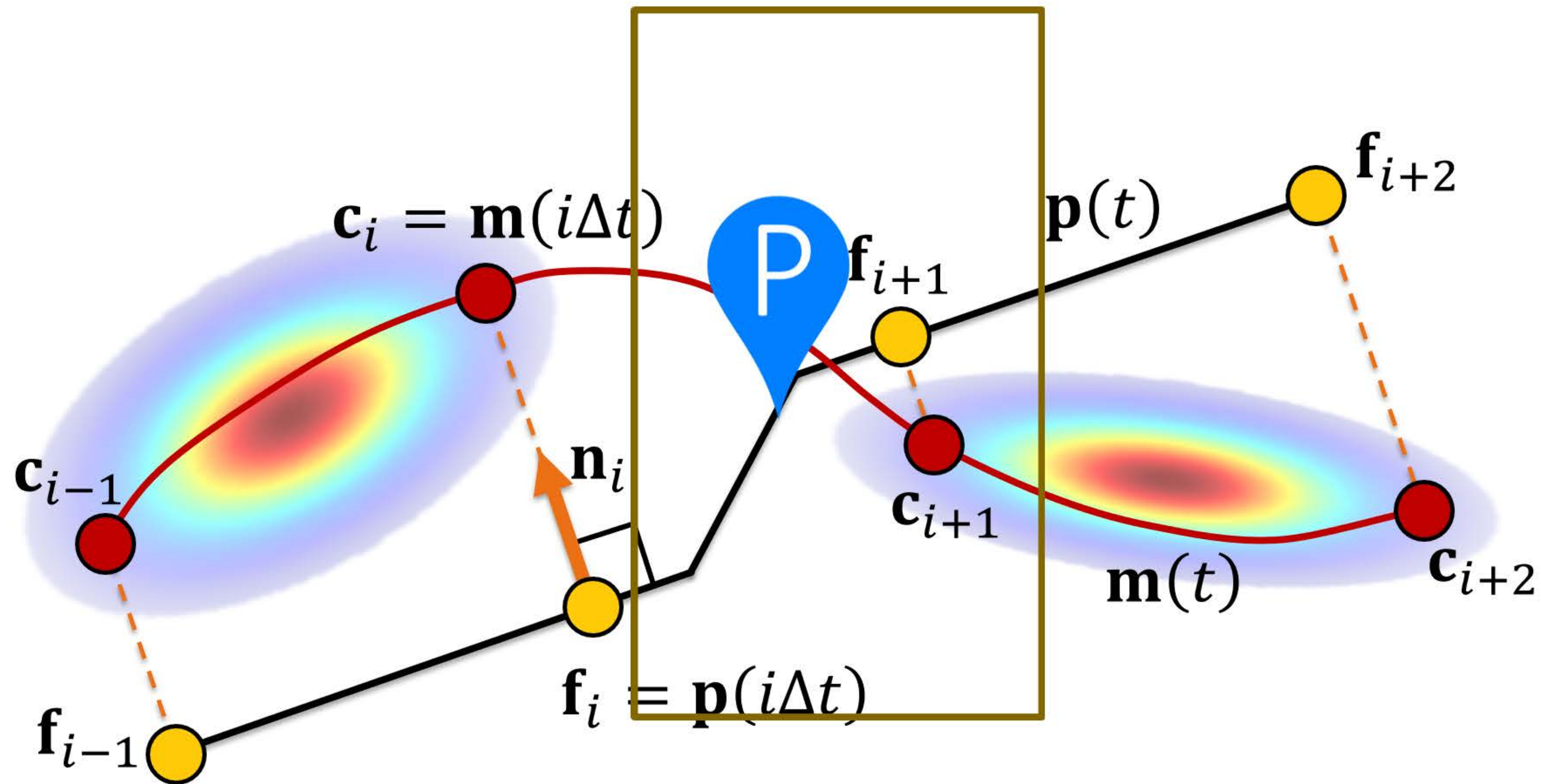
Solutions

1. Define $\mathbf{m}(t)$ as cubic spline
2. Position control points of $\mathbf{m}(t)$ in regions of high POI density
3. Choose suitable parameterization of $\mathbf{m}(t)$

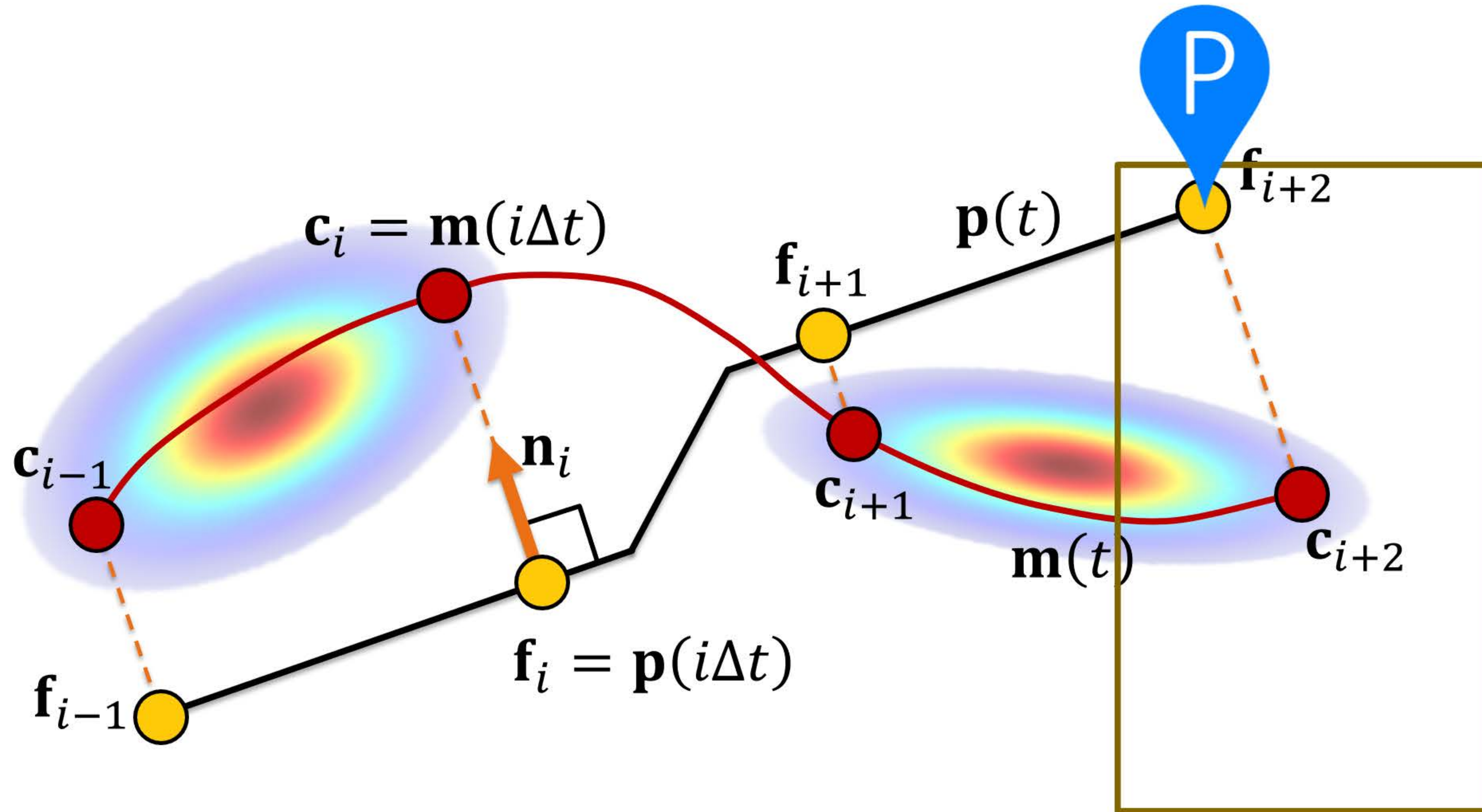
Map Cutout Optimization



Map Cutout Optimization

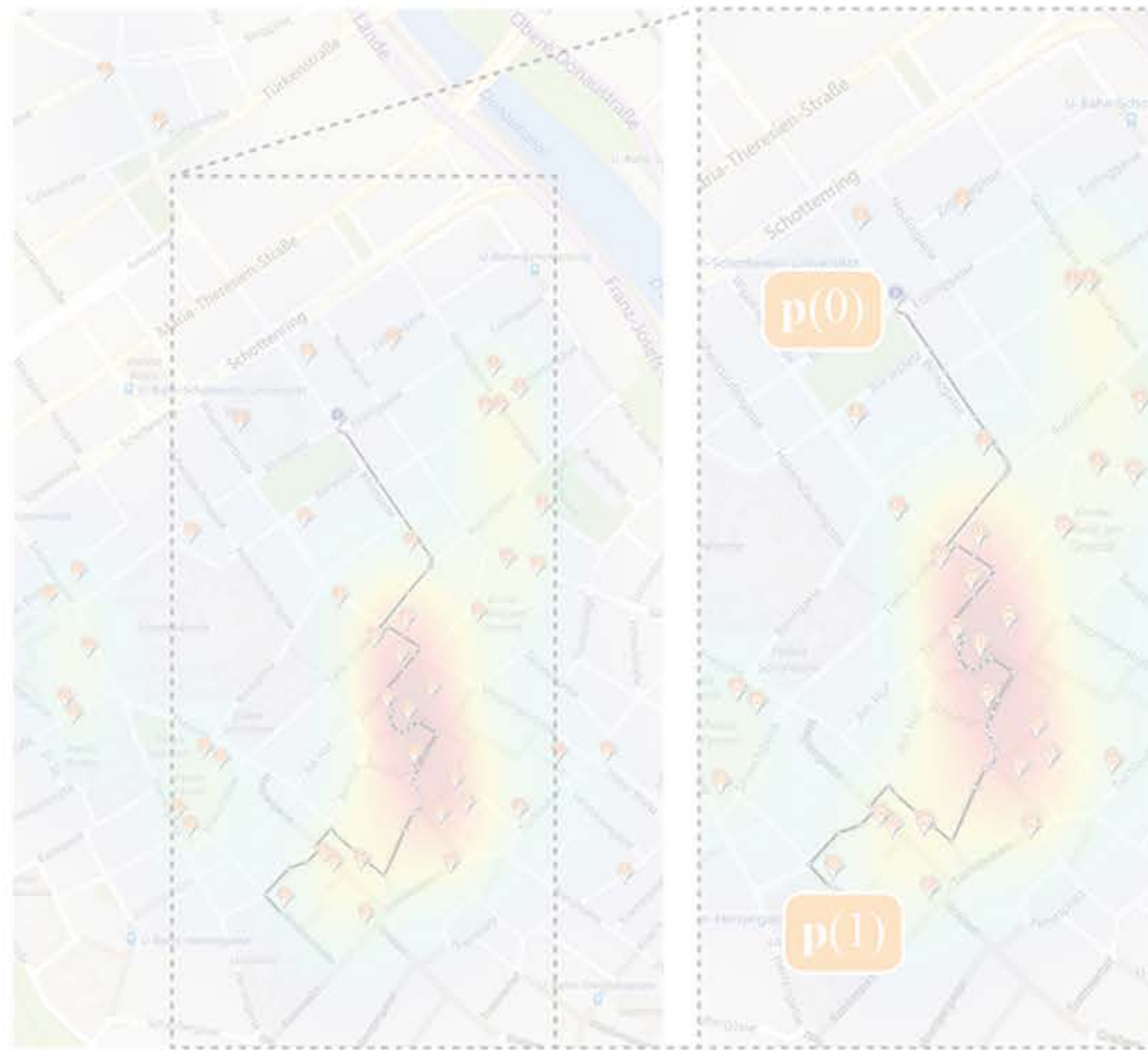


Map Cutout Optimization

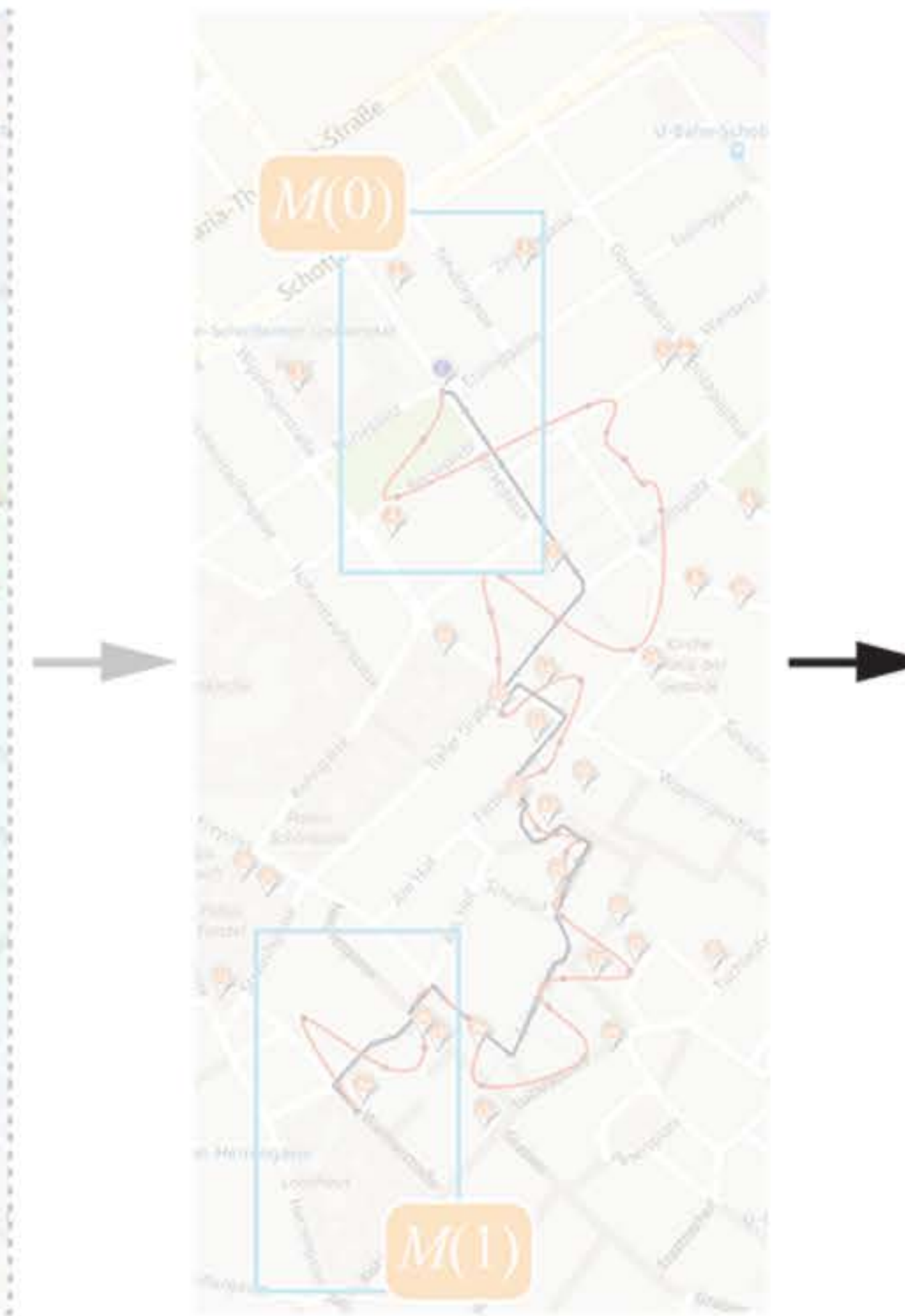


System Overview

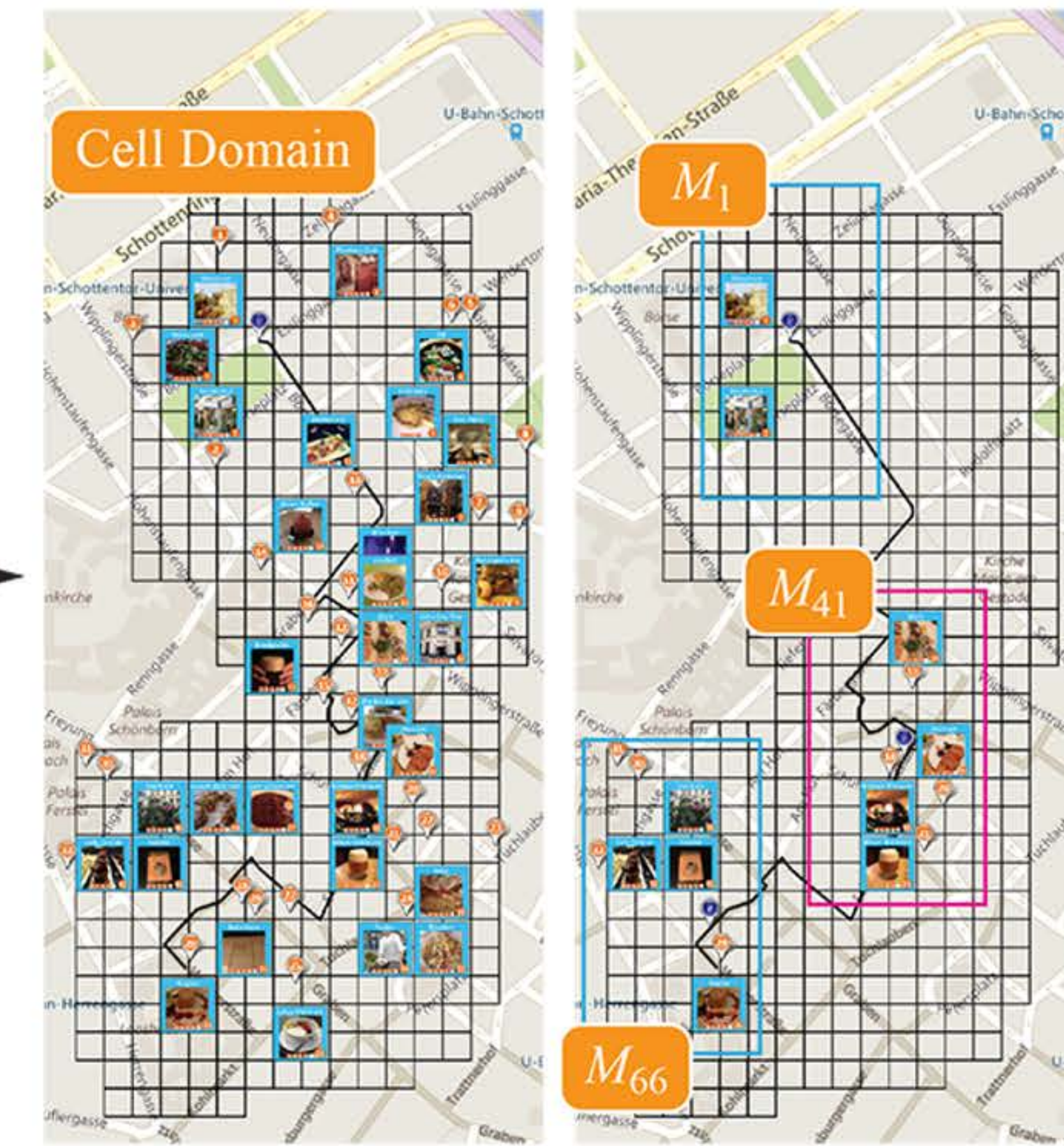
- Input: User position p , Points of Interest $P = \{P_1, P_2, \dots, P_n\}$



Path Computation



Map Cutout Optimization



Detail Lens Placement



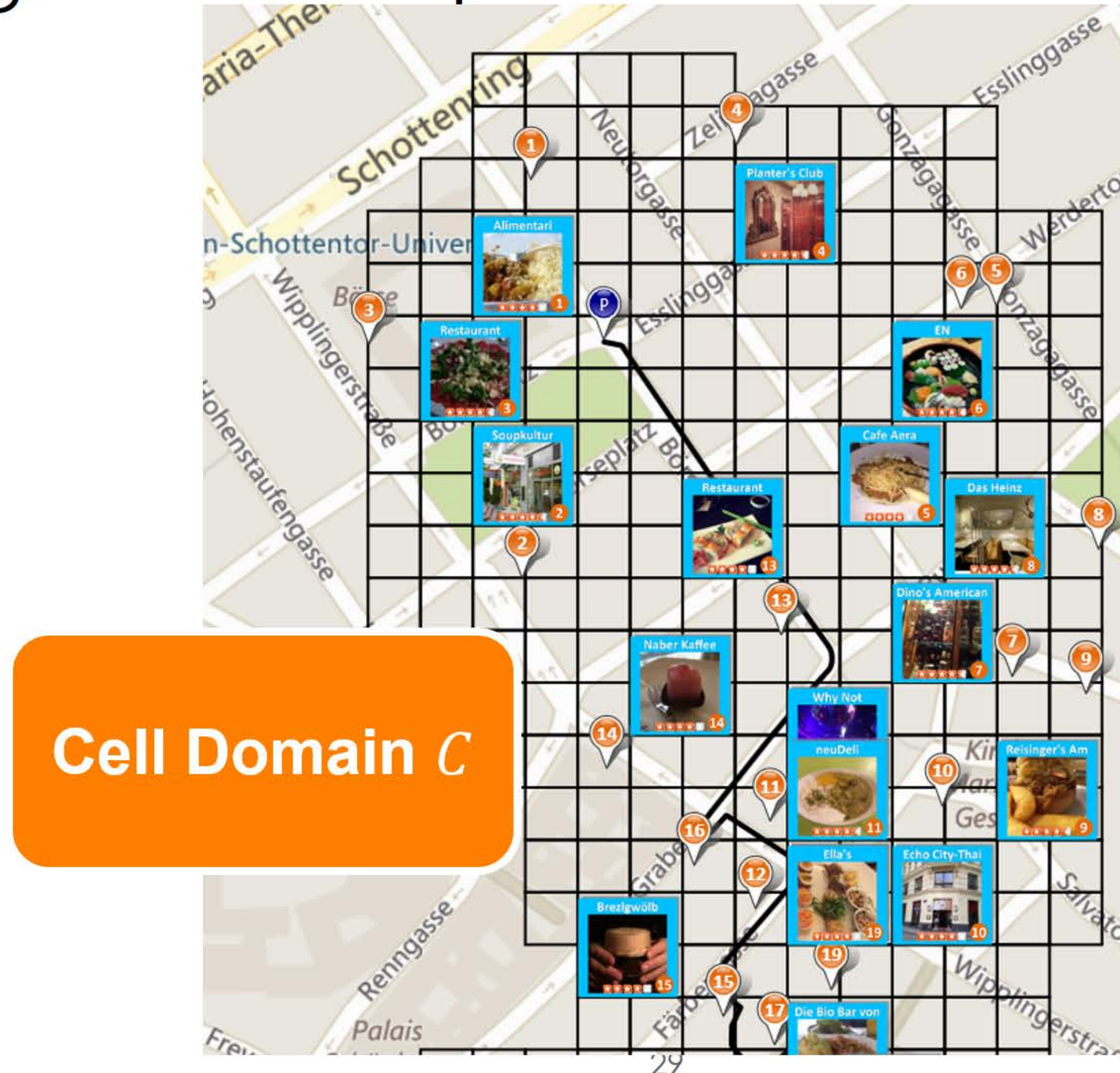
Visualization

Detail Lens Placement

- Motivation
 - Information about POIs is important
- Goal
 - Dynamic Layout following approved design guidelines

Detail Lens Placement

- Partitioning of the map into 2d Cartesian grid



Path Computation



Path

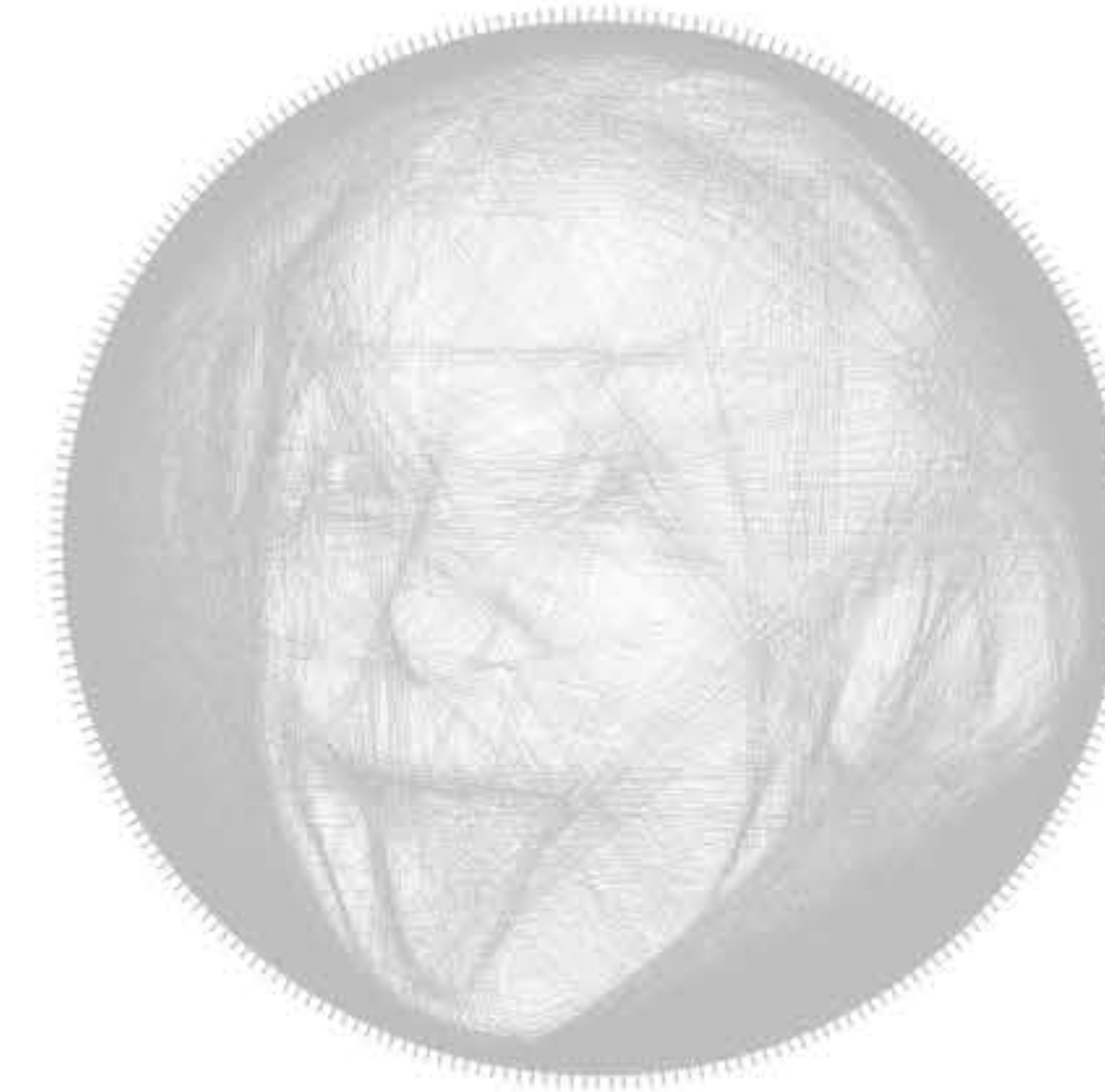
$$\max_{y_i} \sum_{i \in V_P} q(P_i) y_i$$

Layout

$$\max_{v_{\tau,i,x_c,y_c}} \sum_{\tau=1}^{\tau_{\max}} \sum_{i=1}^{n^*} \sum_{c \in C} q_{\tau,i,x_c,y_c} v_{\tau,i,x_c,y_c}$$



GUROBI
OPTIMIZATION



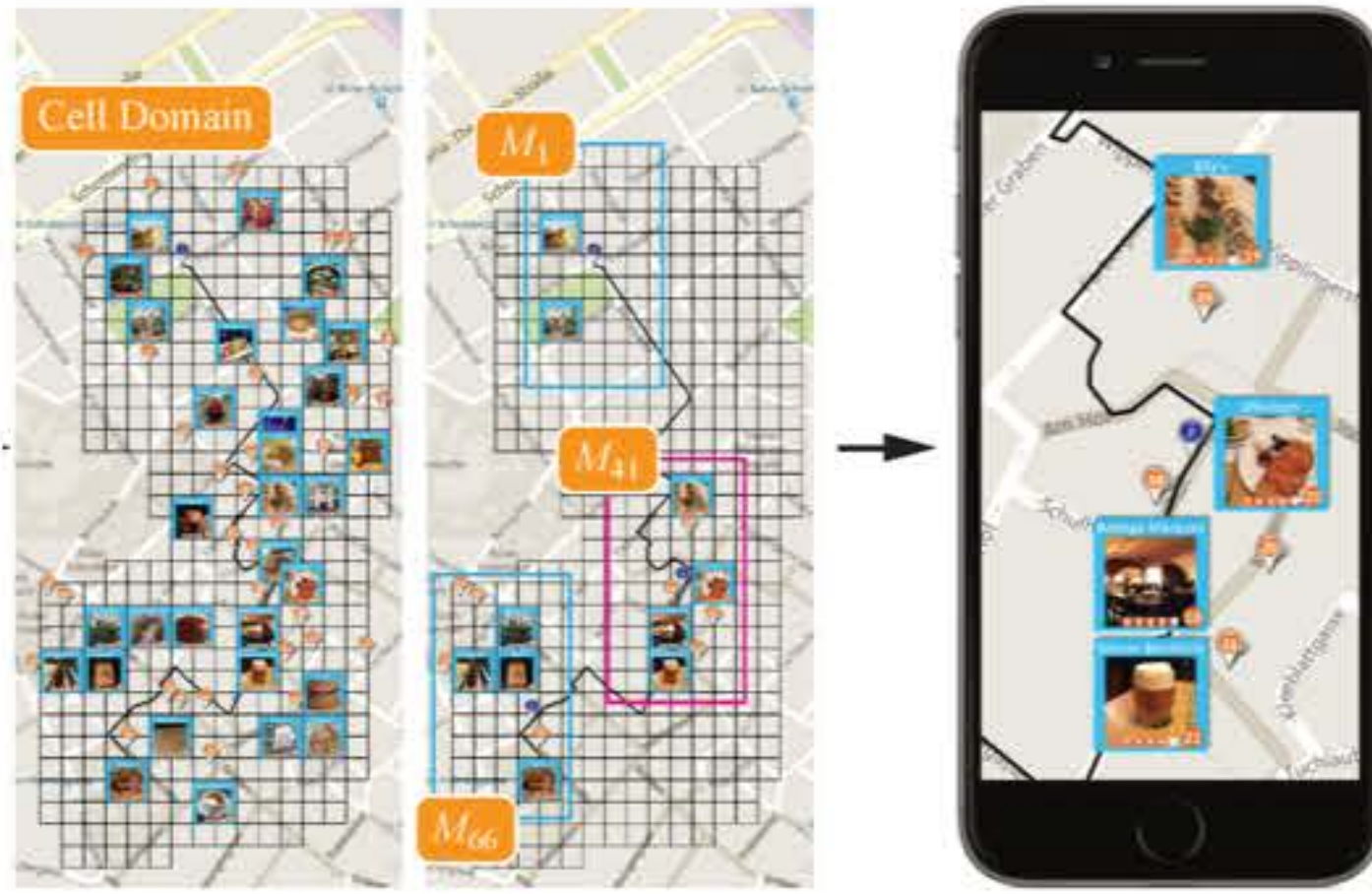
String Selection

$$\min_{x_j} \sum_{i=1}^m \delta_i$$

Not manageable
with Gurobi

➡ Own Greedy Solver

Path Computation



- $v_{\tau,i,x_c,y_c} = 1$ when lens L_i is shown at frame τ at position (x_c, y_c)
- q_{τ,i,x_c,y_c} is the corresponding quality value
- subject to several constraints

Layout

$$\max_{v_{\tau,i,x_c,y_c}} \sum_{\tau=1}^{\tau_{\max}} \sum_{i=1}^{n^*} \sum_{c \in C} q_{\tau,i,x_c,y_c} v_{\tau,i,x_c,y_c}$$



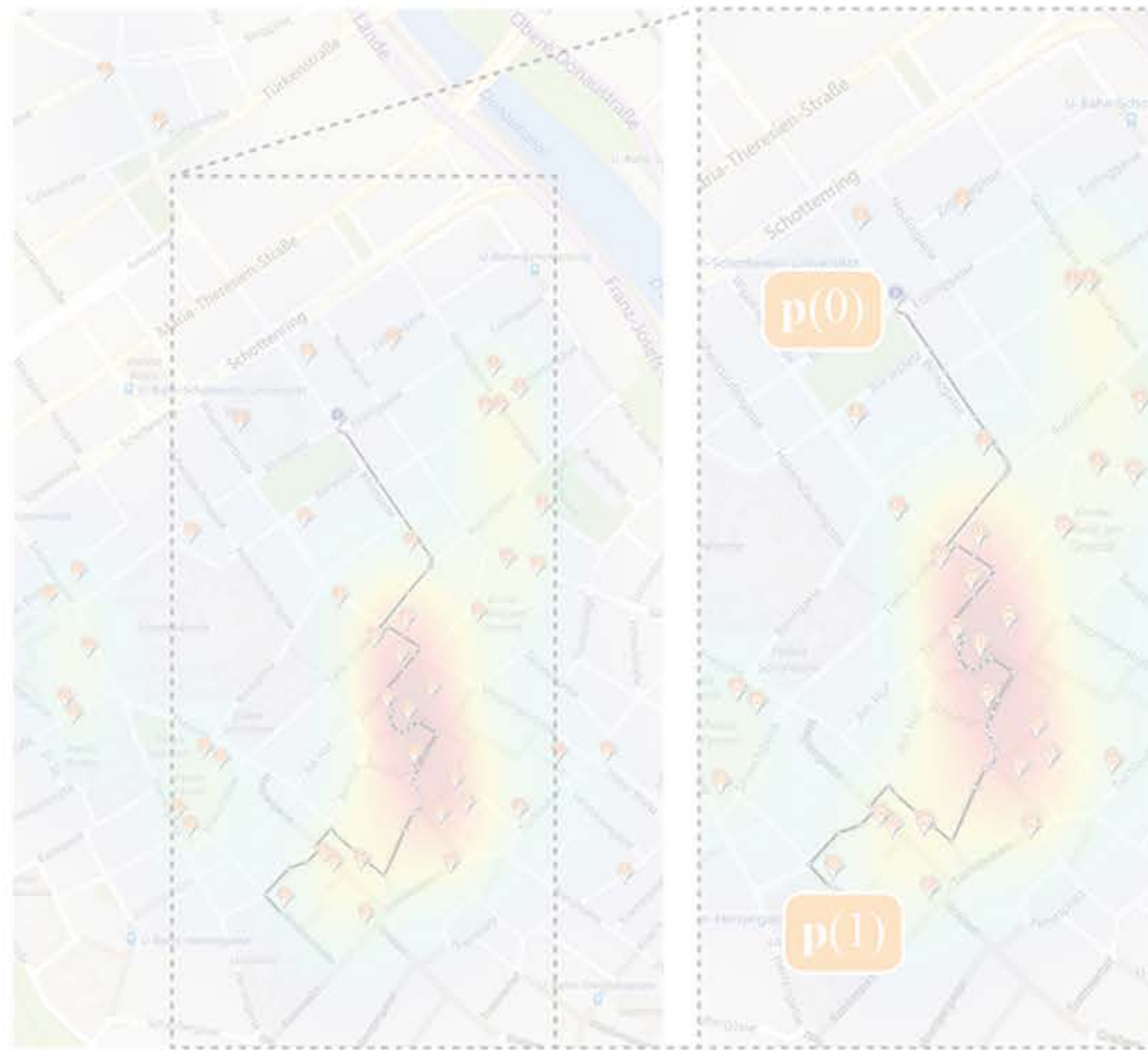
GUROBI
OPTIMIZATION

Detail Lens Placement

- **C1:** Each detail lens L_i is shown at most once at a time
- **C2:** No two detail lenses may overlap
- **C3:** Each detail lens is shown at just one position on the map
- **C4:** Each detail lens is shown in just one contiguous block of frames

System Overview

- Input: User position p , Points of Interest $P = \{P_1, P_2, \dots, P_n\}$



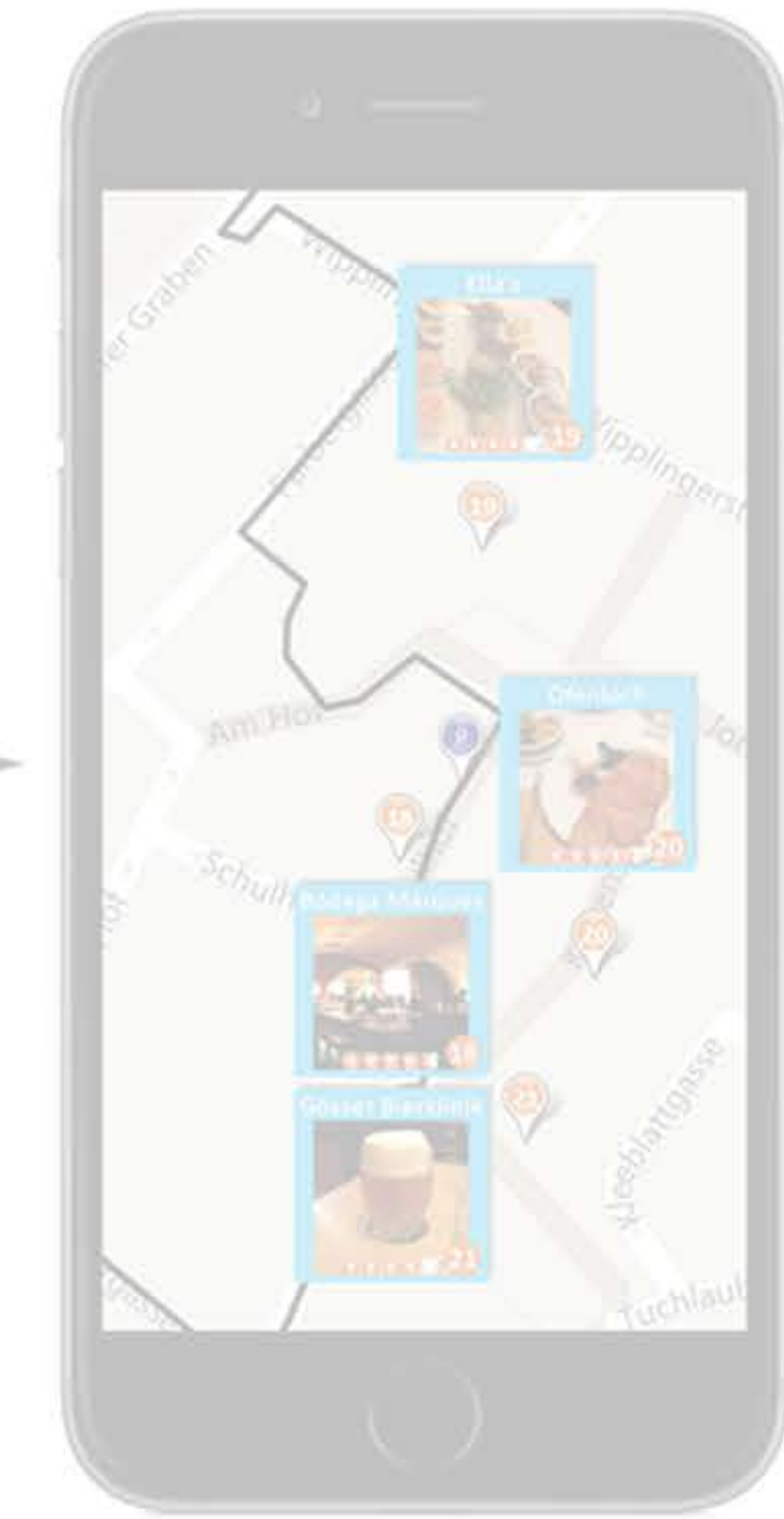
Path Computation



Map Cutout Optimization



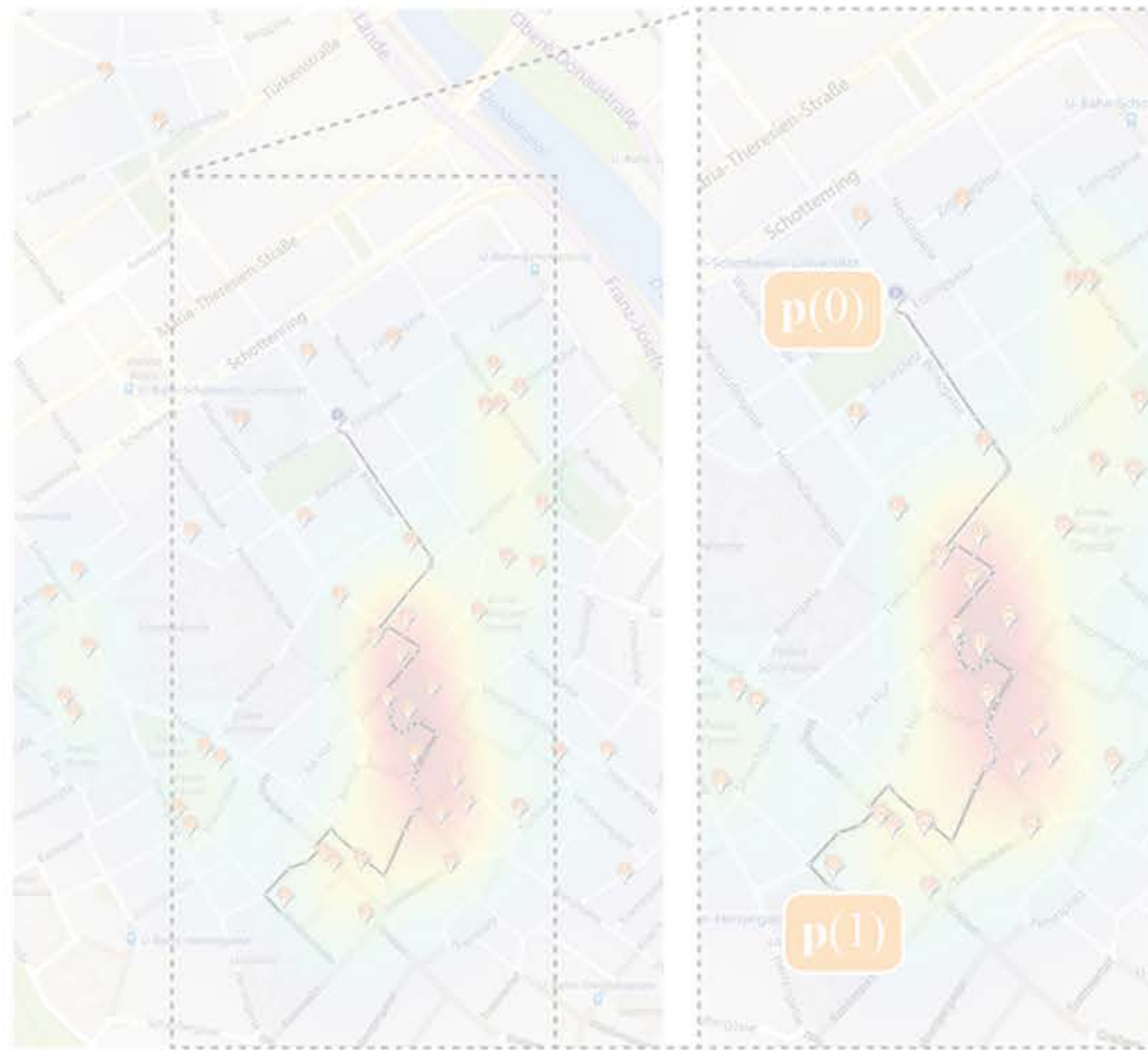
Detail Lens Placement



Visualization

System Overview

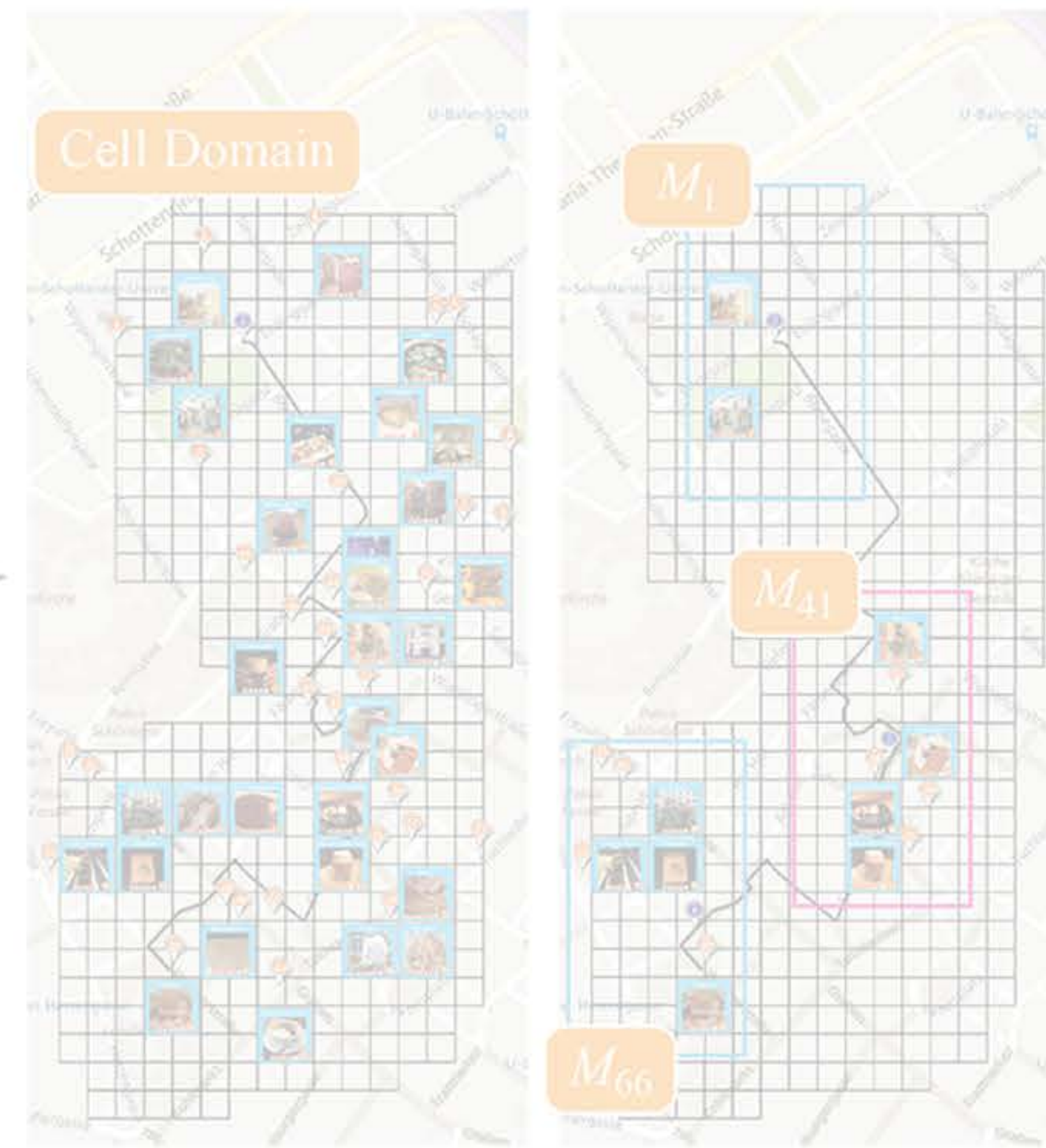
- Input: User position p , Points of Interest $P = \{P_1, P_2, \dots, P_n\}$



Path Computation



Map Cutout
Optimization

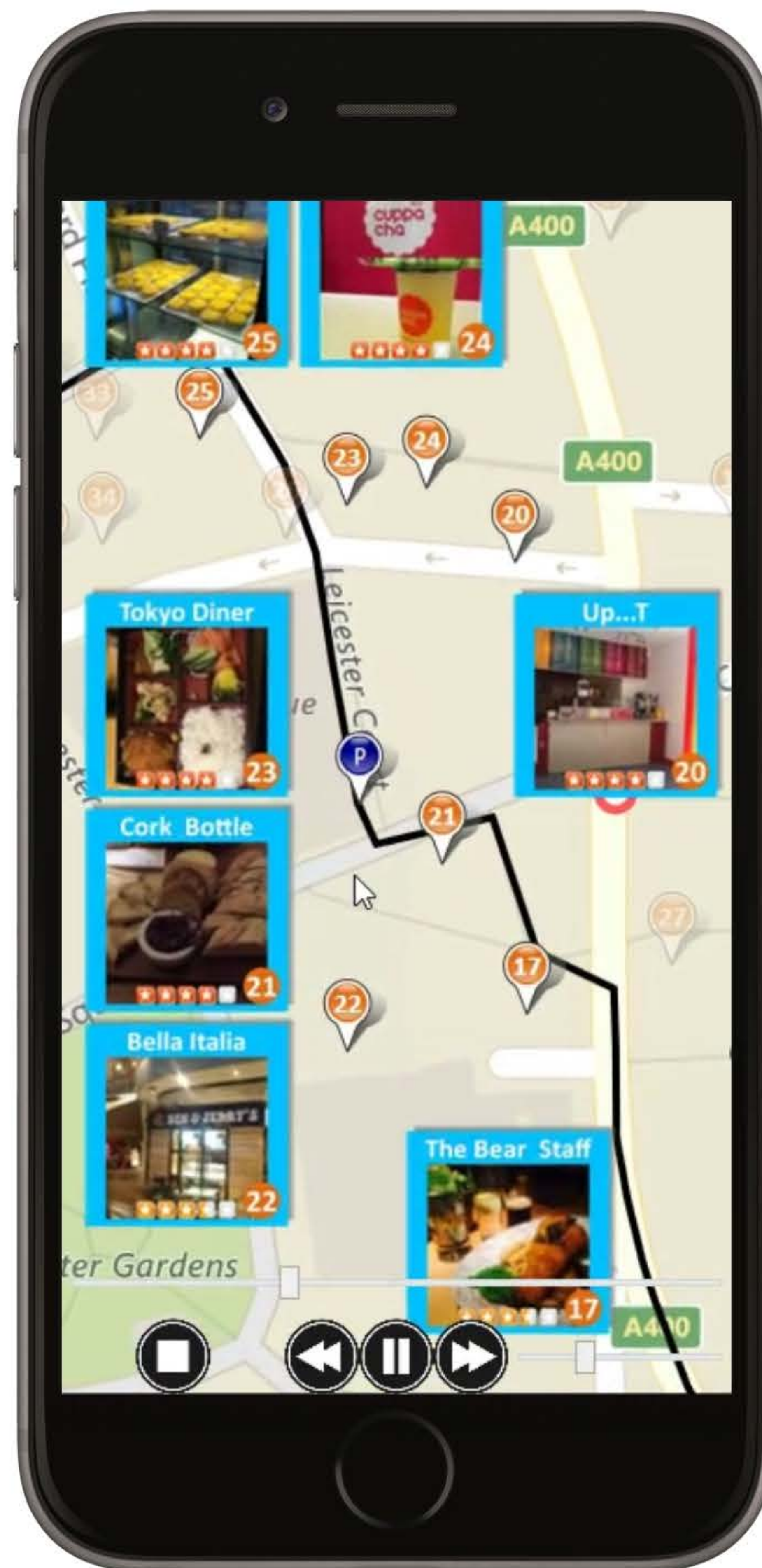


Detail Lens
Placement

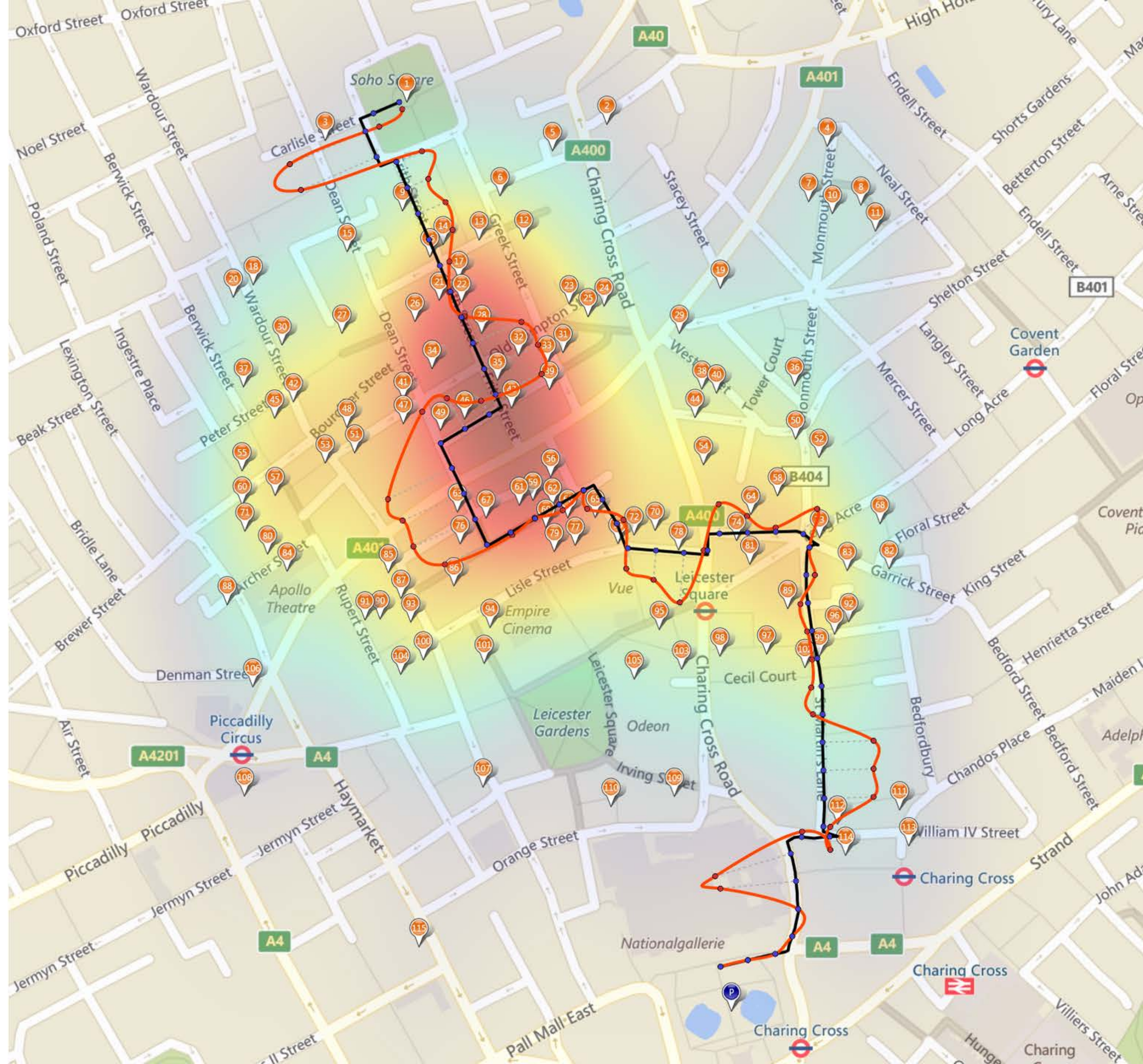


Visualization

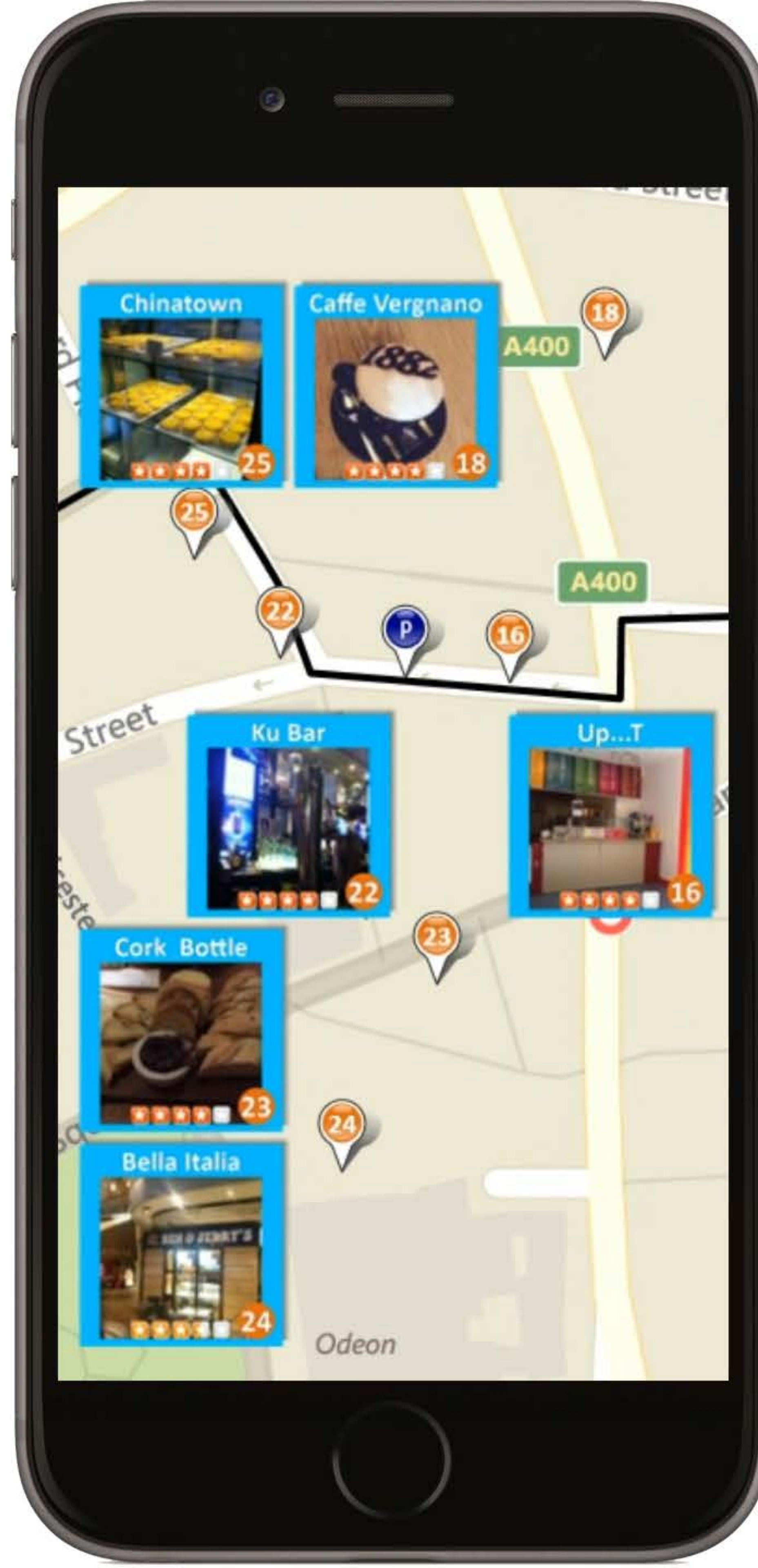
Visualization



London



London



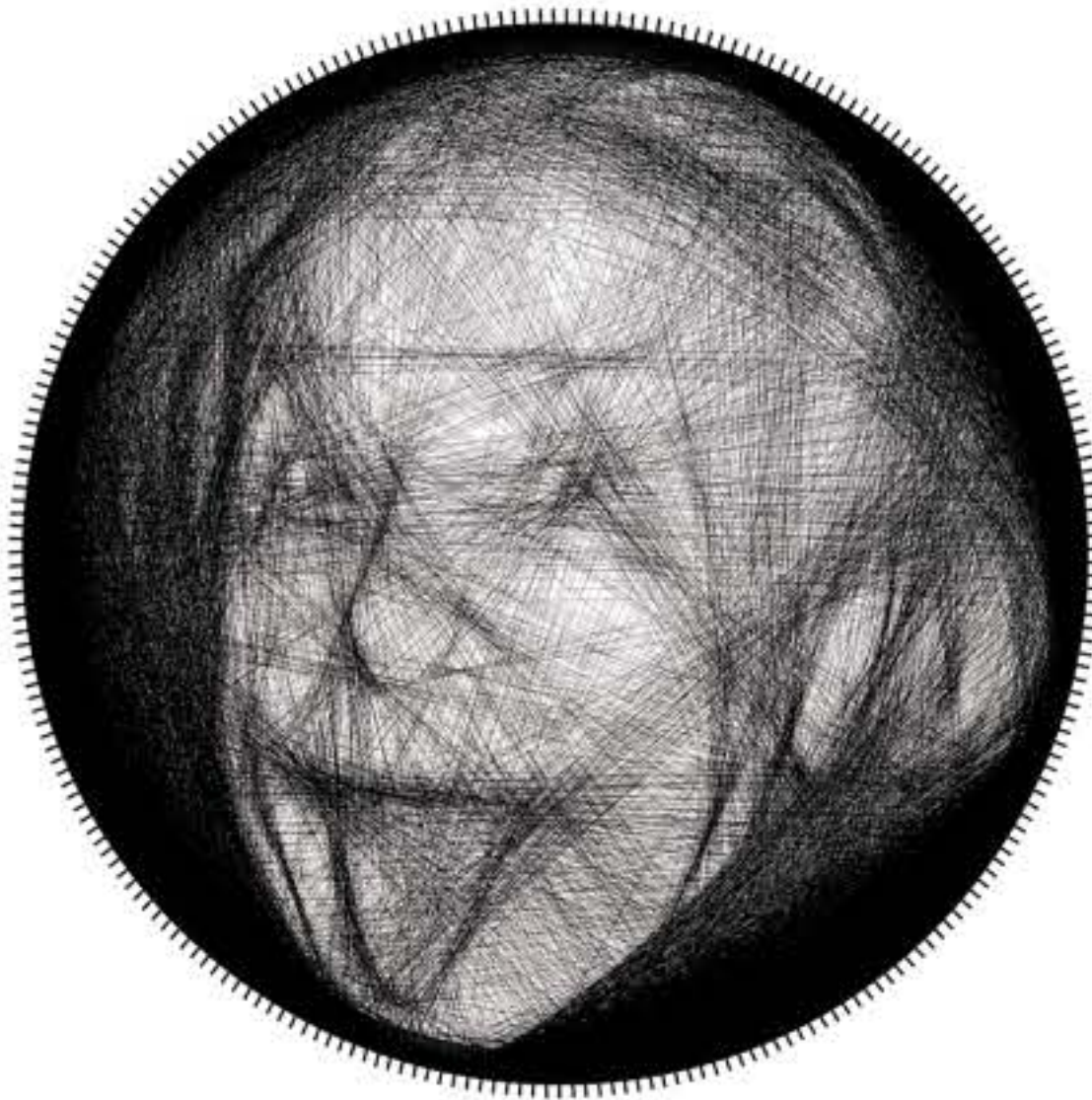
Overview



Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer

Dynamic Path Exploration on Mobile Devices

IEEE Transactions on Visualization and Computer Graphics. May 2018

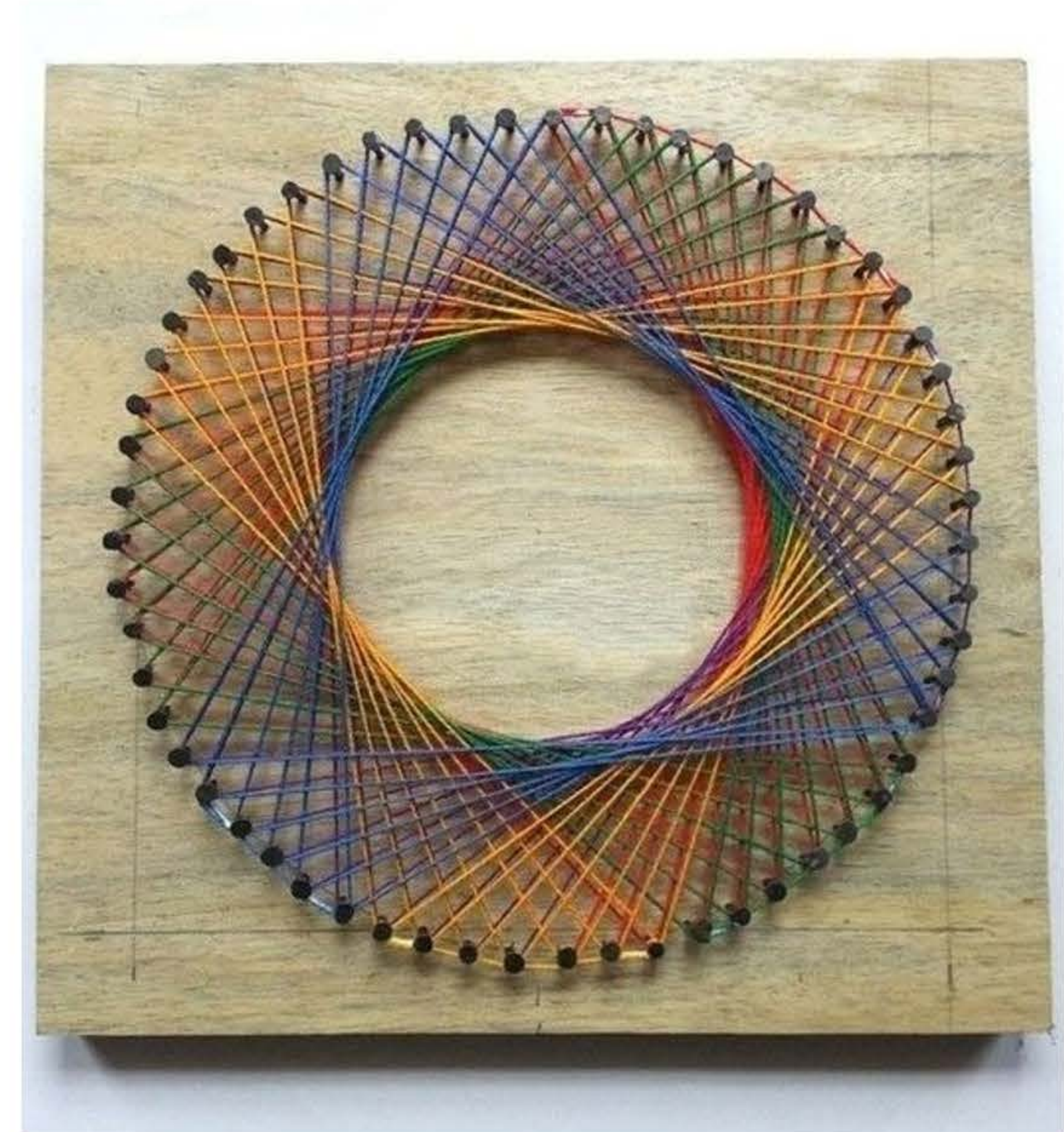
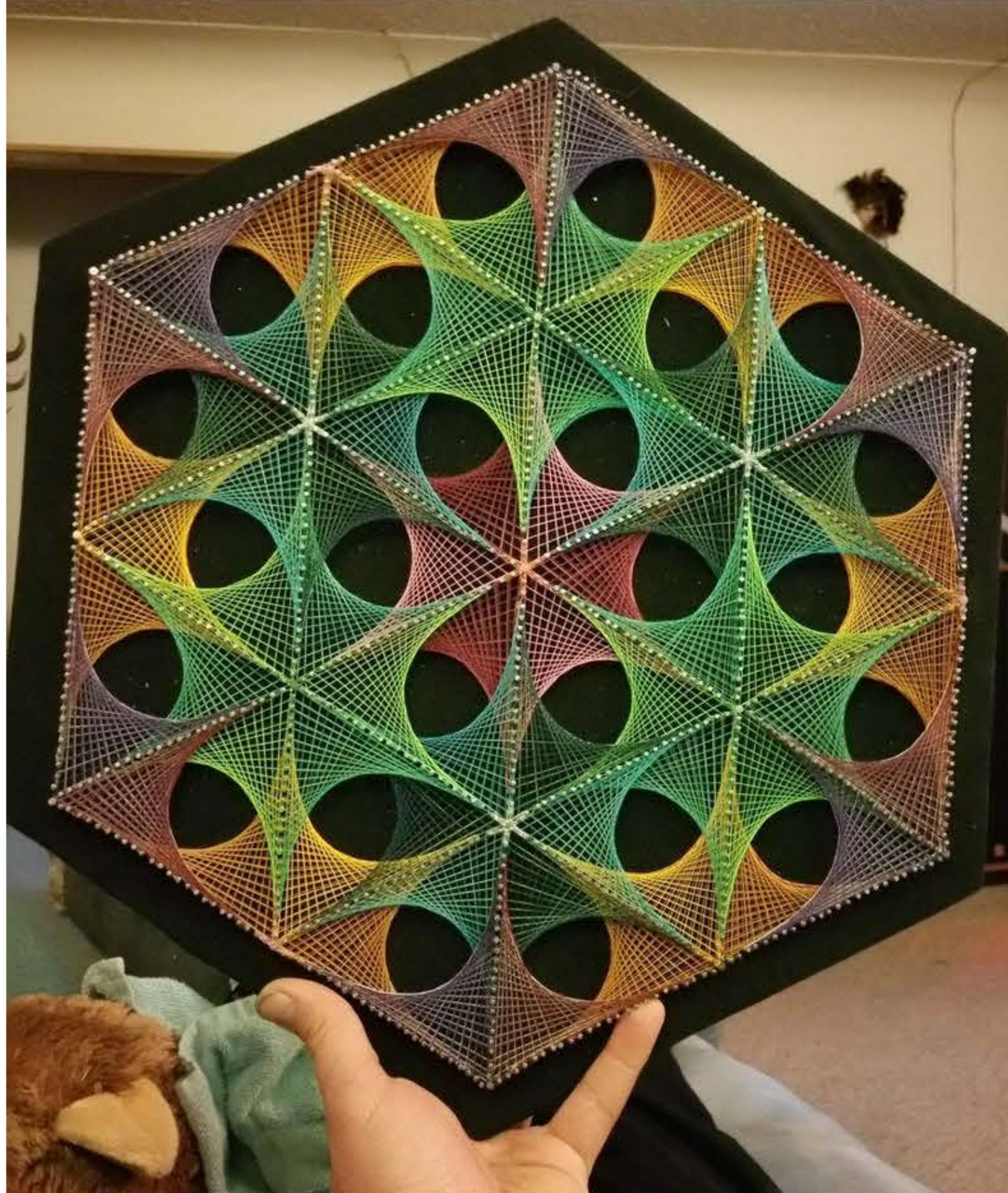


Michael Birsak, Florian Rist, Peter Wonka, and Przemyslaw Musialski

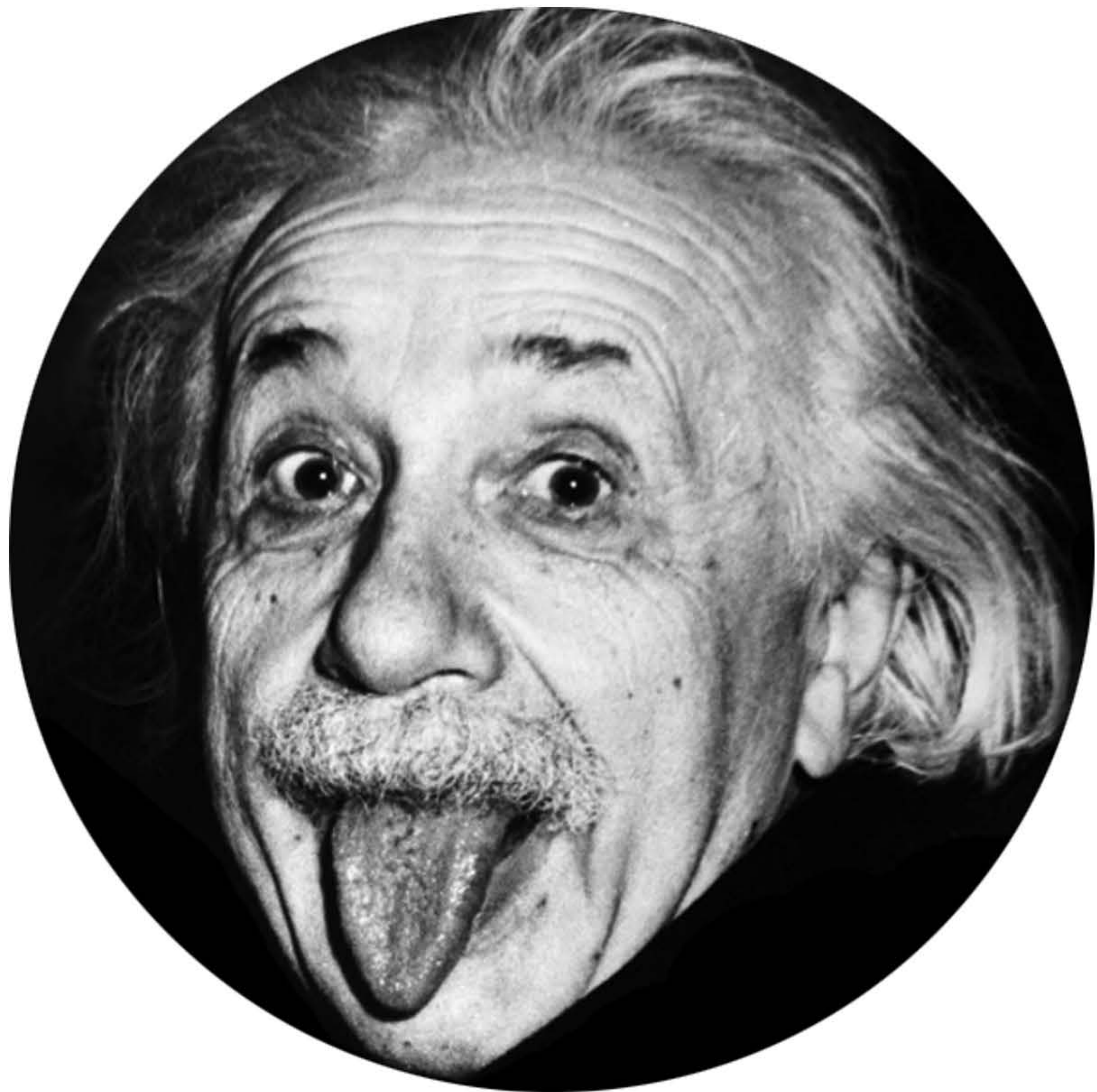
String Art: Towards Computational Fabrication of String Images

Computer Graphics Forum (Proc. EUROGRAPHICS 2018). April 2018

Introduction

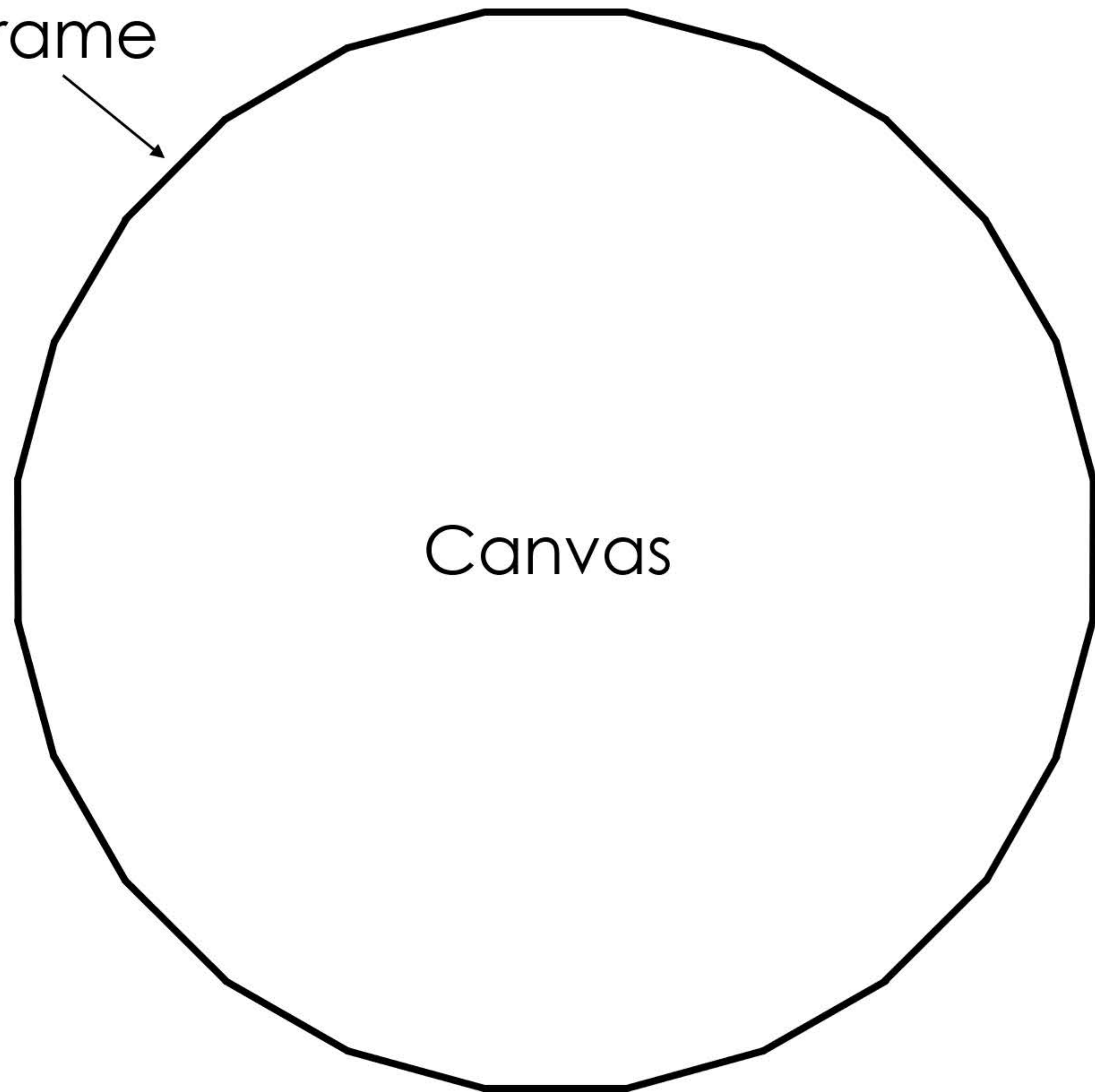


<https://babbledabbledo.com/math-art-idea-explore-geometry-string-art/>

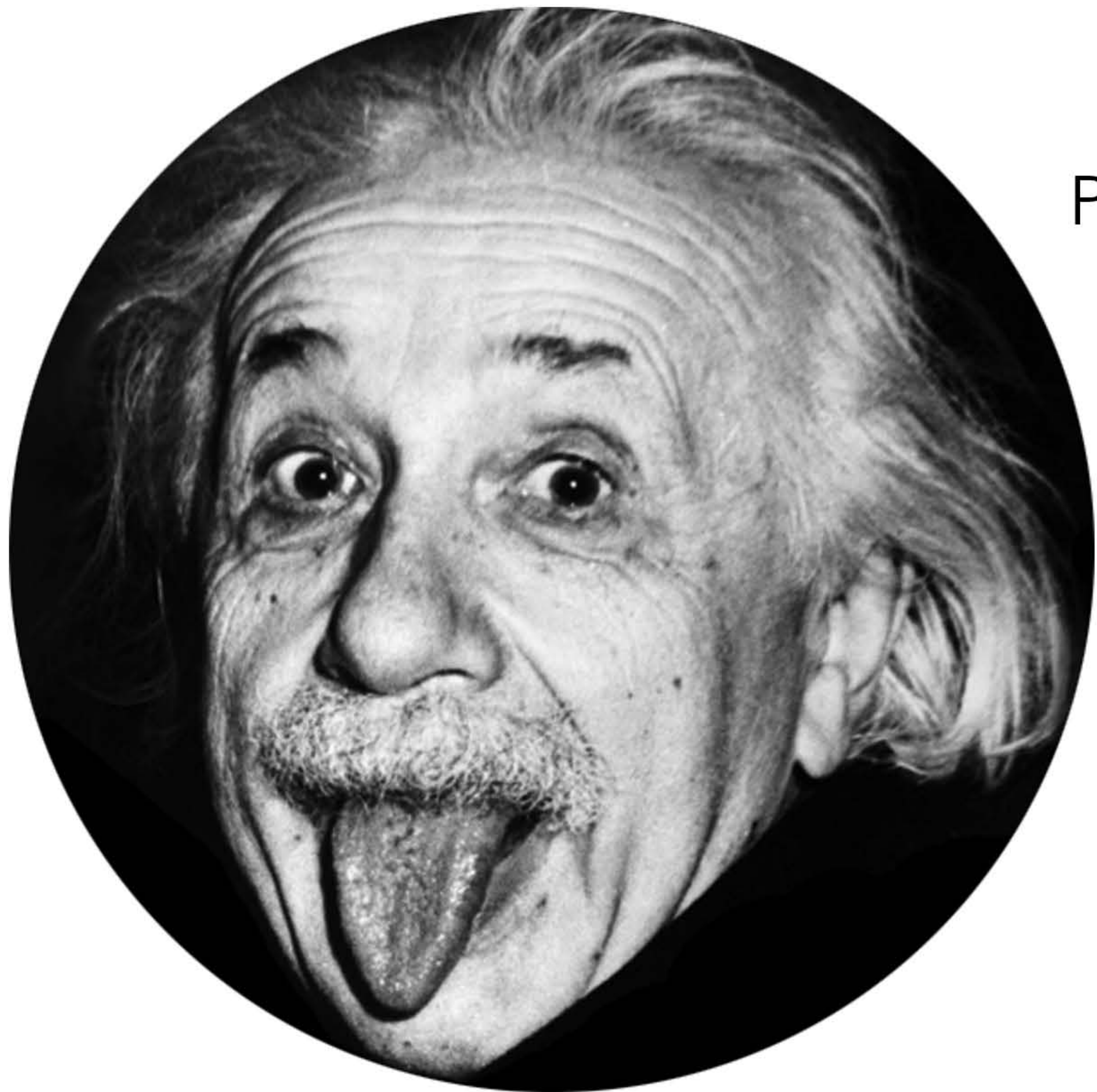


Target

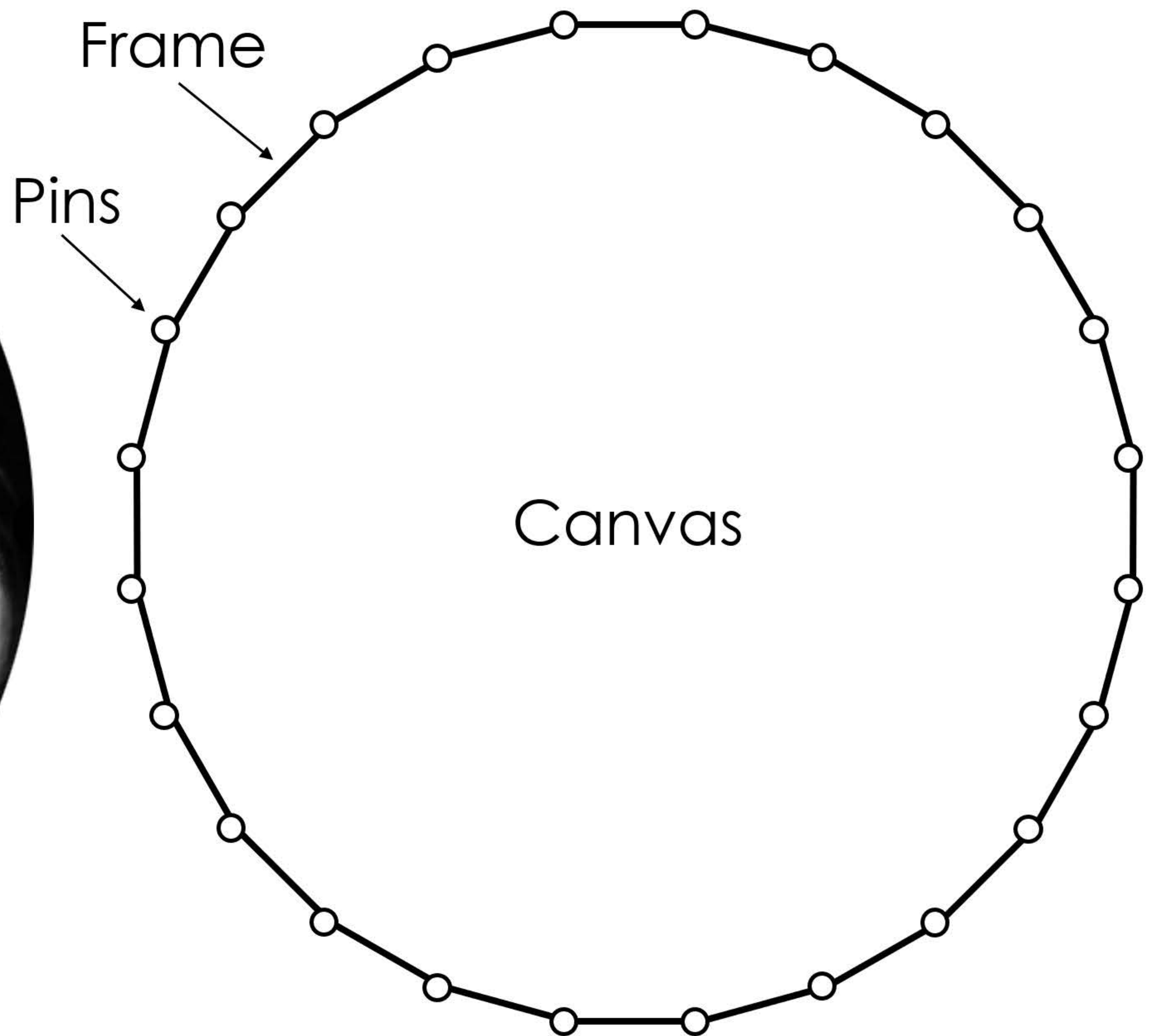
Frame

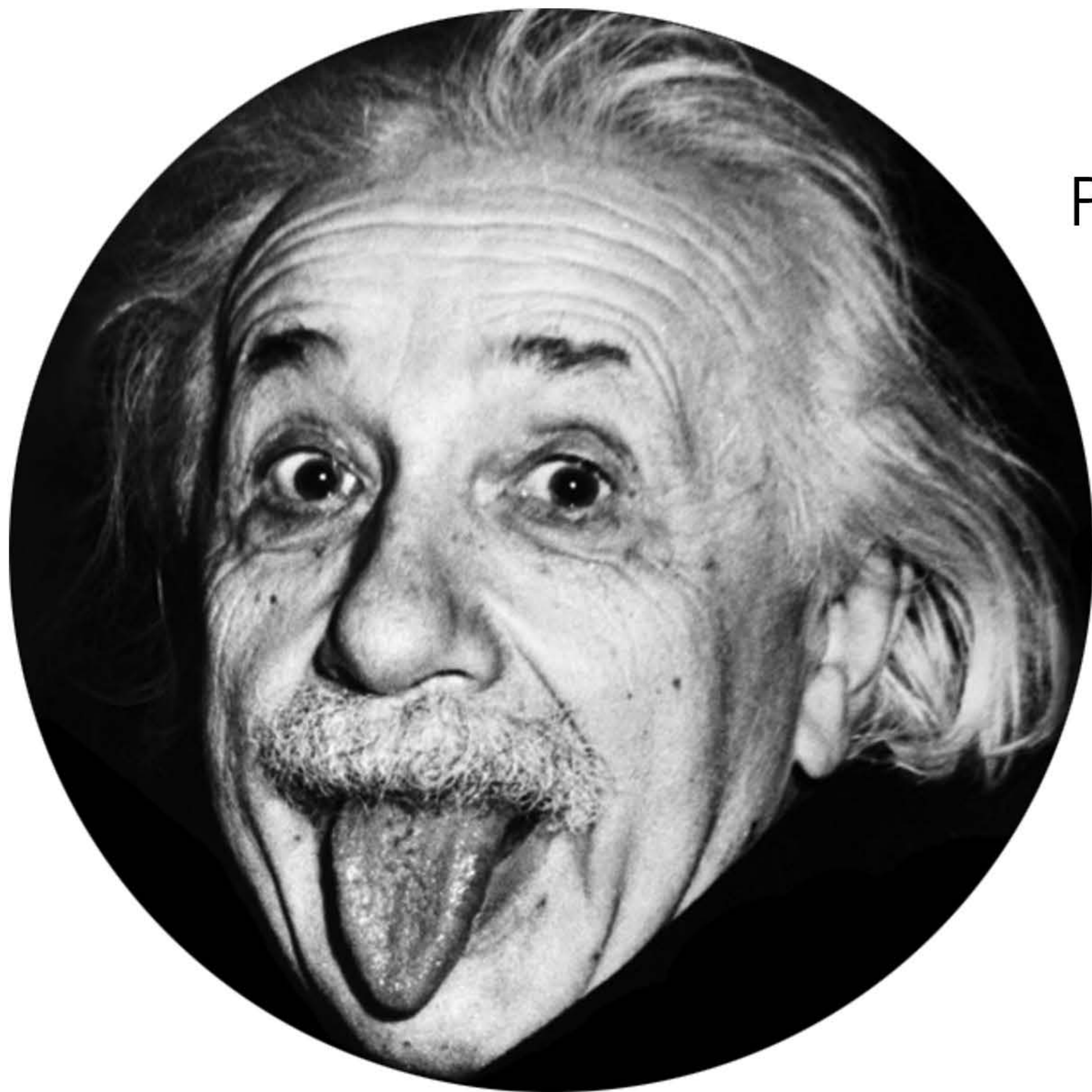


Canvas

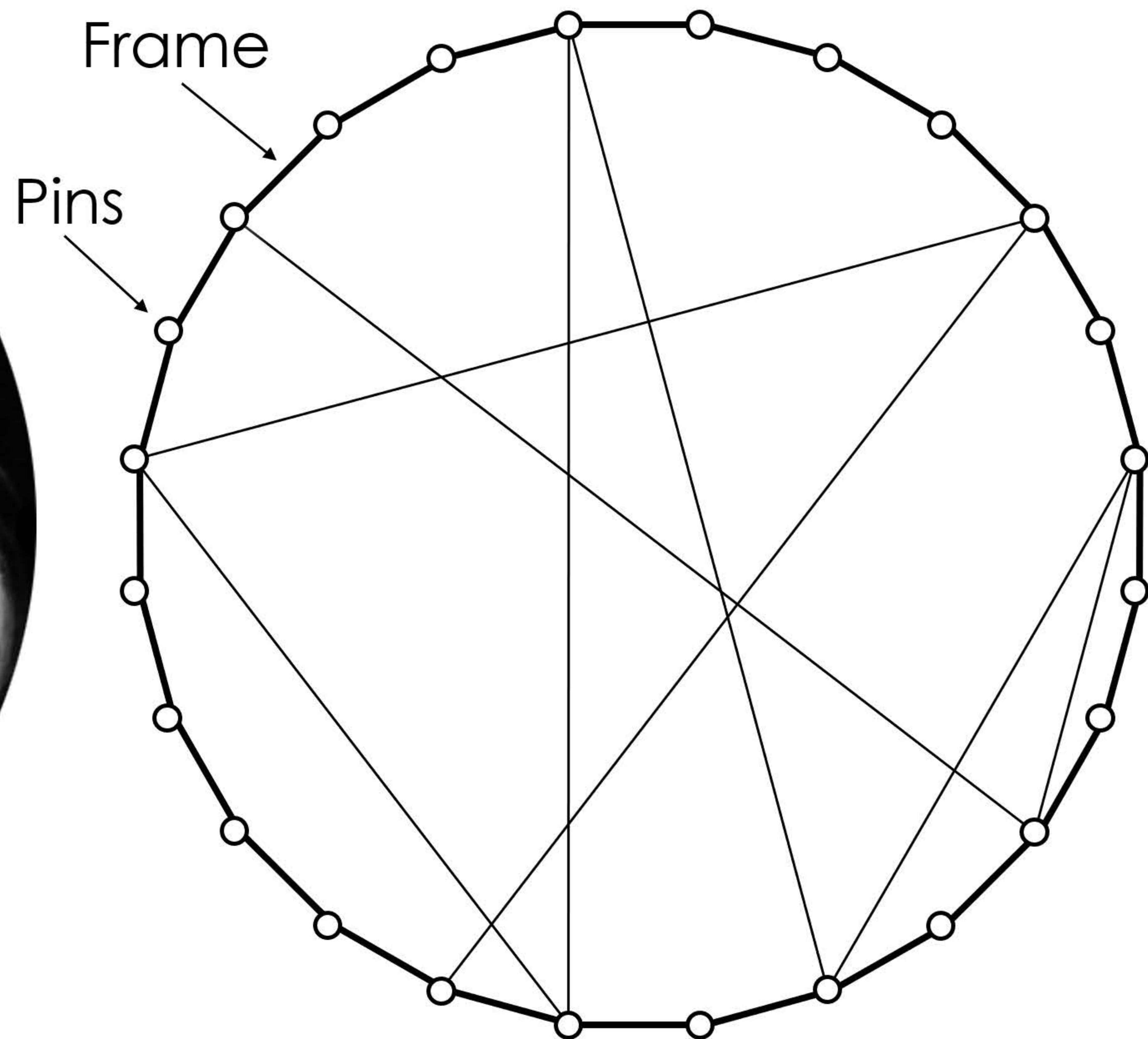


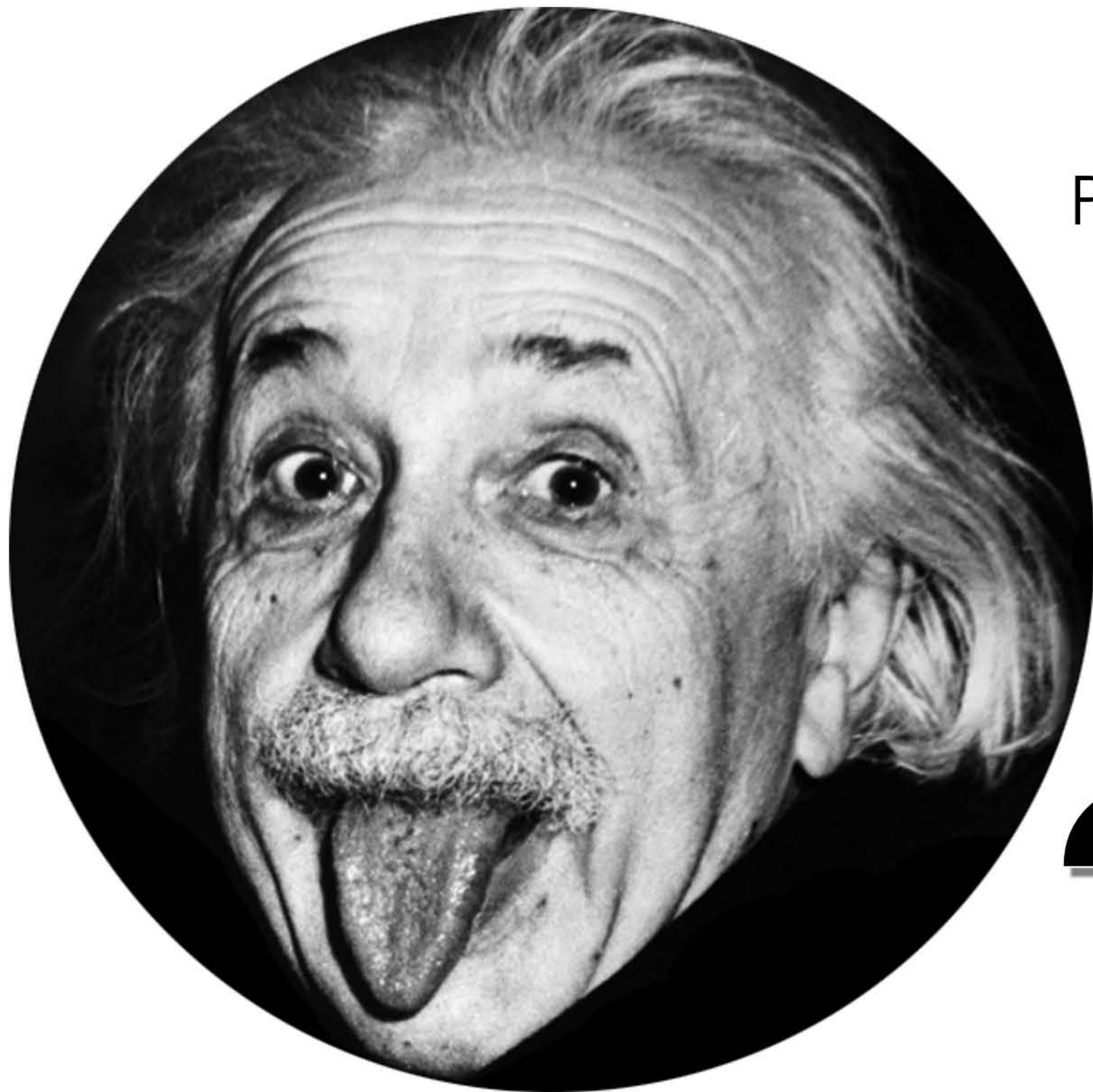
Target



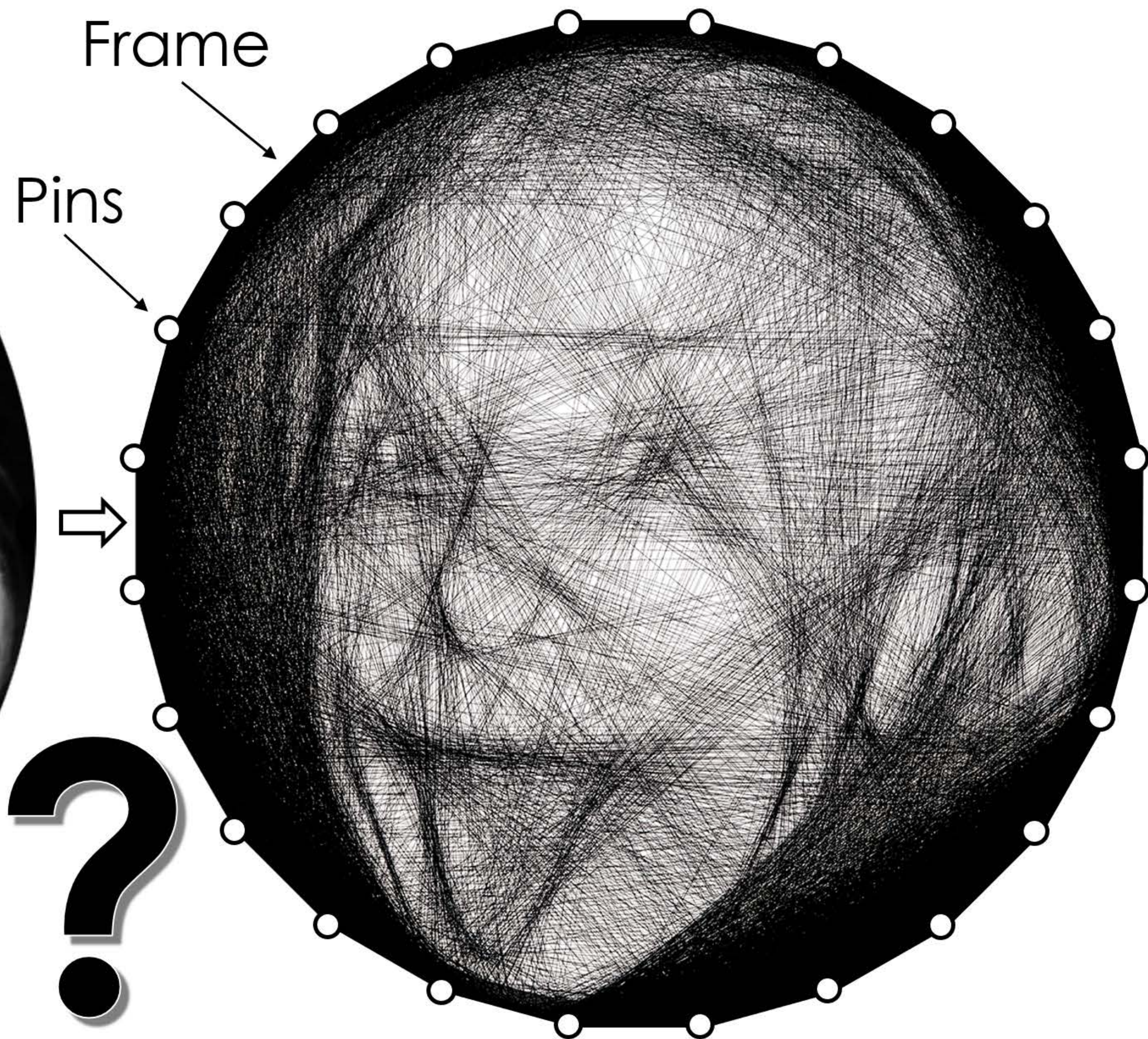


Target

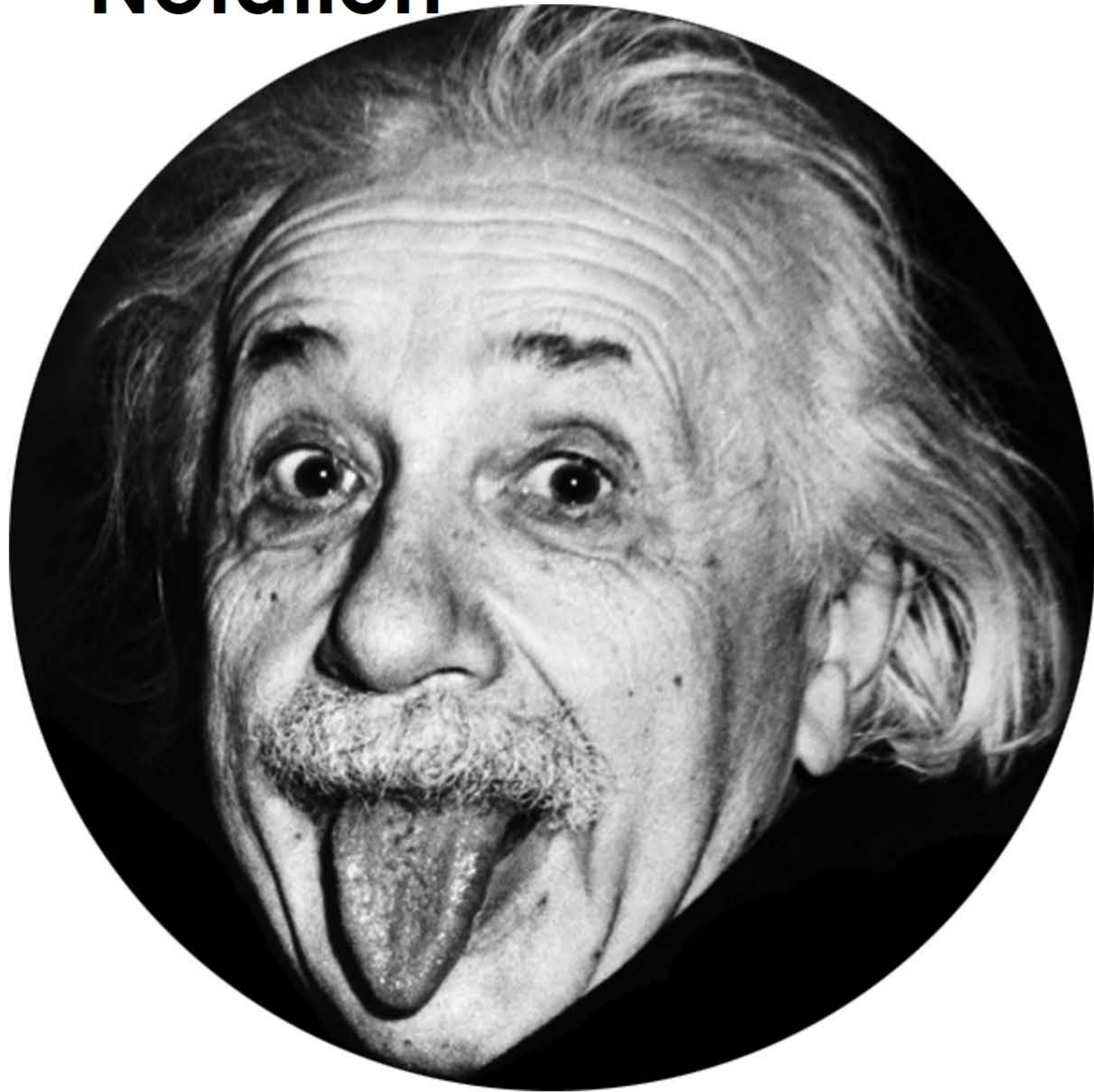




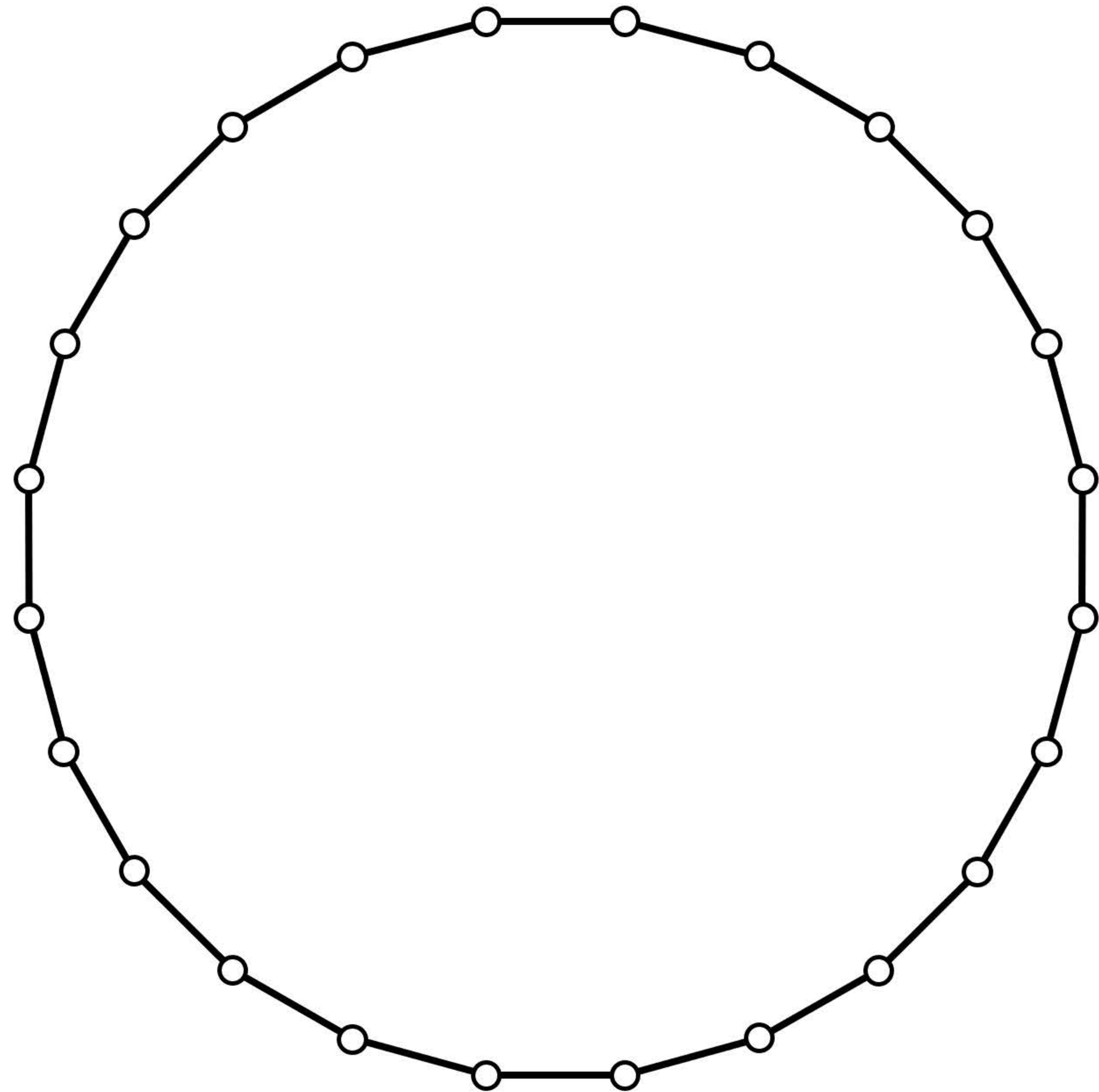
Target



Notation

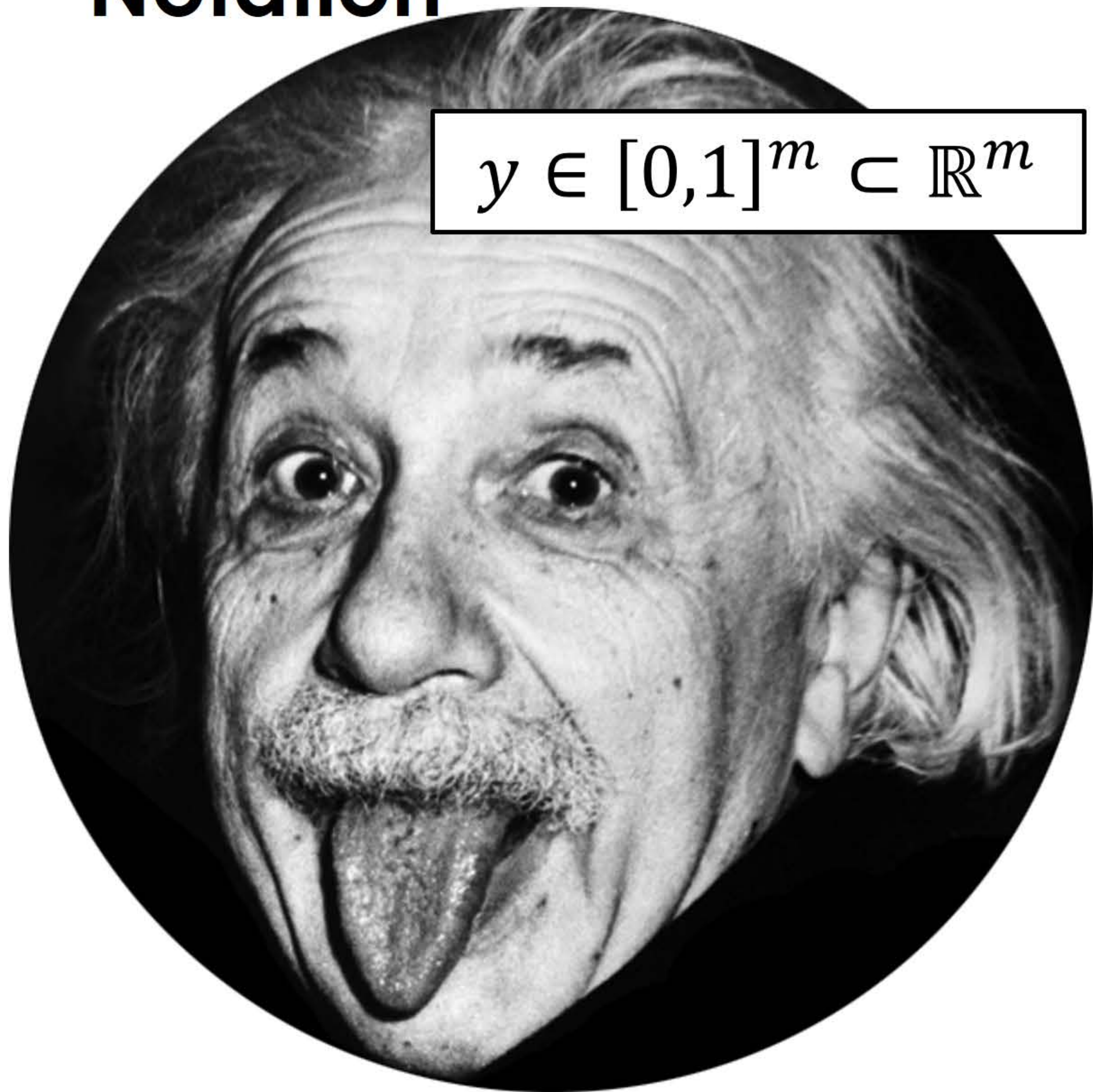


Target

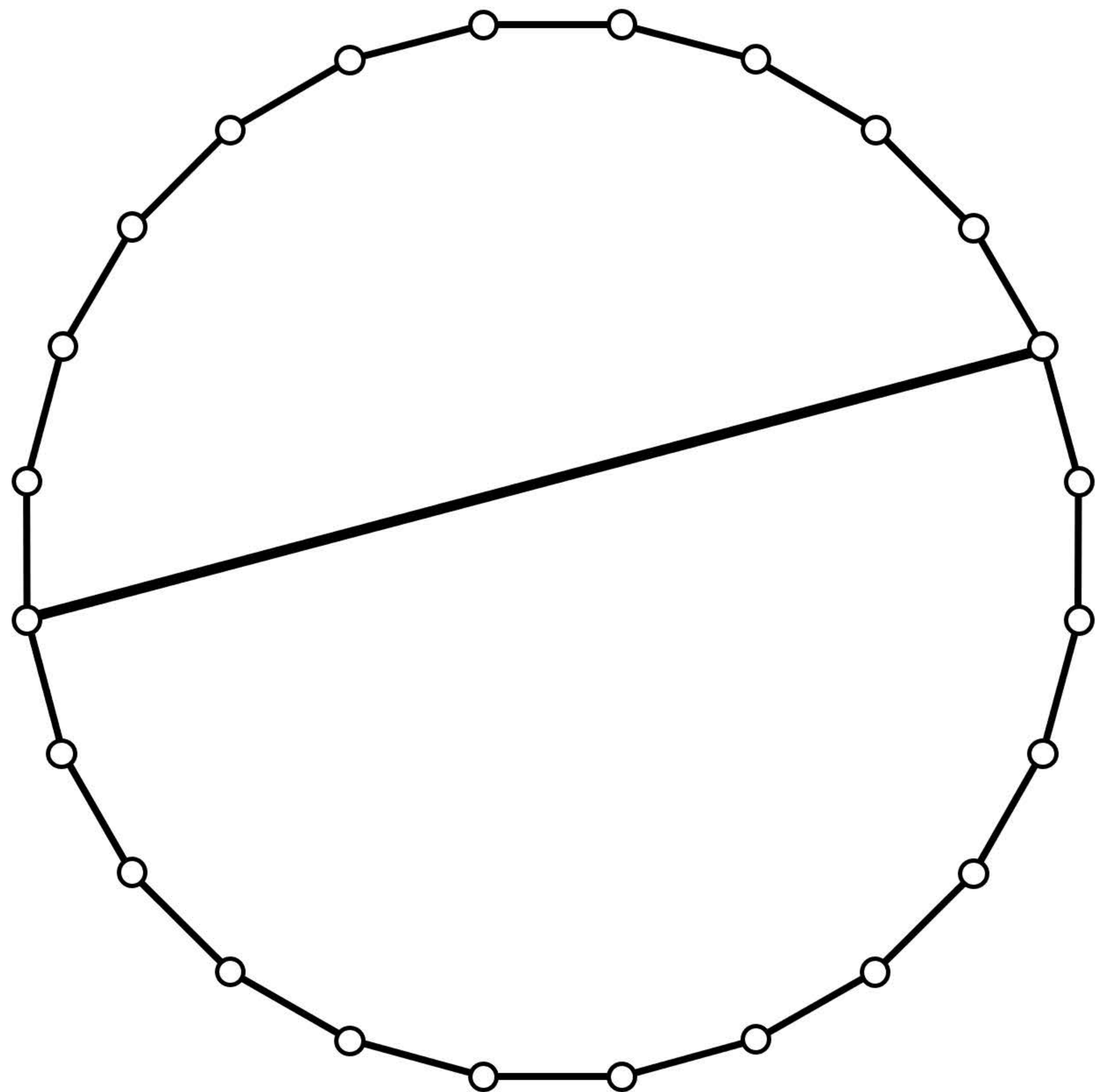


Notation

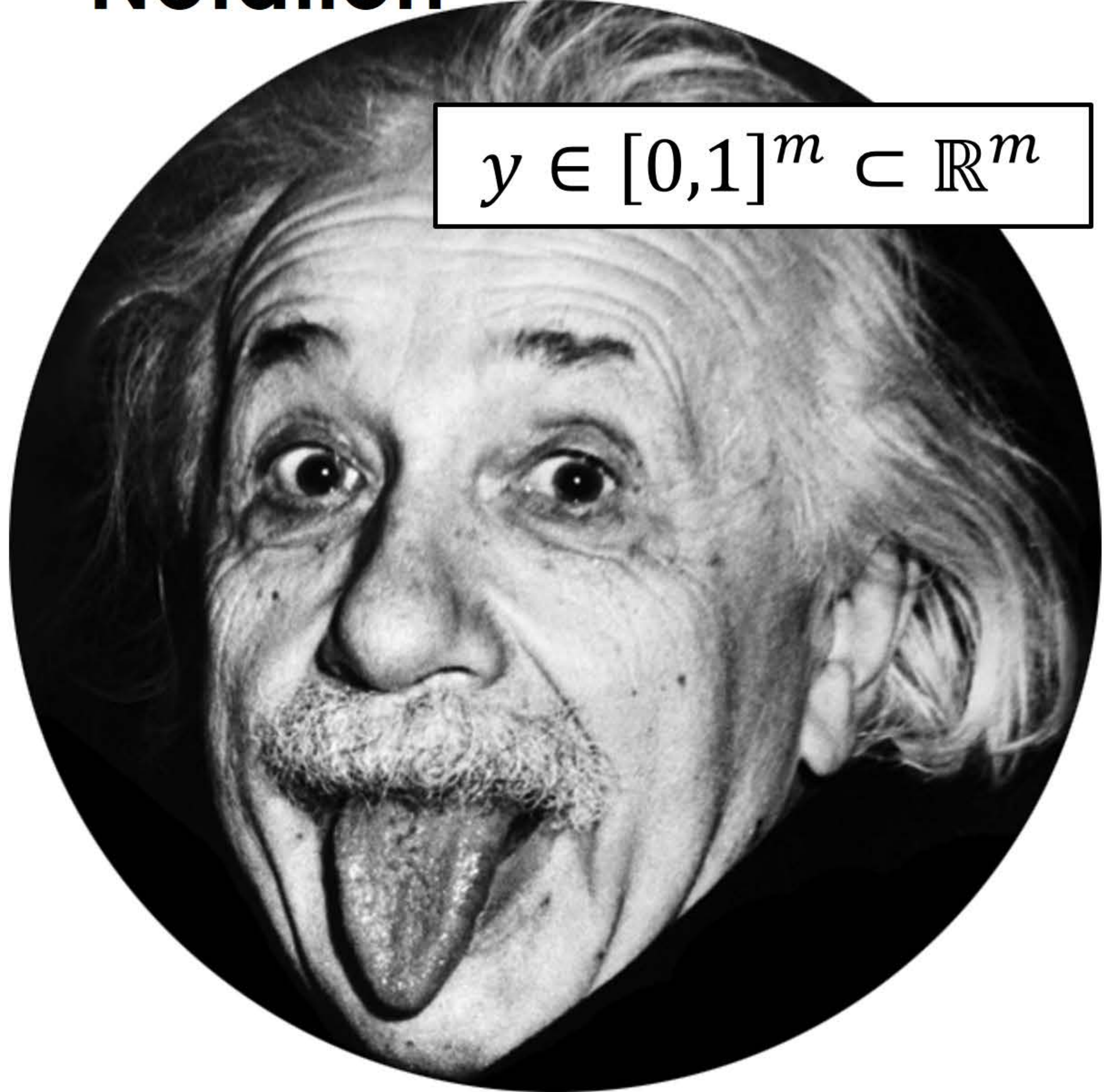
$$y \in [0,1]^m \subset \mathbb{R}^m$$



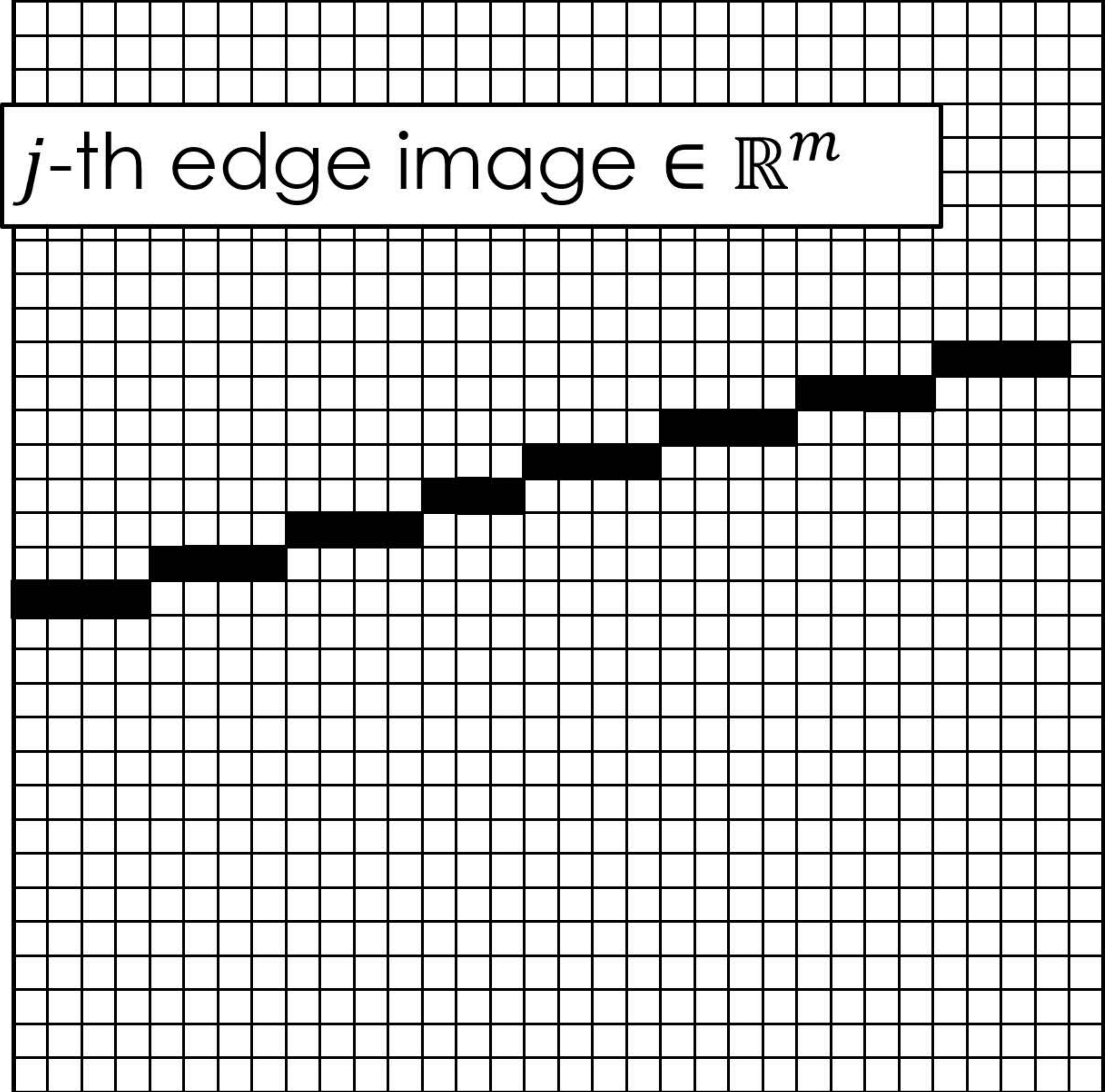
Target



Notation


$$y \in [0,1]^m \subset \mathbb{R}^m$$

Target


$$j\text{-th edge image} \in \mathbb{R}^m$$

Notation

$$y \in [0,1]^m \subset \mathbb{R}^m$$



$$j\text{-th edge image} \in \mathbb{R}^m$$

Notation

$$y \in [0,1]^m \subset \mathbb{R}^m$$



$$j\text{-th edge image} \in \mathbb{R}^m$$

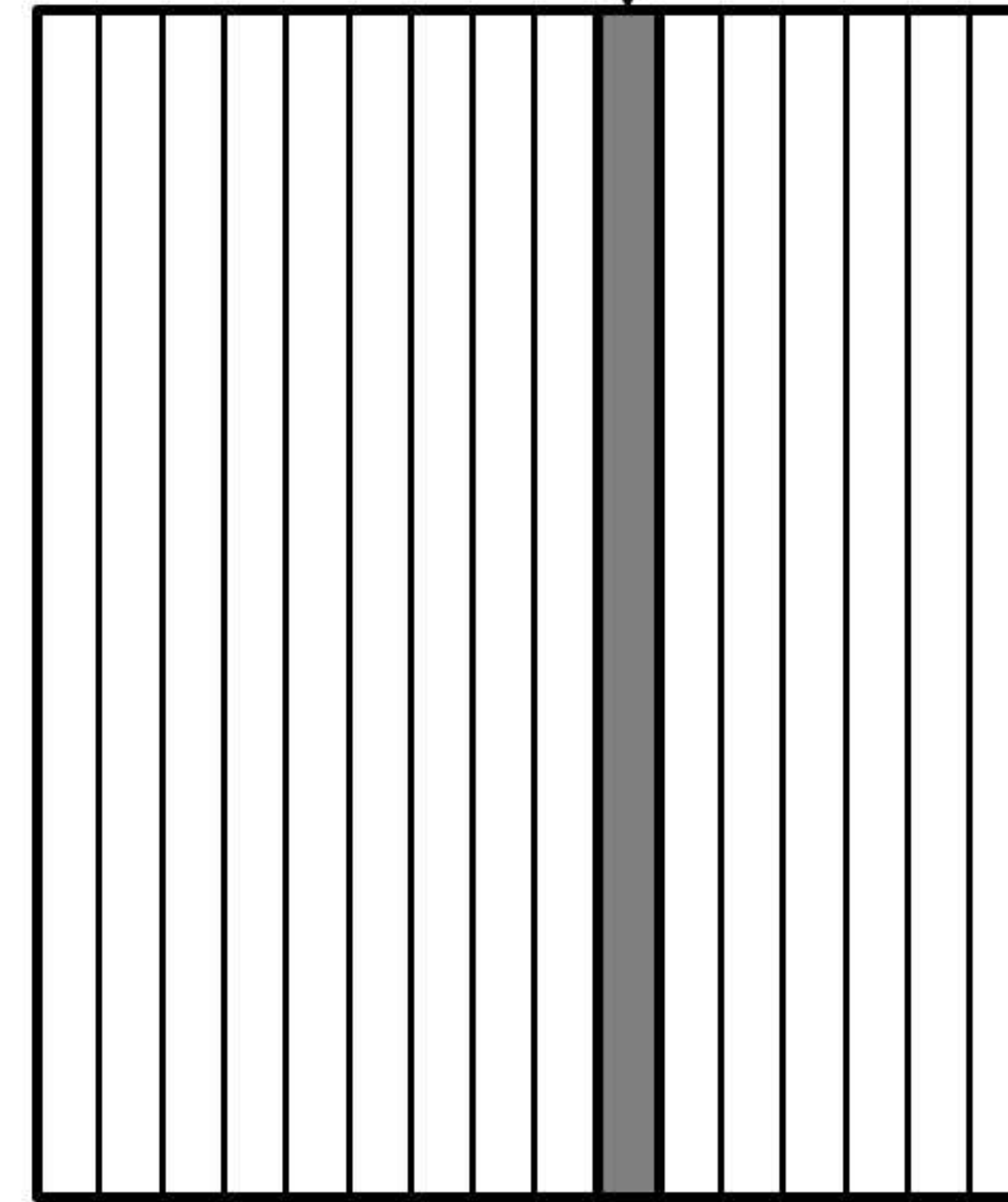


Notation

$$y \in [0,1]^m \subset \mathbb{R}^m$$



$$j\text{-th edge image} \in \mathbb{R}^m$$



$m \times n$ matrix A

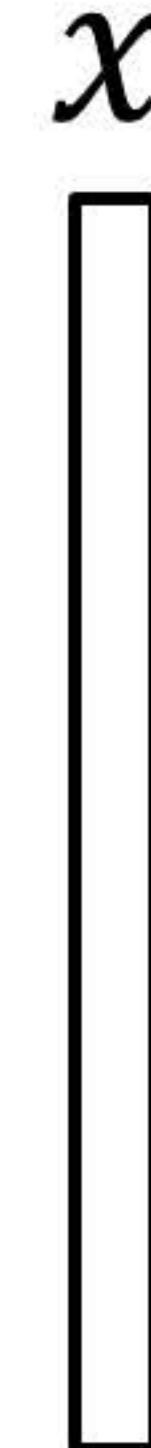
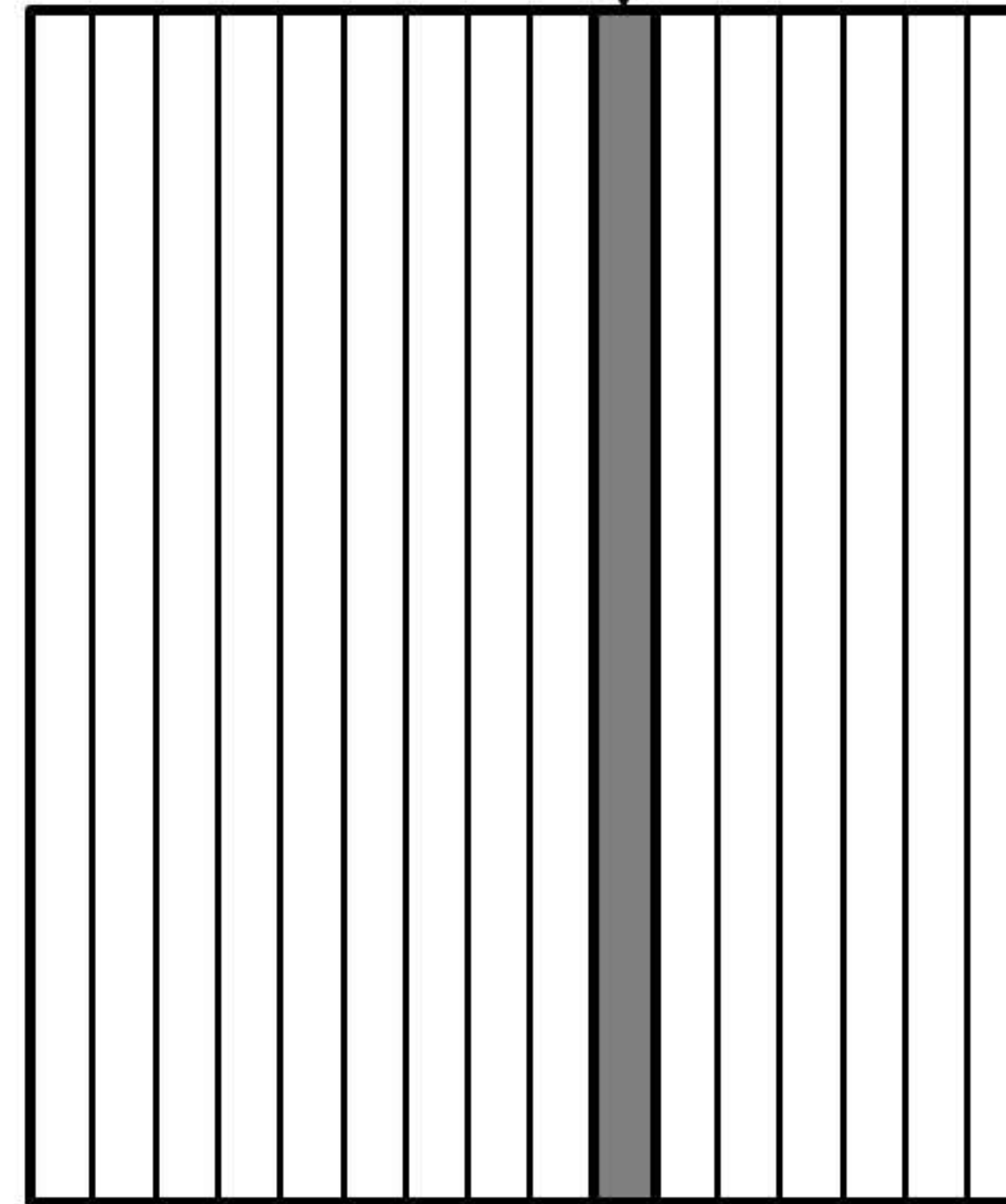
Notation

$$y \in [0,1]^m \subset \mathbb{R}^m$$



=

$$j\text{-th edge image} \in \mathbb{R}^m$$



.

$m \times n$ matrix A

Related Work

- Daniel Varga [Var17]



y

$$\Rightarrow \min_x \|Ax - y\|_2^2 \Rightarrow$$



$z = Ax$

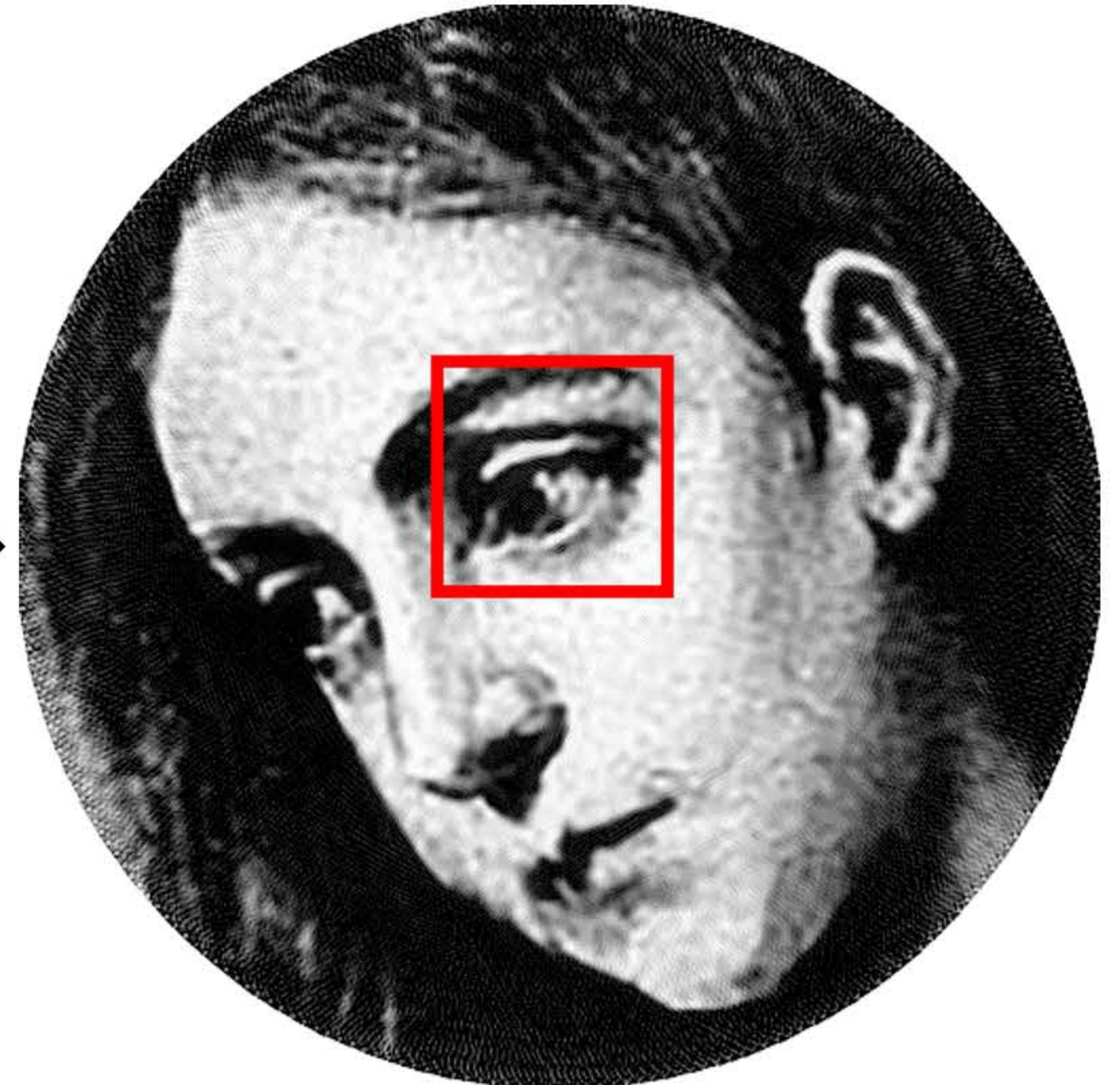
Related Work

- Daniel Varga [Var17]



y

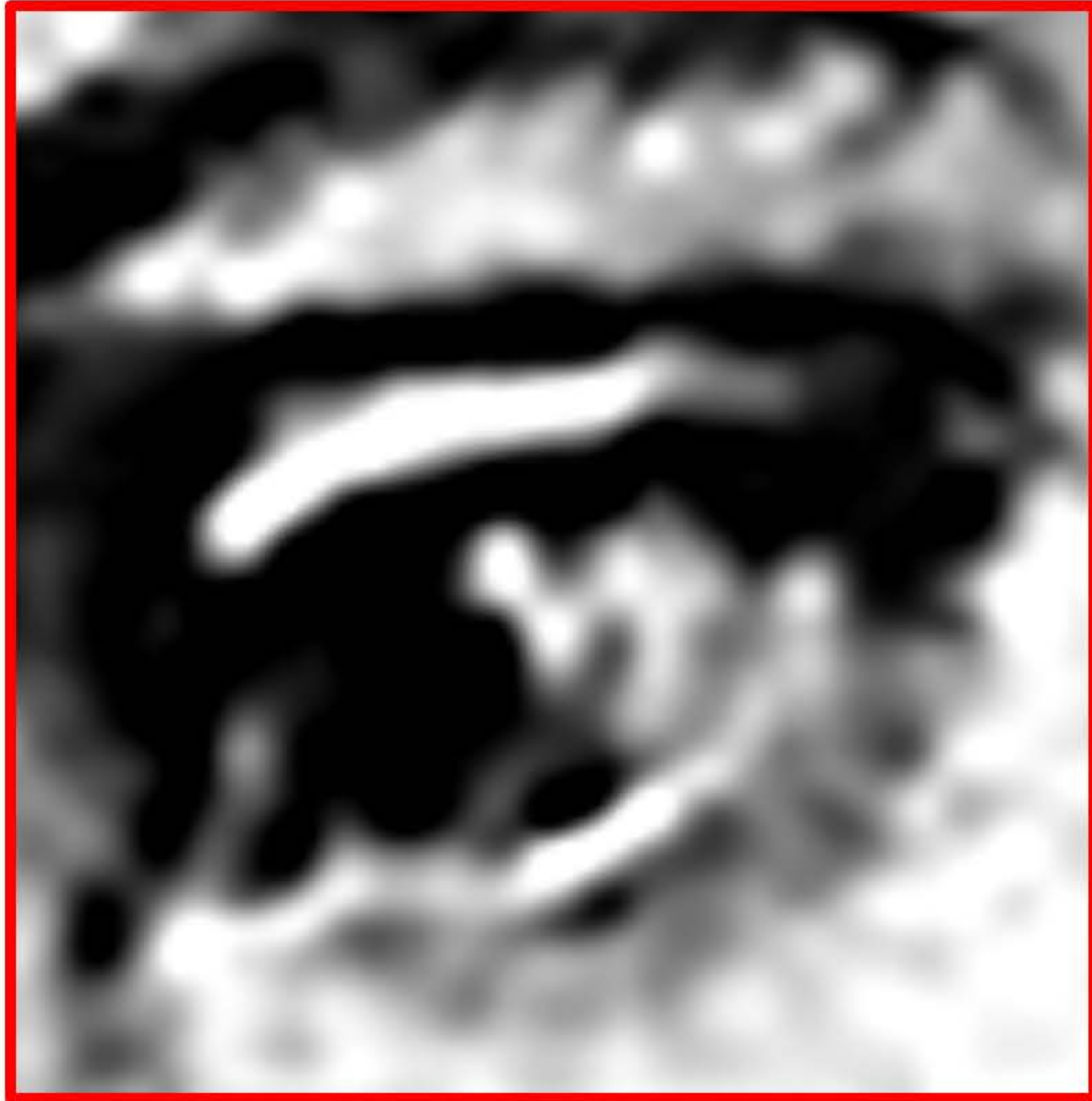
$$\Rightarrow \min_x \|Ax - y\|_2^2 \Rightarrow$$



$z = Ax$

Related Work

- Daniel Varga [Var17]



y

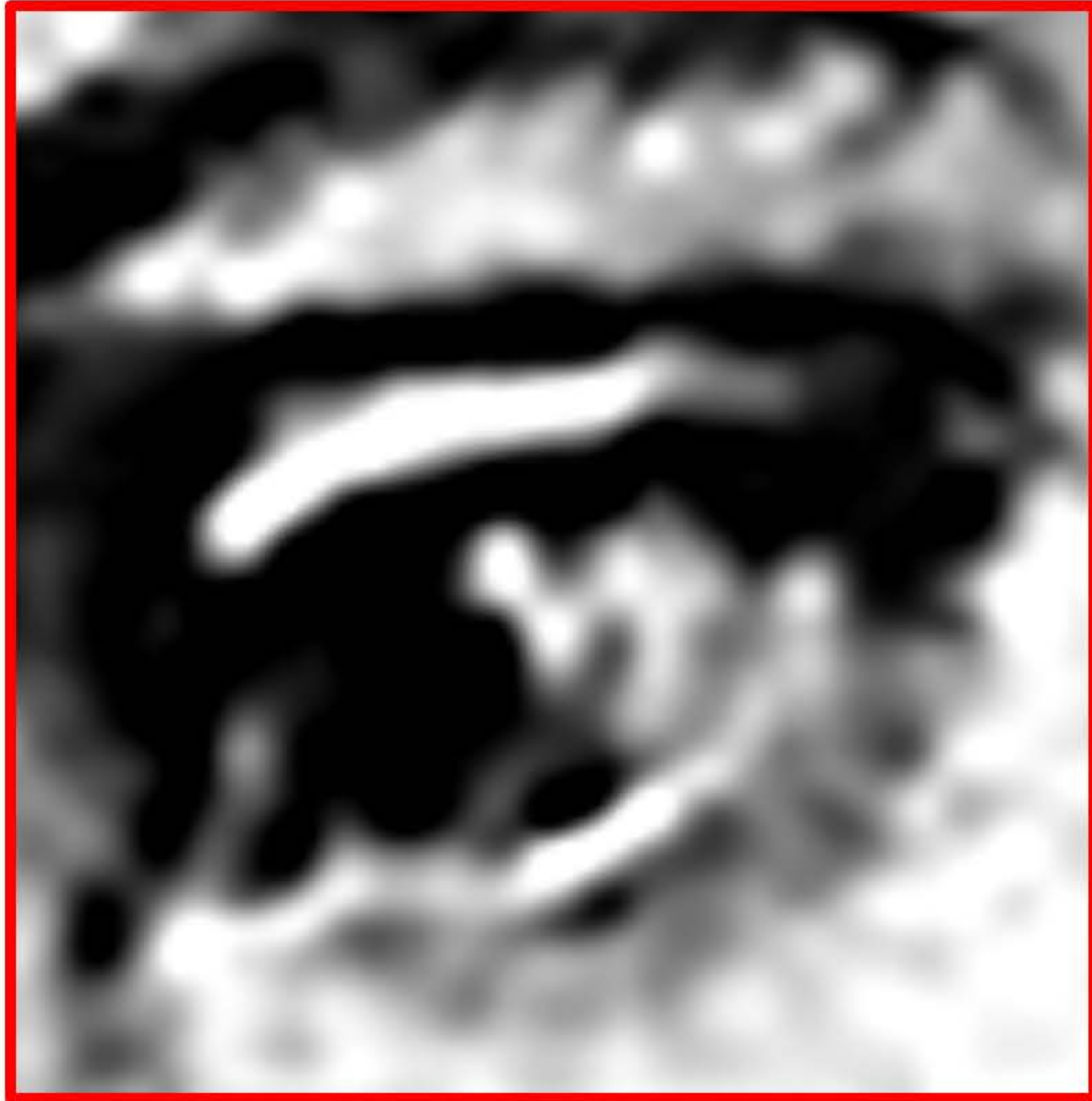
$$\Rightarrow \min_x \|Ax - y\|_2^2 \Rightarrow$$



$z = Ax$

Related Work

- Daniel Varga [Var17]



y

$$\Rightarrow \min_x \|Ax - y\|_2^2 \Rightarrow$$

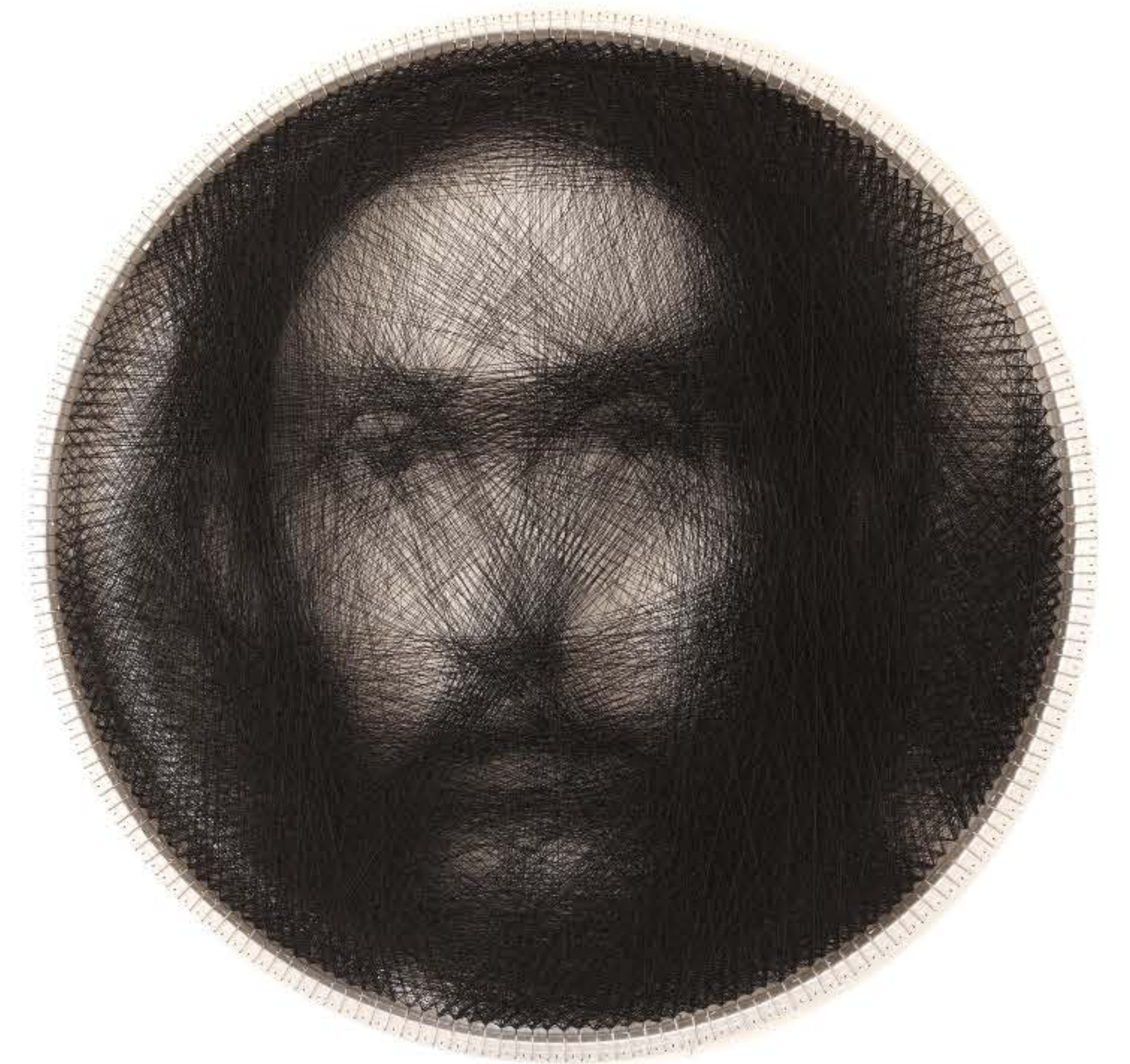
✗ not fabricable



$z = Ax$

Related Work

- „A new way to knit“ by Artist Petros Vrellis [Vre16]

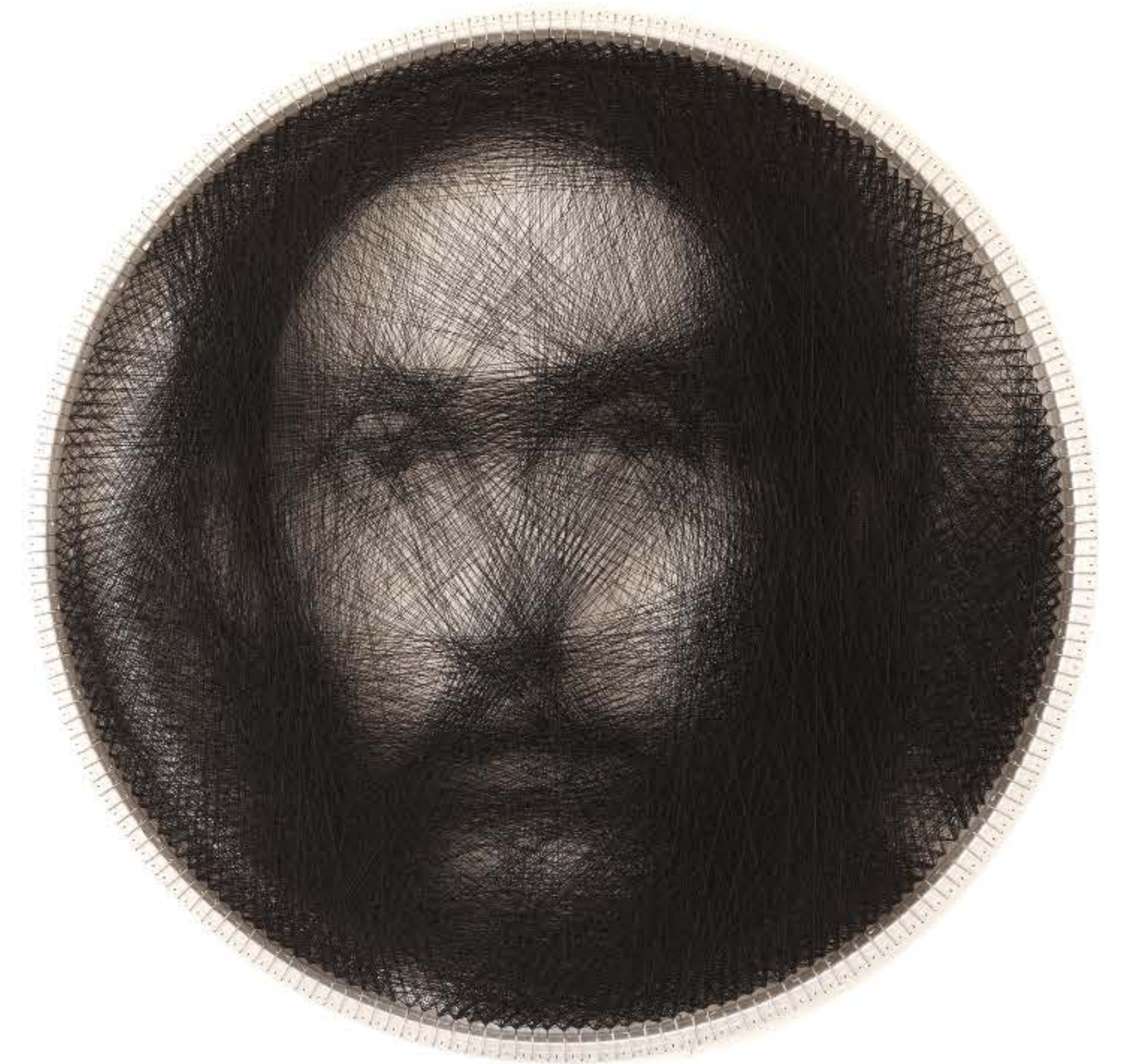


Related Work

- „A new way to knit“ by Artist Petros Vrellis [Vre16]

✗ Proprietary algorithm

✗ Hand crafted

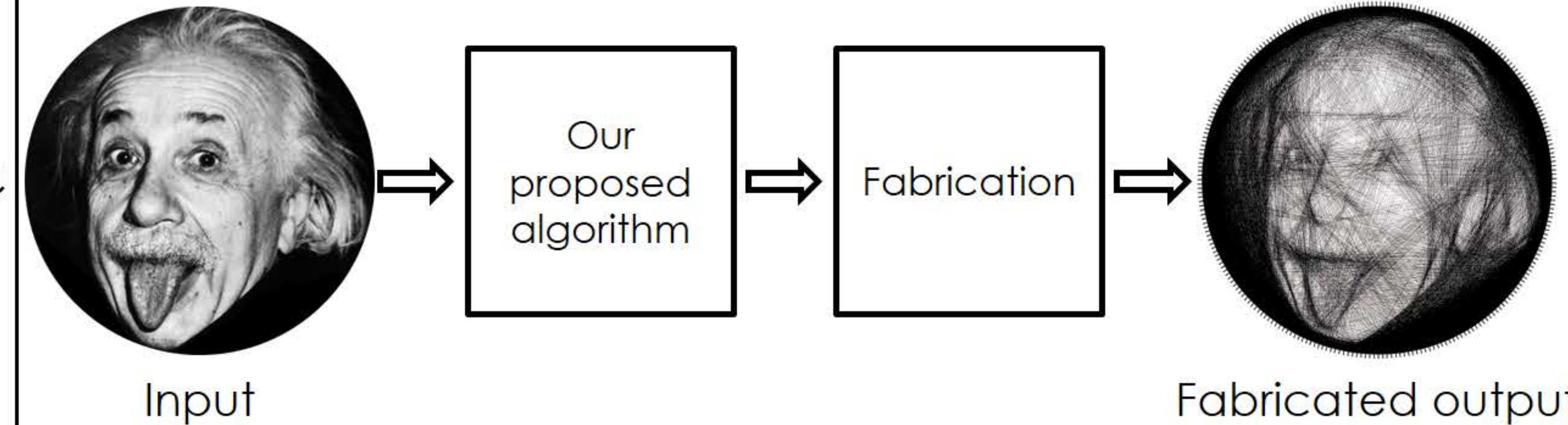


Contributions

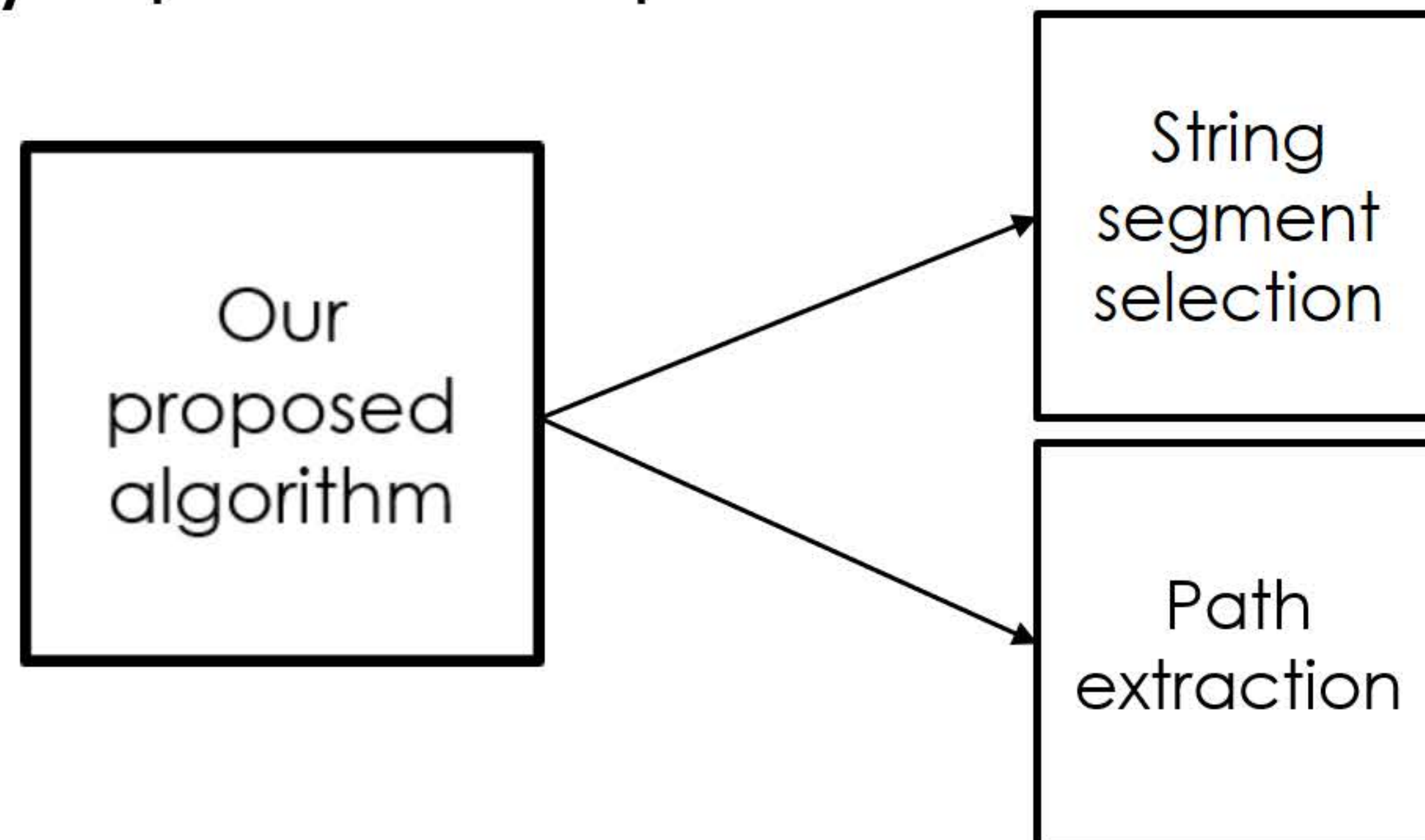
(C1) First formal fabrication-aware treatment of the problem

$$\min_x \|WDC(\tilde{A}x) - Wy\|_2^2 \text{ s. t. } x \in \{0,1\}^n$$

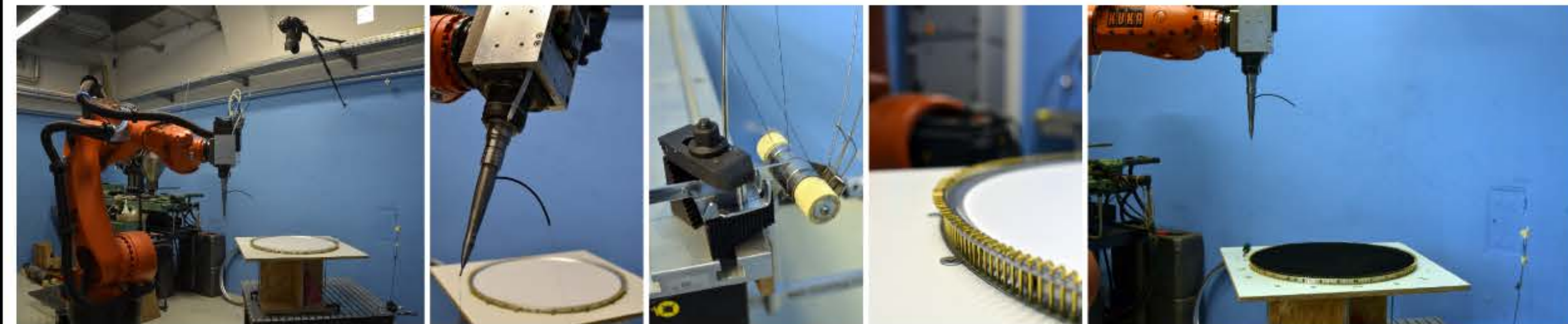
(C2) Automatic computational fabrication pipeline



(C3) Split of the problem

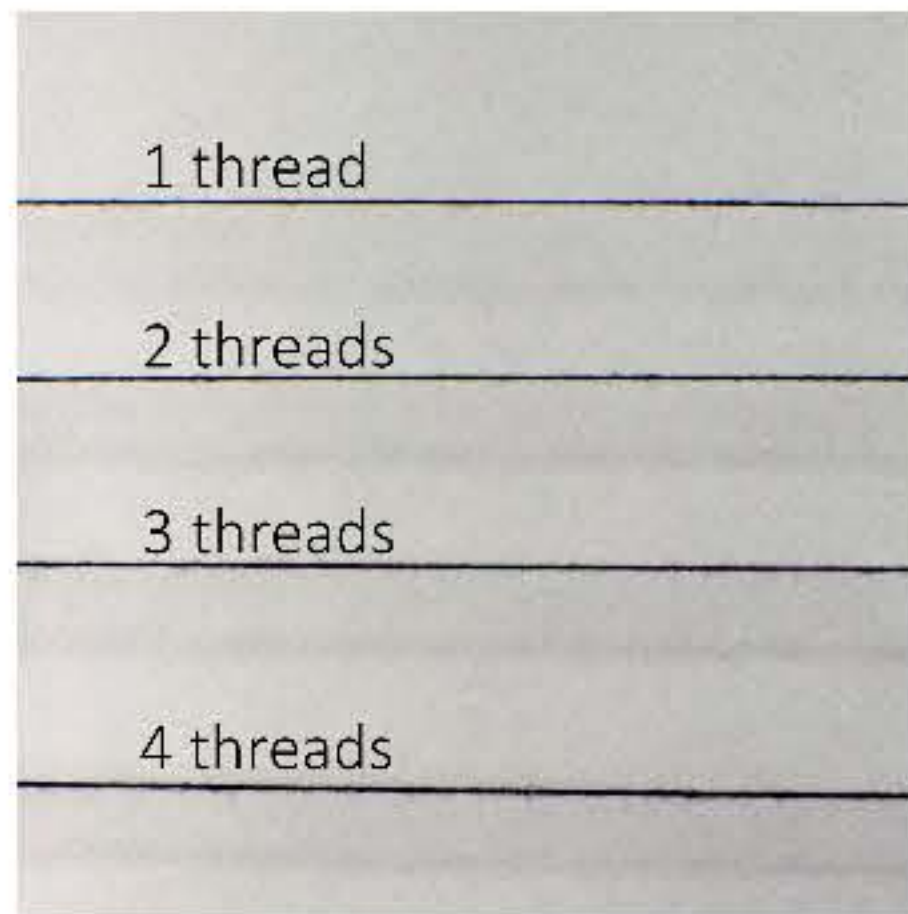


(C4) Automatic fabrication setup



Fabrication Considerations

(R1) Fully opaque string



$$\rightarrow x \in \{0,1\}^n$$

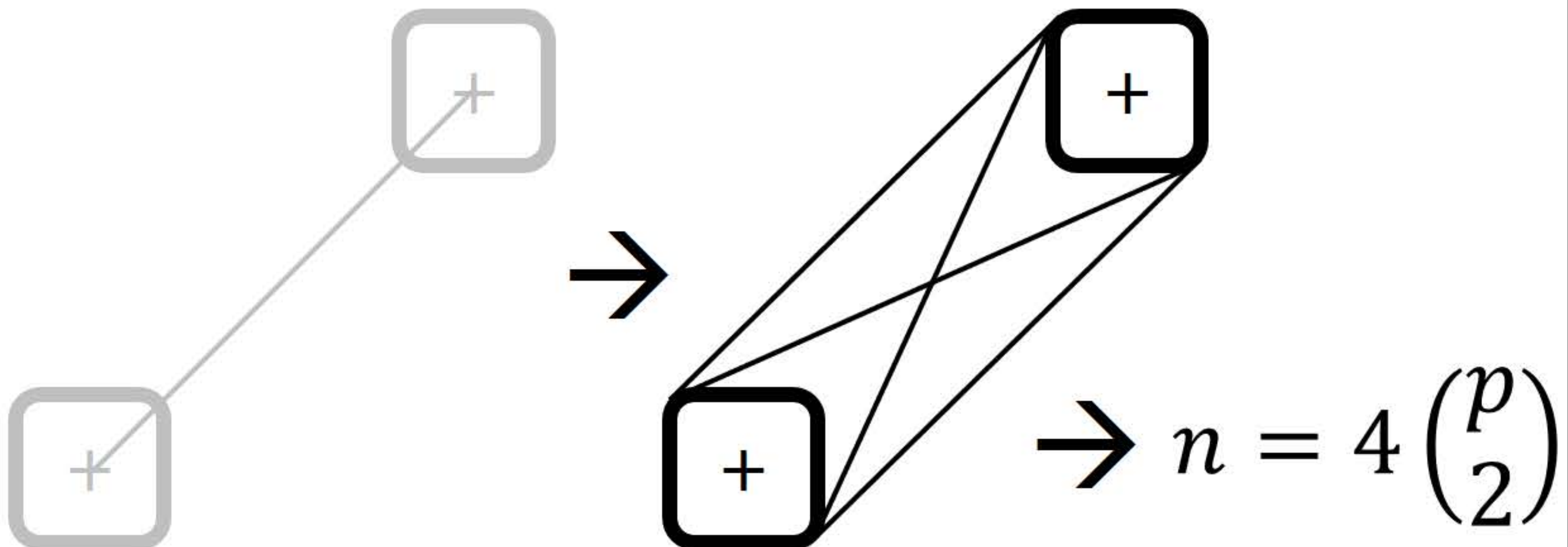
(R2) Choice of canvas resolution

Canvas diameter (e.g. 630 mm)

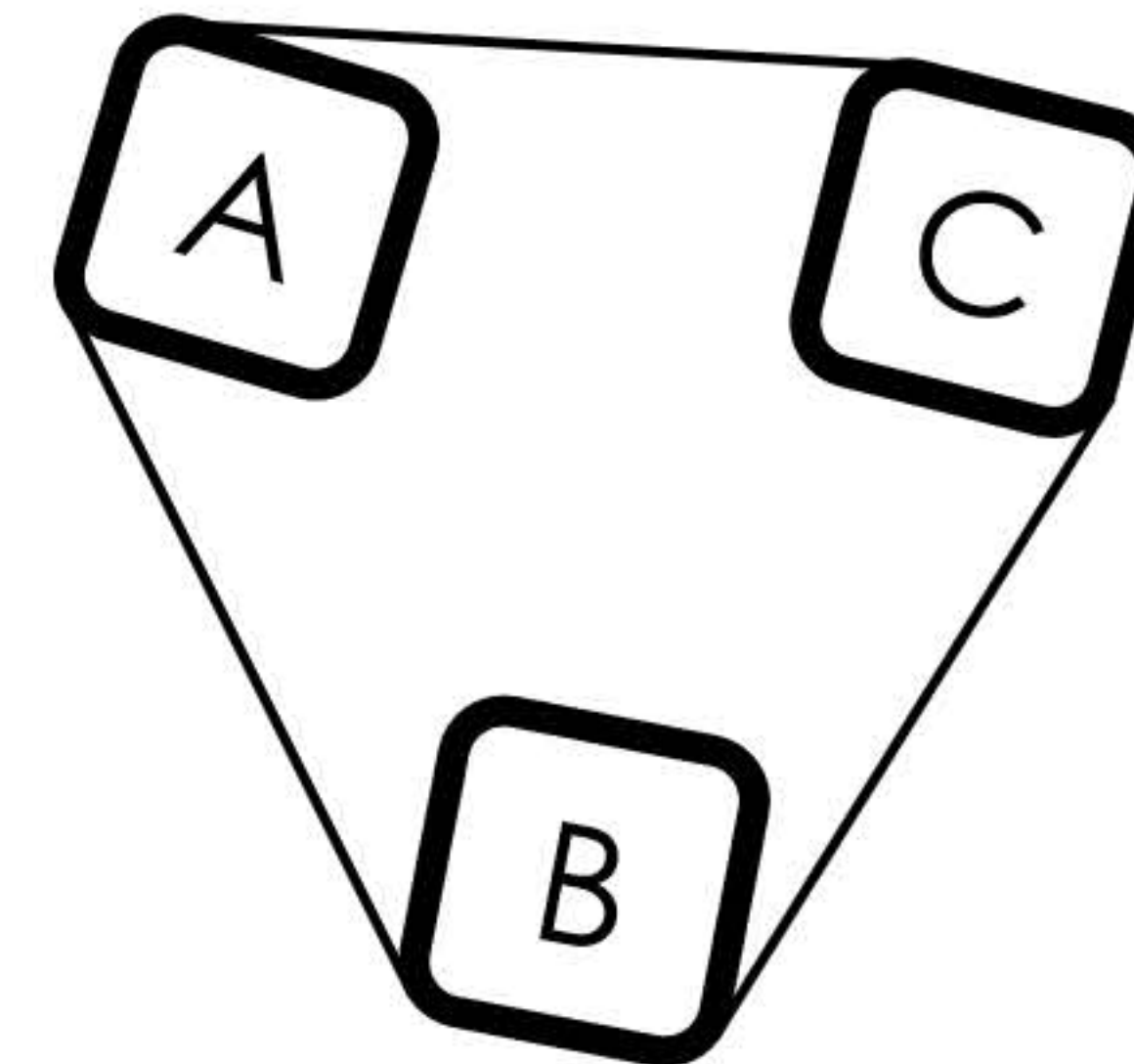
$$Z = \frac{1}{t} d \rightarrow 4096 \times 4096 \text{ pixels}$$

Physical string thickness (e.g. 0.15 mm)

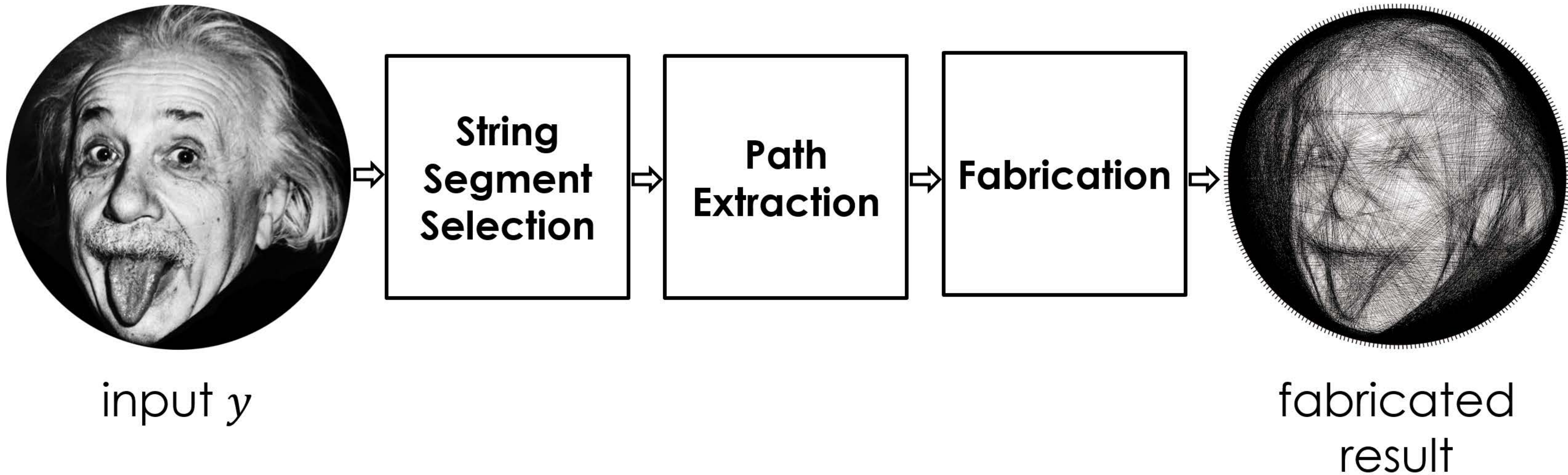
(R3) Account for pin diameter



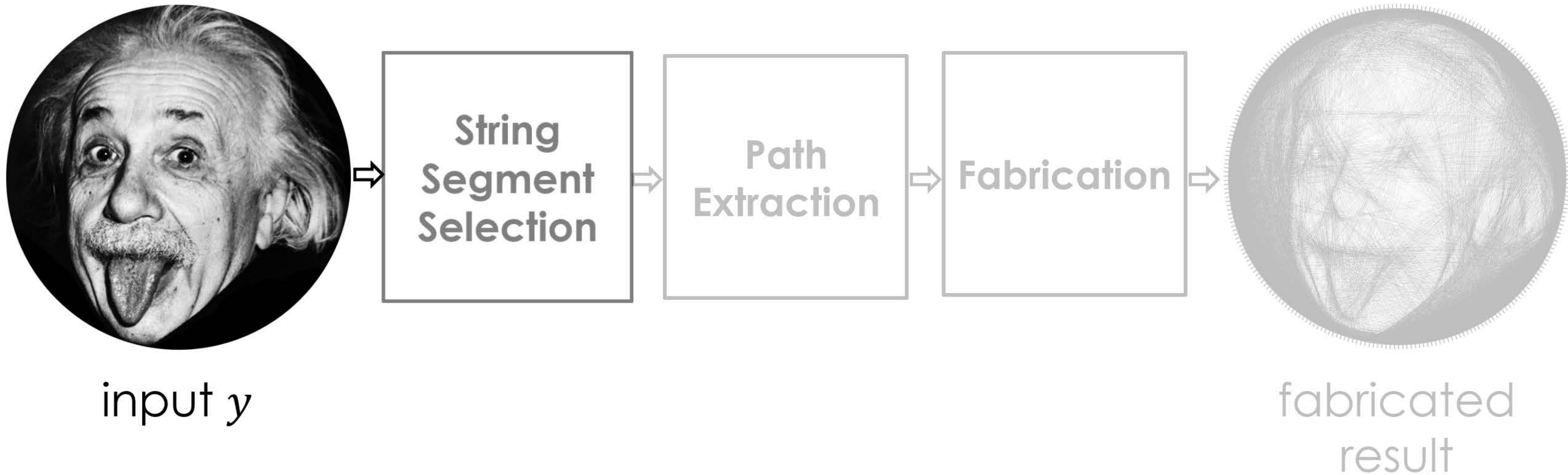
(R4) Graph needs to be Eulerian



System Overview

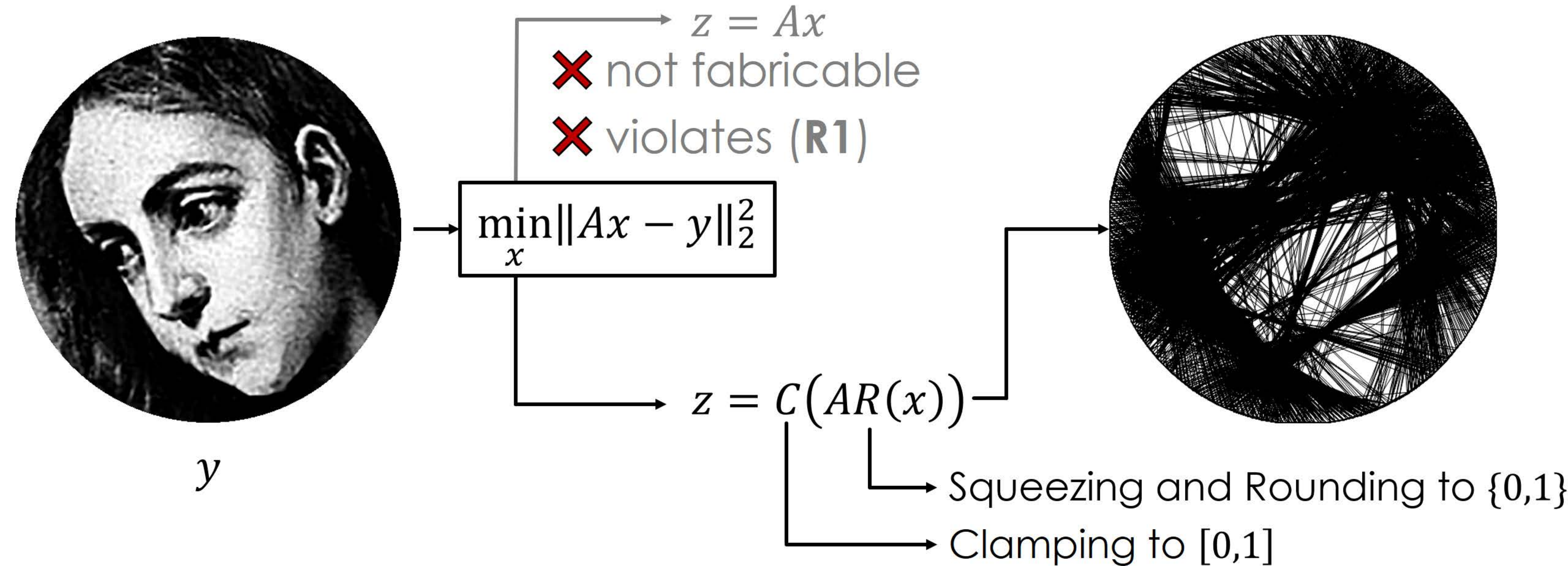


System Overview



Objective Derivation

- Introducing Rounding into the work of [Var17]



Objective Derivation



$$z = C(AR(x))$$

✗ non-satisfying

✗ violates requirement **(R2)** $\left(z = \frac{1}{t}d\right)$

Idea: Upscaling

Introducing $U: \mathbb{R}^m \rightarrow \mathbb{R}^{sm}$

$$\min_x \|\tilde{A}x - Uy\|_2^2 \text{ with } \tilde{A}: \mathbb{R}^n \rightarrow \mathbb{R}^{sm}$$

Objective Derivation

$$\min_x \|\tilde{A}x - Uy\|_2^2 \text{ with } \tilde{A}: \mathbb{R}^n \rightarrow \mathbb{R}^{sm}$$

✗ violates (R1)

$$z = \tilde{A}x$$



$$z = C(\tilde{A}R(x))$$

✗ inaccurate



Objective Derivation

- Crucial disadvantage of least squares approaches is their non-binary output
- **Idea:** Forcing the result x to be binary (NP-hard)
 - L_1 : $\min_x \|DC(\tilde{A}x) - y\|_1 \rightarrow \min_{x_j} \sum_{i=1}^m \delta_i$ not manageable
 - L_2 : Existing approaches [QS08, CZ07] can not handle huge sparse matrices like $\tilde{A}$ (130,560 edges and 4096^2 pixels)

Objective Derivation

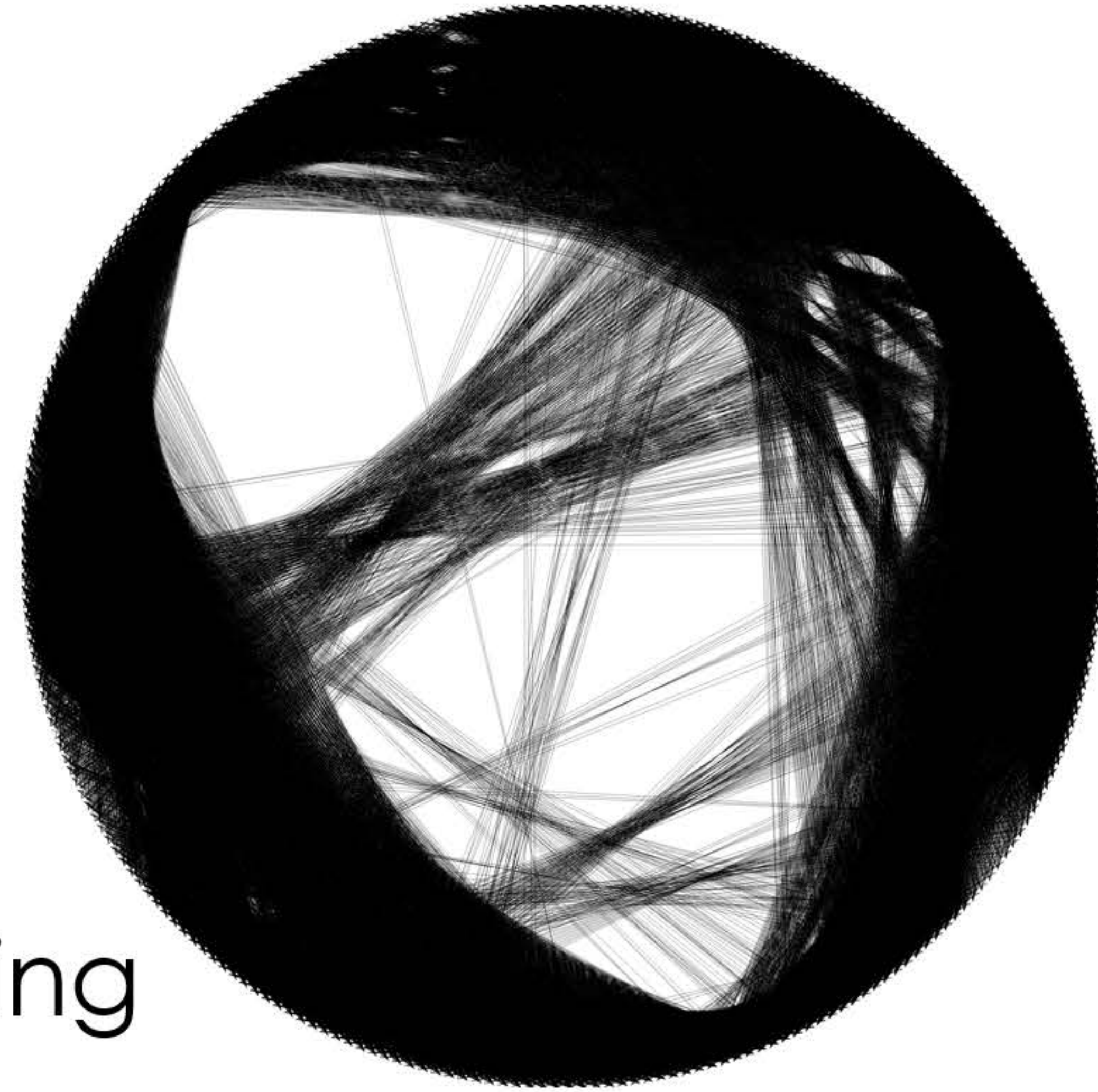
- Our solution: an iterative greedy solver

$$\min_x \|C(Ax) - y\|_2^2 \text{ s.t. } x \in \mathbb{B}^n$$

$$\min_x \|C(\tilde{A}x) - Uy\|_2^2 \text{ s.t. } x \in \mathbb{B}^n$$



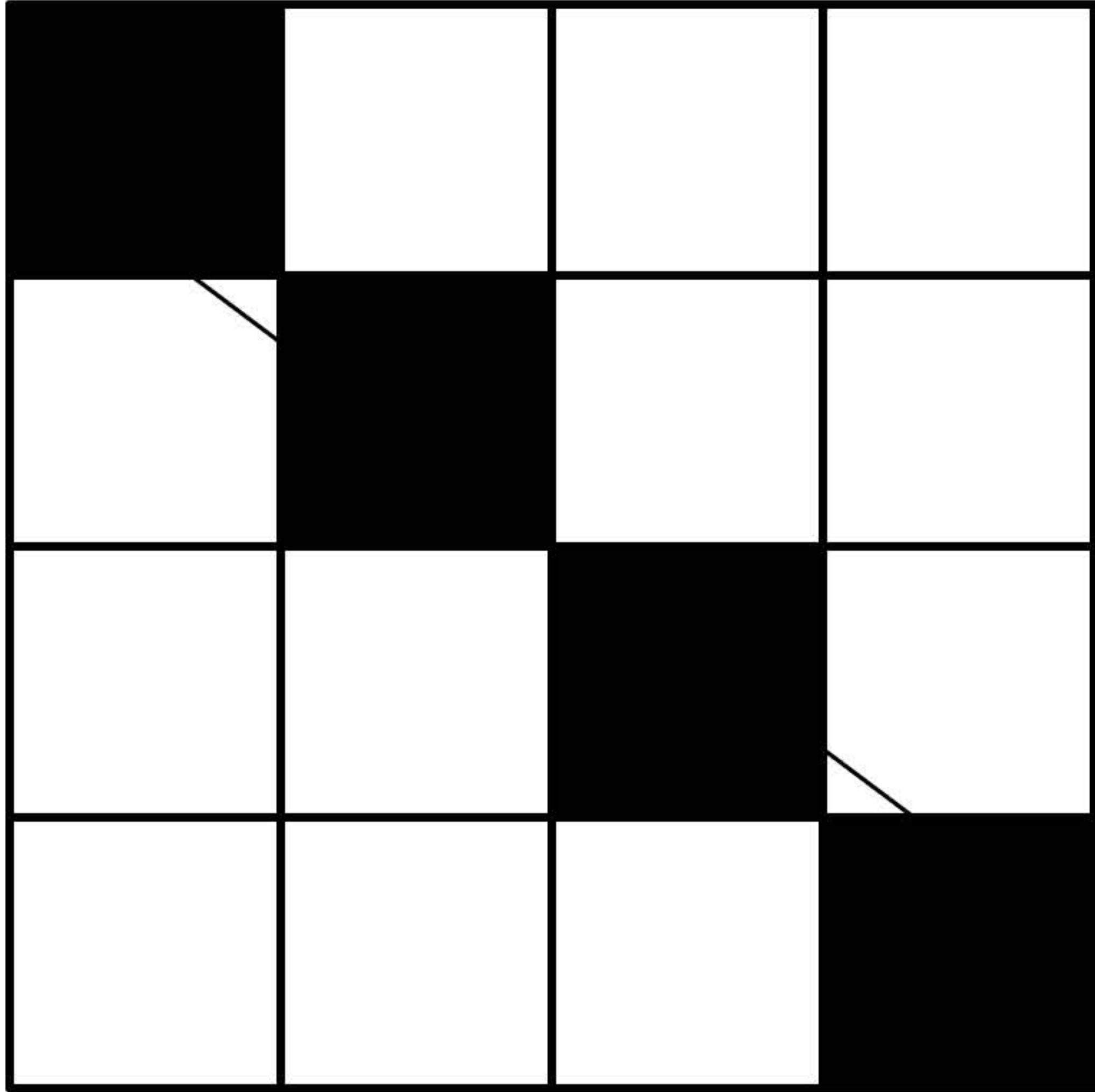
✗ non-satisfying



Range Problem

- Results so far are either too dark or too bright
- **Reason**
 - Output image lacks an expressive range per pixel
 - Each pixel linear combination of binary edge contributions
 - Pixel values almost always ≥ 1

Range Problem

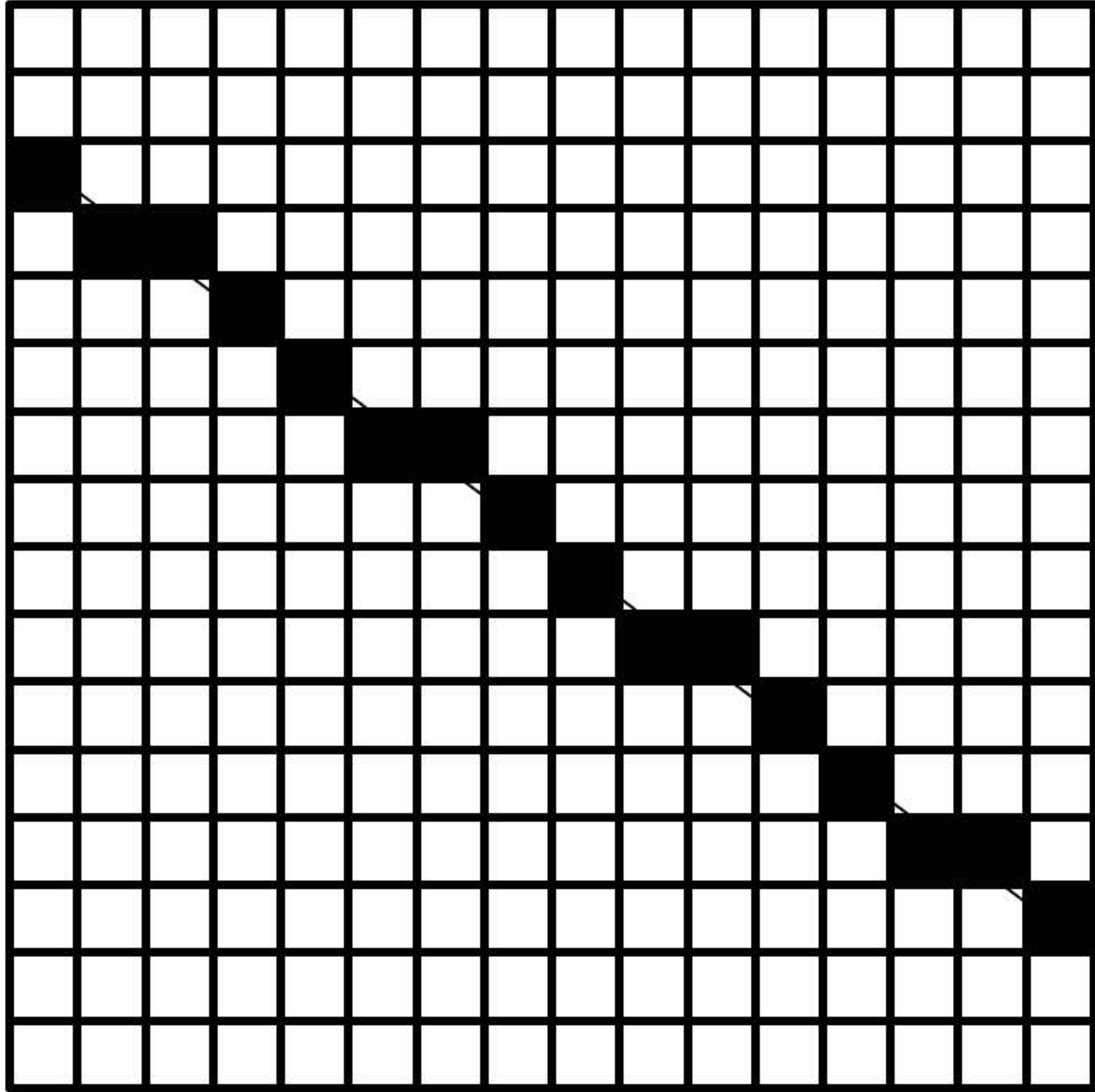


$$\min_x \|C(Ax) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|C(\tilde{A}x) - Uy\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|DC(\tilde{A}x) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

Range Problem

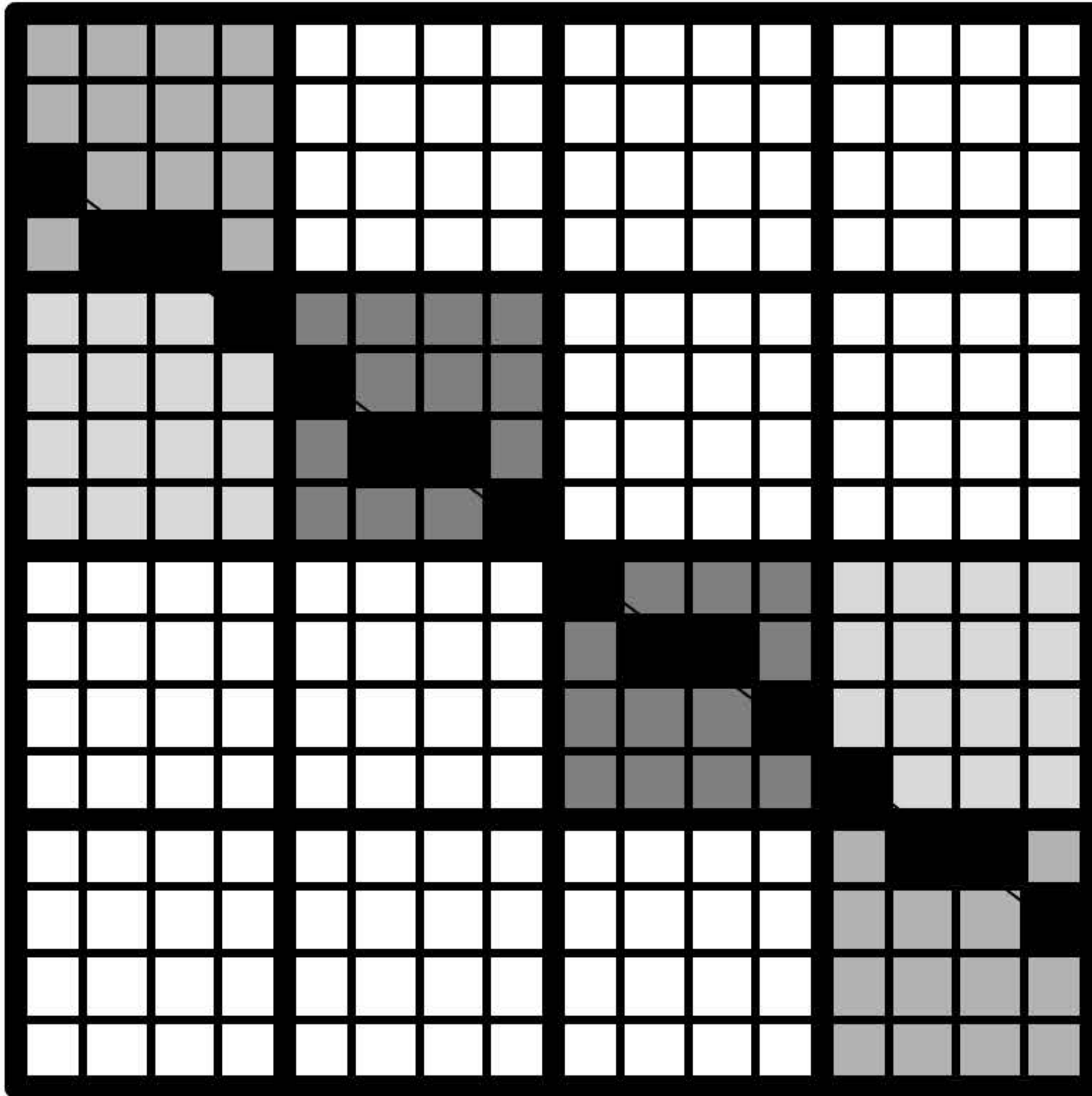


$$\min_x \|C(Ax) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|C(\tilde{A}x) - Uy\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|DC(\tilde{A}x) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

Range Problem

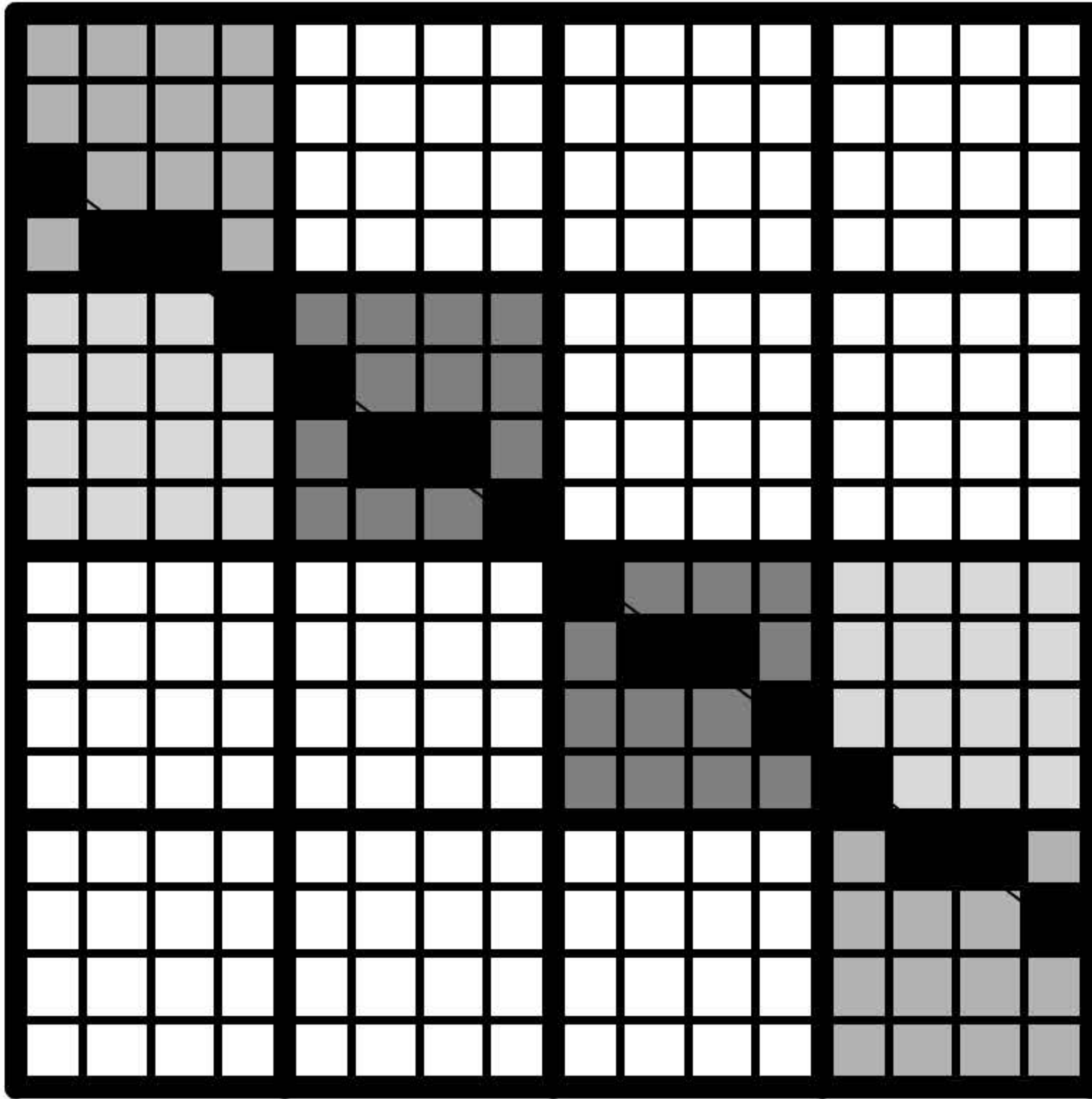


$$\min_x \|C(Ax) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|C(\tilde{A}x) - Uy\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|D_1 C(\tilde{A}x) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

Range Problem



$$\min_x \|C(Ax) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|C(\tilde{A}x) - Uy\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$$\min_x \|DC(\tilde{A}x) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

$\downarrow$
 $D: [0,1]^{sm} \rightarrow [0,1]^m \text{ with } u \mapsto D(u) = Bu$

Range Problem

$$\min_x \|C(\tilde{A}x) - Uy\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$



$$\min_x \|DC(\tilde{A}x) - y\|_2^2 \text{ s. t. } x \in \mathbb{B}^n$$

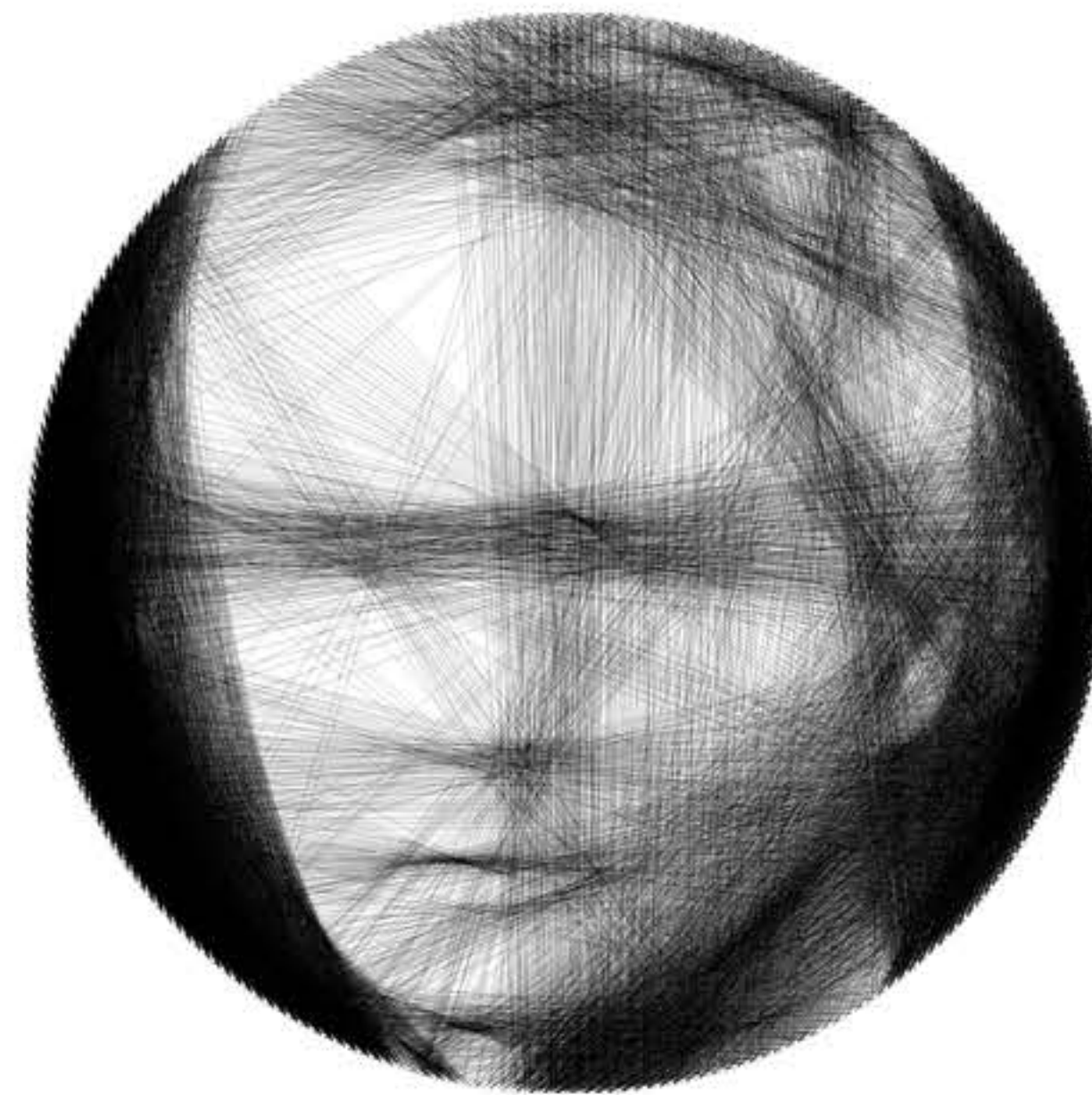


Importance Mapping

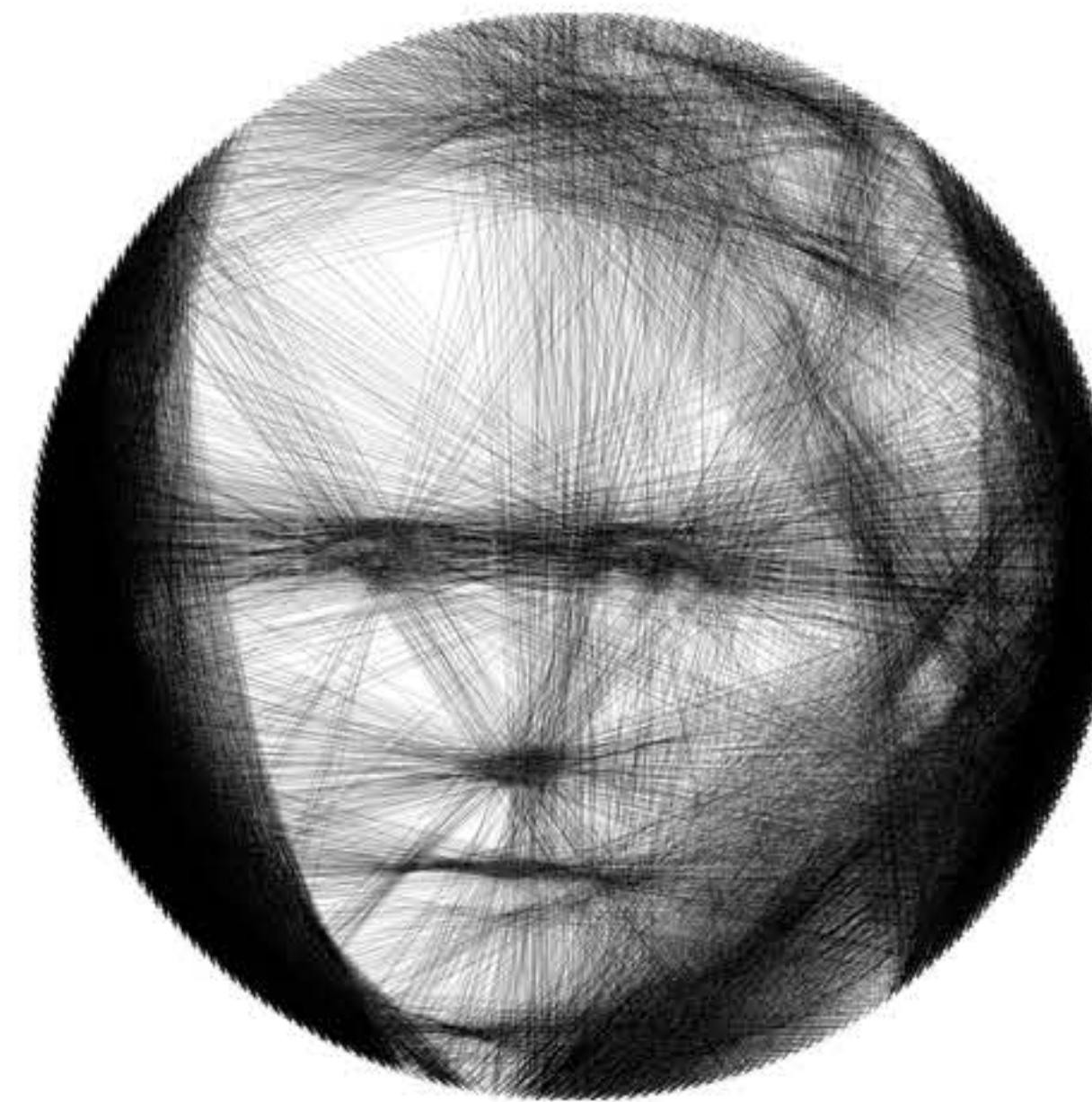
- Real-valued importance map $W: [0,1]^m \rightarrow [0,1]^m$
- Allows to enhance the importance of regions manually



input y



output
(without W)

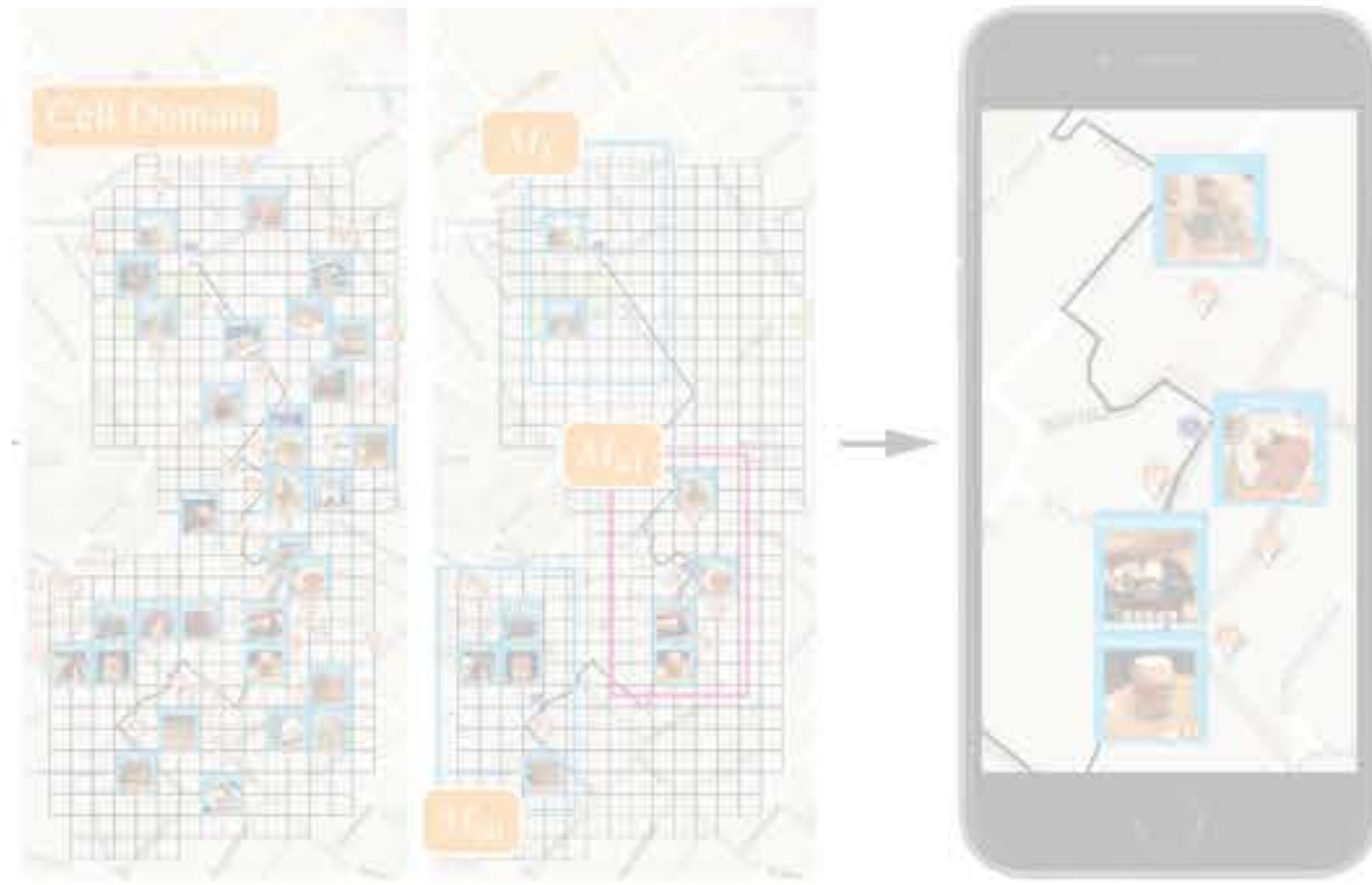


output
(with W)



W

Path Computation

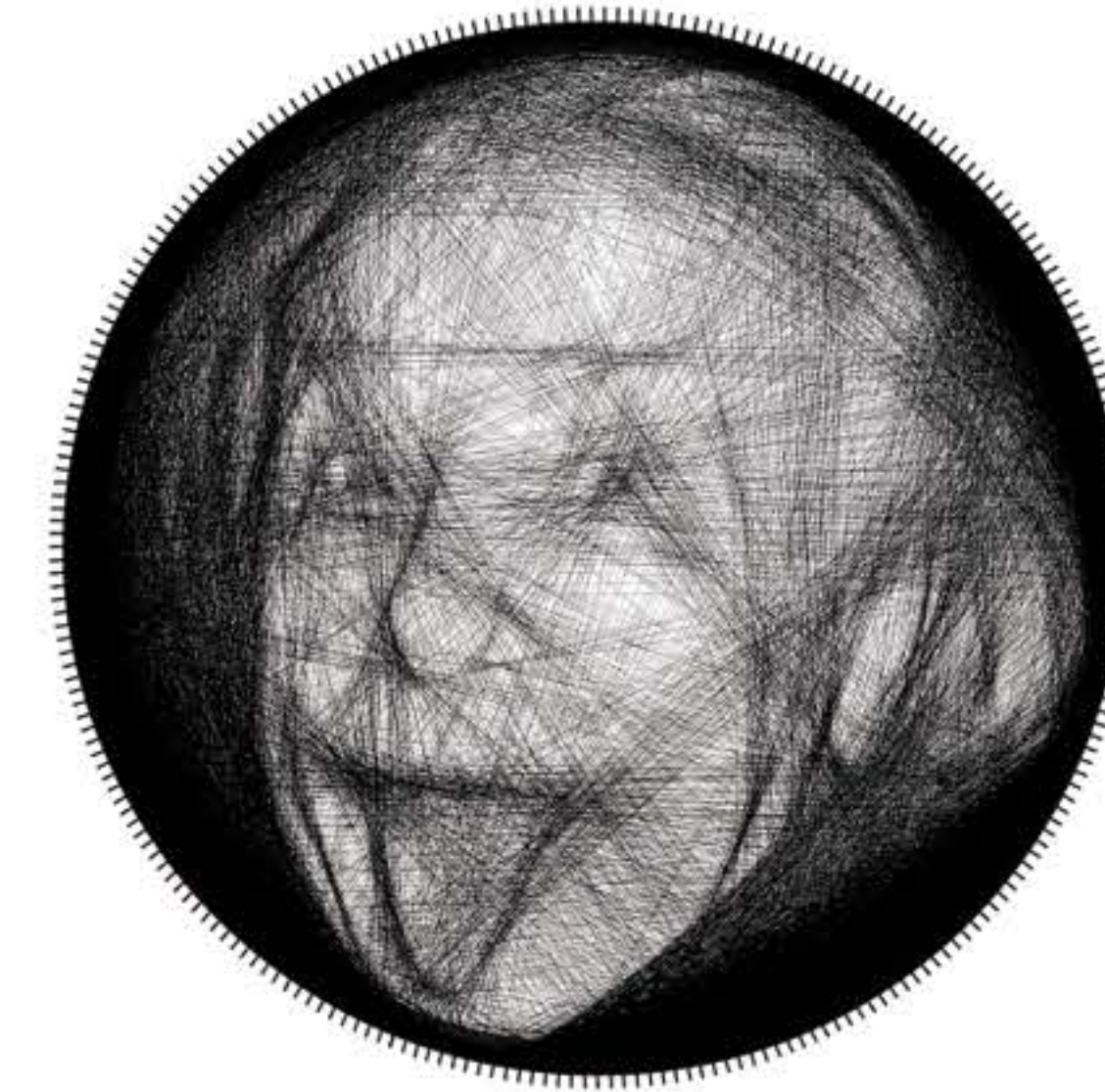


Path

$$\max_{y_i} \sum_{i \in V_P} q(P_i) y_i$$

Layout

$$\max_{v_{\tau,i,x_c,y_c}} \sum_{\tau=1}^{\tau_{\max}} \sum_{i=1}^{n^*} \sum_{c \in C} q_{\tau,i,x_c,y_c} v_{\tau,i,x_c,y_c}$$



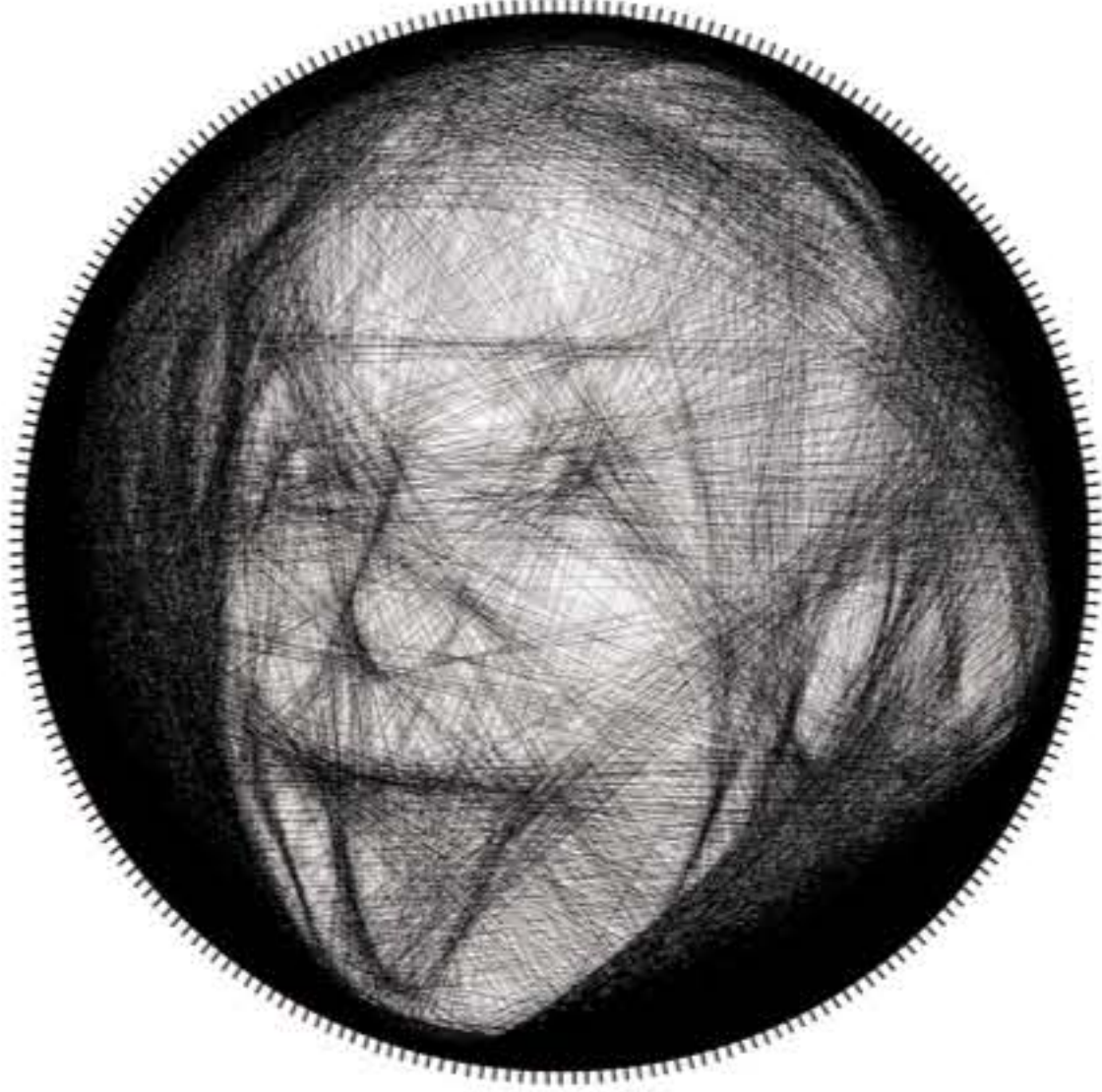
String Selection

$$\min_{x_j} \sum_{i=1}^m \delta_i$$

Not manageable
with Gurobi

➔ Own Greedy Solver

Path Computation



- L_1 minimization: $\min_x \|DC(\tilde{A}x) - y\|_1$ s. t. $x \in \mathbb{B}^n$
- δ_i is the abs. difference to input value y_i
- more than 16M variables and constraints

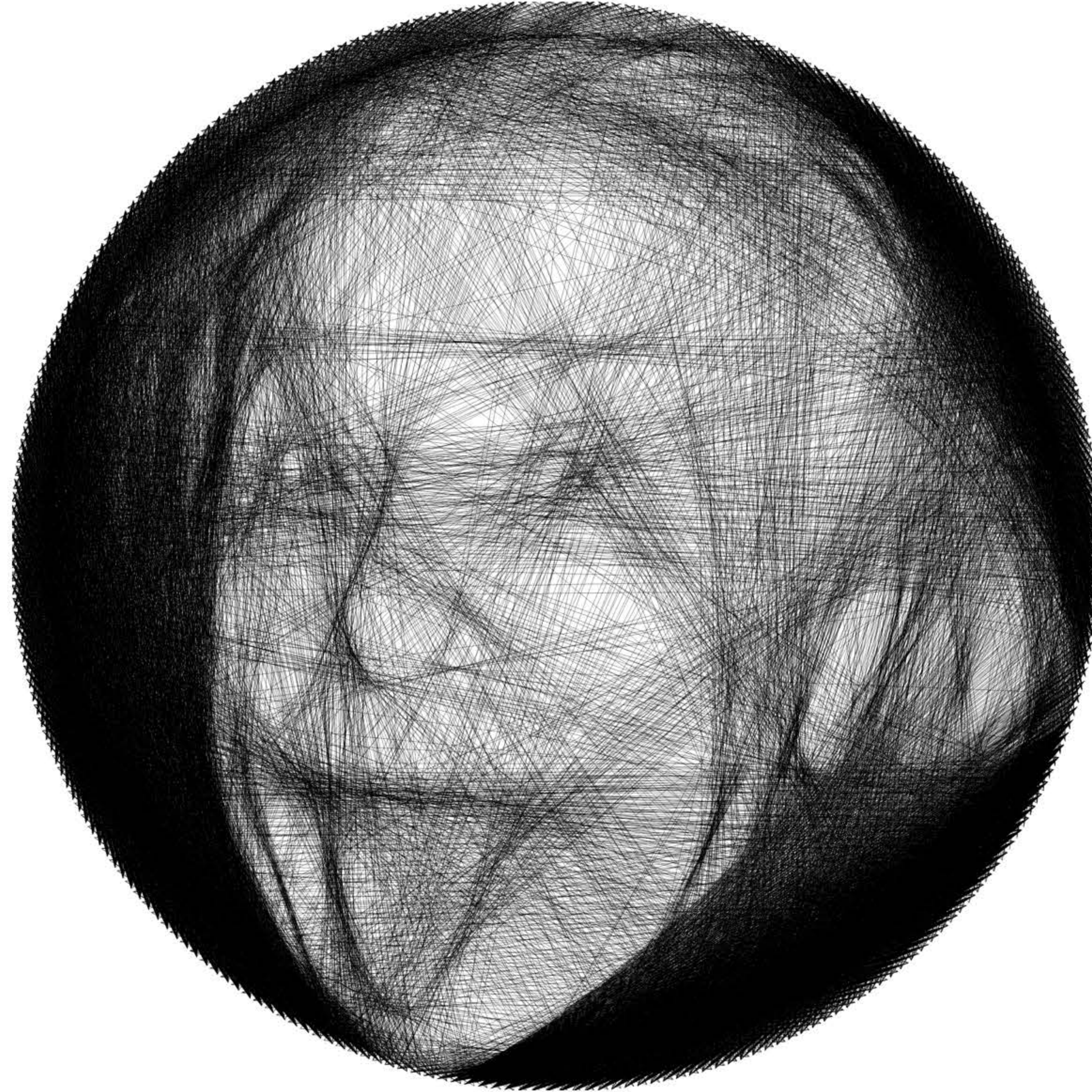
String Selection

$$\min_{x_j} \sum_{i=1}^m \delta_i$$

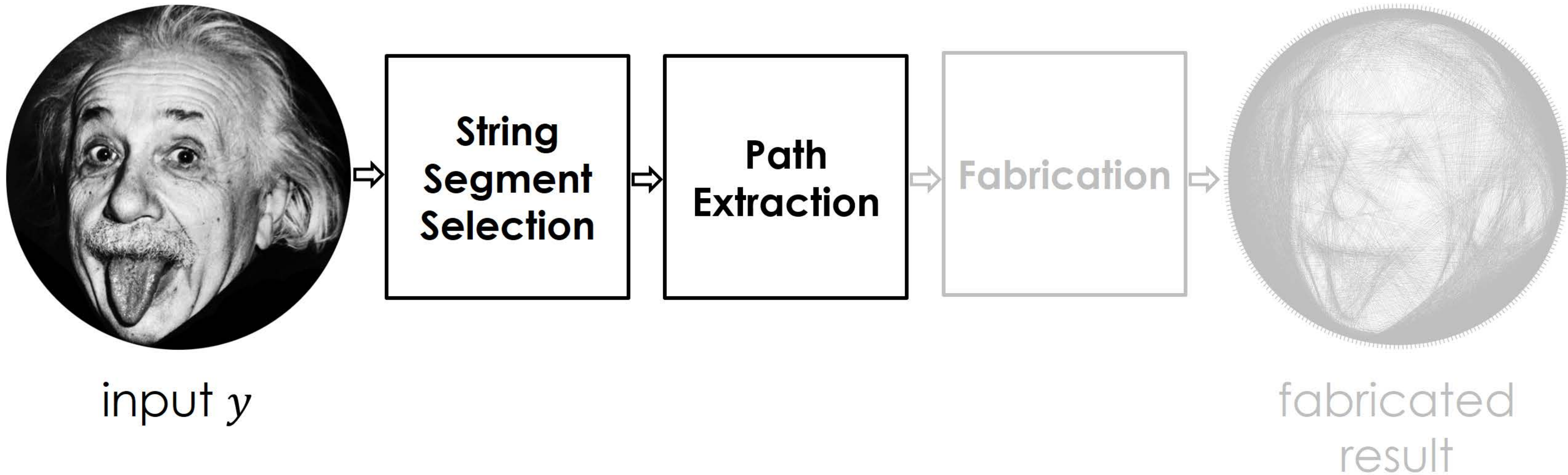
Not managable
with Gurobi

➡ Own Greedy Solver

Iterative Greedy Solver



System Overview

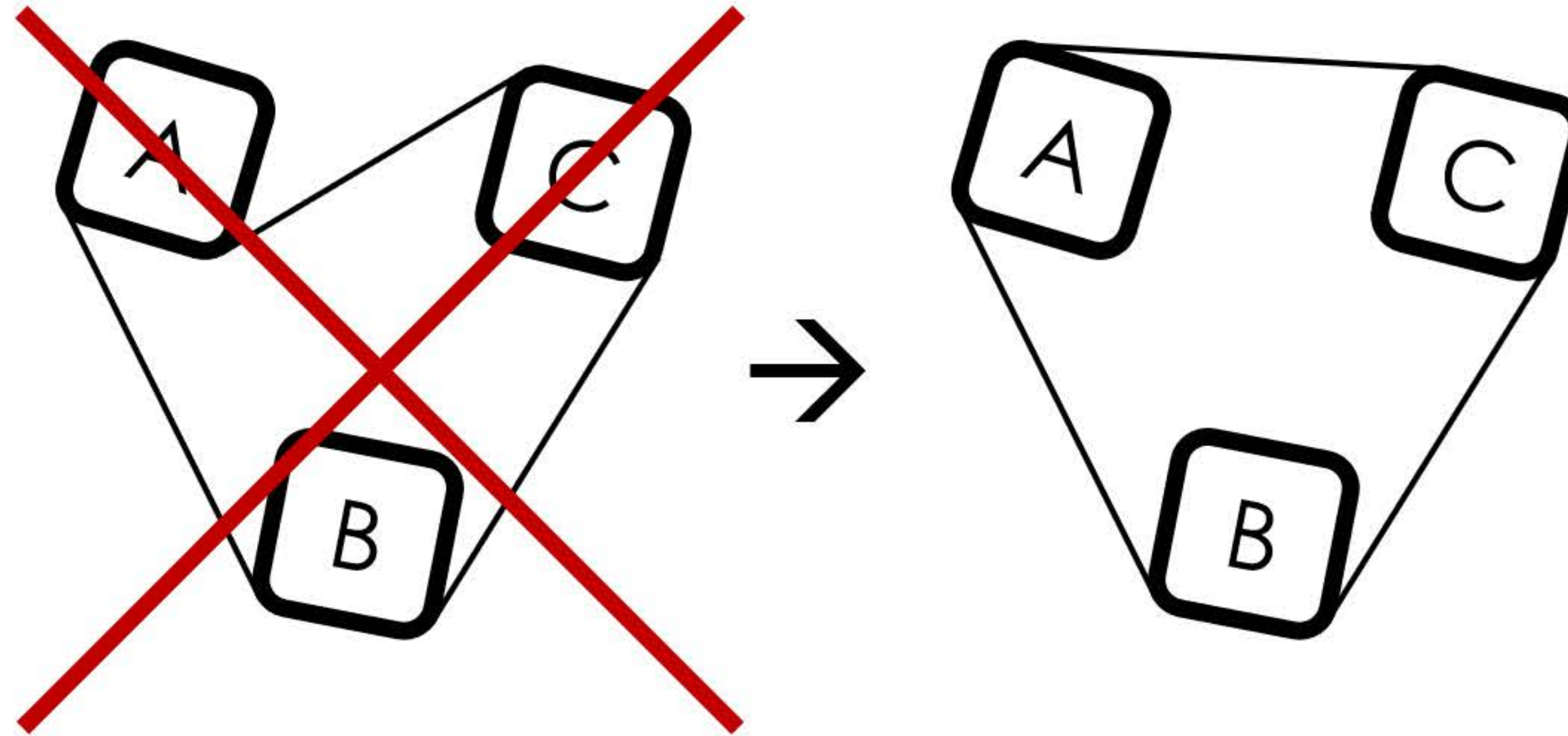


Eulerian Graph Extraction

- Requirement (**R4**)
 - Fabrication from a single piece of thread
 - Graph of string segments has to be (semi-) Eulerian

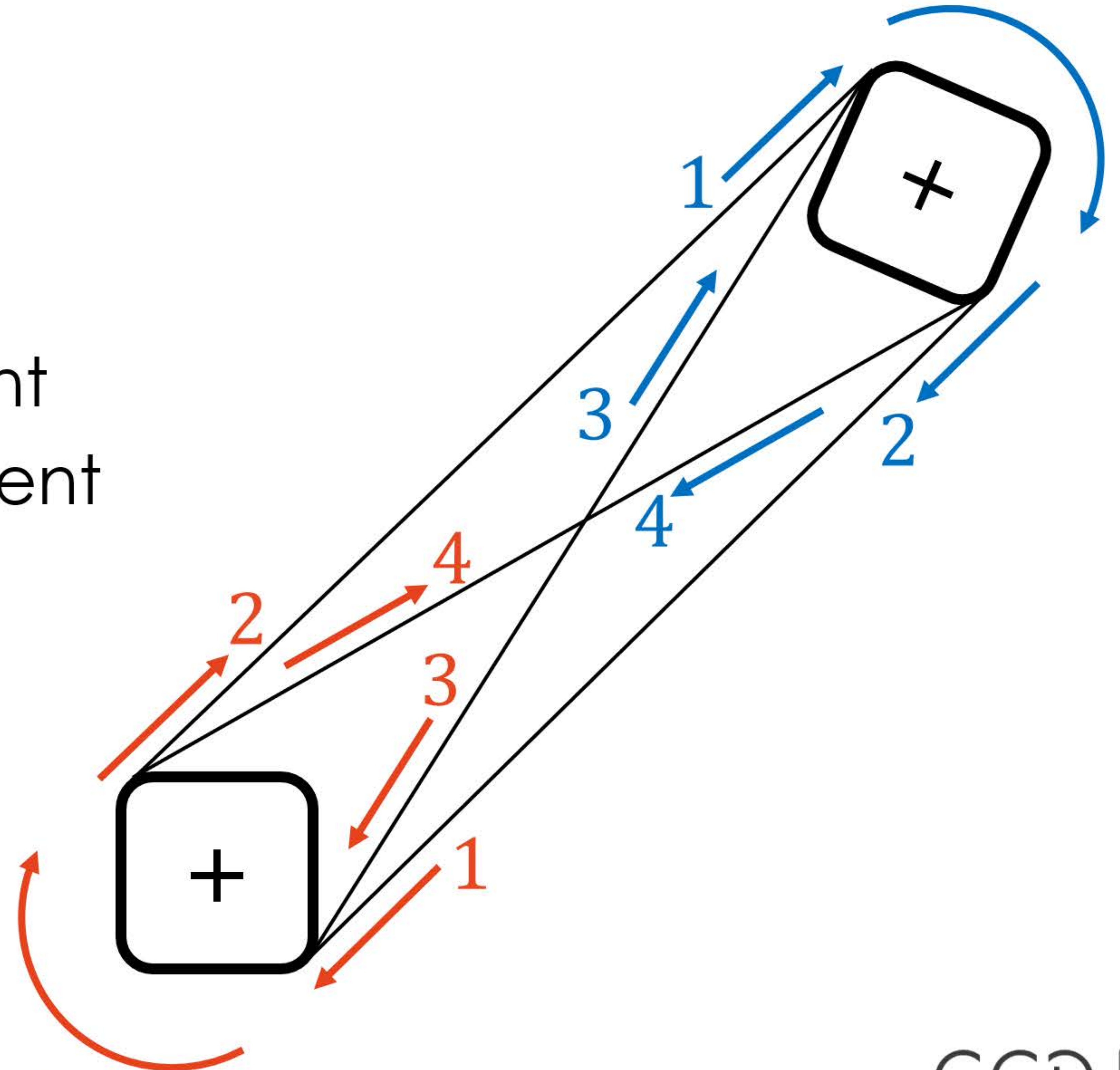
Eulerian Graph Extraction

- Requirement (**R4**)
 - Fabrication from a single piece of thread
 - Graph of string segments has to be (semi-) Eulerian



Eulerian Graph Extraction

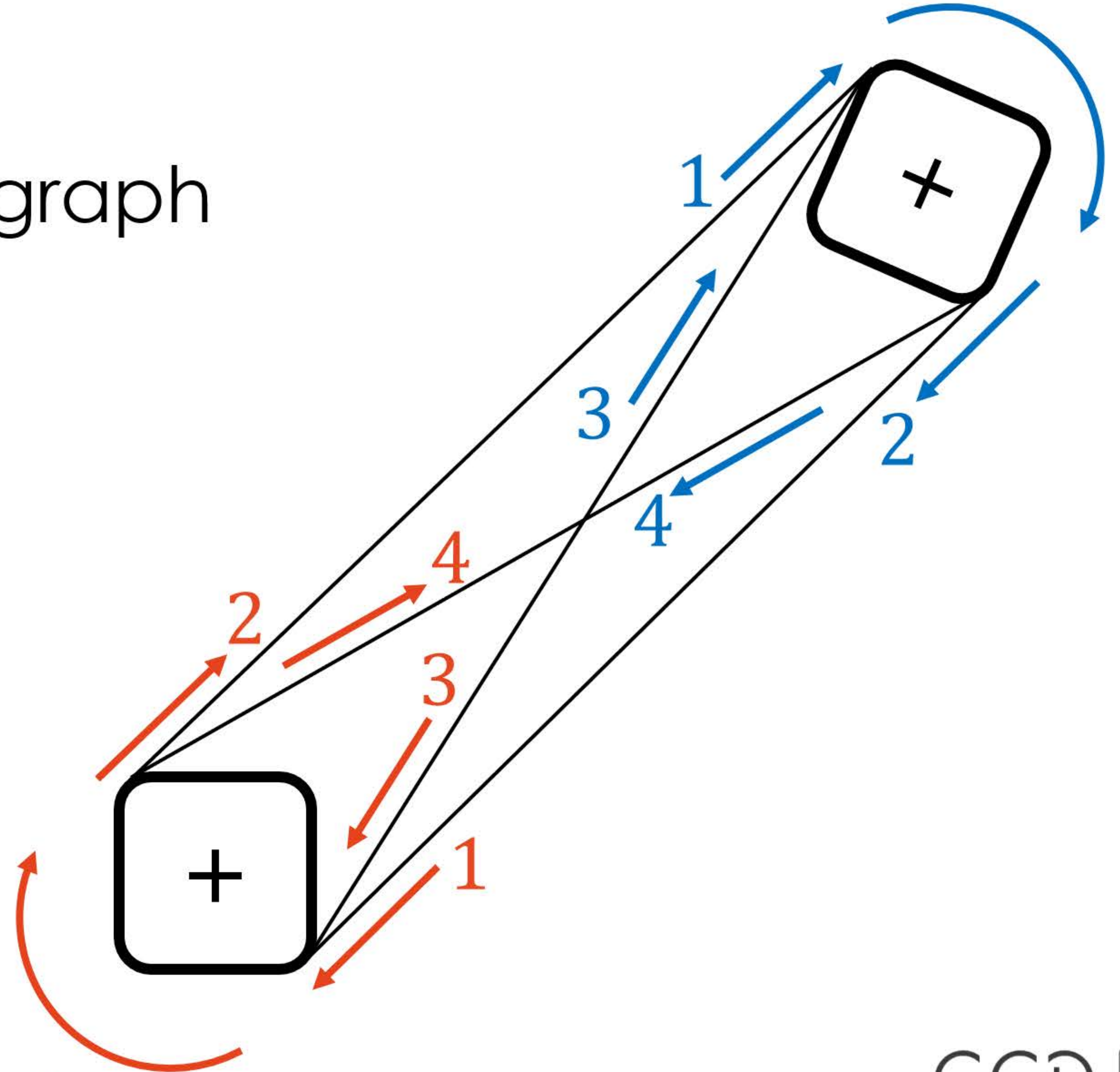
- Edge enumeration
 1. inbound outer tangent
 2. outbound outer tangent
 3. inbound diagonal tangent
 4. diagonal outbound tangent
- Edge grouping
 - 1-3 (inbound)
 - 2-4 (outbound)



Eulerian Graph Extraction

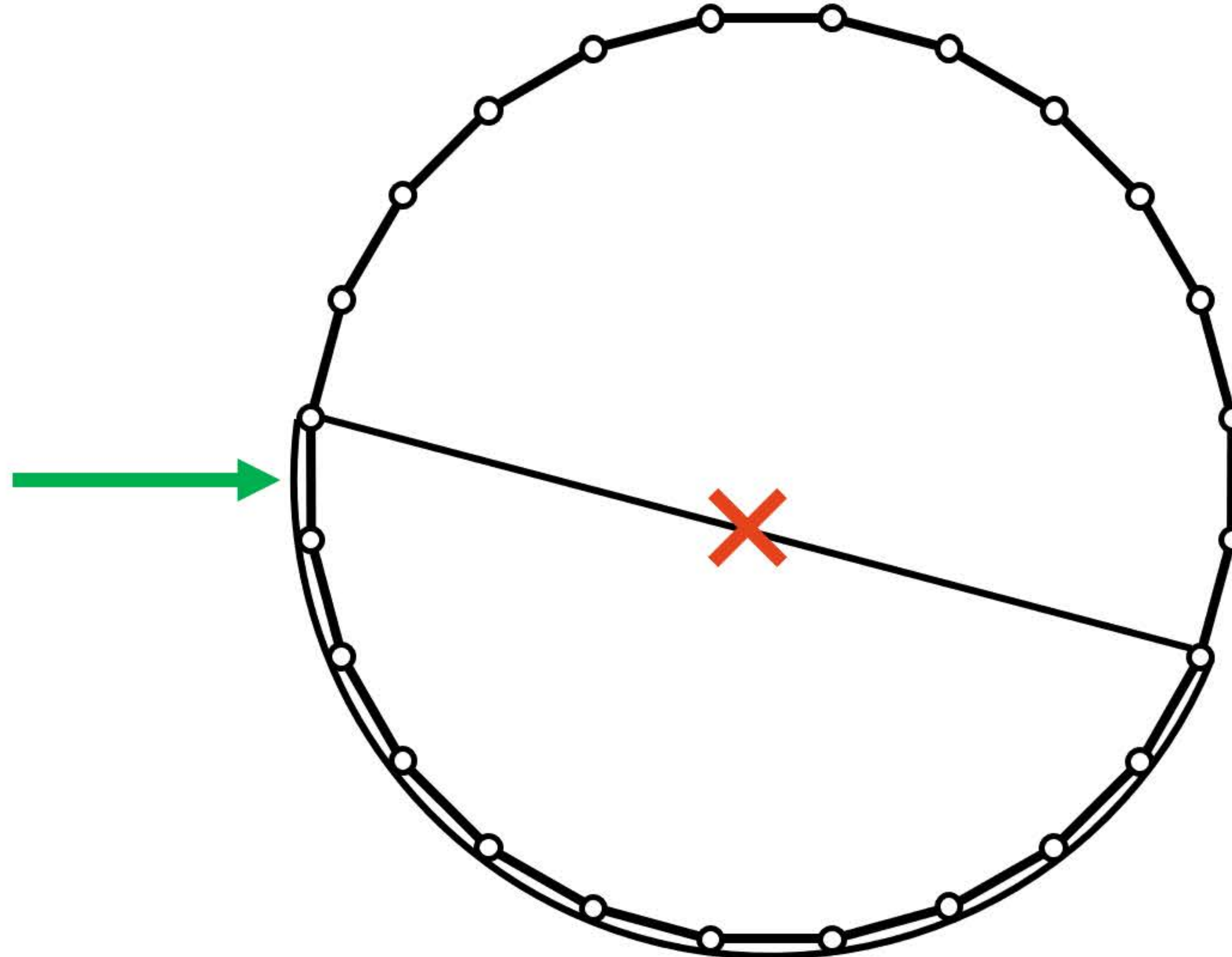
- Key criteria of fabricable graph

$$|G_{13}| = |G_{24}|$$



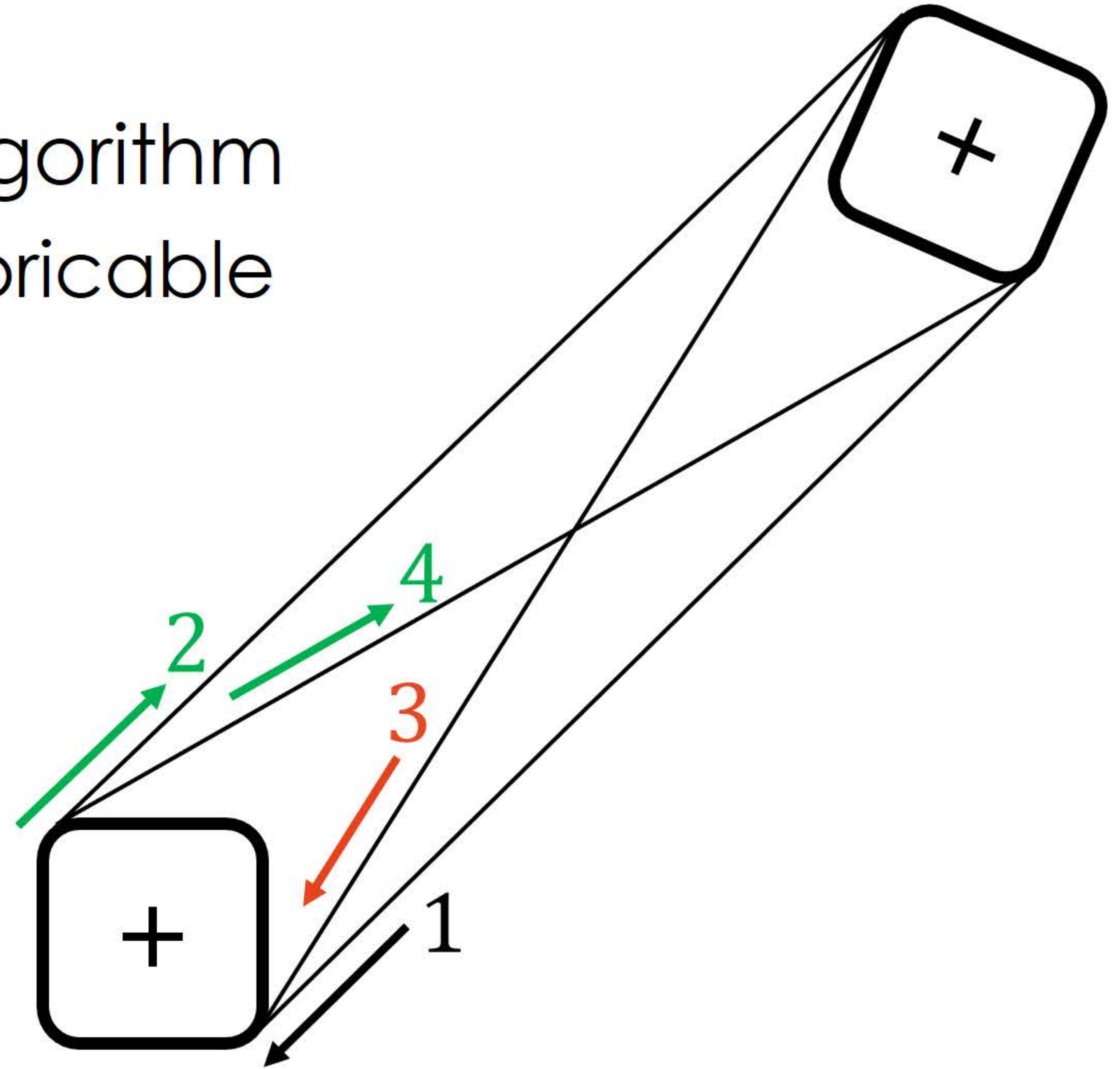
Eulerian Graph Extraction

- String segment selection does not produce graph with equality of groups 1-3 and 2-4 per pin
 - Adding Auxiliary Edges

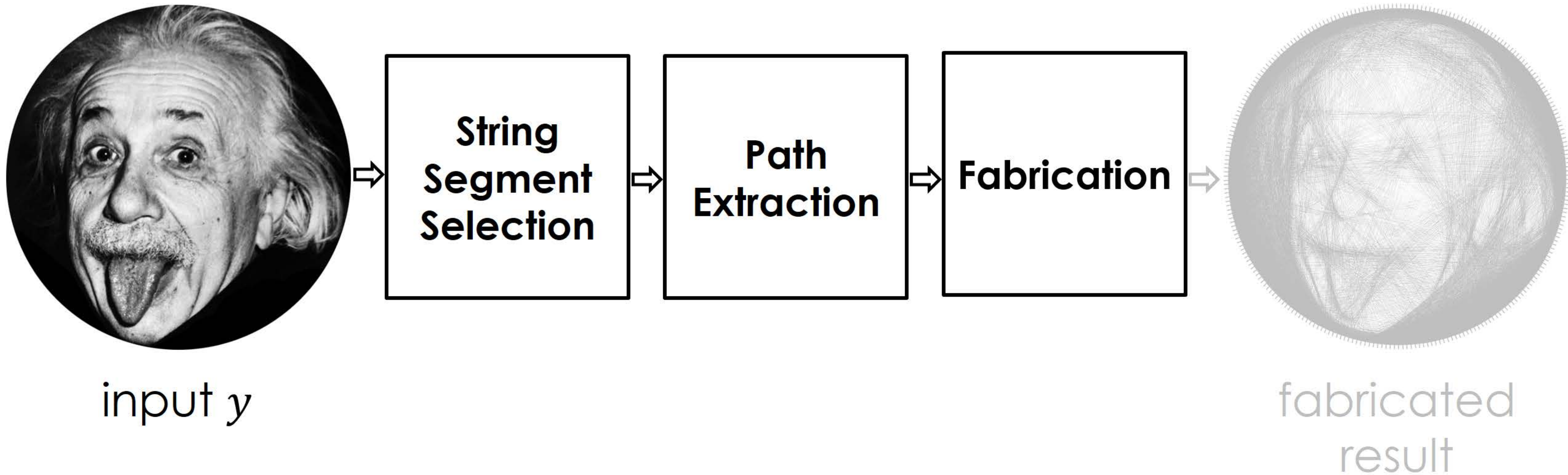


Eulerian Graph Extraction

- Path Identification
 - Adaption of Hierholzer's algorithm
 - all sub-cycles must be fabricable



System Overview

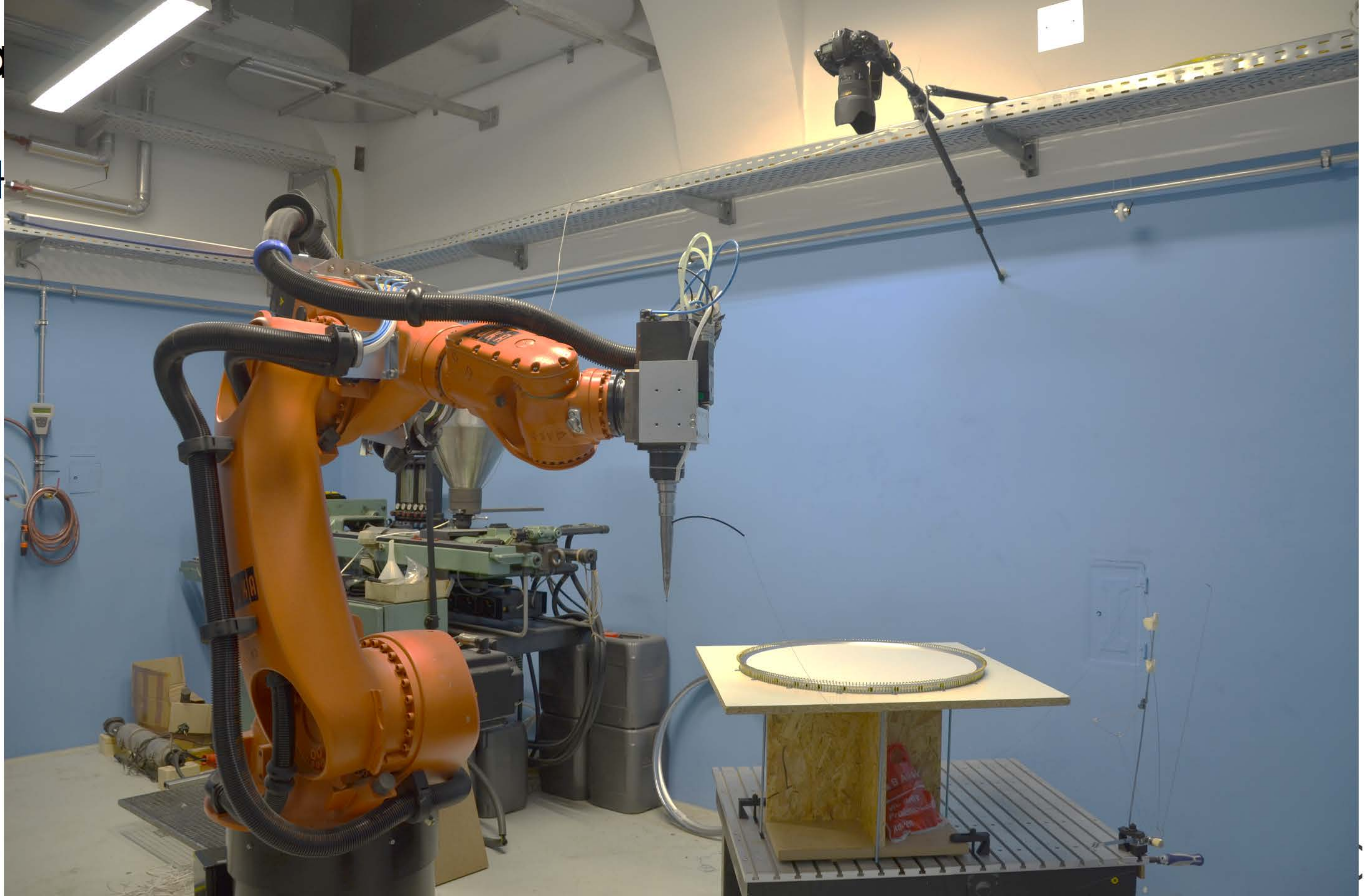


Fabrication

- Hardware Setup

Fo

- F



Fabrication

- Hardware Setup



Industrial Robot
(KUKA KR 60HA)

For

- H

Ir

(



Fabrication

- Hardware Setup



Industrial Robot
(KUKA KR 60HA)



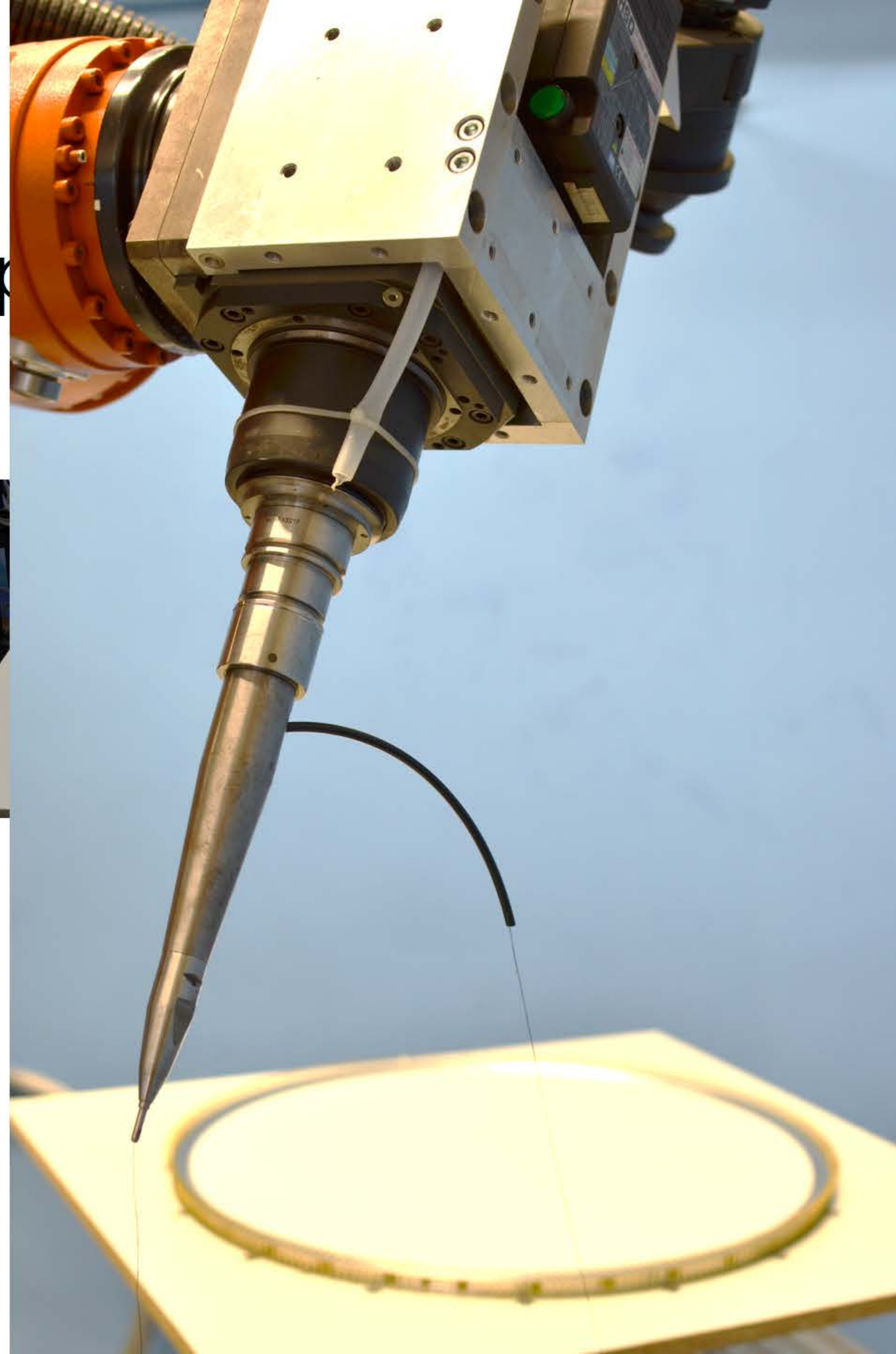
Aluminum
Bicycle Rim
(256 Pins)

Fabrication

- Hardware Setup



Industrial Robot
(KUKA KR 60HA)



Fabrication

- Hardware Setup



Industrial Robot
(KUKA KR 60HA)



Aluminum
Bicycle Rim
(256 Pins)



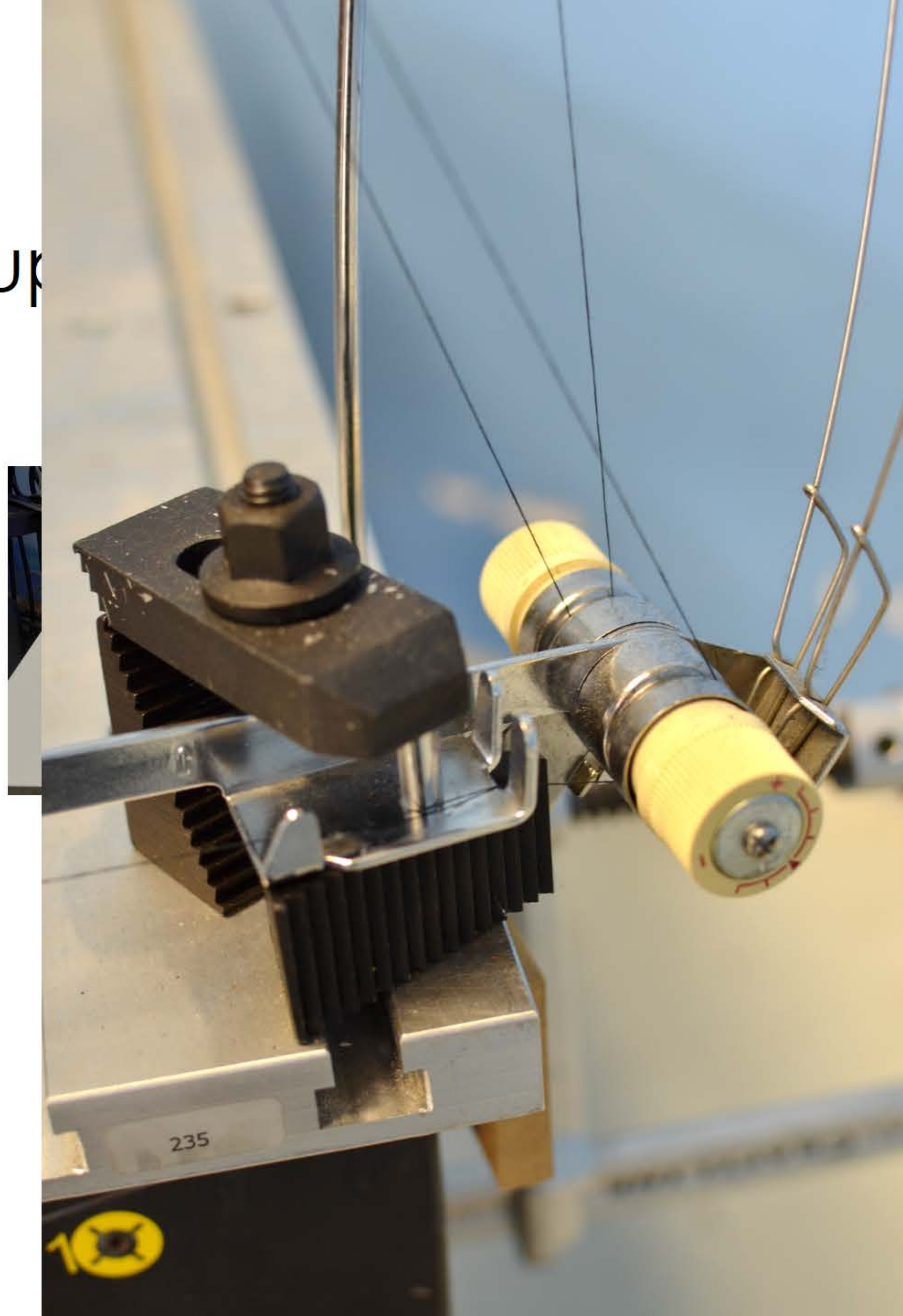
Custom Made
Winding Tool

Fabrication

- Hardware Setup



Industrial Robot
(KUKA KR 60HA)



made
ool

Fabrication

- Hardware Setup



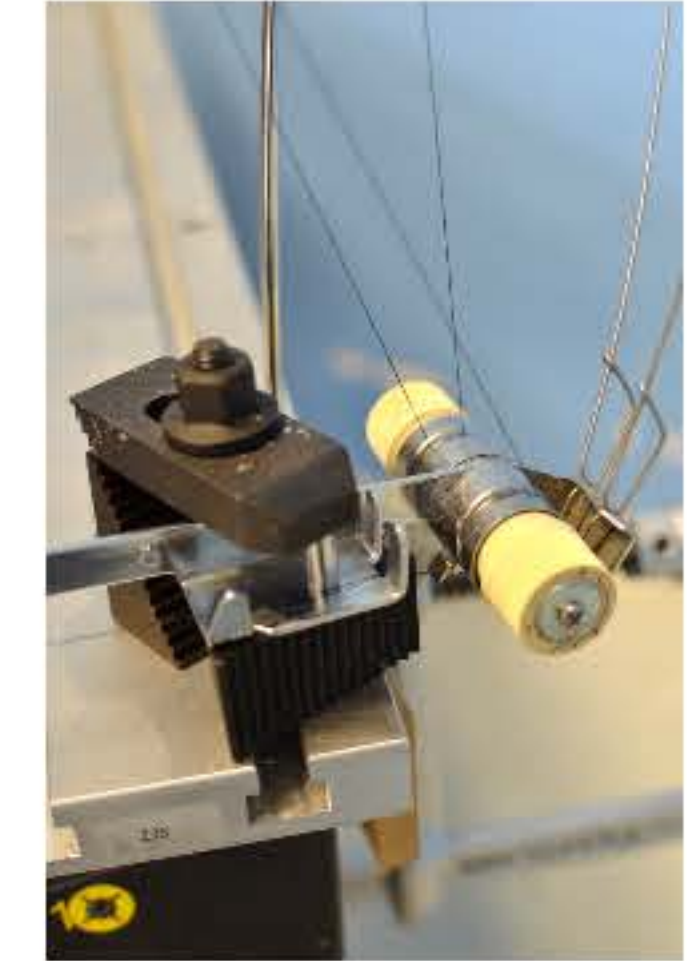
Industrial Robot
(KUKA KR 60HA)



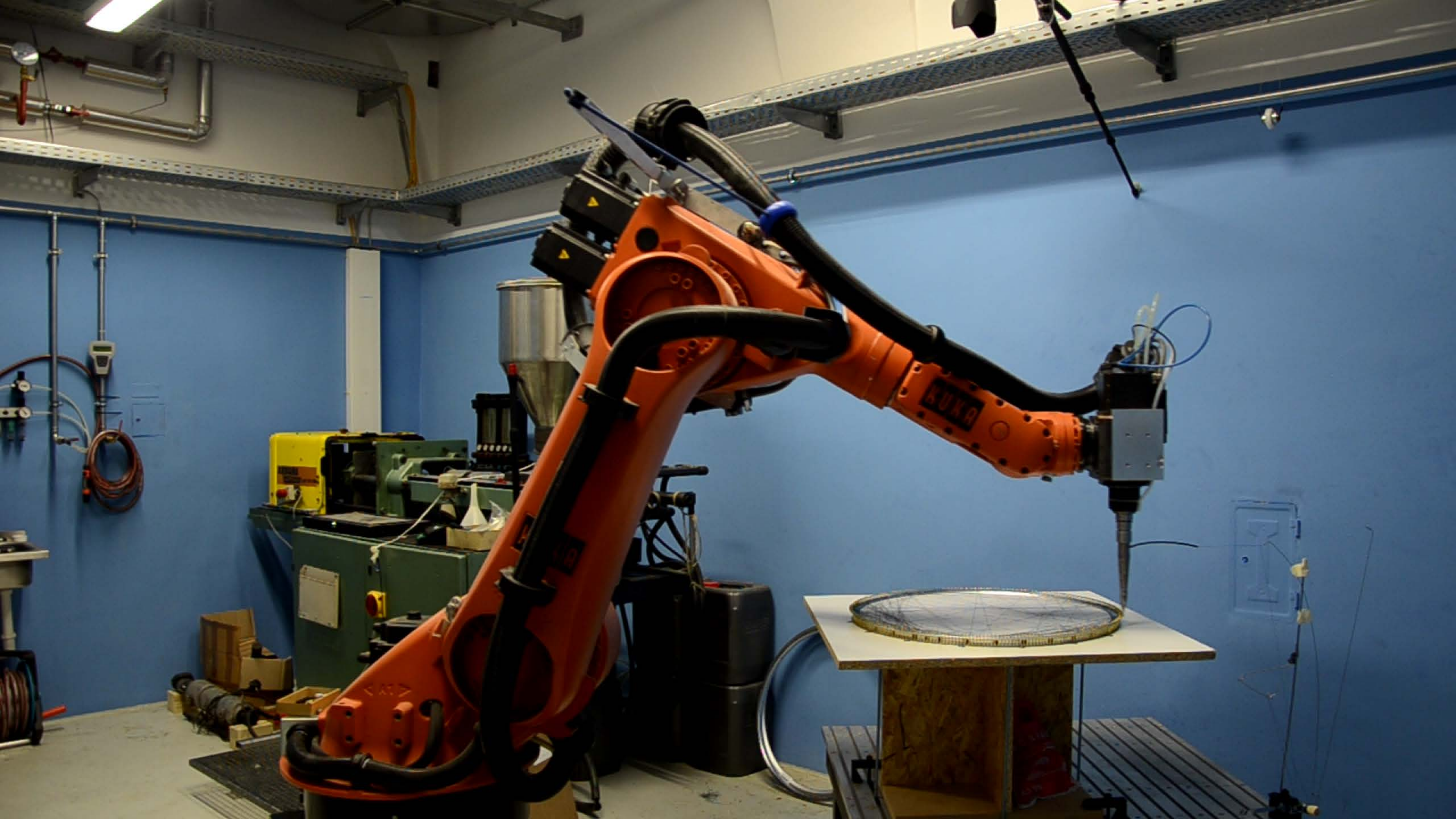
Aluminum
Bicycle Rim
(256 Pins)

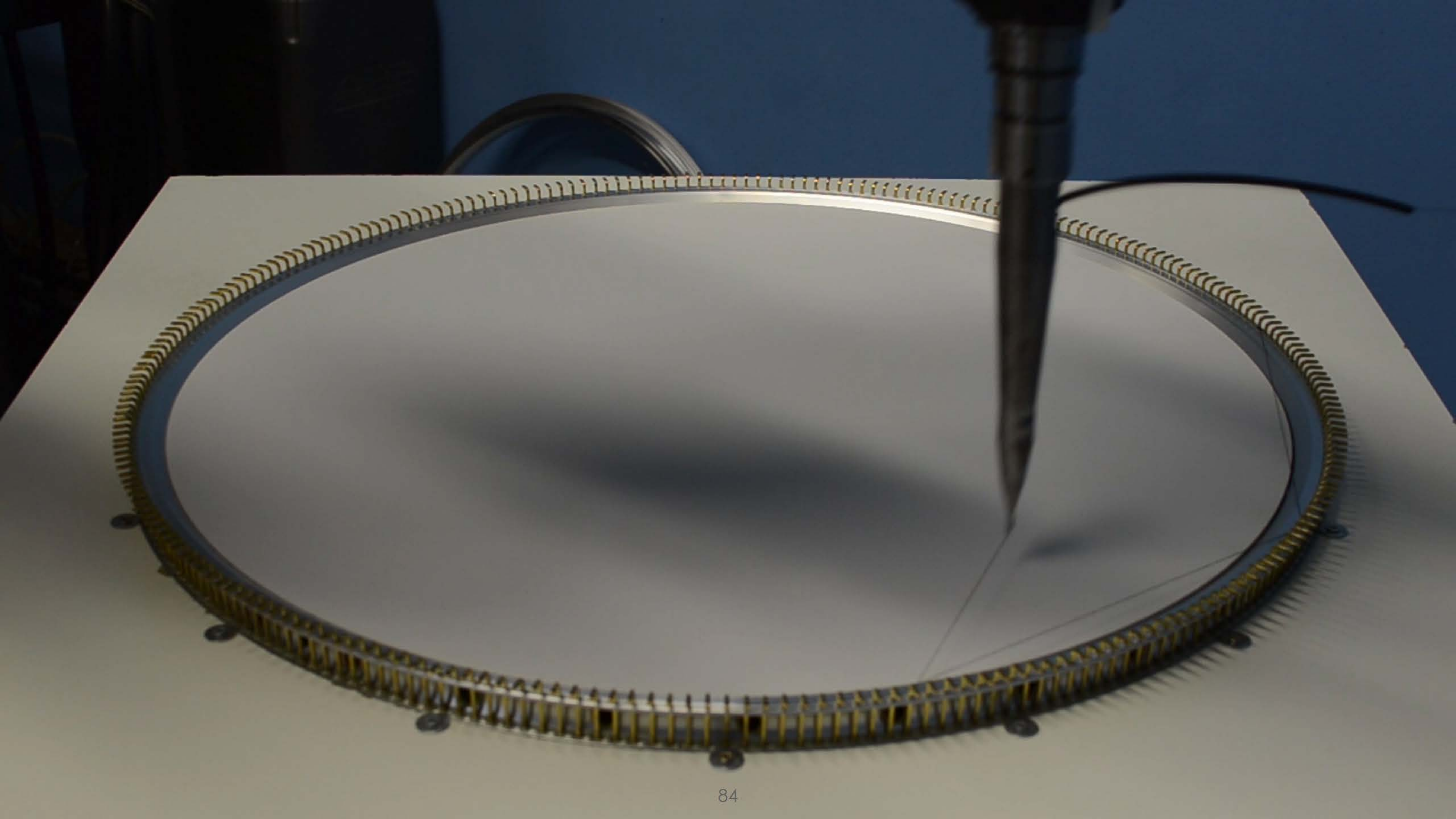


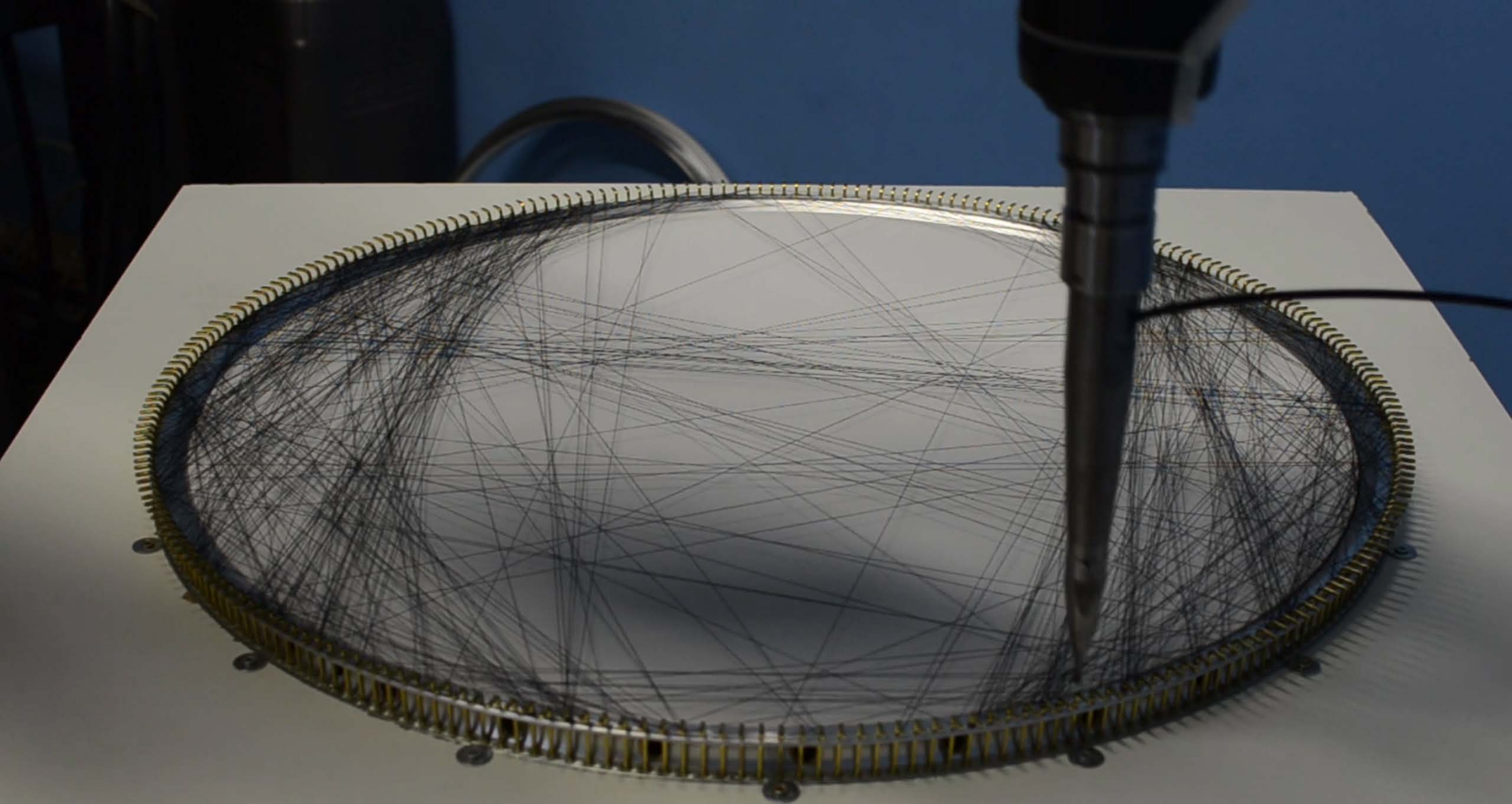
Custom Made
Winding Tool



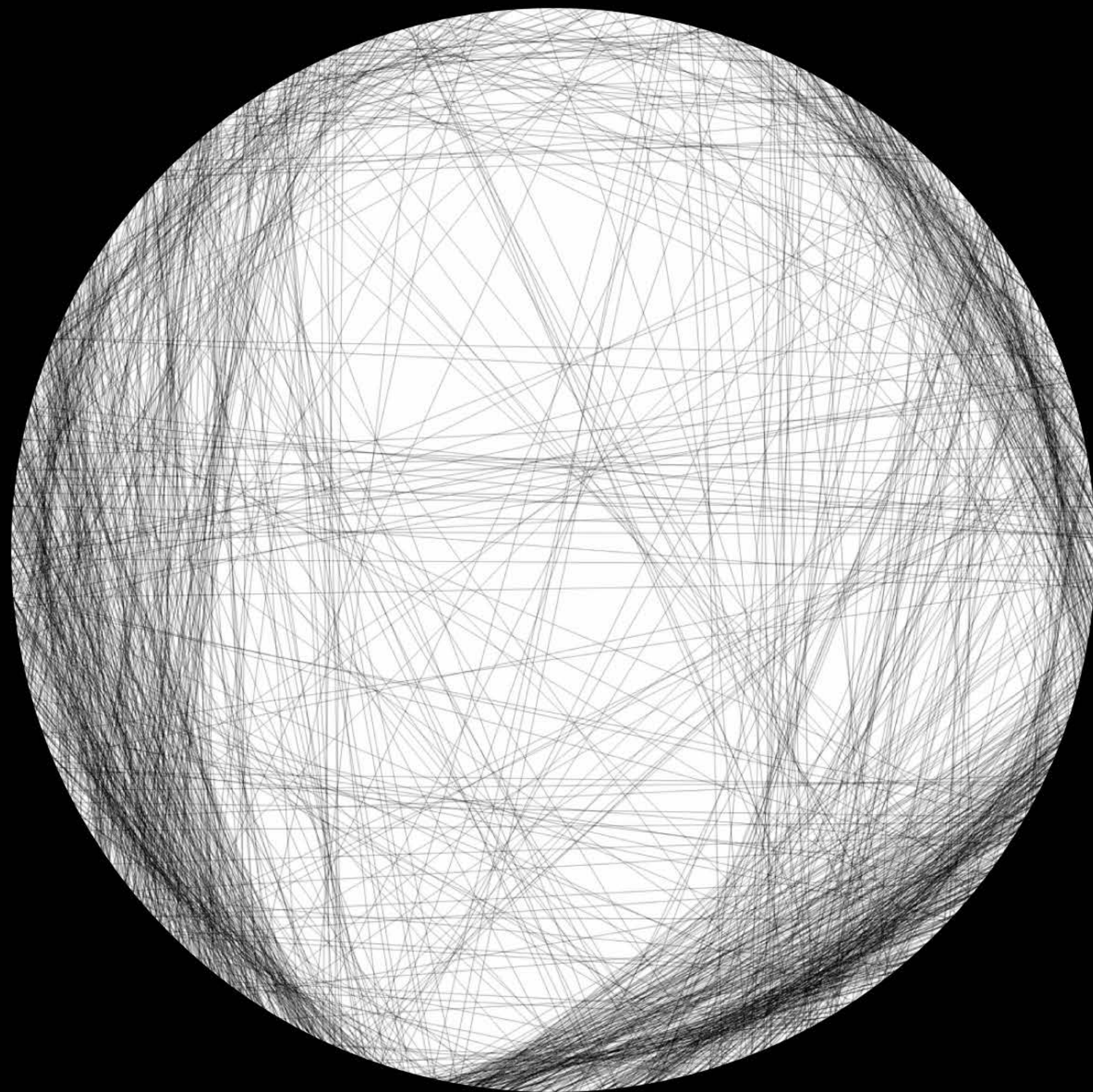
Thread
Tension
Regulator

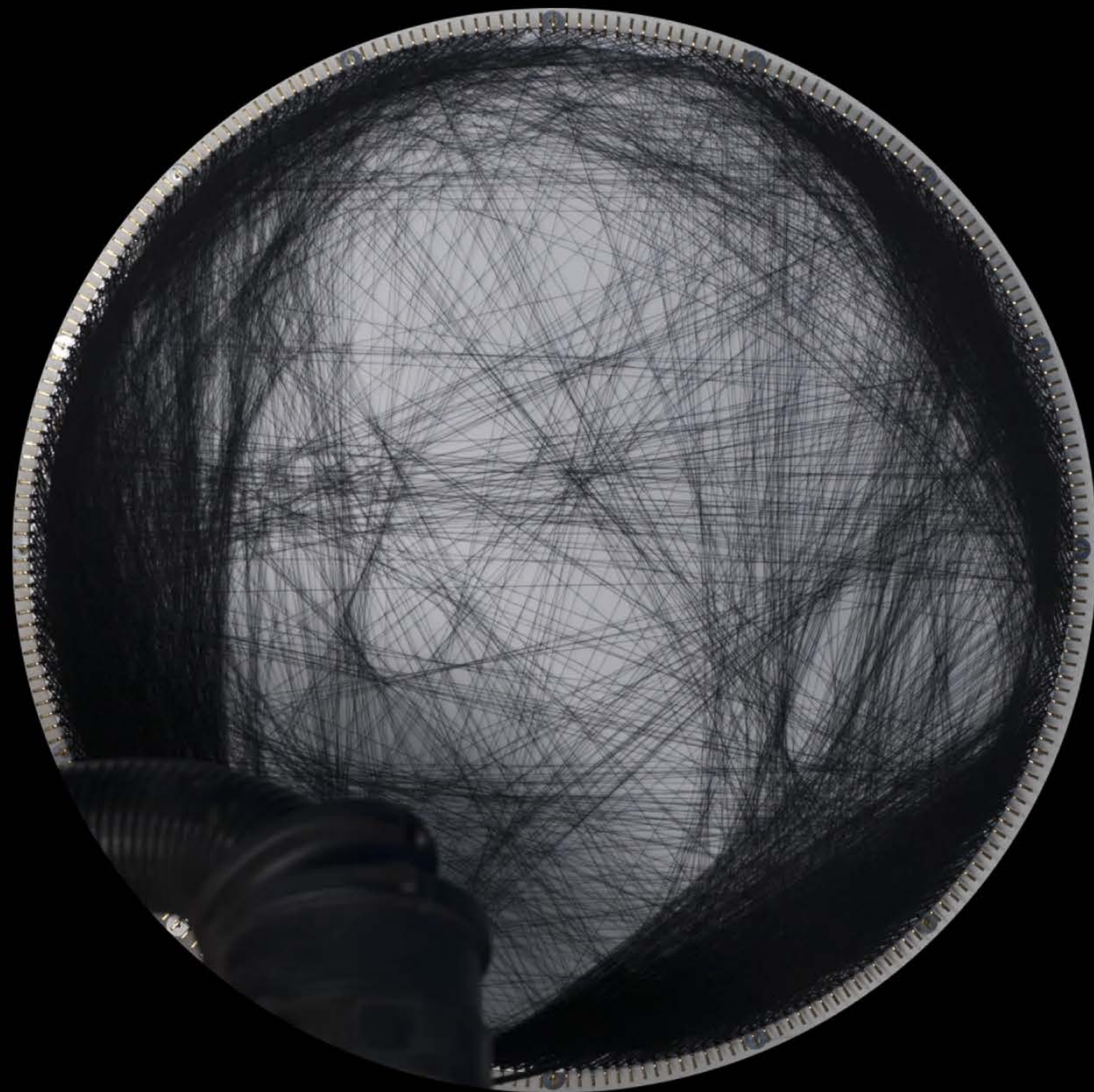
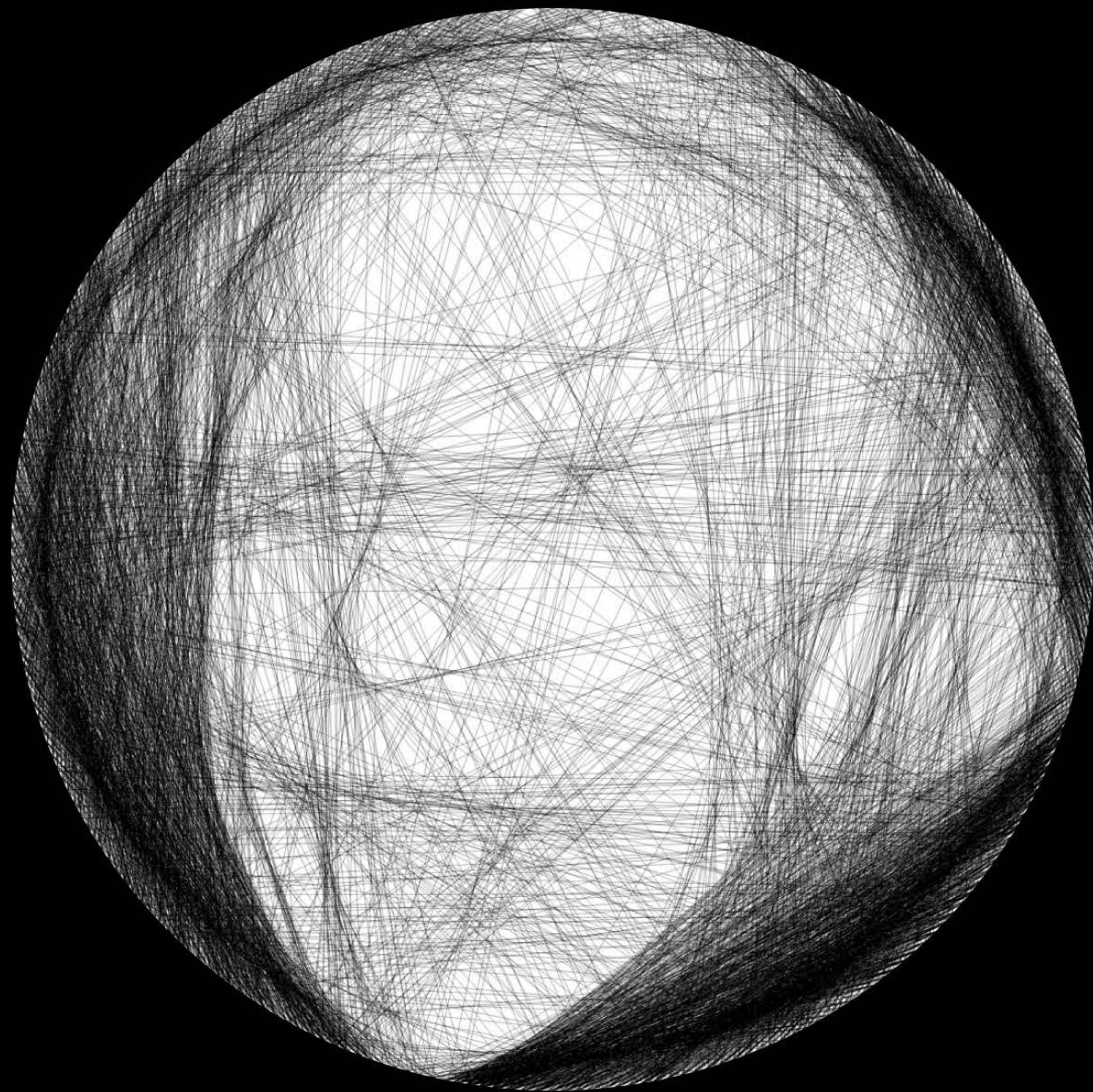


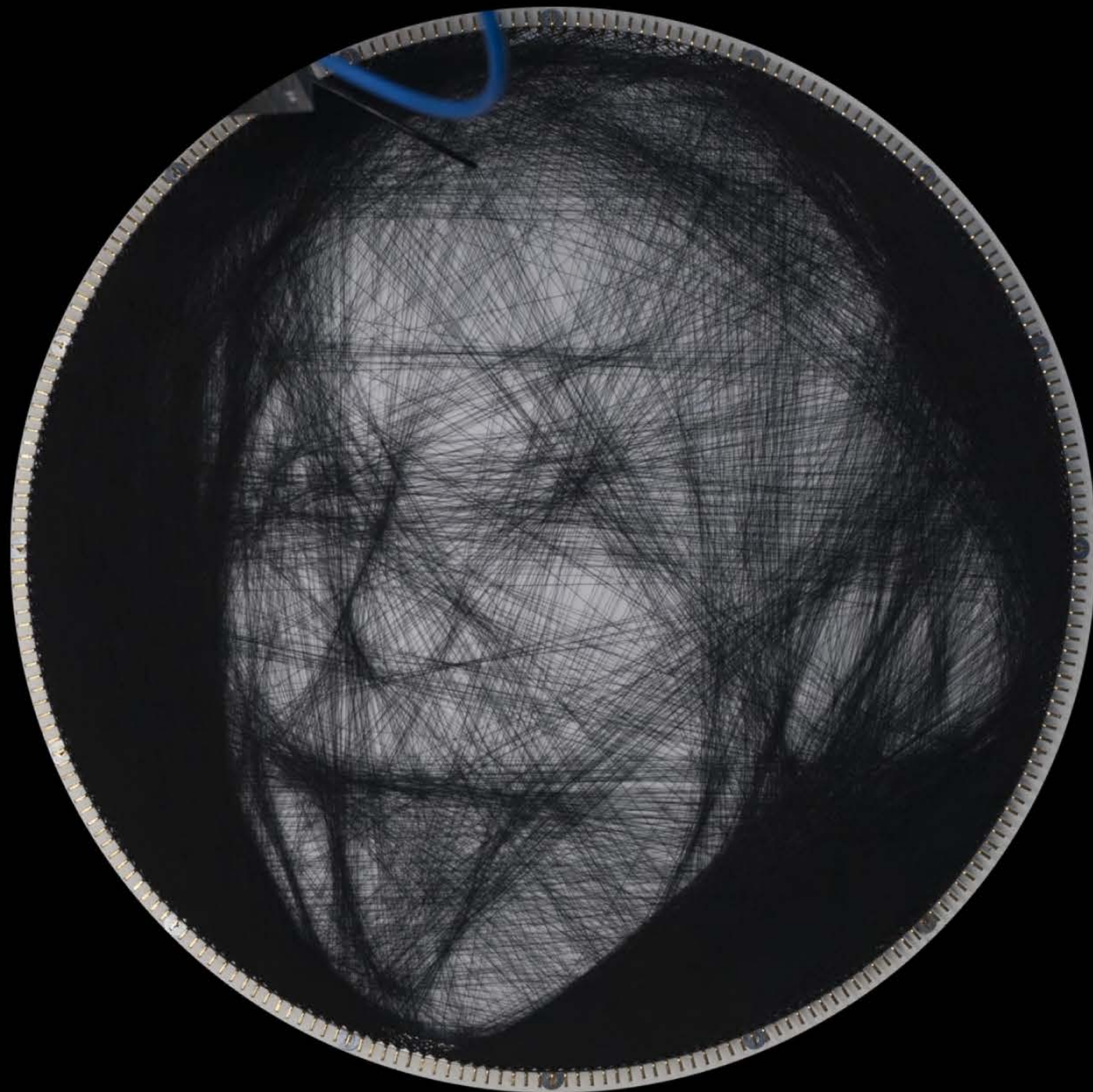
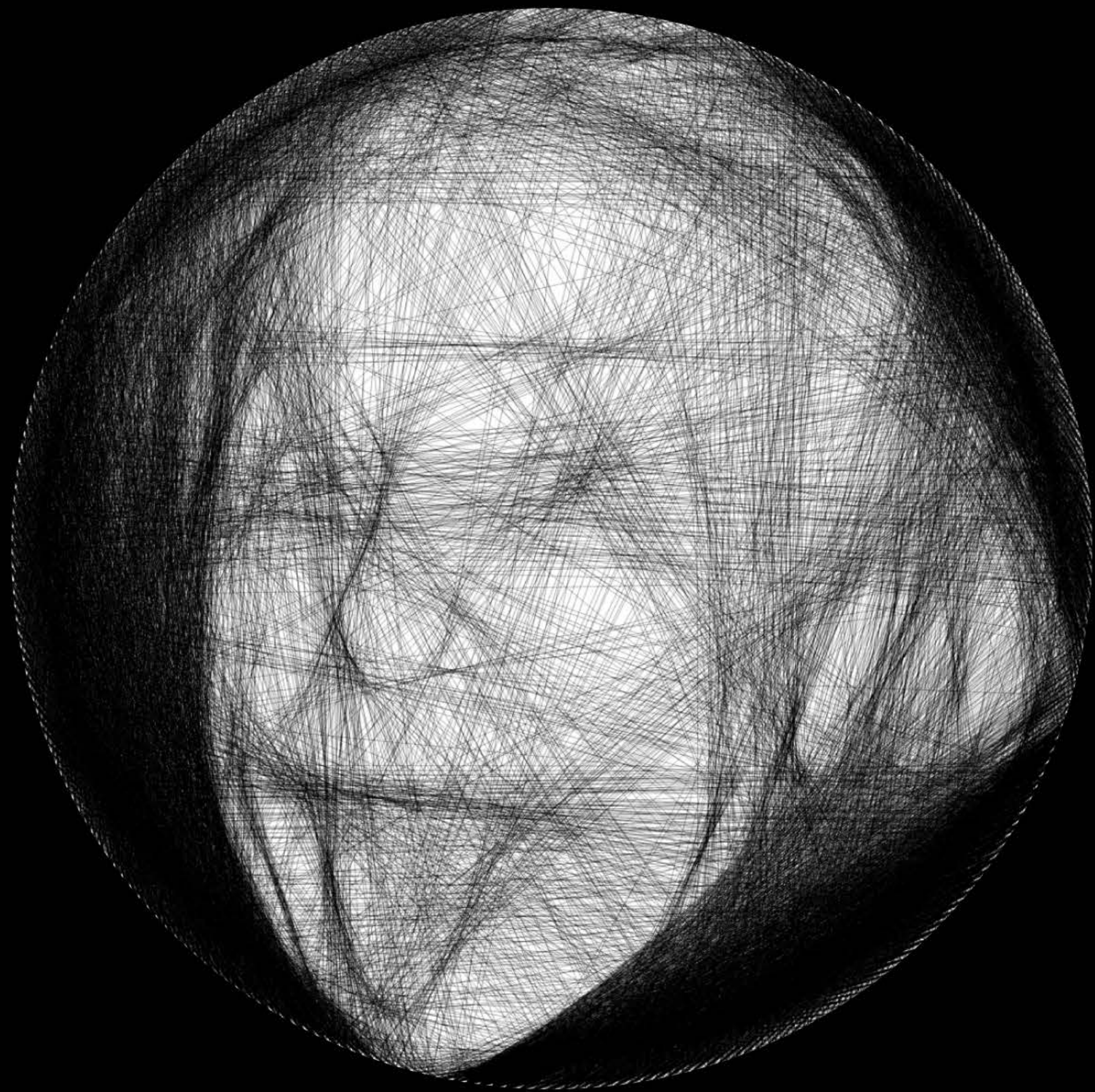




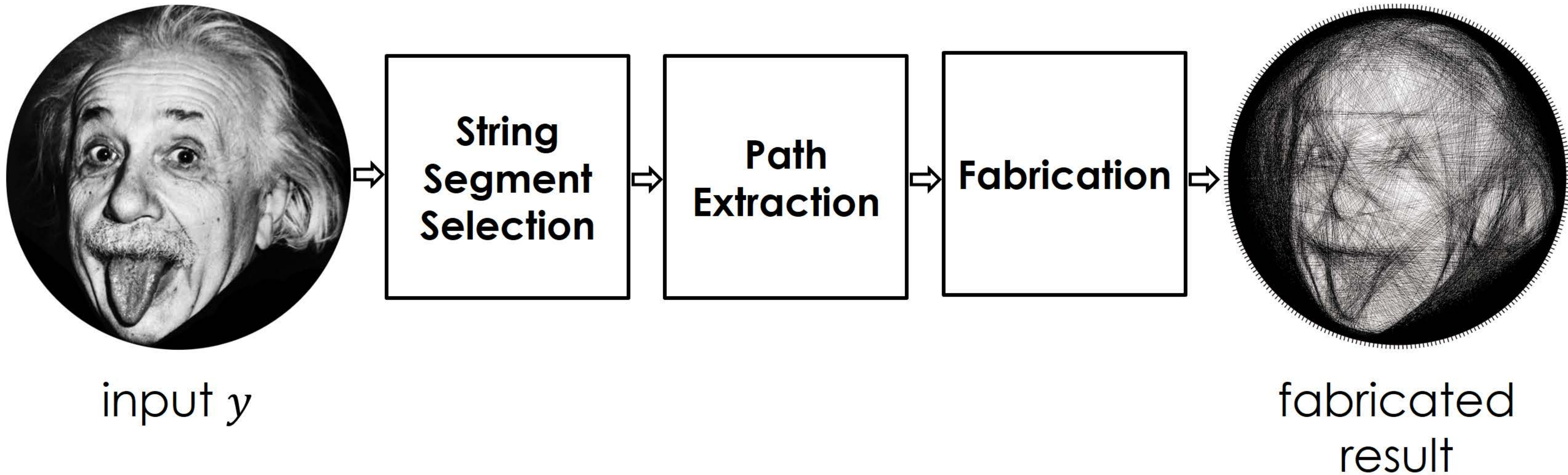




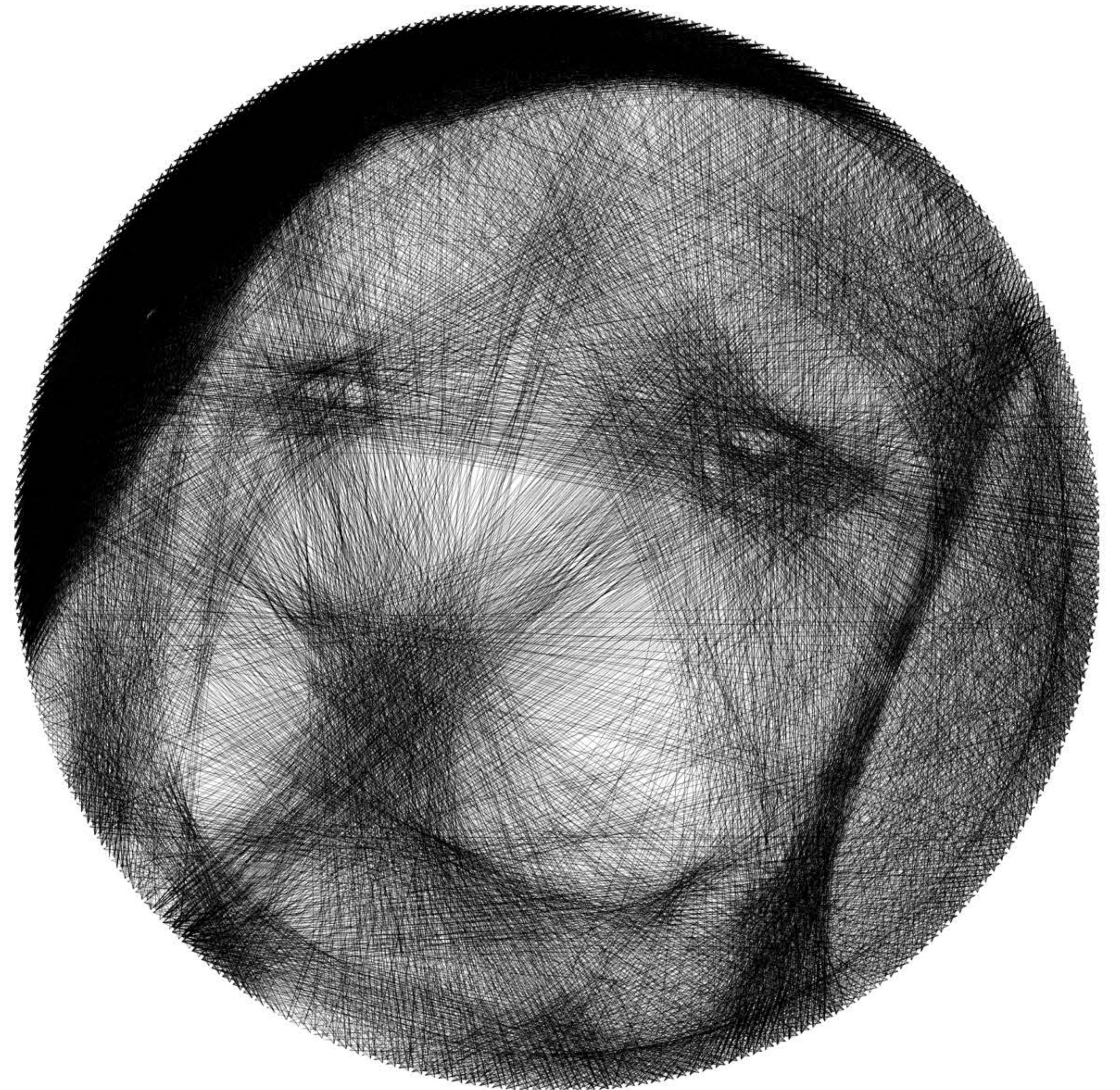




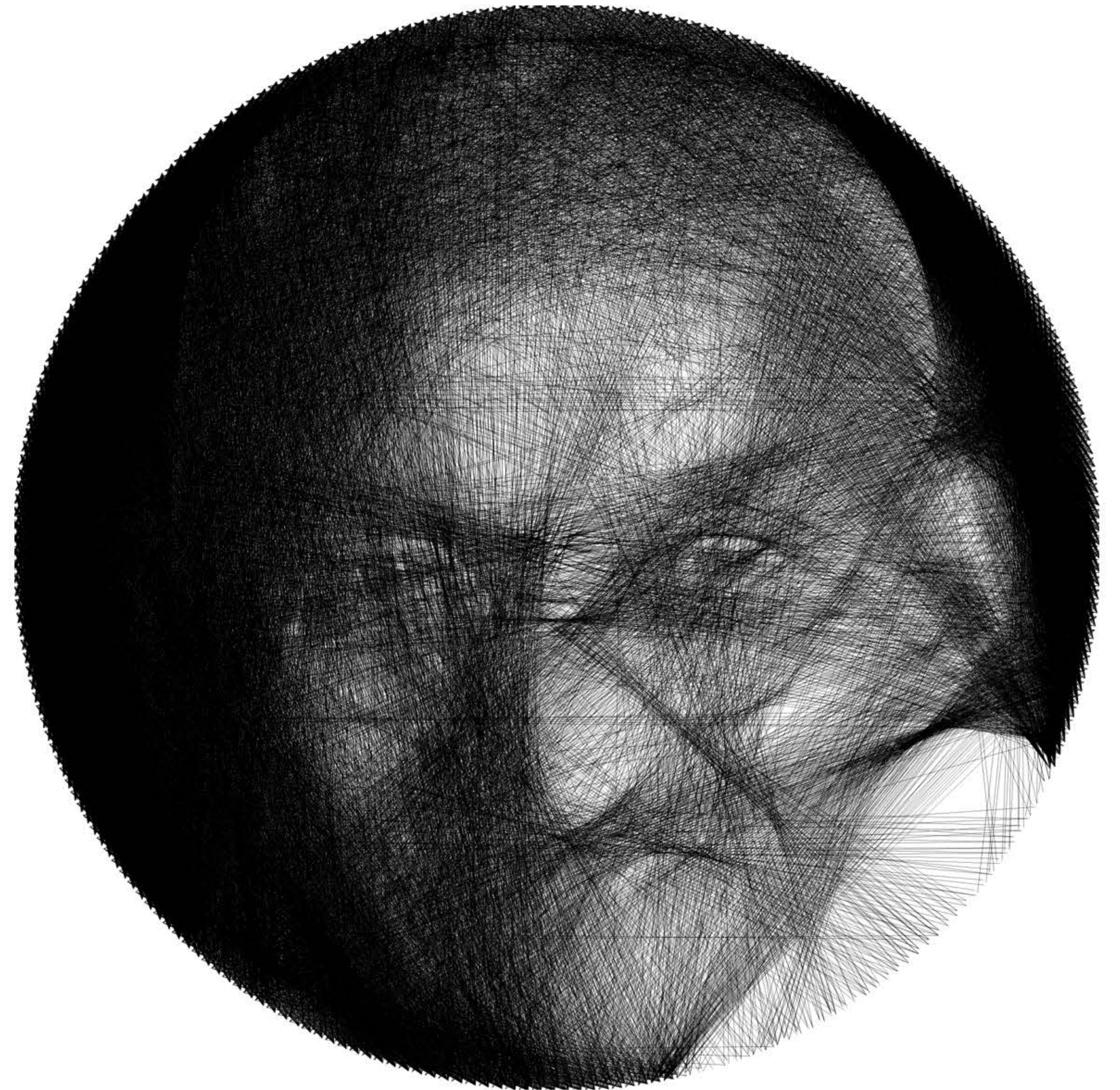
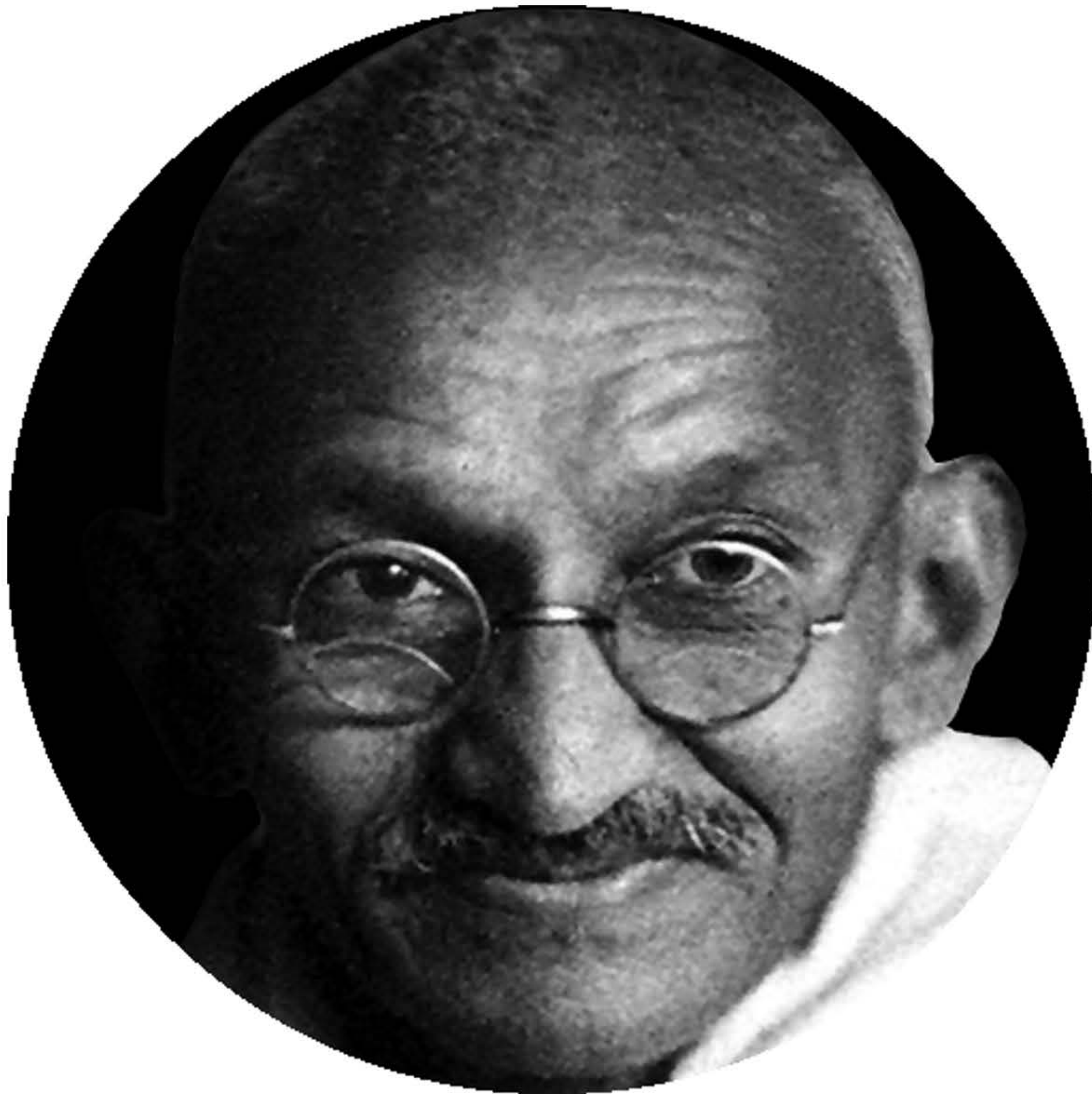
System Overview



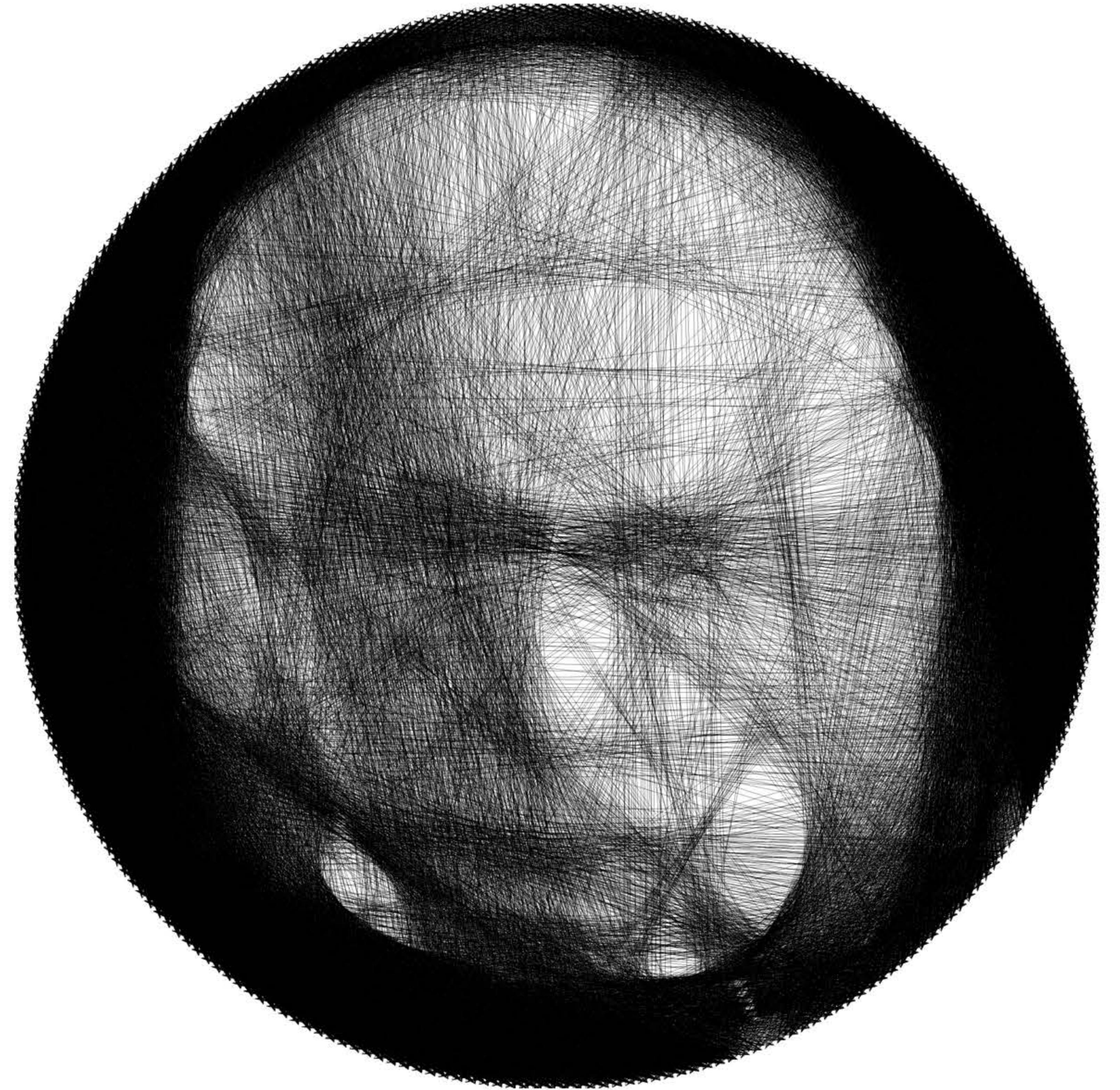
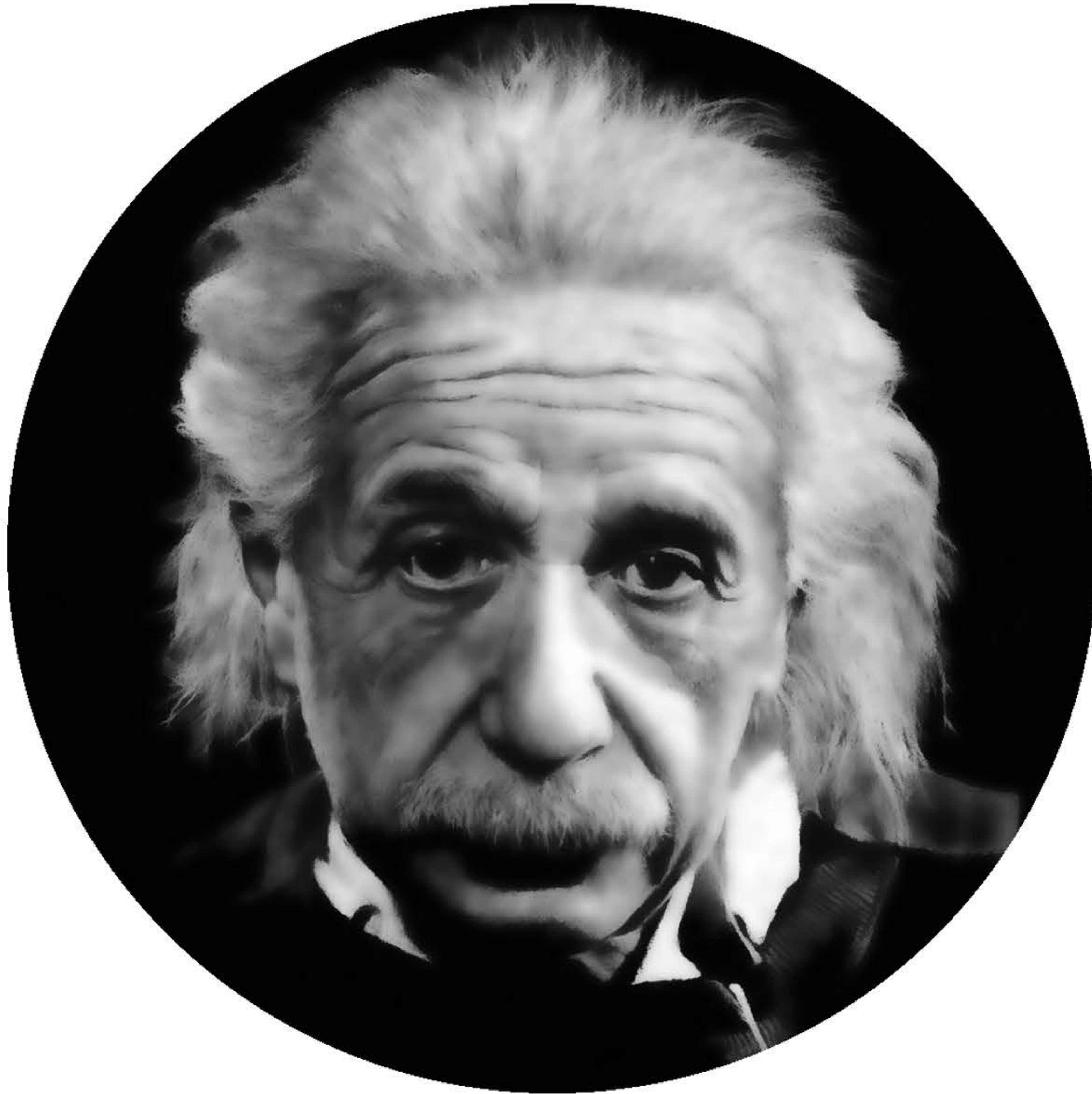
Computed Results



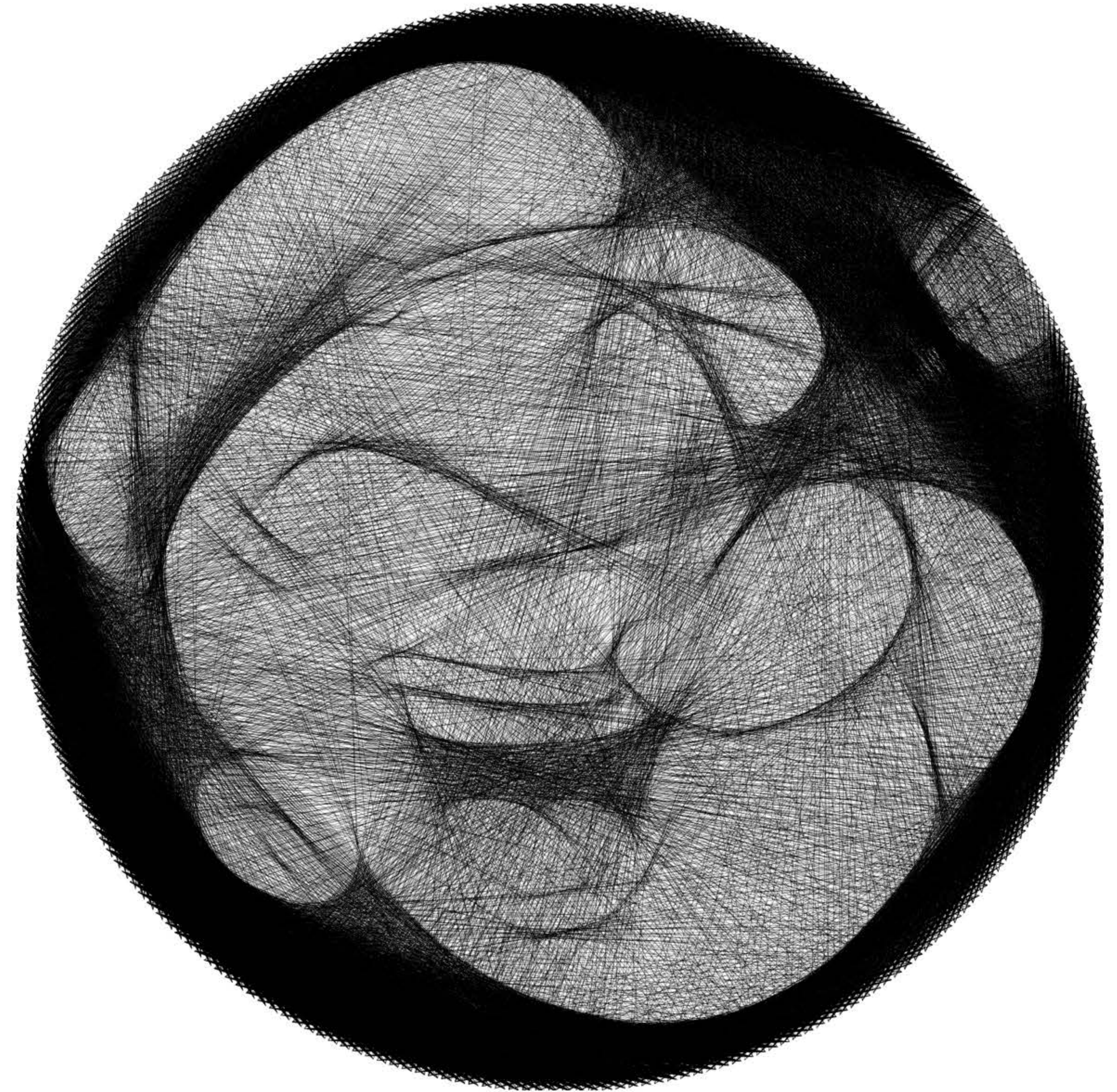
Computed Results



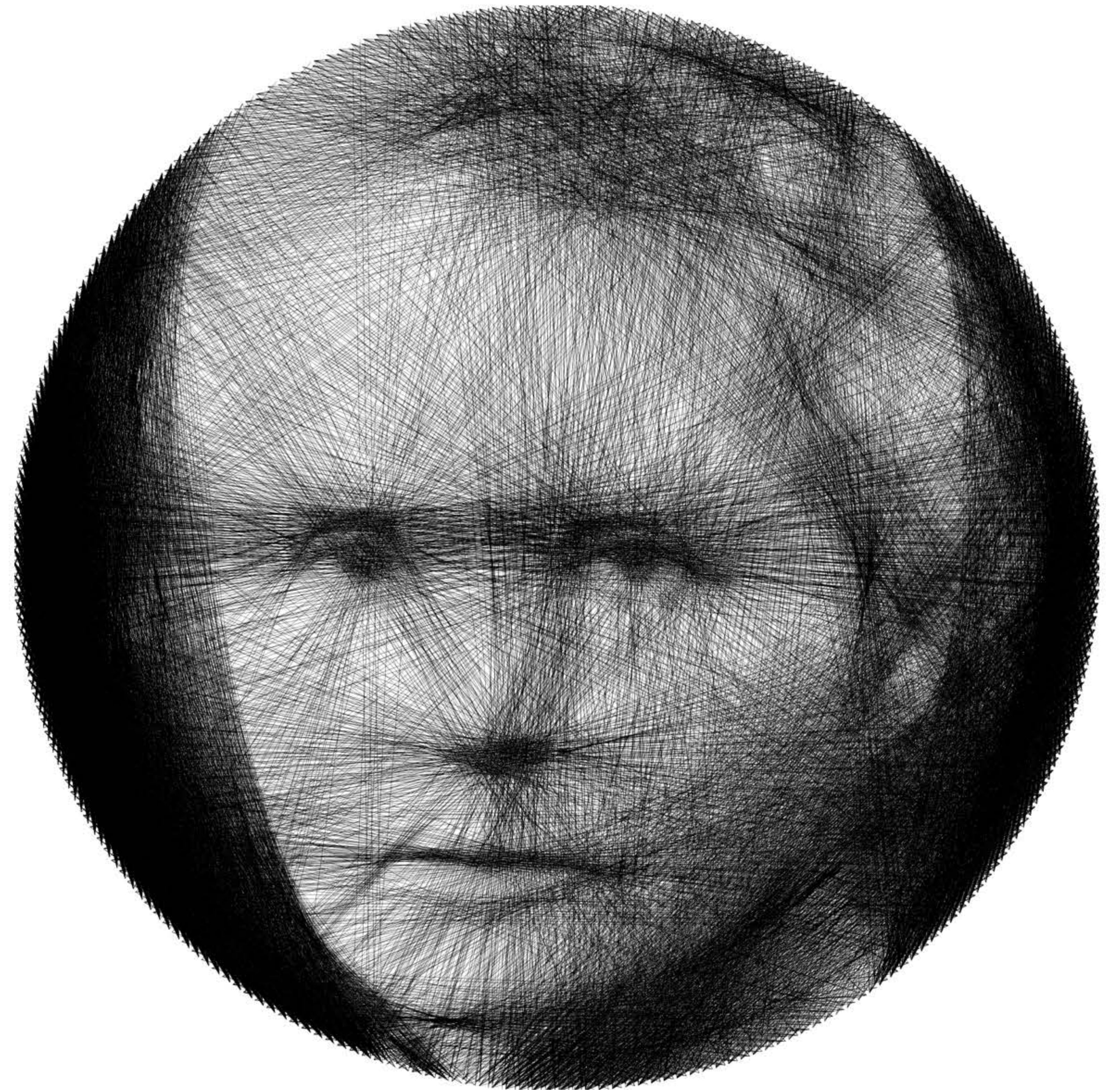
Computed Results



Computed Results



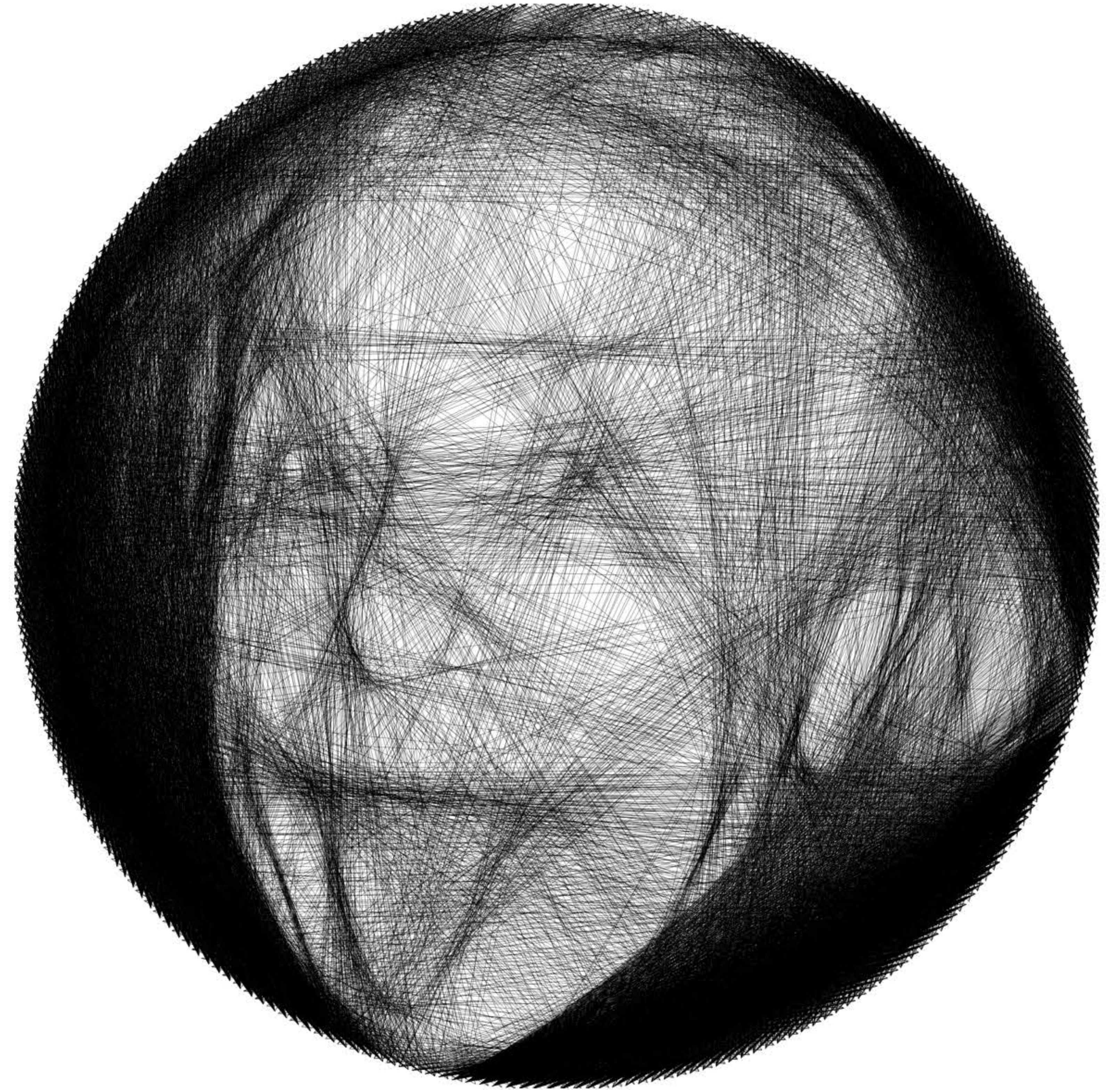
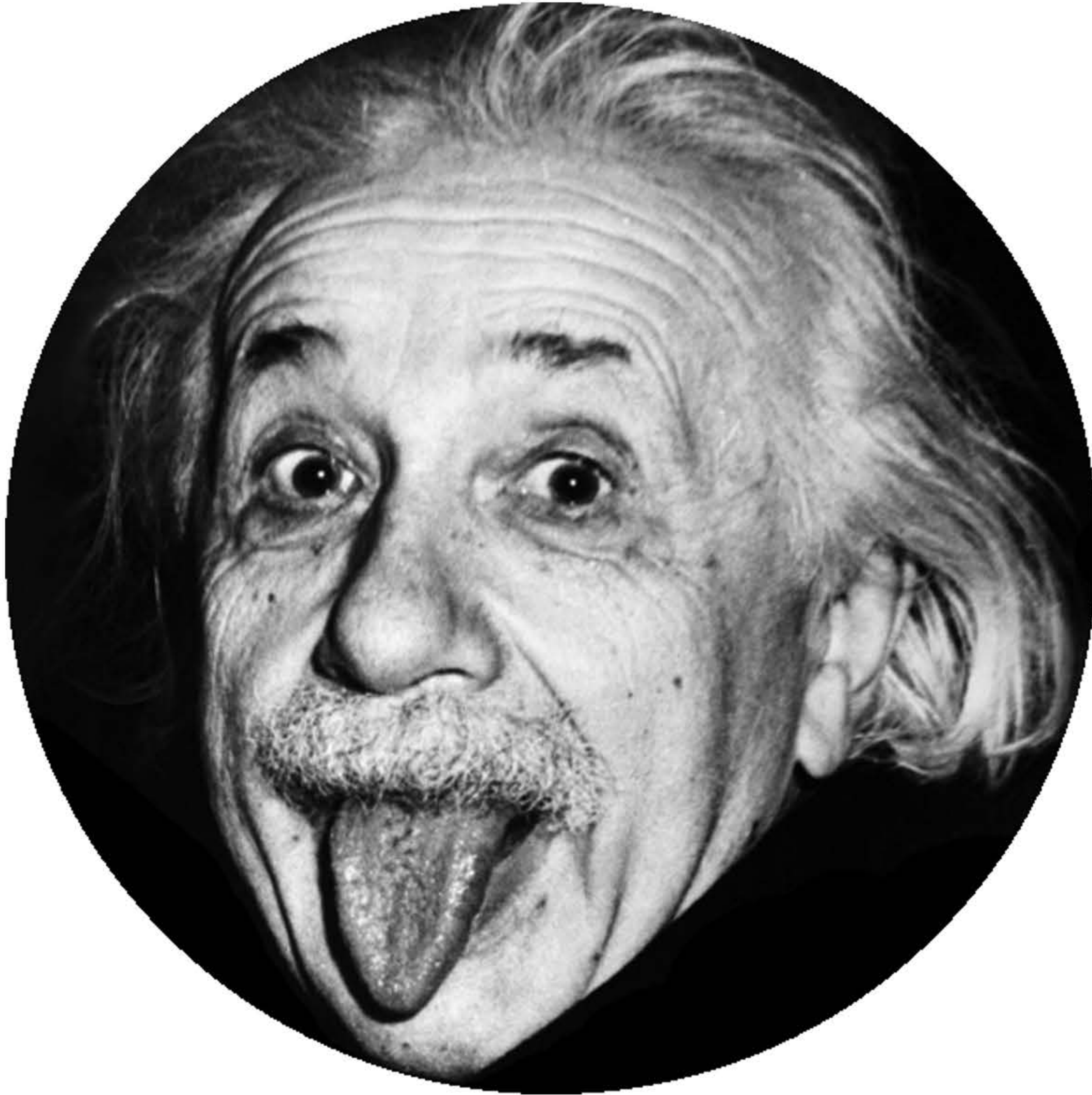
Computed Results



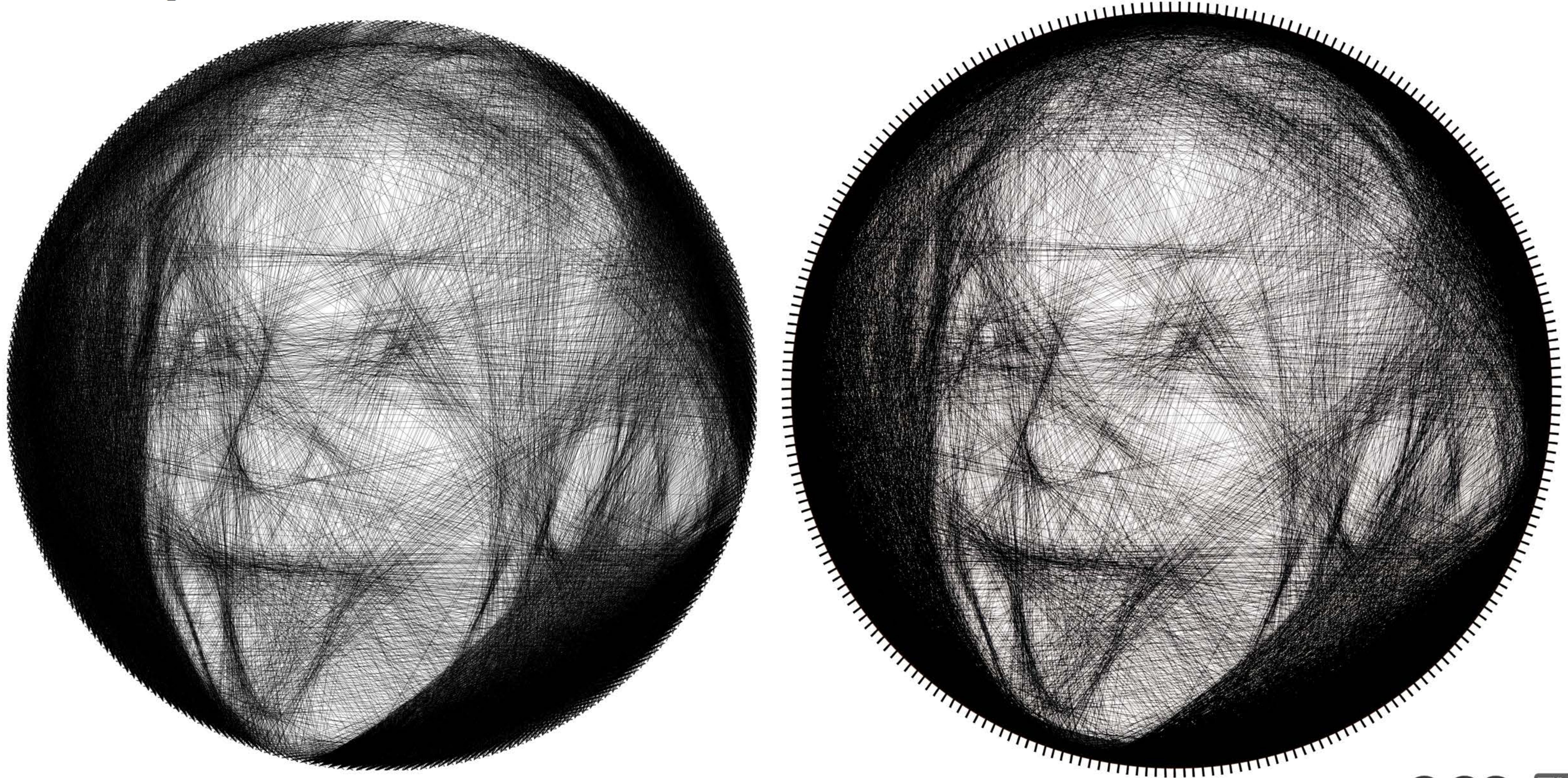
Computed Results



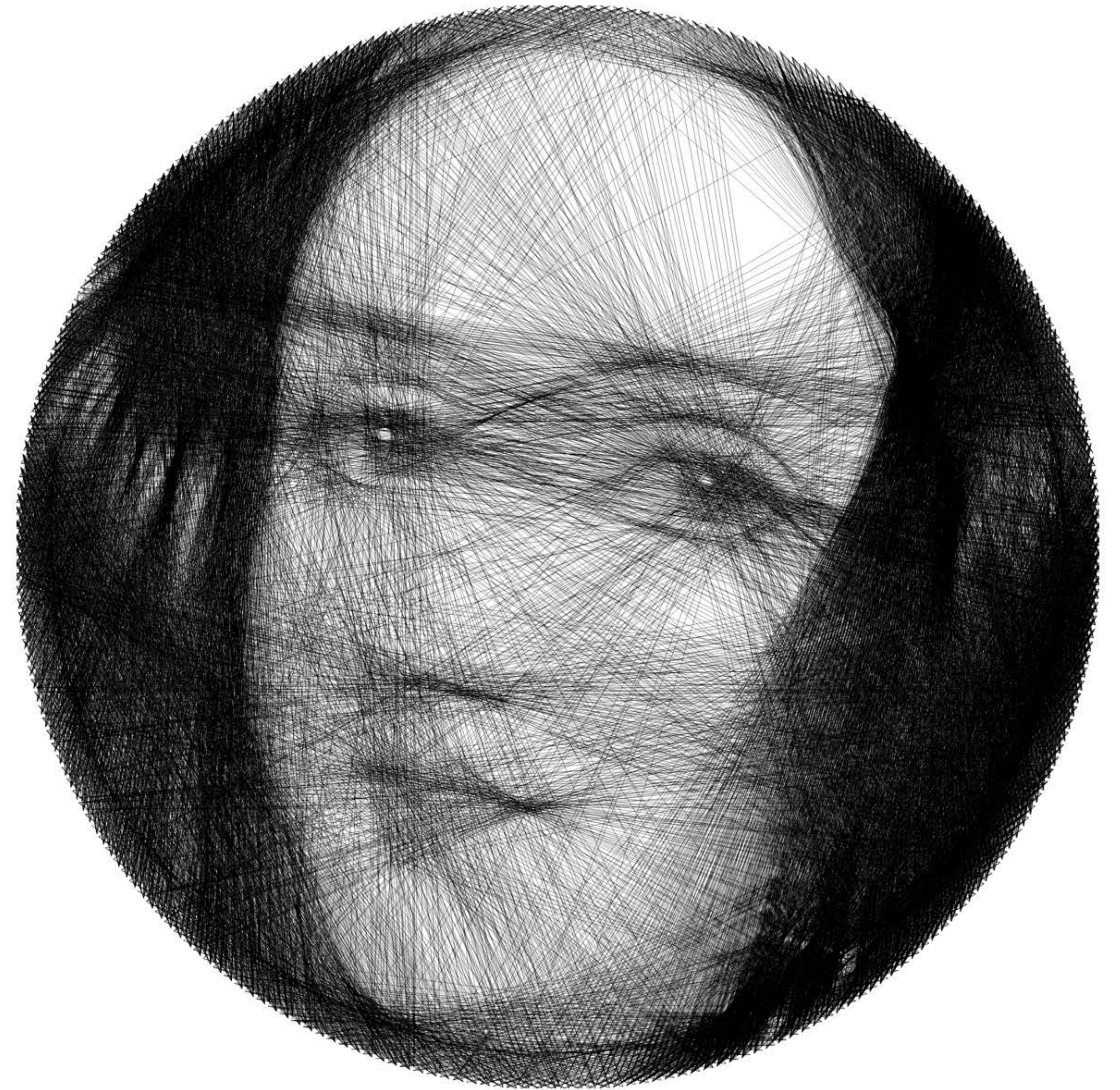
Computed and Fabricated Results



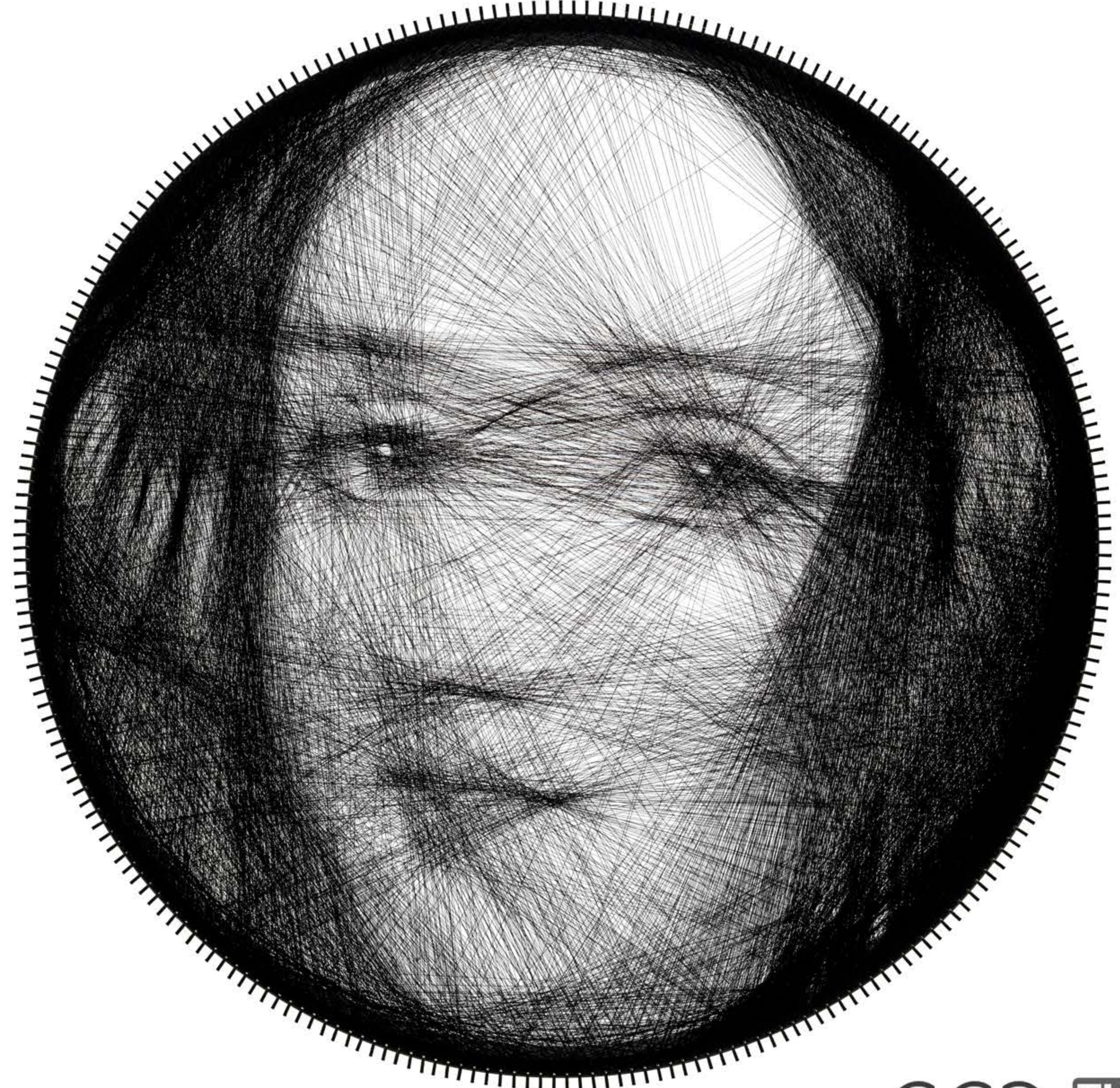
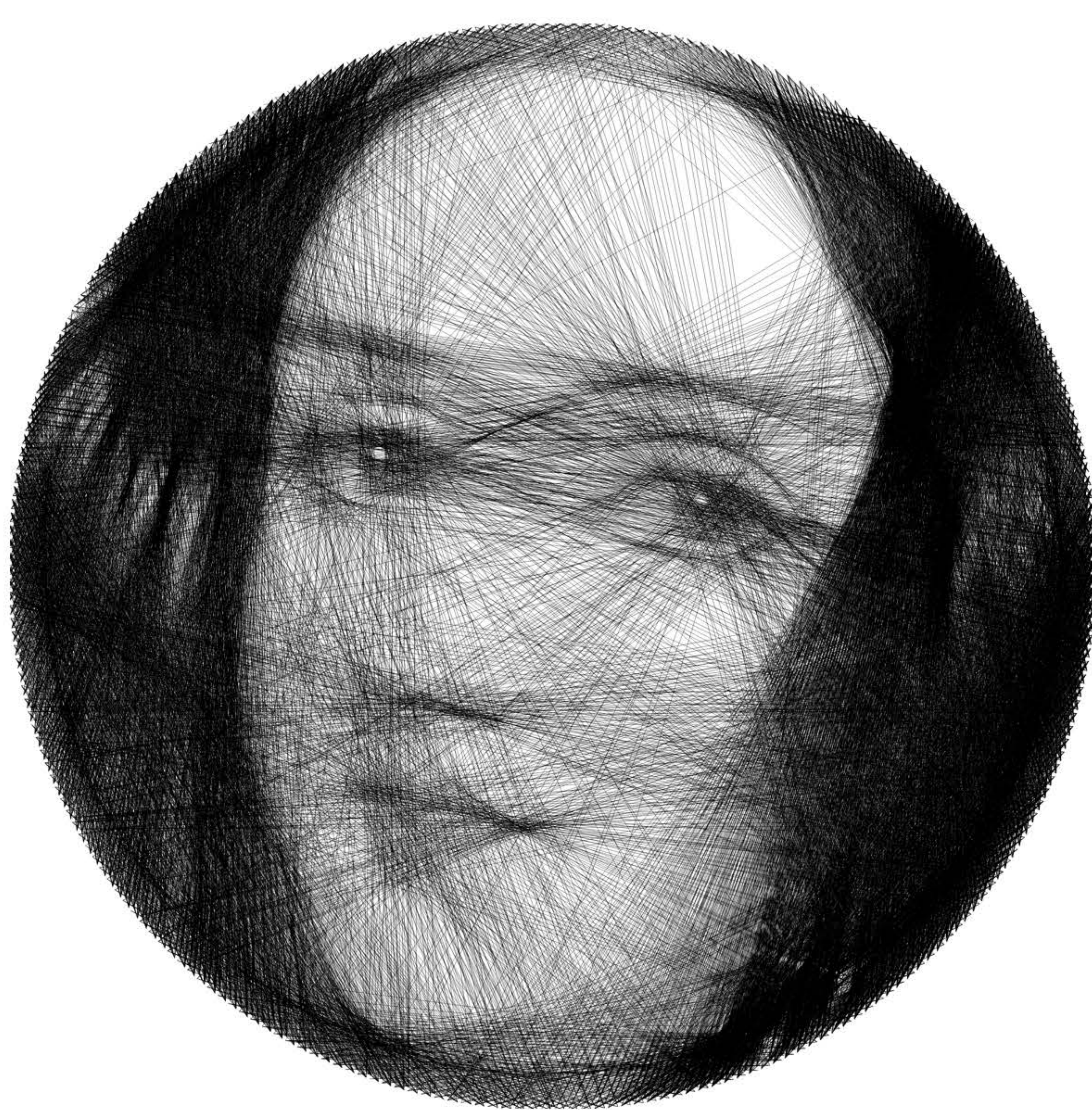
Computed and Fabricated Results



Computed and Fabricated Results



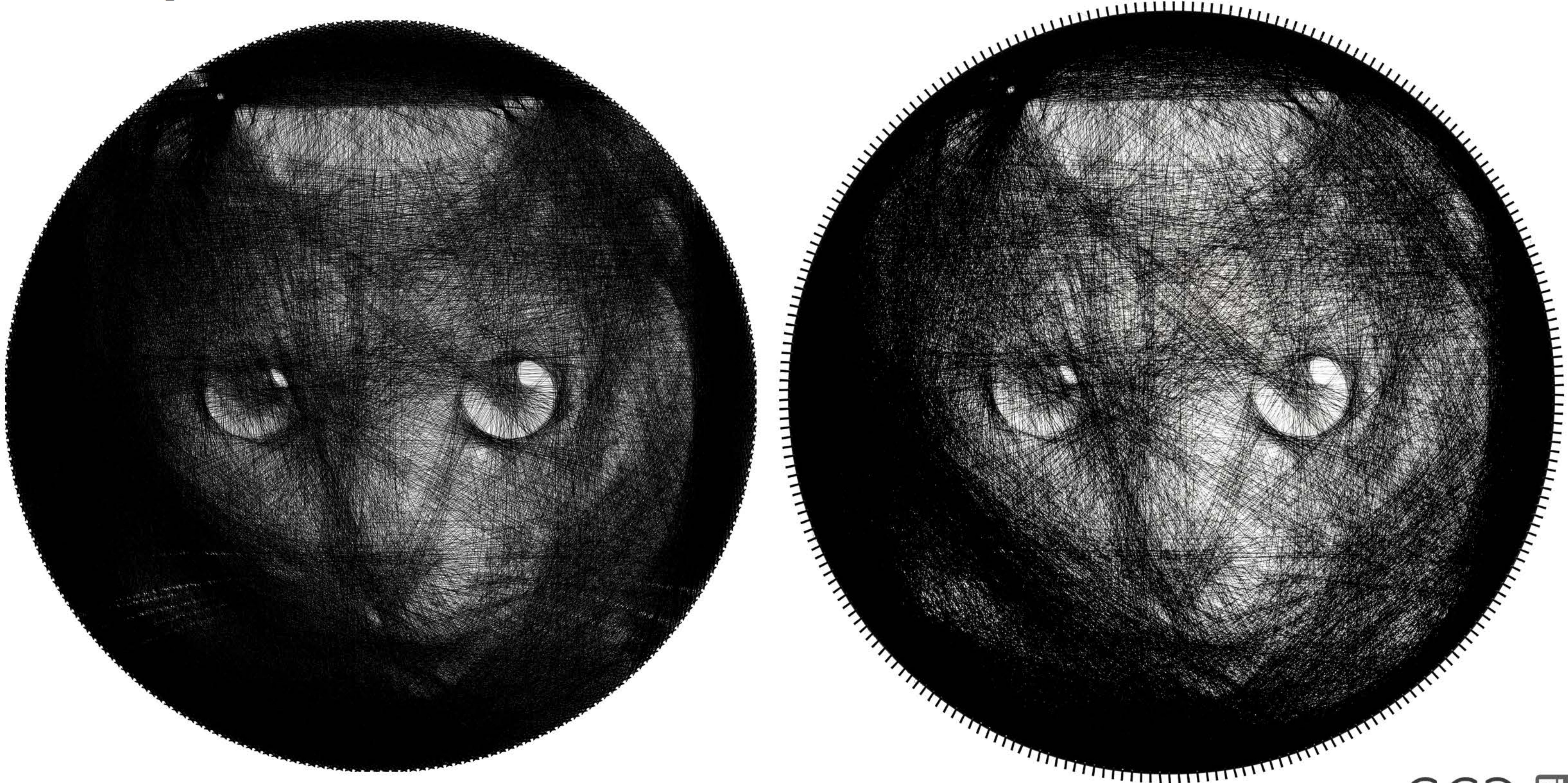
Computed and Fabricated Results



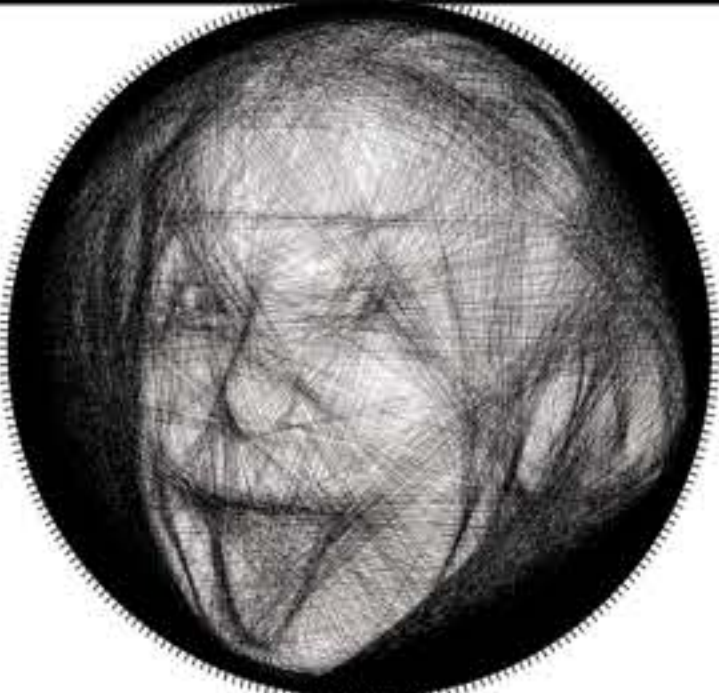
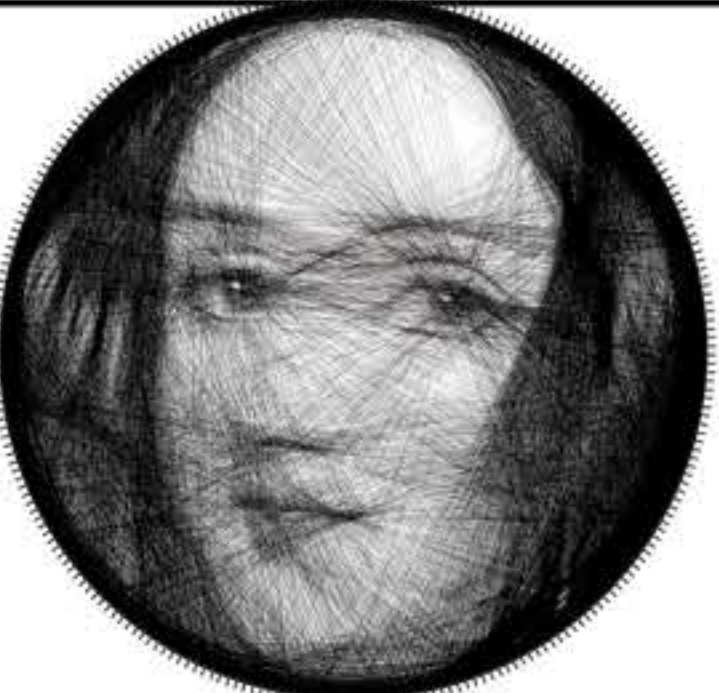
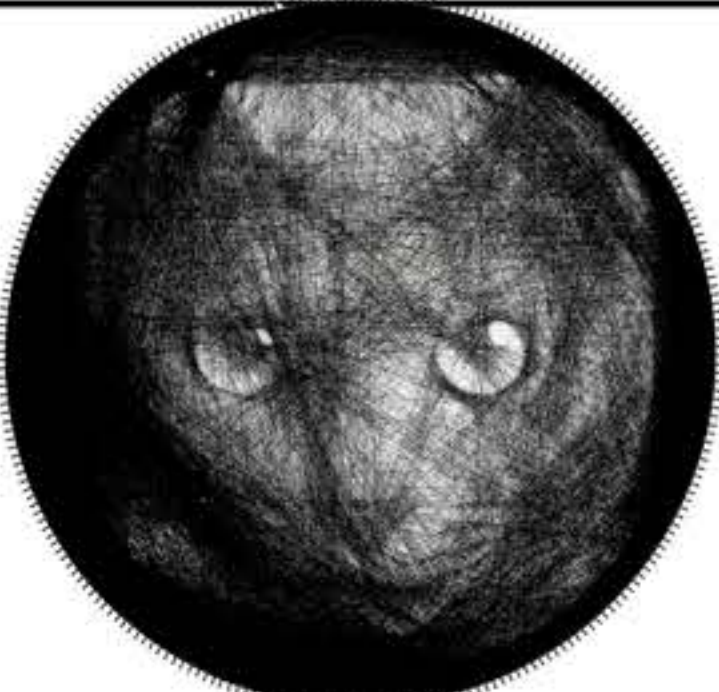
Computed and Fabricated Results



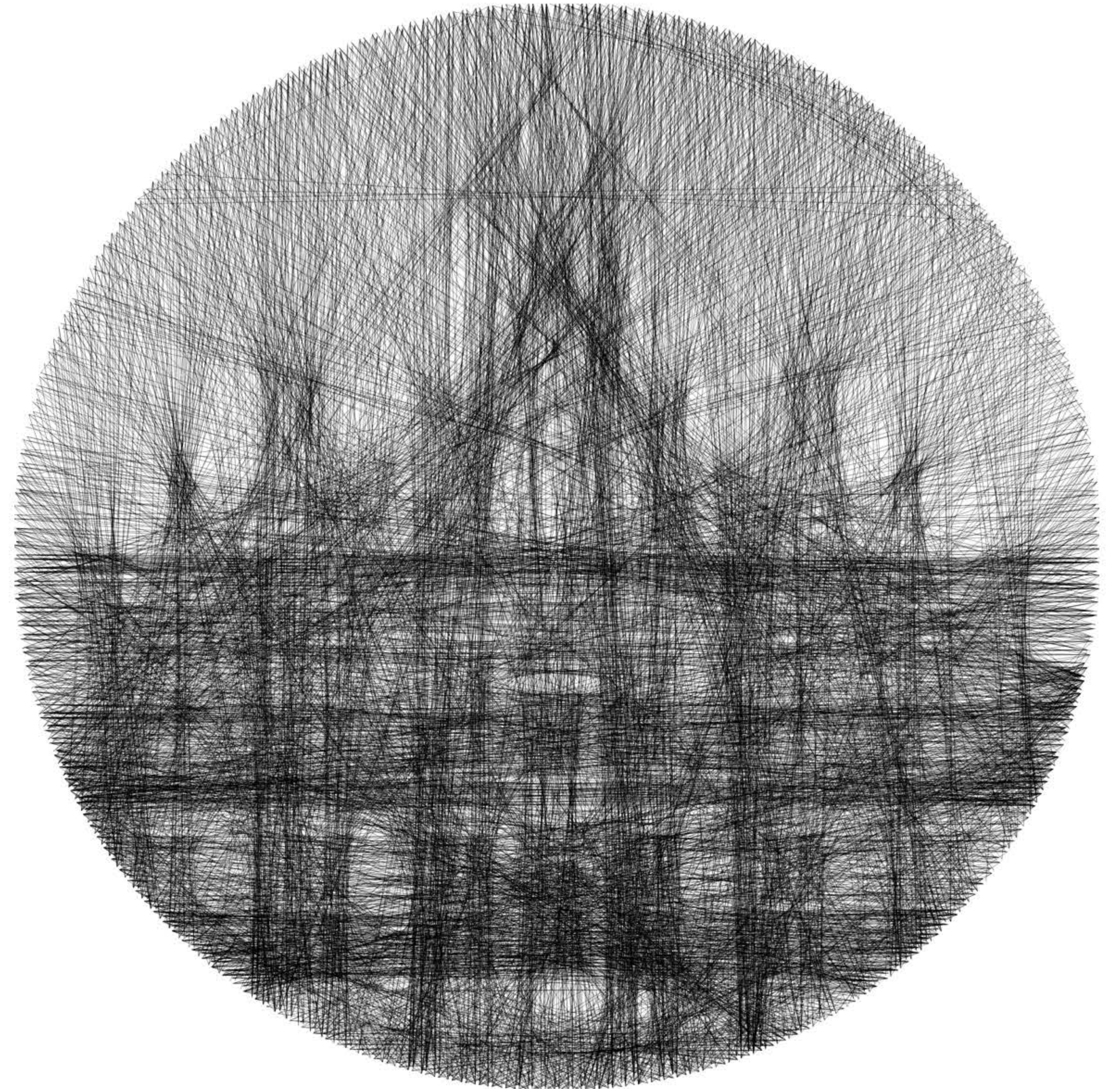
Computed and Fabricated Results



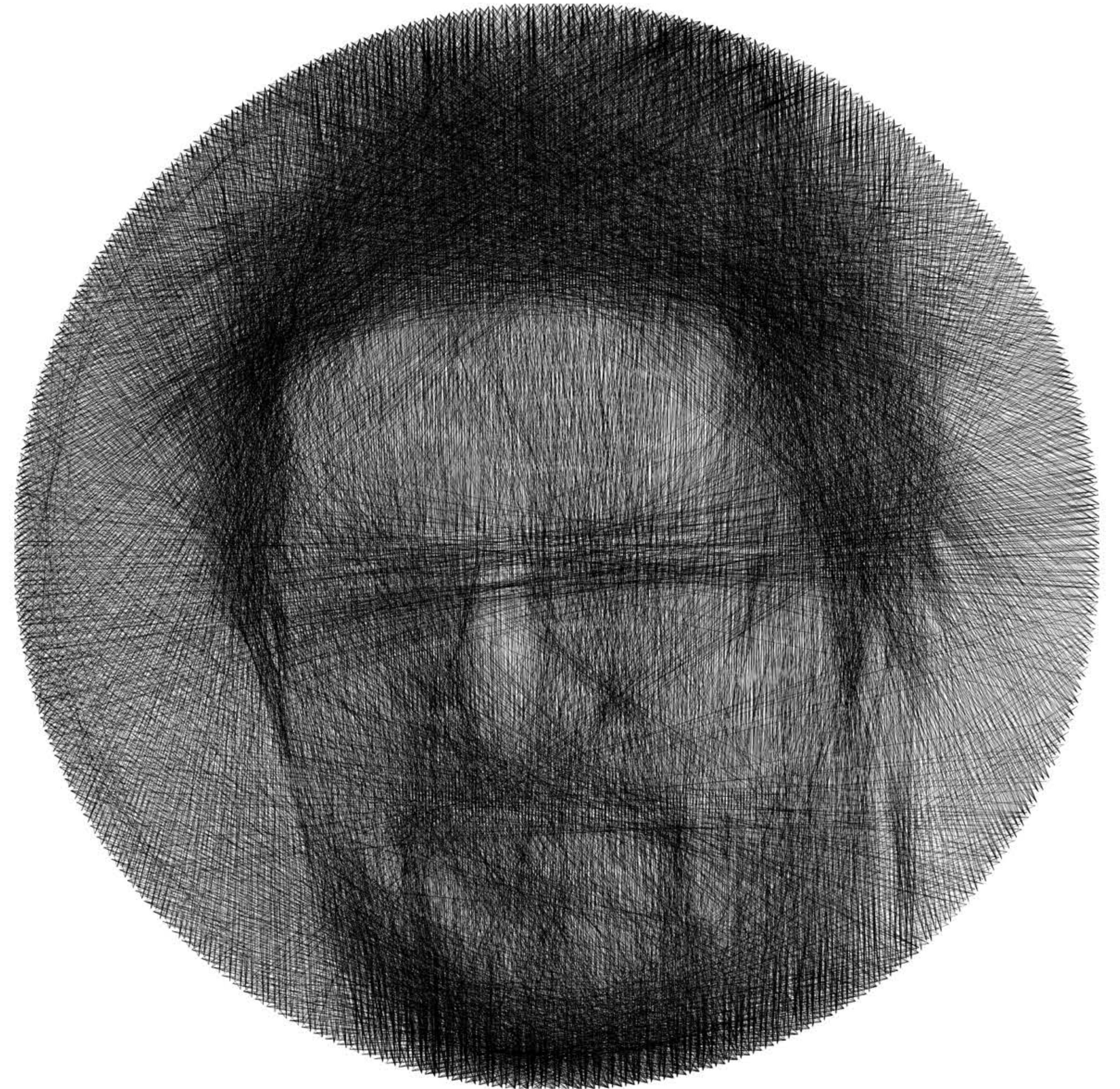
Numbers and Timings

Example	#Edges	Thread [m]	Computation Time [h]	Fabrication Time [h]
	8,324	2,980	1.87	3.4
	6,575	2,679	1.94	2.4
	14,850	5,985	4.39	4.3

Limitations

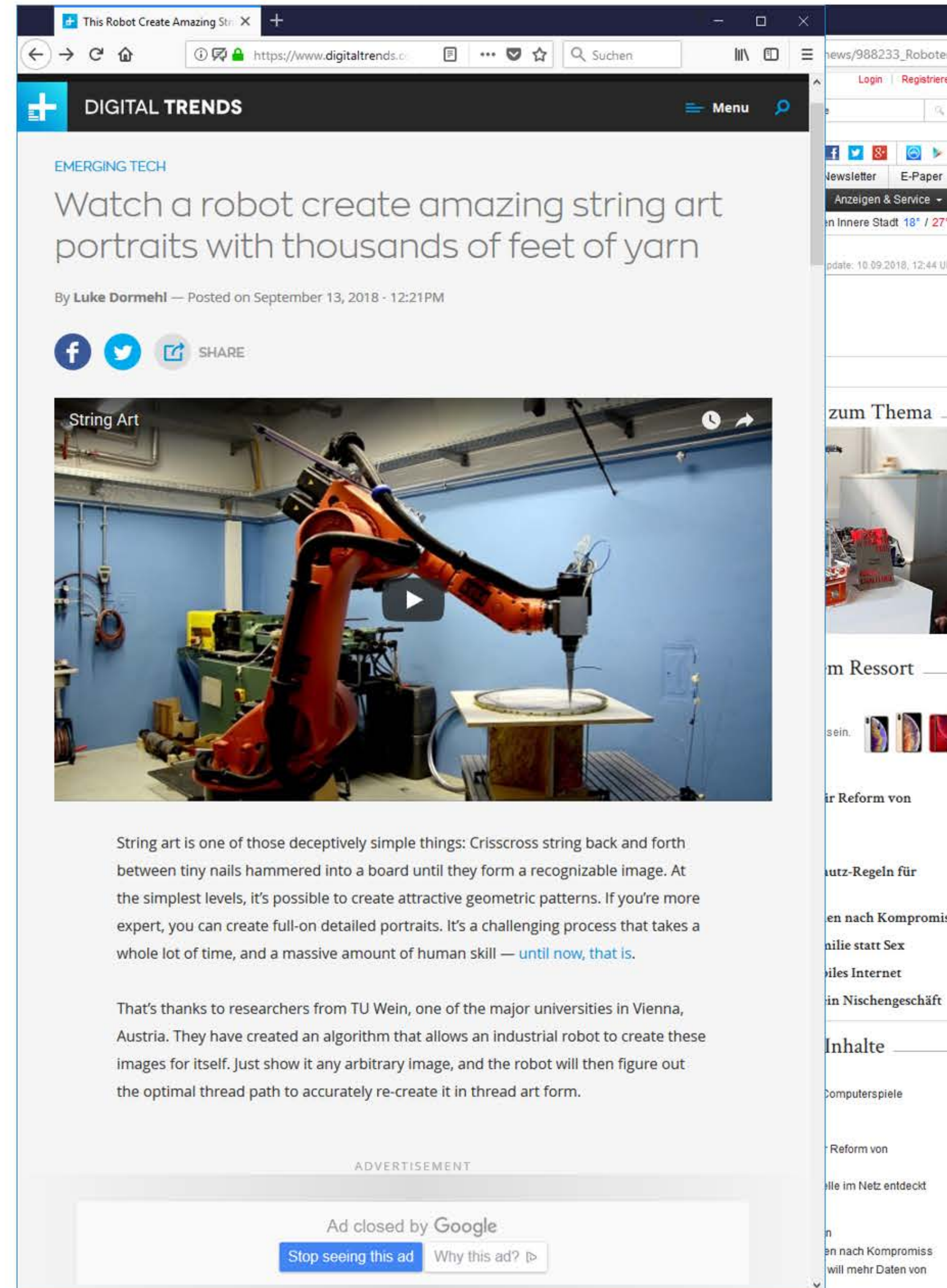


Limitations



Medial Success

- National
 - Der Standard
 - Factorynet
 - Futurezone
 - Kronenzeitung
 - Kurier
 - Wiener Zeitung
- International
 - Digital Trends



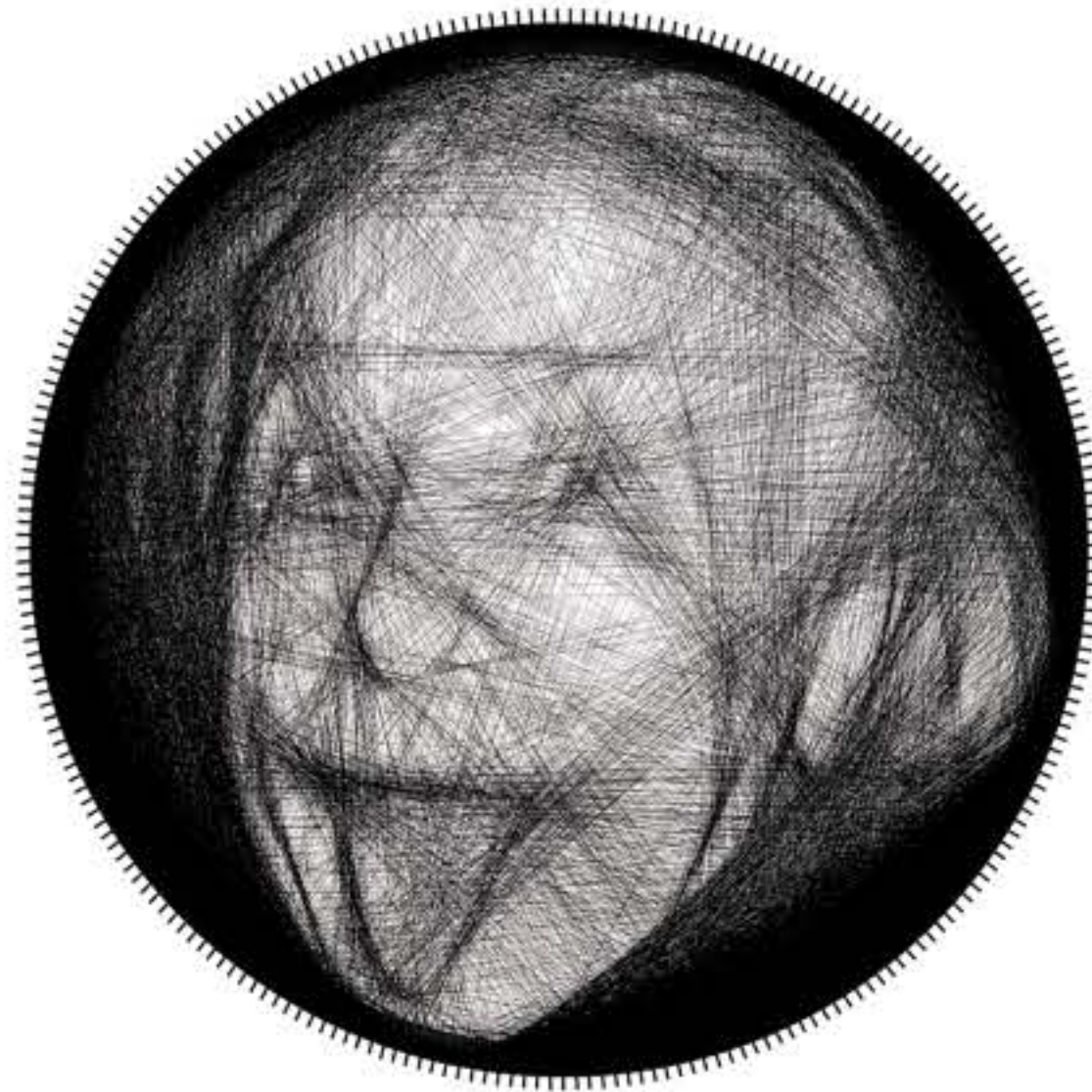
Overview



Michael Birsak, Przemyslaw Musialski, Peter Wonka, and Michael Wimmer

Dynamic Path Exploration on Mobile Devices

IEEE Transactions on Visualization and Computer Graphics. May 2018



Michael Birsak, Florian Rist, Peter Wonka, and Przemyslaw Musialski

String Art: Towards Computational Fabrication of String Images

Computer Graphics Forum (Proc. EUROGRAPHICS 2018). April 2018

Conclusion

- Discrete Optimization and in particular Integer Programming (IP) is an important and powerful tool
- We have shown the versatility of IP for a variety of problems



Diskrete Optimierung für GIS und Kunst



Thank you for
your attention

visit us at
gcd.tuwien.ac.at

