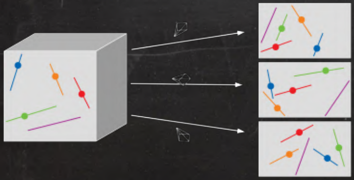
Algebraic Complexity of Computer Vision Problems

Kathlén Kohn



SWEDISH FOUNDATIONS' STARTING GRANT

Ragnar Söderbergs STIFTELSE

Göran Gustafssons Stiftelser

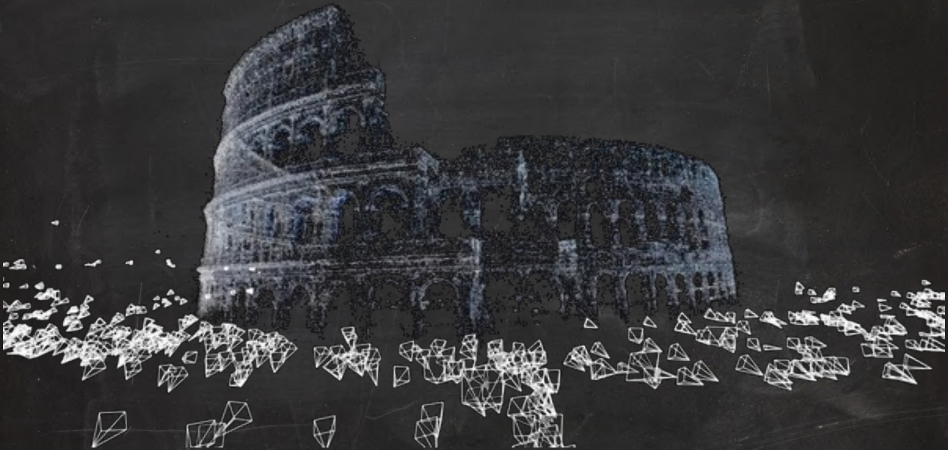
3D reconstruction

2d pictures

given images taken by unknown cameras, want to recover

3d modell

Reconstruct 3D scenes and camera poses from 2D images

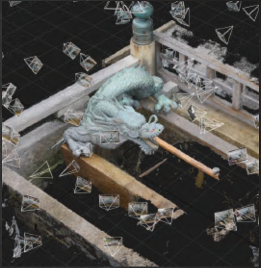


Rome in a Day: S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, S. Seitz, R. Szeliski

3D reconstruction pipeline

Input:
2D images

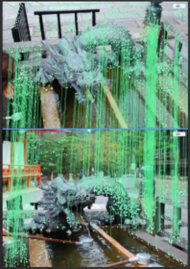
Output:
3D scene & cameras



3D reconstruction pipeline

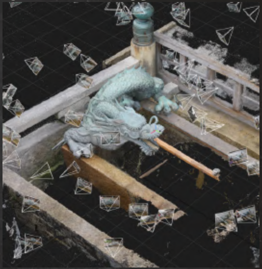
Input:
2D images

Image
matching



Identify common
points and lines
on given images

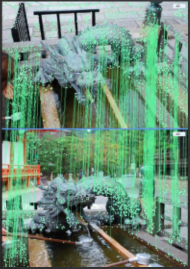
Output:
3D scene & cameras



3D reconstruction pipeline

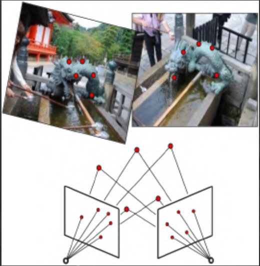
Input:
2D images

Image
matching



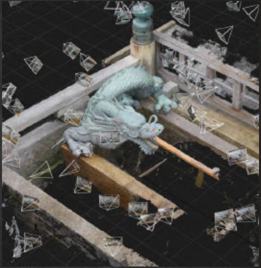
Identify common
points and lines
on given images

**Algebraic
reconstruction**



Reconstruct 3D points and
lines & camera poses

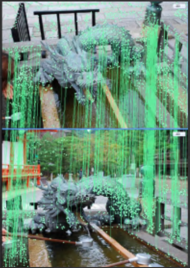
Output:
3D scene & cameras



3D reconstruction pipeline

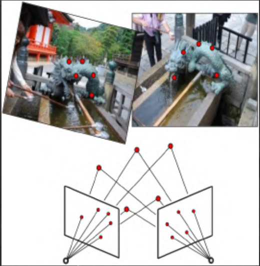
Input:
2D images

Image
matching



Identify common
points and lines
on given images

**Algebraic
reconstruction**



Reconstruct 3D points and
lines & camera poses

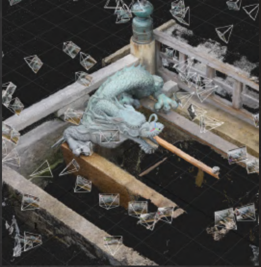
$\Updownarrow$

nonlinear inverse problem:

Compute fiber $\Phi^{-1}(y)$ of algebraic joint camera map

$\Phi : (\mathcal{C} \times \mathcal{X})/G \dashrightarrow \mathcal{Y}$

Output:
3D scene & cameras

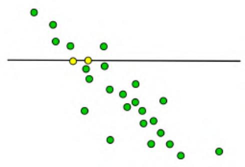


Measurements are noisy, and often corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

Measurements are noisy, and often corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

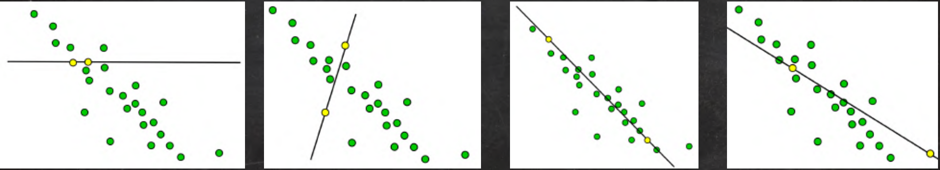
Example: fitting a line to points



Measurements are noisy, and often corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

Example: fitting a line to points



few outliers!

Observations are often noisy, and can even be corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

2d pictures

$\longrightarrow$

3d modell

for general algebraic inverse problems, step 2) means to solve a system of polynomial equations!

Observations are often noisy, and can even be corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

2d pictures

$\longrightarrow$

3d modell

for general algebraic inverse problems, step 2) means to solve a system of polynomial equations!

need to do this very fast, say in < 1 ms! (due to step 4))

Minimal problems

Computer vision engineers call an algebraic map a **minimal problem** if its generic complex fiber is

1) non-empty (otherwise no solution for noisy data) and
2) finite (to have finitely many model candidates in each RANSAC iteration)

Minimal problems

Computer vision engineers call an algebraic map a **minimal problem** if its generic complex fiber is

1) non-empty (otherwise no solution for noisy data) and
2) finite (to have finitely many model candidates in each RANSAC iteration)

For fast solvers, we want generically finite maps of **low degree.**

What is a camera?

A **pinhole camera** is a surjective linear map $\mathbb{P}^3 \to \mathbb{P}^2$.

What is a camera?

A **pinhole camera** is a surjective linear map $\mathbb{P}^3 \to \mathbb{P}^2$.

It is given by a 3×4 matrix

$$P = \underbrace{\begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsic params}} \cdot \underbrace{\begin{bmatrix} R & \mid & t \end{bmatrix}}_{\text{extrinsic params}}, \qquad R \in \mathrm{SO}(3).$$

What is a camera?

A **pinhole camera** is a surjective linear map $\mathbb{P}^3 \to \mathbb{P}^2$.

It is given by a 3×4 matrix

$$P = \underbrace{\begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{intrinsic params}} \cdot \underbrace{\begin{bmatrix} R & \mid & t \end{bmatrix}}_{\text{extrinsic params}}, \qquad R \in \mathrm{SO}(3).$$

If the intrinsic parameters are unknown, P can be (almost) any linear map, and is called an **uncalibrated / projective camera**.

What is a camera?

A **pinhole camera** is a surjective linear map $\mathbb{P}^3 \to \mathbb{P}^2$.

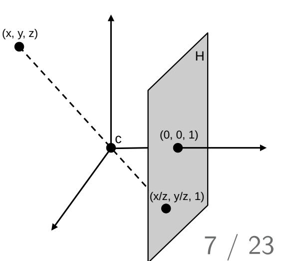
It is given by a 3×4 matrix

$$P = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R & | & t \end{bmatrix}, \qquad R \in \mathrm{SO}(3).$$

intrinsic params extrinsic params

If the intrinsic parameters are unknown, P can be (almost) any linear map, and is called an **uncalibrated / projective camera**.

If the intrinsic parameters are unknown, may assume that they are $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, and so P differs from the "standard camera" by a rotation and translation, called **calibrated camera**.



example minimal problem for projective cameras

Question: Let 2 projective cameras take pictures of k points:

$$\Phi : (\mathbb{P}\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^k \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

cameras points image 1 image 2

example minimal problem for projective cameras

Question: Let 2 projective cameras take pictures of k points:

$$\Phi : (\mathbb{P}\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^k \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

cameras points image 1 image 2

How many points do you need to recover the cameras, i.e., such that Φ has generically finite fibers?

example minimal problem for projective cameras

Question: Let 2 projective cameras take pictures of k points:

$$\underset{\text{cameras}}{\Phi : (\mathbb{P}\,\mathbb{R}^{3\times 4})^2} \times \underset{\text{points}}{(\mathbb{P}^3)^k} \longrightarrow \underset{\text{image 1}}{(\mathbb{P}^2)^k} \times \underset{\text{image 2}}{(\mathbb{P}^2)^k}$$

How many points do you need to recover the cameras, i.e., such that Φ has generically finite fibers?

Observation: We can mod out PGL_4:

$$\Phi : \left((\mathbb{P}\,\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^k \right) / \mathrm{PGL}_4 \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

dim: $\quad 11 \cdot 2 \quad + 3k \quad - 15 \qquad\qquad 2k + 2k$

example minimal problem for projective cameras

Question: Let 2 projective cameras take pictures of k points:

$$\Phi : (\mathbb{P}\,\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^k \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

cameras points image 1 image 2

How many points do you need to recover the cameras, i.e., such that Φ has generically finite fibers?

Observation: We can mod out PGL_4:

$$\Phi : \left((\mathbb{P}\,\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^k \right) / \mathrm{PGL}_4 \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

dim: $11 \cdot 2 \quad + 3k \quad - 15 \qquad 2k + 2k$

Domain and codomain have equal dimension for $k = 7$, and indeed, in that case, the generic fiber is finite of cardinality

example minimal problem for projective cameras

Question: Let 2 projective cameras take pictures of k points:

$$\Phi : (\mathbb{P}\,\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^k \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

cameras points image 1 image 2

How many points do you need to recover the cameras, i.e., such that Φ has generically finite fibers?

Observation: We can mod out PGL_4:

$$\Phi : \left((\mathbb{P}\,\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^k\right) / \mathrm{PGL}_4 \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

dim: $11 \cdot 2 \quad + 3k \quad - 15 \qquad 2k + 2k$

Domain and codomain have equal dimension for $k = 7$, and indeed, in that case, the generic fiber is finite of cardinality **3**.

example minimal problem for calibrated cameras

Question: Let 2 calibrated cameras take pictures of k points:

$$\Phi : (\mathrm{SO}(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

cameras points image 1 image 2

How many points do you need to recover the cameras, i.e., such that Φ has generically finite fibers?

example minimal problem for calibrated cameras

Question: Let 2 calibrated cameras take pictures of k points:

$$\Phi : (\mathrm{SO}(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

cameras points image 1 image 2

How many points do you need to recover the cameras, i.e., such that Φ has generically finite fibers?

Observation: We can mod out $G = \{ \left[\begin{smallmatrix} R & t \\ 0 & \lambda \end{smallmatrix}\right] \in \mathrm{GL}_4 \mid R \in \mathrm{SO}(3)\}$:

$$\Phi : \left(\mathrm{SO}(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k\right) / G \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

dim: $(3+3) \cdot 2 \quad + 3k \quad - 7 \qquad 2k + 2k$

example minimal problem for calibrated cameras

Question: Let 2 calibrated cameras take pictures of k points:

$$\Phi : (\mathrm{SO}(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

cameras points image 1 image 2

How many points do you need to recover the cameras, i.e., such that Φ has generically finite fibers?

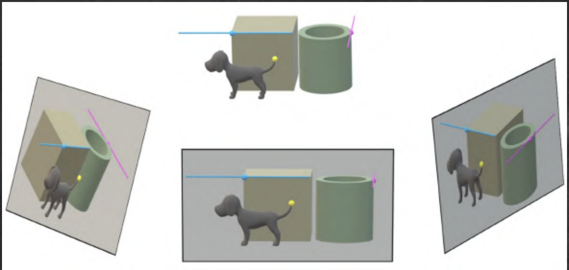
Observation: We can mod out $G = \{ \left[\begin{smallmatrix} R & t \\ 0 & \lambda \end{smallmatrix} \right] \in \mathrm{GL}_4 \mid R \in \mathrm{SO}(3) \}$:

$$\Phi : \left(\mathrm{SO}(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^k \right) / G \longrightarrow (\mathbb{P}^2)^k \times (\mathbb{P}^2)^k$$

dim: $(3+3) \cdot 2 \quad + 3k \quad - 7 \qquad 2k + 2k$

Domain and codomain have equal dimension for $k = 5$, and indeed, in that case, the generic fiber is finite of cardinality **20**.

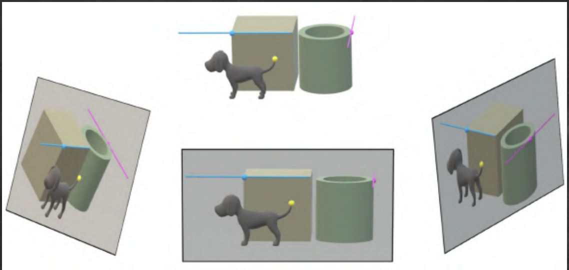
another minimal example for calibrated cameras



another minimal example for calibrated cameras

Given: point, point on line & point on line on each 2d-image

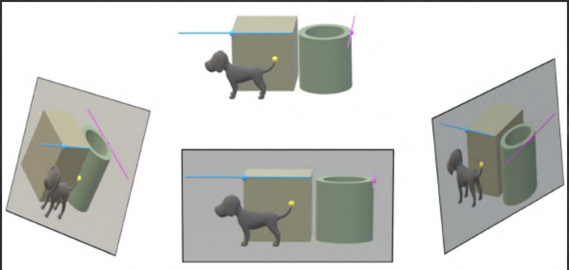
Goal: compute point, point on line & point on line in 3-space, and positions $c_1, c_2, c_3 \in \mathbb{R}^3$ & orientations $R_1, R_2, R_3 \in \mathrm{SO}(3)$ of cameras



another minimal example for calibrated cameras

Given: point, point on line & point on line on each 2d-image

Goal: compute point, point on line & point on line in 3-space, and positions $c_1, c_2, c_3 \in \mathbb{R}^3$ & orientations $R_1, R_2, R_3 \in \mathrm{SO}(3)$ of cameras

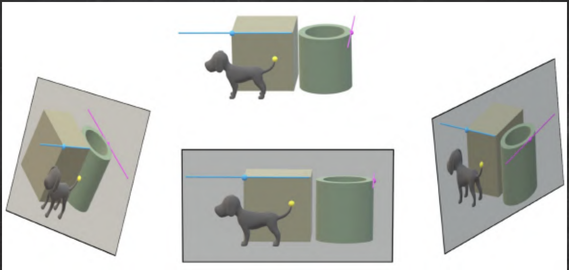


Generally has 312 complex solutions (modulo G).

another minimal example for calibrated cameras

Given: point, point on line & point on line on each 2d-image

Goal: compute point, point on line & point on line in 3-space, and positions $c_1, c_2, c_3 \in \mathbb{R}^3$ & orientations $R_1, R_2, R_3 \in \mathrm{SO}(3)$ of cameras



Generally has 312 complex solutions (modulo G).

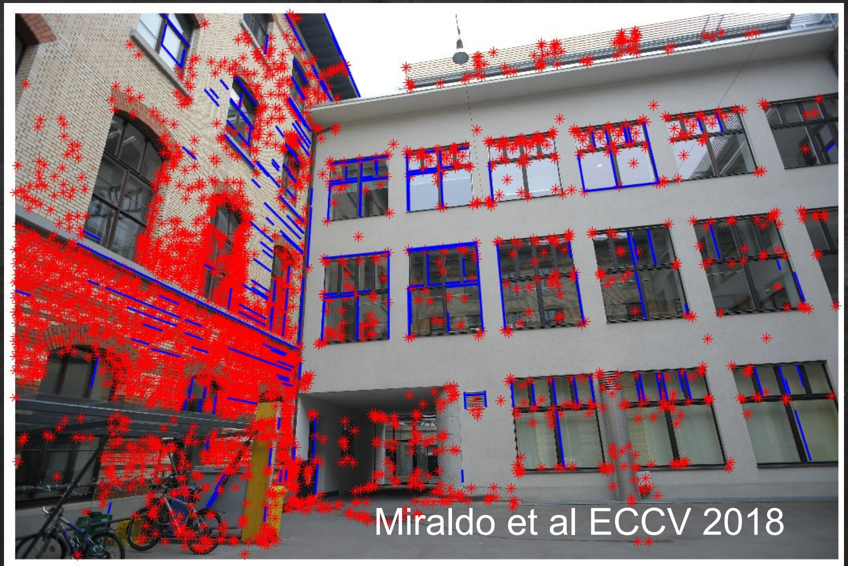
Gröbner basis methods won't terminate ...

Homotopy continuation can solve in 660ms on average on Intel core i7-7920HQ processor with 4 threads Fabbri et. al.: TRPLP – Trifocal Relative Pose from Lines at Points, CVPR 2020

Fundamental Research Questions

1. Can we list **all** minimal problems?
2. How many solutions do they have?

We do not only want to work with points,
but also with lines and their incidences!

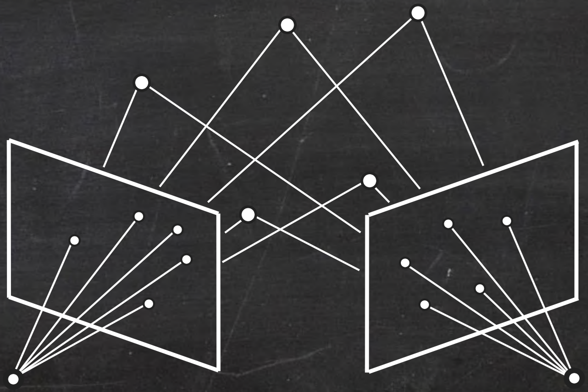


Our Result

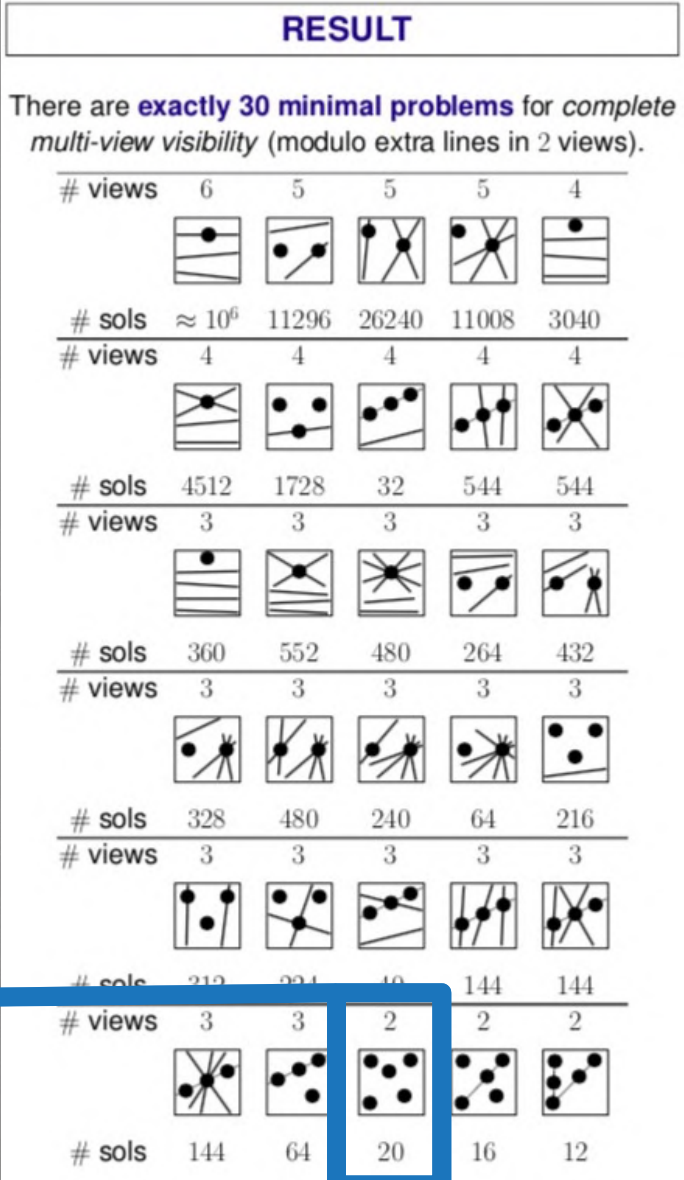
We provide the **first complete classification of all minimal problems** when all points and lines are visible in each given image.

Our Result

We provide the **first complete classification of all minimal problems** when all points and lines are visible in each given image.



The 5-point problem has 20 solutions.

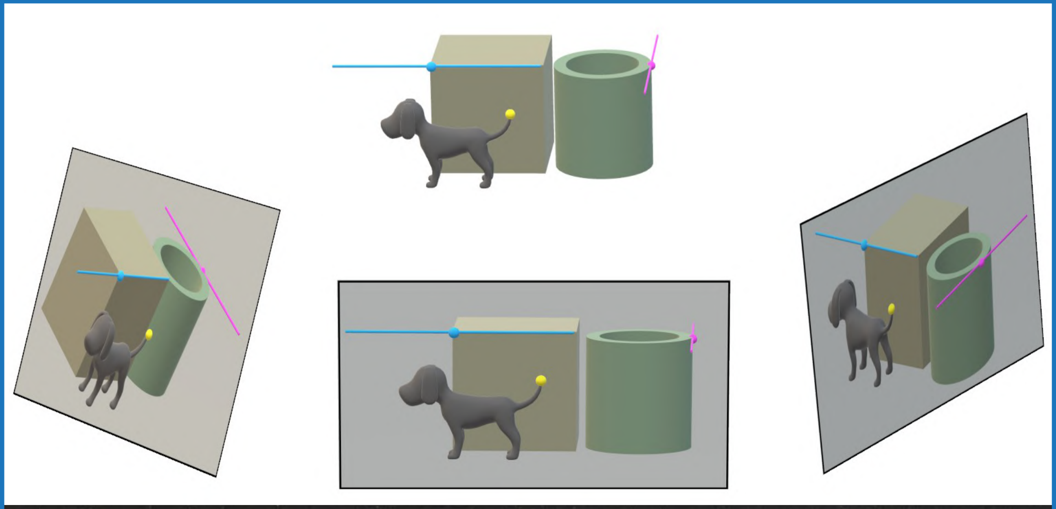


Our Result

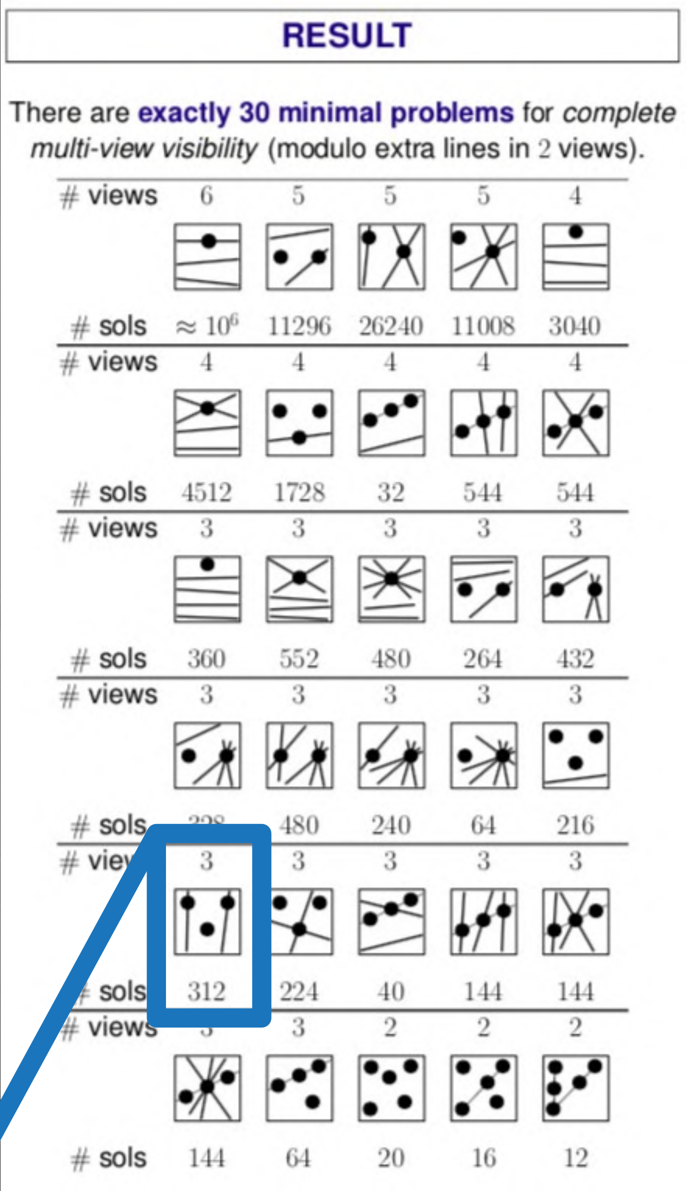
We provide the **first complete classification of all minimal problems** when all points and lines are visible in each given image.

First solver for such a high-degree problem based on state-of-the-art algorithms from **numerical algebraic geometry:**

TRPLP – Trifocal Relative Pose from Lines at Points, Fabbri et. al., CVPR 2020



This problem has 312 solutions
(counted over the complex numbers).



RESULT

There are **exactly 30 minimal problems** for *complete multi-view visibility* (modulo extra lines in 2 views).

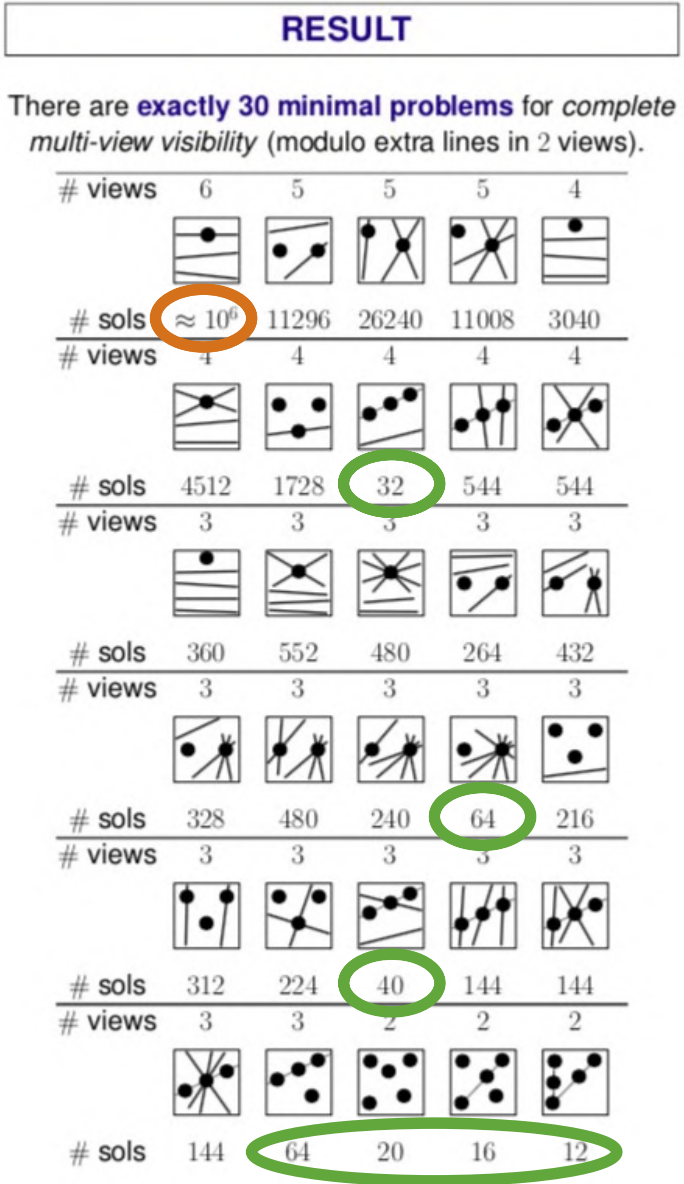
# views	6	5	5	5	4
# sols	≈ 10^6	11296	26240	11008	3040
# views	4	4	4	4	4
# sols	4512	1728	32	544	544
# views	3	3	3	3	3
# sols	360	552	480	264	432
# views	3	3	3	3	3
# sols	[illegible]	480	240	64	216
# views	3	3	3	3	3
# sols	312	224	40	144	144
# views	3	3	2	2	2
# sols	144	64	20	16	12

Our Result

We provide the **first complete classification of all minimal problems** when all points and lines are visible in each given image.

We **measure the complexity of each minimal problem** by computing its number of solutions

(counted over the complex numbers).



What about projective cameras?

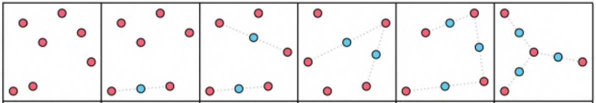
Theorem (K. Kiehn, A. Ahlbäck, K. Kohn): For projective cameras, all minimal problems involving points and lines are:

a) 2 cameras viewing one of the point-line arrangements in Table 1, plus arbitrarily many additional lines;

b) at least 2 cameras observing one of the 2 right-most point-line arrangements in Table 1;

c) one of the 285 PLPs in SM Section E (with 3–9 views).

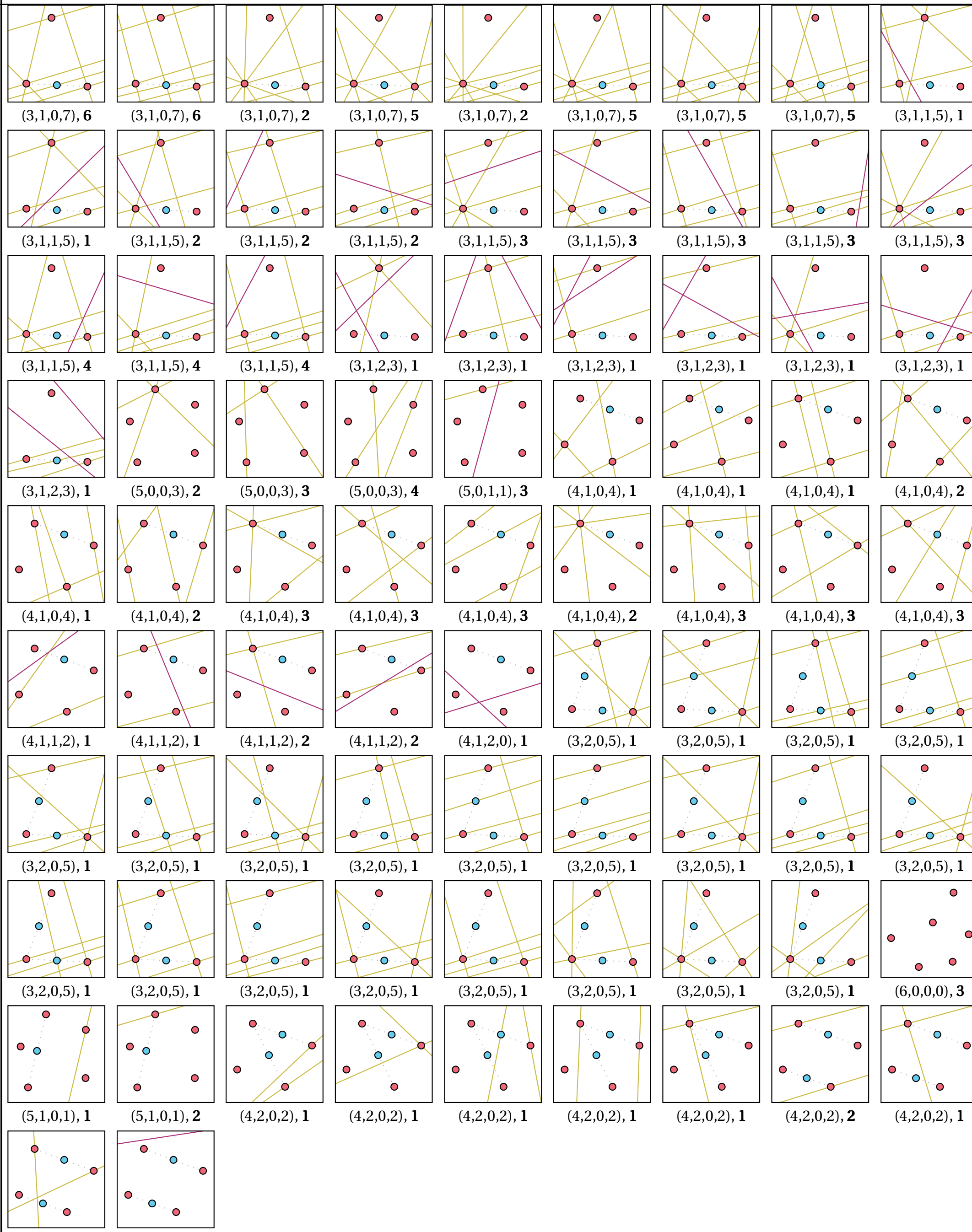
Their degrees are given in Table 1 and SM Section E.



3	2	2	1	1	1

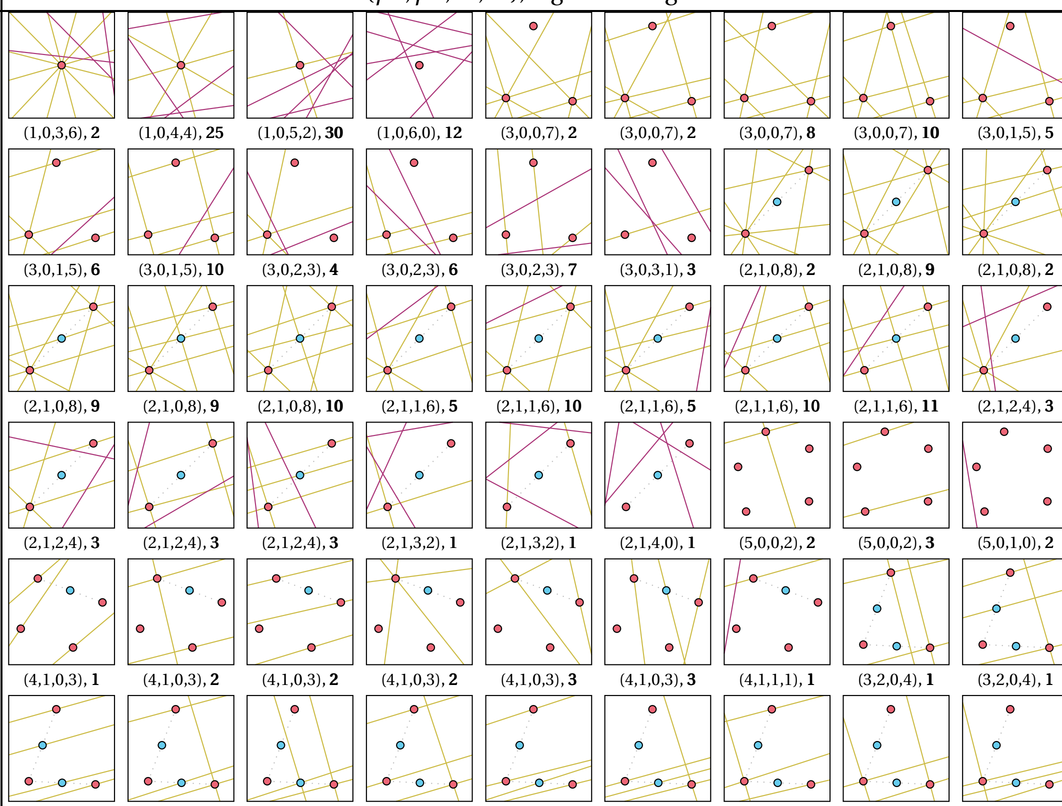
m	(p^f, p^d, l^f, l^a), algebraic degree								
3	(0,0,9,0), **36**	(1,0,4,7), **6**	(1,0,5,5), **23**	(1,0,6,3), **23**	(1,0,7,1), **15**	(2,0,0,12), **4**	(2,0,1,10), **6**	(2,0,1,10), **16**	(2,0,2,8), **4**
	(2,0,2,8), **12**	(2,0,2,8), **16**	(2,0,3,6), **2**	(2,0,3,6), **9**	(2,0,3,6), **15**	(2,0,3,6), **17**	(2,0,4,4), **9**	(2,0,4,4), **12**	(2,0,4,4), **13**
	(2,0,5,2), **8**	(2,0,5,2), **9**	(2,0,6,0), **7**	(3,0,0,9), **4**	(3,0,0,9), **4**	(3,0,0,9), **4**	(3,0,0,9), **10**	(3,0,0,9), **10**	(3,0,0,9), **12**
	(3,0,1,7), **2**	(3,0,1,7), **7**	(3,0,1,7), **2**	(3,0,1,7), **7**	(3,0,1,7), **10**	(3,0,1,7), **11**	(3,0,2,5), **2**	(3,0,2,5), **5**	(3,0,2,5), **7**
	(3,0,2,5), **8**	(3,0,2,5), **9**	(3,0,3,3), **6**	(3,0,3,3), **6**	(3,0,3,3), **6**	(3,0,4,1), **3**	(2,1,0,10), **4**	(2,1,0,10), **4**	(2,1,0,10), **4**
	(2,1,0,10), **4**	(2,1,0,10), **10**	(2,1,0,10), **10**	(2,1,0,10), **10**	(2,1,0,10), **10**	(2,1,1,8), **2**	(2,1,1,8), **7**	(2,1,1,8), **10**	(2,1,1,8), **2**
	(2,1,1,8), **7**	(2,1,1,8), **10**	(2,1,1,8), **10**	(2,1,1,8), **11**	(2,1,2,6), **2**	(2,1,2,6), **5**	(2,1,2,6), **5**	(2,1,2,6), **5**	(2,1,2,6), **5**
	(2,1,2,6), **5**	(2,1,2,6), **5**	(2,1,3,4), **2**	(2,1,3,4), **2**	(2,1,3,4), **2**	(2,1,3,4), **2**	(2,1,4,2), **1**	(2,1,4,2), **1**	(2,1,5,0), **1**
	(4,0,0,6), **2**	(4,0,0,6), **5**	(4,0,0,6), **2**	(4,0,0,6), **5**	(4,0,0,6), **6**	(4,0,0,6), **5**	(4,0,0,6), **7**	(4,0,1,4), **3**	(4,0,1,4), **5**
	(4,0,1,4), **5**	(4,0,1,4), **6**	(4,0,2,2), **3**	(4,0,2,2), **4**	(4,0,3,0), **3**	(3,1,0,7), **2**	(3,1,0,7), **2**	(3,1,0,7), **2**	(3,1,0,7), **2**
	(3,1,0,7), **2**	(3,1,0,7), **5**	(3,1,0,7), **5**	(3,1,0,7), **5**	(3,1,0,7), **5**	(3,1,0,7), **2**	(3,1,0,7), **5**	(3,1,0,7), **6**	(3,1,0,7), **5**

Table 7: Minimal problems with their associated degree.



m	(p^f, p^d, l^f, l^a), algebraic degree								
3	(3,1,0,7), **6**	(3,1,0,7), **6**	(3,1,0,7), **2**	(3,1,0,7), **5**	(3,1,0,7), **2**	(3,1,0,7), **5**	(3,1,0,7), **5**	(3,1,0,7), **5**	(3,1,1,5), **1**
	(3,1,1,5), **1**	(3,1,1,5), **2**	(3,1,1,5), **2**	(3,1,1,5), **2**	(3,1,1,5), **3**	(3,1,1,5), **3**	(3,1,1,5), **3**	(3,1,1,5), **3**	(3,1,1,5), **3**
	(3,1,1,5), **4**	(3,1,1,5), **4**	(3,1,1,5), **4**	(3,1,2,3), **1**	(3,1,2,3), **1**	(3,1,2,3), **1**	(3,1,2,3), **1**	(3,1,2,3), **1**	(3,1,2,3), **1**
	(3,1,2,3), **1**	(5,0,0,3), **2**	(5,0,0,3), **3**	(5,0,0,3), **4**	(5,0,1,1), **3**	(4,1,0,4), **1**	(4,1,0,4), **1**	(4,1,0,4), **1**	(4,1,0,4), **2**
	(4,1,0,4), **1**	(4,1,0,4), **2**	(4,1,0,4), **3**	(4,1,0,4), **3**	(4,1,0,4), **3**	(4,1,0,4), **2**	(4,1,0,4), **3**	(4,1,0,4), **3**	(4,1,0,4), **3**
	(4,1,1,2), **1**	(4,1,1,2), **1**	(4,1,1,2), **2**	(4,1,1,2), **2**	(4,1,2,0), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**
	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**
	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(3,2,0,5), **1**	(6,0,0,0), **3**
	(5,1,0,1), **1**	(5,1,0,1), **2**	(4,2,0,2), **1**	(4,2,0,2), **1**	(4,2,0,2), **1**	(4,2,0,2), **1**	(4,2,0,2), **1**	(4,2,0,2), **2**	(4,2,0,2), **1**
	(4,2,0,2), **1**	(4,2,1,0), **1**							

Table 8: Minimal problems with their associated degree.



m	(p^f, p^d, l^f, l^a), algebraic degree								
4	(1,0,3,6), **2**	(1,0,4,4), **25**	(1,0,5,2), **30**	(1,0,6,0), **12**	(3,0,0,7), **2**	(3,0,0,7), **2**	(3,0,0,7), **8**	(3,0,0,7), **10**	(3,0,1,5), **5**
	(3,0,1,5), **6**	(3,0,1,5), **10**	(3,0,2,3), **4**	(3,0,2,3), **6**	(3,0,2,3), **7**	(3,0,3,1), **3**	(2,1,0,8), **2**	(2,1,0,8), **9**	(2,1,0,8), **2**
	(2,1,0,8), **9**	(2,1,0,8), **9**	(2,1,0,8), **10**	(2,1,1,6), **5**	(2,1,1,6), **10**	(2,1,1,6), **5**	(2,1,1,6), **10**	(2,1,1,6), **11**	(2,1,2,4), **3**
	(2,1,2,4), **3**	(2,1,2,4), **3**	(2,1,2,4), **3**	(2,1,3,2), **1**	(2,1,3,2), **1**	(2,1,4,0), **1**	(5,0,0,2), **2**	(5,0,0,2), **3**	(5,0,1,0), **2**
	(4,1,0,3), **1**	(4,1,0,3), **2**	(4,1,0,3), **2**	(4,1,0,3), **2**	(4,1,0,3), **3**	(4,1,0,3), **3**	(4,1,1,1), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**
	(3,2,0,4), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**	(3,2,0,4), **1**

Table 9: Minimal problems with their associated degree.

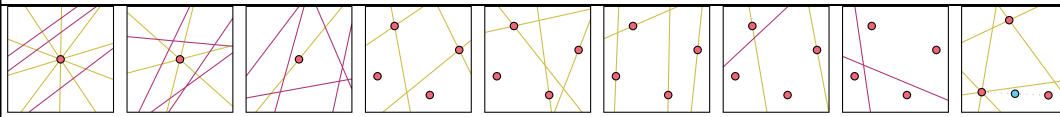
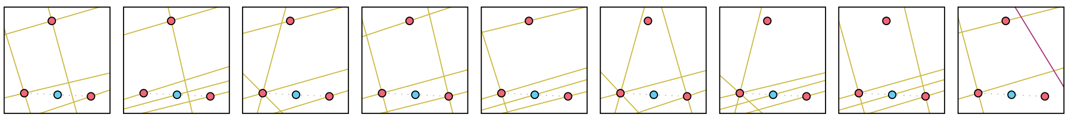
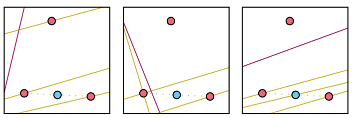
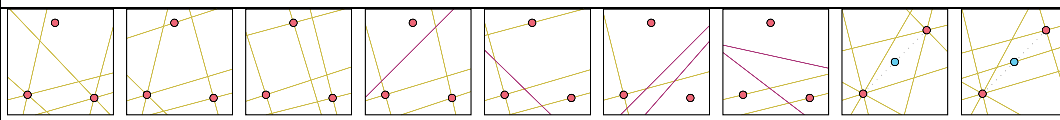
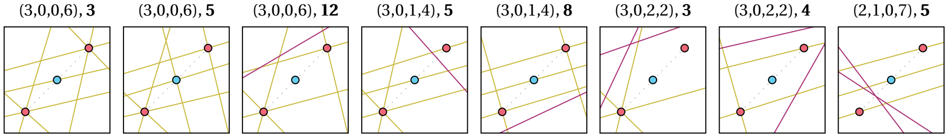
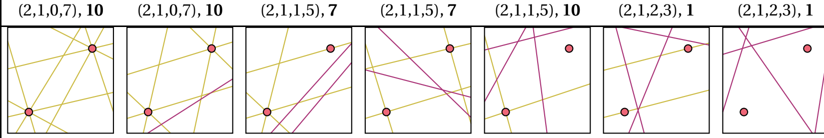
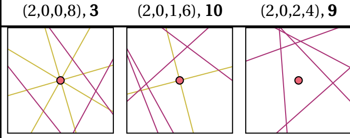
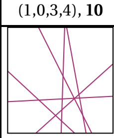
m	(p^f, p^d, l^f, l^a), algebraic degree
5	 (1,0,3,5), **6**; (1,0,4,3), **35**; (1,0,5,1), **20**; (4,0,0,4), **3**; (4,0,0,4), **4**; (4,0,0,4), **7**; (4,0,1,2), **3**; (4,0,2,0), **2**; (3,1,0,5), **2**
	 (3,1,0,5), **2**; (3,1,0,5), **2**; (3,1,0,5), **4**; (3,1,0,5), **6**; (3,1,0,5), **6**; (3,1,0,5), **4**; (3,1,0,5), **4**; (3,1,0,5), **5**; (3,1,1,3), **1**
	 (3,1,1,3), **1**; (3,1,1,3), **2**; (3,1,1,3), **2**
6	 (3,0,0,6), **3**; (3,0,0,6), **5**; (3,0,0,6), **12**; (3,0,1,4), **5**; (3,0,1,4), **8**; (3,0,2,2), **3**; (3,0,2,2), **4**; (2,1,0,7), **5**; (2,1,0,7), **5**
	 (2,1,0,7), **10**; (2,1,0,7), **10**; (2,1,1,5), **7**; (2,1,1,5), **7**; (2,1,1,5), **10**; (2,1,2,3), **1**; (2,1,2,3), **1**; (2,1,2,3), **1**
7	 (2,0,0,8), **3**; (2,0,1,6), **10**; (2,0,2,4), **9**; (2,0,2,4), **20**; (2,0,3,2), **6**; (2,0,3,2), **9**; (2,0,4,0), **3**
8	 (1,0,3,4), **10**; (1,0,4,2), **38**; (1,0,5,0), **8**
9	 (0,0,6,0), **114**

Table 10: Minimal problems with their associated degree.

Is the number of solutions an accurate complexity measure?

Is the number of solutions an accurate complexity measure?

A Galois-Theoretic Complexity Measure for Solving Systems of Algebraic Equations

Timothy Duff

March 25, 2025

Abstract

Motivated by applications of algebraic geometry, we introduce the *Galois width*, a quantity characterizing the complexity of solving algebraic equations in a restricted model of computation allowing only field arithmetic and adjoining polynomial roots. We explain why practical heuristics such as monodromy give (at least) lower bounds on this quantity, and discuss problems in geometry, optimization, statistics, and computer vision for which knowledge of the Galois width either leads to improvements over standard solution techniques or rules out this possibility entirely.

Galois width example

The Galois width of finding the roots of a univariate polynomial of degree n is

$$\begin{cases} 3 \quad , & \text{if } n = 4 \\ n \quad , & \text{else} \end{cases}$$

Galois width example

The Galois width of finding the roots of a univariate polynomial of degree n is

$$\begin{cases} 3 & , \quad \text{if } n = 4 \\ n & , \quad \text{else} \end{cases}$$

The roots of a general quartic can be expressed in terms of the roots of its resolvent cubic and additional square roots thereof!

Galois width of vision minimal problems

Let 2 projective cameras take pictures of 7 points:

$$\Phi : \left((\mathbb{P}\,\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^7 \right) / \mathrm{PGL}_4 \longrightarrow (\mathbb{P}^2)^7 \times (\mathbb{P}^2)^7$$

has generic fibers of size 3 and $\mathrm{GaloisWidth}(\Phi) = 3$.

Galois width of vision minimal problems

Let 2 projective cameras take pictures of 7 points:

$$\Phi : \left((\mathbb{P}\,\mathbb{R}^{3\times 4})^2 \times (\mathbb{P}^3)^7 \right) / \mathrm{PGL}_4 \longrightarrow (\mathbb{P}^2)^7 \times (\mathbb{P}^2)^7$$

has generic fibers of size 3 and $\mathrm{GaloisWidth}(\Phi) = 3$.

Let 2 calibrated cameras take pictures of 5 points:

$$\Phi : \left((\mathrm{SO}(3) \times \mathbb{R}^3)^2 \times (\mathbb{P}^3)^5 \right) / G \longrightarrow (\mathbb{P}^2)^5 \times (\mathbb{P}^2)^5$$

has generic fibers of size 20 and $\mathrm{GaloisWidth}(\Phi) = 10$.

Joint-image varieties

well-known theorem:

For a fixed camera pair (P_1, P_2) with distinct kernels, the image of their joint picture-taking map

$$\begin{aligned} \Phi_{P_1,P_2} : \mathbb{P}^3 &\longrightarrow \mathbb{P}^2 \times \mathbb{P}^2, \\ X &\longmapsto (P_1 X, P_2 X) \end{aligned}$$

is a hypersurface.

Joint-image varieties

well-known theorem:
For a fixed camera pair (P_1, P_2) with distinct kernels, the image of their joint picture-taking map

$$\begin{aligned} \Phi_{P_1,P_2} : \mathbb{P}^3 &\longrightarrow \mathbb{P}^2 \times \mathbb{P}^2, \\ X &\longmapsto (P_1 X, P_2 X) \end{aligned}$$

is a hypersurface. It is defined by a bilinear equation, i.e., of the form $\{(x, y) \in \mathbb{P}^2 \times \mathbb{P}^2 \mid x^\top F y = 0\}$ for some 3×3 matrix $F = F_{P_1,P_2}$, called **fundamental matrix** of the camera pair.

Joint-image varieties

well-known theorem:
For a fixed camera pair (P_1, P_2) with distinct kernels, the image of their joint picture-taking map

$$\Phi_{P_1,P_2} : \mathbb{P}^3 \longrightarrow \mathbb{P}^2 \times \mathbb{P}^2, \quad X \longmapsto (P_1 X, P_2 X)$$

is a hypersurface. It is defined by a bilinear equation, i.e., of the form $\{(x, y) \in \mathbb{P}^2 \times \mathbb{P}^2 \mid x^\top F y = 0\}$ for some 3×3 matrix $F = F_{P_1,P_2}$, called **fundamental matrix** of the camera pair.

- pairs of **projective** cameras (P_1, P_2) mod PGL_4 are **1-to-1** with rank-2 matrices F_{P_1,P_2}.

Joint-image varieties

well-known theorem:
For a fixed camera pair (P_1, P_2) with distinct kernels, the image of their joint picture-taking map

$$\begin{aligned} \Phi_{P_1,P_2} : \mathbb{P}^3 &\longrightarrow \mathbb{P}^2 \times \mathbb{P}^2, \\ X &\longmapsto (P_1X, P_2X) \end{aligned}$$

is a hypersurface. It is defined by a bilinear equation, i.e., of the form $\{(x, y) \in \mathbb{P}^2 \times \mathbb{P}^2 \mid x^\top Fy = 0\}$ for some 3×3 matrix $F = F_{P_1,P_2}$, called **fundamental matrix** of the camera pair.

- pairs of **projective** cameras (P_1, P_2) mod PGL_4 are **1-to-1** with rank-2 matrices F_{P_1,P_2}.
- pairs of **calibrated** cameras (P_1, P_2) mod G are **2-to-1** with rank-2 matrices F_{P_1,P_2} that have coinciding singular values.

Joint-image varieties

well-known theorem:

For a fixed camera pair (P_1, P_2) with distinct kernels, the image of their joint picture-taking map

$$\Phi_{P_1,P_2} : \mathbb{P}^3 \longrightarrow \mathbb{P}^2 \times \mathbb{P}^2, \quad X \longmapsto (P_1 X, P_2 X)$$

is a hypersurface. It is defined by a bilinear equation, i.e., of the form $\{(x, y) \in \mathbb{P}^2 \times \mathbb{P}^2 \mid x^\top F y = 0\}$ for some 3×3 matrix $F = F_{P_1,P_2}$, called **fundamental matrix** of the camera pair.

- pairs of **projective** cameras (P_1, P_2) mod PGL_4 are **1-to-1** with rank-2 matrices F_{P_1,P_2}.
- pairs of **calibrated** cameras (P_1, P_2) mod G are **2-to-1** with rank-2 matrices F_{P_1,P_2} that have coinciding singular values.

First reconstructing F_{P_1,P_2} and afterwards (P_1, P_2) explains $\deg(\Phi) = 20 = 10 \cdot 2 = \mathrm{GaloisWidth}(\Phi) \cdot 2$ in the calibrated case.

Triangulation

Typically, we need up to **thousands** of points to reconstruct the cameras involved in a 3D scene, but want to reconstruct up to **millions** of points to obtain a dense point cloud.

Triangulation

Typically, we need up to **thousands** of points to reconstruct the cameras involved in a 3D scene, but want to reconstruct up to **millions** of points to obtain a dense point cloud.

For a known camera pair (P_1, P_2), noisy image points $(\tilde{x}, \tilde{y}) \in \mathbb{P}^2 \times \mathbb{P}^2$ do **not** lie on their joint image, i.e., $\tilde{x}^\top F \tilde{y} \neq 0$ for $F = F_{P_1,P_2}$.

Triangulation

Typically, we need up to **thousands** of points to reconstruct the cameras involved in a 3D scene, but want to reconstruct up to **millions** of points to obtain a dense point cloud.

For a known camera pair (P_1, P_2), noisy image points $(\tilde{x}, \tilde{y}) \in \mathbb{P}^2 \times \mathbb{P}^2$ do **not** lie on their joint image, i.e., $\tilde{x}^\top F \tilde{y} \neq 0$ for $F = F_{P_1,P_2}$. **Triangulation** is the problem of finding the best $X \in \mathbb{P}^3$ such that $(\tilde{x}, \tilde{y}) \approx (P_1 X, P_2 X)$:

Triangulation

Typically, we need up to **thousands** of points to reconstruct the cameras involved in a 3D scene, but want to reconstruct up to **millions** of points to obtain a dense point cloud.

For a known camera pair (P_1, P_2), noisy image points $(\tilde{x}, \tilde{y}) \in \mathbb{P}^2 \times \mathbb{P}^2$ do **not** lie on their joint image, i.e., $\tilde{x}^\top F \tilde{y} \neq 0$ for $F = F_{P_1,P_2}$. **Triangulation** is the problem of finding the best $X \in \mathbb{P}^3$ such that $(\tilde{x}, \tilde{y}) \approx (P_1 X, P_2 X)$:

To make sense of the latter, we pass to affine charts $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, 1)$ and $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, 1)$:

Triangulation

Typically, we need up to **thousands** of points to reconstruct the cameras involved in a 3D scene, but want to reconstruct up to **millions** of points to obtain a dense point cloud.

For a known camera pair (P_1, P_2), noisy image points $(\tilde{x}, \tilde{y}) \in \mathbb{P}^2 \times \mathbb{P}^2$ do **not** lie on their joint image, i.e., $\tilde{x}^\top F \tilde{y} \neq 0$ for $F = F_{P_1,P_2}$. **Triangulation** is the problem of finding the best $X \in \mathbb{P}^3$ such that $(\tilde{x}, \tilde{y}) \approx (P_1 X, P_2 X)$:

To make sense of the latter, we pass to affine charts $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, 1)$ and $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, 1)$:

$$\min_{x_1,x_2,y_1,y_2} (x_1 - \tilde{x}_1)^2 + (x_2 - \tilde{x}_2)^2 + (y_1 - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2,$$

$$(x_1, x_2, 1)\, F\, (y_1, y_2, 1)^\top = 0$$

Triangulation

Typically, we need up to **thousands** of points to reconstruct the cameras involved in a 3D scene, but want to reconstruct up to **millions** of points to obtain a dense point cloud.

For a known camera pair (P_1, P_2), noisy image points $(\tilde{x}, \tilde{y}) \in \mathbb{P}^2 \times \mathbb{P}^2$ do **not** lie on their joint image, i.e., $\tilde{x}^\top F \tilde{y} \neq 0$ for $F = F_{P_1,P_2}$. **Triangulation** is the problem of finding the best $X \in \mathbb{P}^3$ such that $(\tilde{x}, \tilde{y}) \approx (P_1 X, P_2 X)$:

To make sense of the latter, we pass to affine charts $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, 1)$ and $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, 1)$:

$$\min_{x_1,x_2,y_1,y_2} (x_1 - \tilde{x}_1)^2 + (x_2 - \tilde{x}_2)^2 + (y_1 - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2,$$
$$(x_1, x_2, 1)\, F\, (y_1, y_2, 1)^\top = 0$$

This optimization problem has 6 critical points generically,

Triangulation

Typically, we need up to **thousands** of points to reconstruct the cameras involved in a 3D scene, but want to reconstruct up to **millions** of points to obtain a dense point cloud.

For a known camera pair (P_1, P_2), noisy image points $(\tilde{x}, \tilde{y}) \in \mathbb{P}^2 \times \mathbb{P}^2$ do **not** lie on their joint image, i.e., $\tilde{x}^\top F \tilde{y} \neq 0$ for $F = F_{P_1,P_2}$. **Triangulation** is the problem of finding the best $X \in \mathbb{P}^3$ such that $(\tilde{x}, \tilde{y}) \approx (P_1 X, P_2 X)$:

To make sense of the latter, we pass to affine charts $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, 1)$ and $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, 1)$:

$$\min_{x_1,x_2,y_1,y_2} (x_1 - \tilde{x}_1)^2 + (x_2 - \tilde{x}_2)^2 + (y_1 - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2,$$
$$(x_1, x_2, 1)\, F\, (y_1, y_2, 1)^\top = 0$$

This optimization problem has 6 critical points generically, and its Galois width is 6.

Weighted triangulation

Can we find $\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0$ such that

$$\min_{x_1,x_2,y_1,y_2} \lambda_1(x_1 - \tilde{x}_1)^2 + \lambda_2(x_2 - \tilde{x}_2)^2 + \lambda_3(y_1 - \tilde{y}_1)^2 + \lambda_4(y_2 - \tilde{y}_2)^2,$$

$$(x_1, x_2, 1)\, F\, (y_1, y_2, 1)^\top = 0$$

has less critical points?

Weighted triangulation

Can we find $\lambda_1, \lambda_2, \lambda_3, \lambda_4 > 0$ such that

$$\min_{x_1, x_2, y_1, y_2} \lambda_1 (x_1 - \tilde{x}_1)^2 + \lambda_2 (x_2 - \tilde{x}_2)^2 + \lambda_3 (y_1 - \tilde{y}_1)^2 + \lambda_4 (y_2 - \tilde{y}_2)^2,$$
$$(x_1, x_2, 1)\, F\, (y_1, y_2, 1)^\top = 0$$

has less critical points?

Yes!
First, after a coordinate change (via rotating and translating), the original unweighted problem becomes

$$\min_{z_1, \ldots, z_4} (z_1 - \tilde{z}_1)^2 + (z_2 - \tilde{z}_2)^2 + (z_3 - \tilde{z}_3)^2 + (z_4 - \tilde{z}_4)^2,$$
$$a_1 z_1^2 - a_1 z_2^2 + a_2 z_3^2 - a_2 z_4^2 = 0.$$

Weighted triangulation

Theorem (F. Rydell, G. Bökman, F. Kahl, K. Kohn):
The number of critical points of

$$\min_{z_1,\dots,z_4} \lambda_1(z_1-\tilde{z}_1)^2+\lambda_2(z_2-\tilde{z}_2)^2+\lambda_3(z_3-\tilde{z}_3)^2+\lambda_4(z_4-\tilde{z}_4)^2,$$
$$a_1z_1^2-a_1z_2^2+a_2z_3^2-a_2z_4^2=0$$

is generically

- 2 if $\boldsymbol{\lambda}=(\mu a_1,\nu a_1,\mu a_2,\nu a_2)$ for some $\mu,\nu>0$,
- 4 if $(\lambda_1,\lambda_3)=\mu(a_1,a_2)$ for $\mu>0$ or $(\lambda_2,\lambda_4)=\nu(a_1,a_2)$ for $\nu>0$,
- 6 otherwise.

Open problem

The analogous problem for camera triples (P_1, P_2, P_3) given noisy image points $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2$ is

$$\min_{x_1, x_2, y_1, y_2, z_1, z_2} (x_1 - \tilde{x}_1)^2 + (x_2 - \tilde{x}_2)^2 + (y_1 - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2 + (z_1 - \tilde{z}_1)^2 + (z_2 - \tilde{z}_2)^2,$$

$$(x_1, x_2, 1) \equiv P_1 X$$

$$(y_1, y_2, 1) \equiv P_2 X$$

$$(z_1, z_2, 1) \equiv P_3 X$$

$$\text{for some } X \in \mathbb{P}^3.$$

Open problem

The analogous problem for camera triples (P_1, P_2, P_3) given noisy image points $(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2$ is

$$\min_{x_1,x_2,y_1,y_2,z_1,z_2} (x_1 - \tilde{x}_1)^2 + (x_2 - \tilde{x}_2)^2 + (y_1 - \tilde{y}_1)^2 + (y_2 - \tilde{y}_2)^2 + (z_1 - \tilde{z}_1)^2 + (z_2 - \tilde{z}_2)^2,$$

$$(x_1, x_2, 1) \equiv P_1 X$$
$$(y_1, y_2, 1) \equiv P_2 X$$
$$(z_1, z_2, 1) \equiv P_3 X$$
$$\text{for some } X \in \mathbb{P}^3.$$

It has 47 critical points generically, and its Galois width is 47.

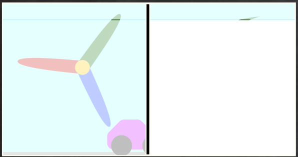
Can you find weights $\lambda_1, \ldots, \lambda_6 > 0$ such that the generic number of critical points is as low as possible? How low can it even get?

important challenge: algebra-geometry foundations of

rolling-shutter cameras that are the vast majority of today's cameras:
take pictures by scanning across the scene, capturing the image row by row

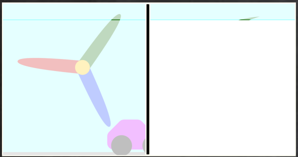
important challenge: algebra-geometry foundations of

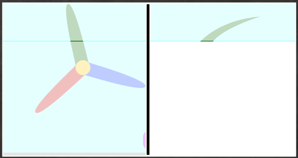
rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row



important challenge: algebra-geometry foundations of

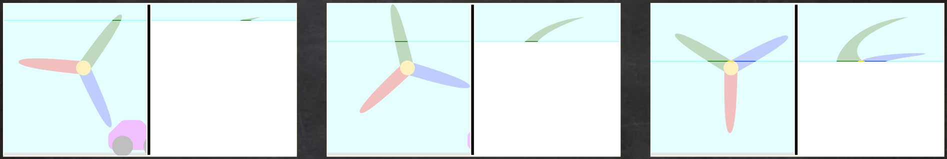
rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row





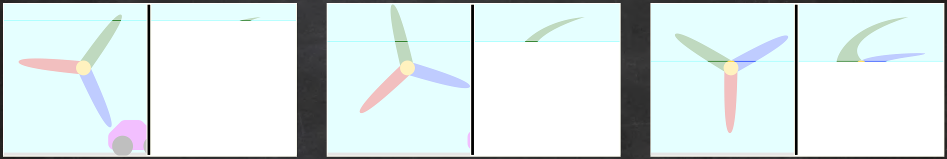
important challenge: algebra-geometry foundations of

rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row



important challenge: algebra-geometry foundations of

rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row

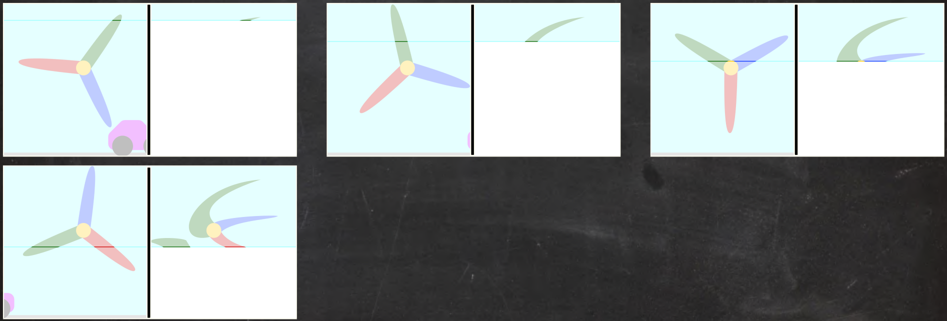


Algebraically:

- The image of a line is typically a higher-degree curve.

important challenge: algebra-geometry foundations of

rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row

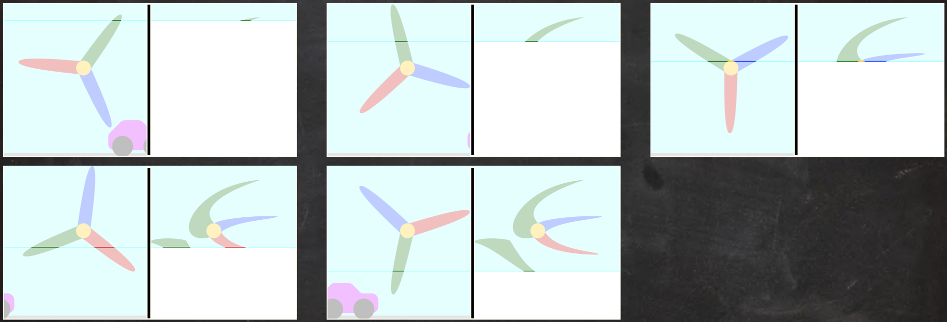


Algebraically:

- The image of a line is typically a higher-degree curve.

important challenge: algebra-geometry foundations of

rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row

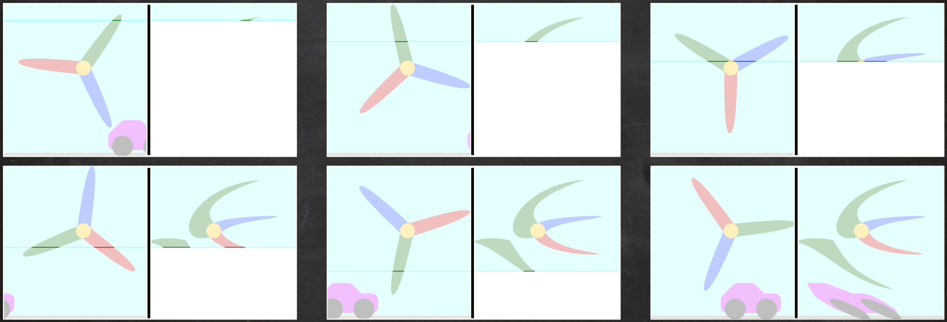


Algebraically:

- The image of a line is typically a higher-degree curve.

important challenge: algebra-geometry foundations of

rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row



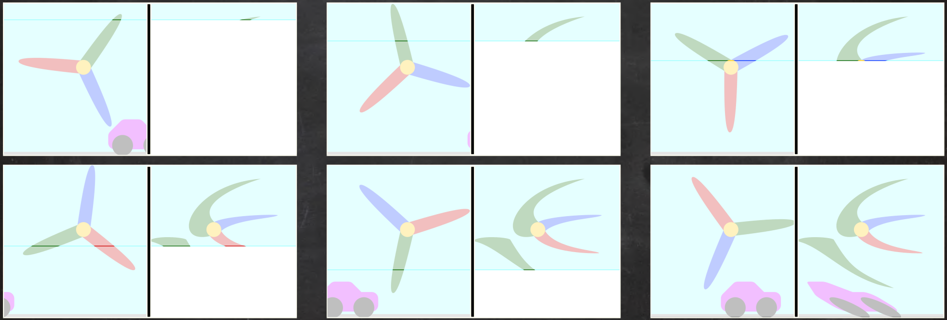
(by Cmglee @ Wikipedia
https://creativecommons.org/licenses/by-sa/3.0/deed.en
changes: added black separating line)

Algebraically:

- The image of a line is typically a higher-degree curve.

important challenge: algebra-geometry foundations of

rolling-shutter cameras that are the vast majority of today's cameras: take pictures by scanning across the scene, capturing the image row by row



(by Cmglee @ Wikipedia

changes: added black separating line)

Algebraically:

- The image of a line is typically a higher-degree curve.
- A 3D point can appear more than once in the image.

Algebraic Neural Network Theory . . .

is the study of neural networks with polynomial (or more generally, piecewise rational) activation function.

Note: They can approximate arbitrary neural networks by Weierstrass approximation.

Algebraic Neural Network Theory . . .

is the study of neural networks with polynomial (or more generally, piecewise rational) activation function.

Note: They can approximate arbitrary neural networks by Weierstrass approximation.

An Algebraic Complexity Theory Problem:
Fix a polynomial $\sigma \in \mathbb{K}[x]$ of degree $r > 1$. A **Multi-Layer Perception (MLP)** with weights $\boldsymbol{W} = (W_1, \ldots, W_L)$, where $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$, is the map $\varphi_{\boldsymbol{W}} : \mathbb{K}^{d_0} \to \mathbb{K}^{d_L}$ given by the composition

$$\varphi_{\boldsymbol{W}} = W_L \circ \sigma \circ \ldots \circ \sigma \circ W_1,$$

where σ is applied coordinate-wise.

Algebraic Neural Network Theory . . .

is the study of neural networks with polynomial (or more generally, piecewise rational) activation function.

Note: They can approximate arbitrary neural networks by Weierstrass approximation.

An Algebraic Complexity Theory Problem:
Fix a polynomial $\sigma \in \mathbb{K}[x]$ of degree $r > 1$. A **Multi-Layer Perception (MLP)** with weights $\boldsymbol{W} = (W_1, \ldots, W_L)$, where $W_i \in \mathbb{R}^{d_i \times d_{i-1}}$, is the map $\varphi_{\boldsymbol{W}} : \mathbb{K}^{d_0} \to \mathbb{K}^{d_L}$ given by the composition

$$\varphi_{\boldsymbol{W}} = W_L \circ \sigma \circ \ldots \circ \sigma \circ W_1,$$

where σ is applied coordinate-wise.

Theorem (V. Shahverdi, G. Marchetti): Let $\mathrm{char}(\mathbb{K}) = 0$ or $> r$. For every $f \in \mathbb{K}[x_1, \ldots, x_{d_0}]^{d_L}$, there is an MLP such that $\varphi_{\boldsymbol{W}} = f$.

What is the smallest such MLP?

Also...

linear convolutional networks arise by composing **convolutional tensors** (generalization of sparse Toeplitz matrices), which is equivalent to multipliying certain **sparse multivariate polynomials**.

Also...

linear convolutional networks arise by composing **convolutional tensors** (generalization of sparse Toeplitz matrices), which is equivalent to multipliying certain **sparse multivariate polynomials**.

Toeplitz matrices correspond to univariate polynomials: For $S \in \mathbb{Z}_{>0}$, let

$$\begin{aligned} \pi_S : \mathbb{R}^k &\longrightarrow \mathbb{R}[x^S]_{\leq k-1}, \\ v &\longmapsto v_0 x^{S(k-1)} + v_1 x^{S(k-2)} + \ldots + v_{k-2} x^S + v_{k-1}. \end{aligned}$$

Then, composing Toeplitz matrices of strides $s_L, \ldots, s_1$ is equivalent to

$$\pi_{S_L}(w_L) \cdots \pi_{S_1}(w_1), \text{ where } S_i := s_1 \cdots s_{i-1}.$$

Open PhD Position in my group on Algebraic Geometry in Neural Network Theory !!!

machine learning	algebraic geometry
sample complexity & expressivity	dimension, degree, covering number
subnetworks & implicit bias	singularities
identifiability & hidden symmetries	fibers of the parametrization
optimization & gradient descent	critical point theory, discriminants, dynamical invariants

An Invitation to Neuroalgebraic Geometry

Giovanni Luca Marchetti [*1] Vahid Shahverdi [*1] Stefano Mereta [*1] Matthew Trager [*2] Kathlén Kohn [*1]

Abstract

In this expository work, we promote the study of function spaces parameterized by machine learning models through the lens of algebraic geometry. To this end, we focus on algebraic models, such as neural networks with polynomial activations, whose associated function spaces are semi-algebraic varieties. We outline a dictionary between algebro-geometric invariants of these varieties, such as dimension, degree, and singularities, and fundamental aspects of machine learning, such as sample complexity, expressivity, training dynamics, and implicit bias.

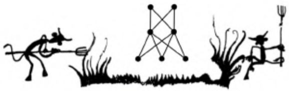


Figure 1. A neural variation of a celebrated doodle from the algebraic geometry literature (Grothendieck, 1968).