New Complexity Bounds for Image Matching under Rotation and Scaling

Christian Hundt[1] Maciej Liśkiewicz[2]

Institut für Informatik, Universität Rostock, Germany

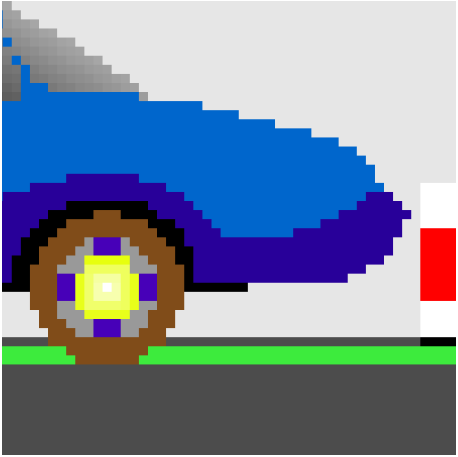
Institut für Theoretische Informatik, Universität zu Lübeck, Germany

22.06.2009

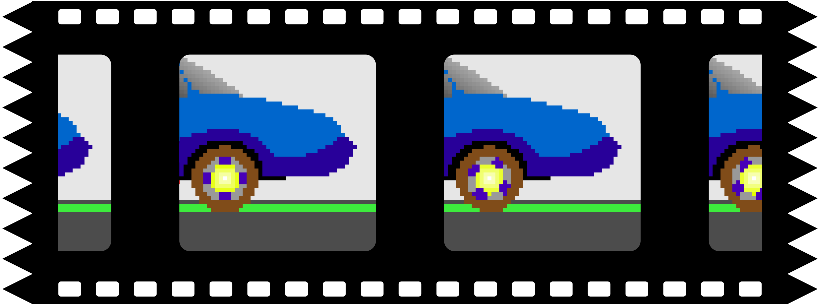
The Image Matching Problem

Remember last year's CPM ...

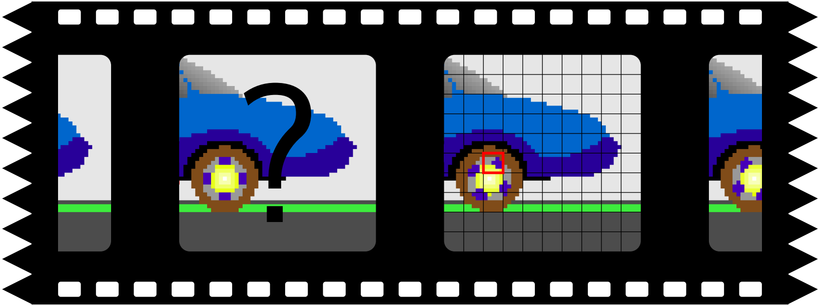
The Image Matching Problem



The Image Matching Problem



The Image Matching Problem



The Image Matching Problem

Given:

any subimage B

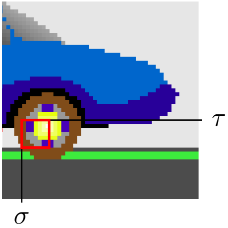
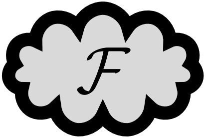
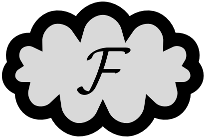


image A



The Image Matching Problem

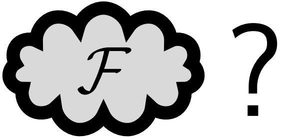
Given: B  $\updownarrow n$ A  $\updownarrow m$



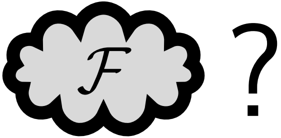
Find: $f \in \mathcal{F}$ $f(A)$ 

Goal: $\Delta(f(A), B) \rightarrow \min$

Transformation Classes



Transformation Classes



Rotation - $\mathcal{F}_{\mathtt{r}}$

$$f(x,y) = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Scaling and Rotation - $\mathcal{F}_{\mathtt{sr}}$

$$f(x,y) = \begin{pmatrix} s\cos\phi & s\sin\phi \\ -s\sin\phi & s\cos\phi \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix}$$

Continuous Problem?

Image Matching seems to be a continuous problem by $\mathcal{F}_{\mathtt{sr}}$.

Continuous Problem?

But it is not!

Continuous Problem?

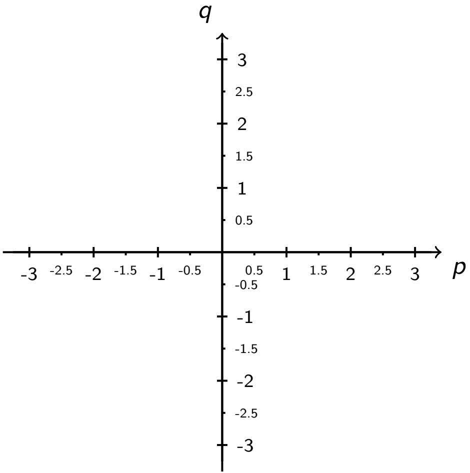
$D(A) = \{f(A) \mid f \in \mathcal{F}_{\mathtt{sr}}\}$ is always finite.

Continuous Problem?

For  $D(A)$ contains

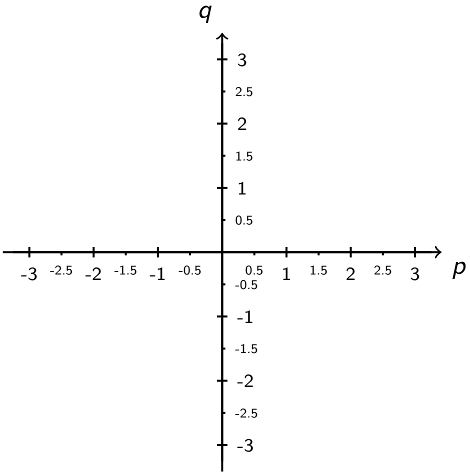
... only 1250 out of hundreds and thousands!

Our Characterization



Let $p = s \cos \phi$ and $q = s \sin \phi$,

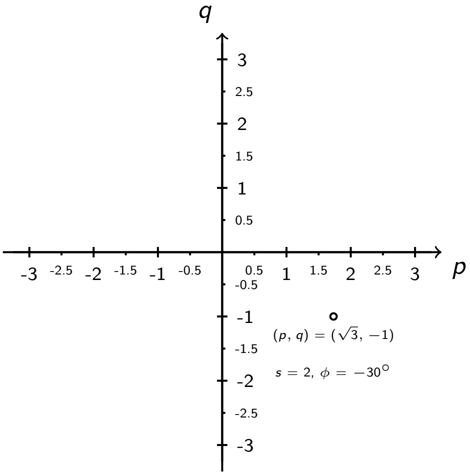
Our Characterization



Let $p = s\cos\phi$ and $q = s\sin\phi$, then (p, q) stands for $f(p, q) \in \mathcal{F}_{\mathtt{sr}}$:

$$f(p, q) = \begin{pmatrix} p & -q \\ q & p \end{pmatrix}$$

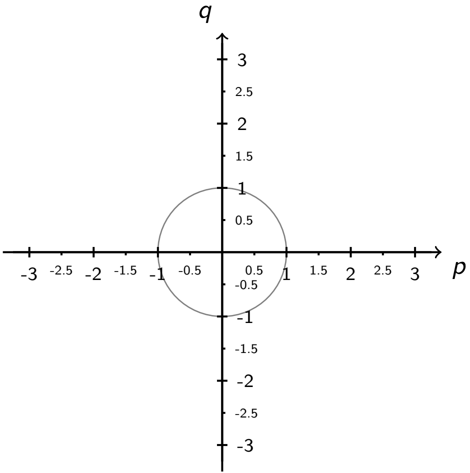
Our Characterization



Let $p = s\cos\phi$ and $q = s\sin\phi$, then (p, q) stands for $f(p, q) \in \mathcal{F}_{\mathtt{sr}}$:

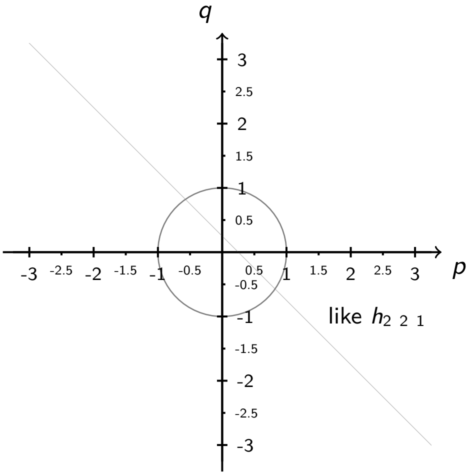
$$f(p, q) = \begin{pmatrix} p & -q \\ q & p \end{pmatrix}$$

Our Characterization



The unit circle (C) stands for transformations with $s = 1$, i.e., for $\mathcal{F}_{\mathtt{r}}$.

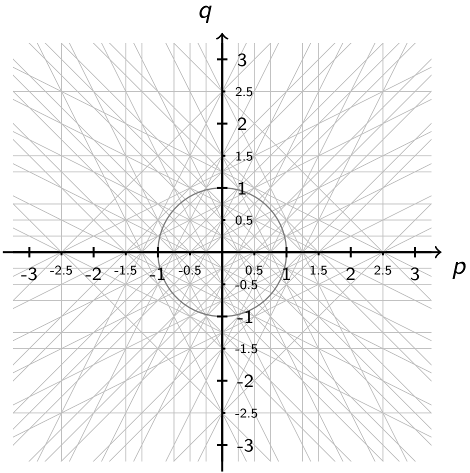
Our Characterization



There are lines

$h_{ijk} : ip + jq = k - 0.5$

Our Characterization

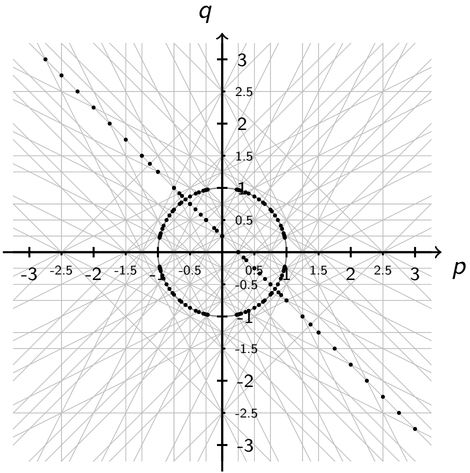


There are lines

$h_{ijk} : ip + jq = k - 0.5$

for $i, j \in \{-n, \ldots, n\}$
and for
$k \in \{-m, \ldots, m+1\}$

Our Characterization



There are lines

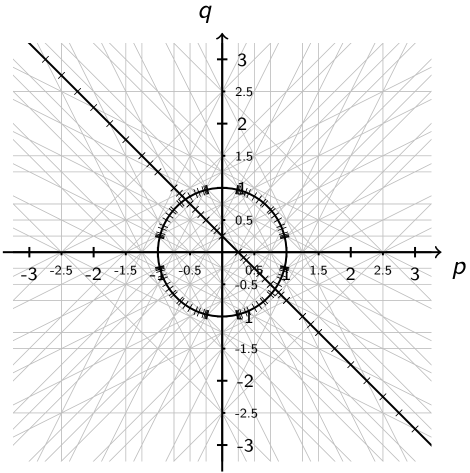
$h_{ijk} : ip + jq = k - 0.5$

for $i, j \in \{-n, \ldots, n\}$
and for
$k \in \{-m, \ldots, m+1\}$

cutting $\mathbb{R}^2$ and C into sets $\mathcal{A}(m,n)$ and $\mathcal{A}_C(m,n)$ of

- points,
- line segments,
- convex regions.

Our Characterization



There are lines

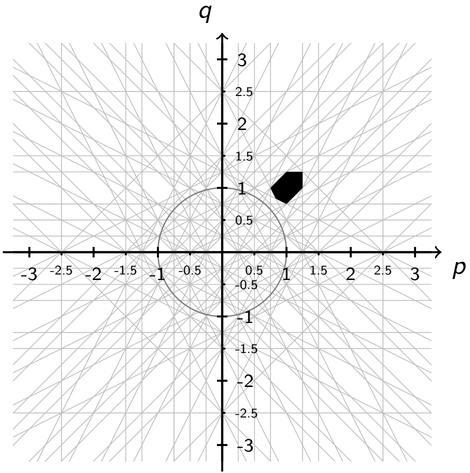
$h_{ijk} : ip + jq = k - 0.5$

for $i, j \in \{-n, \ldots, n\}$
and for
$k \in \{-m, \ldots, m + 1\}$

cutting $\mathbb{R}^2$ and C into sets $\mathcal{A}(m, n)$ and $\mathcal{A}_C(m, n)$ of

- points,
- line segments,
- convex regions.

Our Characterization



There are lines

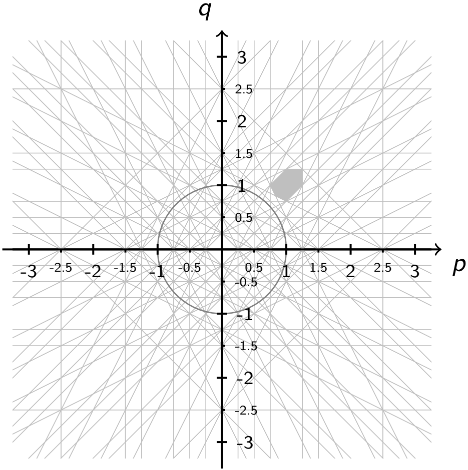
$h_{ijk} : ip + jq = k - 0.5$

for $i, j \in \{-n, \ldots, n\}$
and for
$k \in \{-m, \ldots, m+1\}$

cutting $\mathbb{R}^2$ and C into sets $\mathcal{A}(m,n)$ and $\mathcal{A}_C(m,n)$ of

- points,
- line segments,
- convex regions.

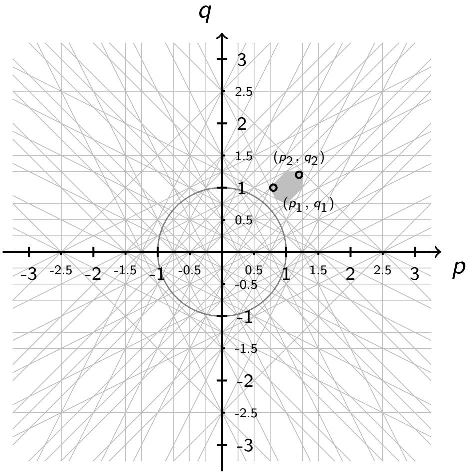
Our Characterization



Theorem

Let $\varphi \in \mathcal{A}(m, n)$

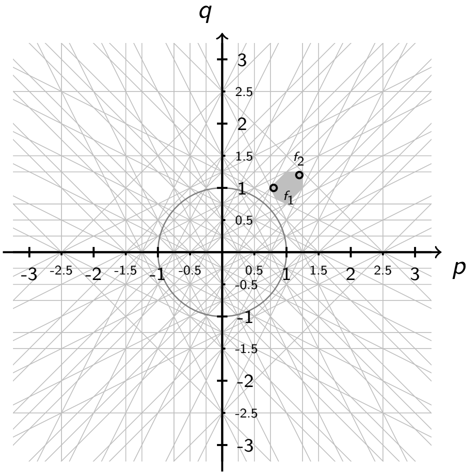
Our Characterization



Theorem

Let $\varphi \in \mathcal{A}(m, n)$ *and* $(p_1, q_1), (p_2, q_2) \in \varphi$

Our Characterization

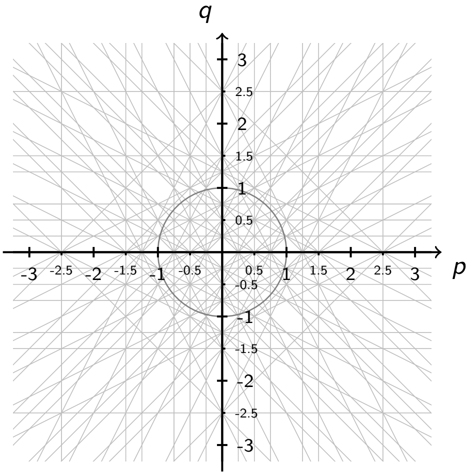


Theorem

Let $\varphi \in \mathcal{A}(m, n)$ *and* $(p_1, q_1), (p_2, q_2) \in \varphi$ *represent* $f_1 = f(p_1, q_1)$ *and* $f_2 = f(p_2, q_2)$.

Then $f_1(A) = f_2(A)$.

Our Characterization



Theorem

Let $\varphi \in \mathcal{A}(m, n)$ *and* $(p_1, q_1), (p_2, q_2) \in \varphi$ *represent* $f_1 = f(p_1, q_1)$ *and* $f_2 = f(p_2, q_2)$.

Then $f_1(A) = f_2(A)$.

The same holds for $\mathcal{A}_C(m, n)$.

Our Characterization

Theorem

If $m \approx n$ *then*

1. $|\mathcal{A}_C(m, n)| \in \Theta(n^3)$,
2. $|\mathcal{A}(m, n)| \in O(n^6) \cap \Omega(n^5)$.

Our Characterization

Theorem

If $m \approx n$ then

1. $|\mathcal{A}_C(m, n)| \in \Theta(n^3)$,
2. $|\mathcal{A}(m, n)| \in O(n^6) \cap \Omega(n^5)$.

Our Characterization

Theorem

If $m \approx n$ *then*

1. $|\mathcal{A}_C(m, n)| \in \Theta(n^3)$,
2. $|\mathcal{A}(m, n)| \in O(n^6) \cap \Omega(n^5)$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.
2.
3.
4.
5.
6.

[1]We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}(m, n)$	$O(n^6)$ (Edelsbrunner) [1]
2.		
3.		
4.		
5.		
6.		

[1] We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}(m, n)$	$O(n^6)$ (Edelsbrunner) [1]
2.	traverse all faces φ of $\mathcal{A}(m, n)$	$O(n^6)$ (DFS)
3.		
4.		
5.		
6.		

[1] We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}(m, n)$	$O(n^6)$ (Edelsbrunner) [1]
2.	traverse all faces φ of $\mathcal{A}(m, n)$	$O(n^6)$ (DFS)
3.	find $(p, q) \in \varphi$	$O(1)$
4.		
5.		
6.		

[1] We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}(m,n)$	$O(n^6)$ (Edelsbrunner) [1]
2.	traverse all faces φ of $\mathcal{A}(m,n)$	$O(n^6)$ (DFS)
3.	find $(p,q) \in \varphi$	$O(1)$
4.	get $f = f(p,q)$ and compute $f(A)$	$O(1)$ (amortized)
5.		
6.		

[1] We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}(m, n)$	$O(n^6)$ (Edelsbrunner) [1]
2.	traverse all faces φ of $\mathcal{A}(m, n)$	$O(n^6)$ (DFS)
3.	find $(p, q) \in \varphi$	$O(1)$
4.	get $f = f(p, q)$ and compute $f(A)$	$O(1)$ (amortized)
5.	obtain $\Delta = (f(A), B)$	$O(1)$ (amortized)
6.		

[1] We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}(m, n)$	$O(n^6)$ (Edelsbrunner) [1]
2.	traverse all faces φ of $\mathcal{A}(m, n)$	$O(n^6)$ (DFS)
3.	find $(p, q) \in \varphi$	$O(1)$
4.	get $f = f(p, q)$ and compute $f(A)$	$O(1)$ (amortized)
5.	obtain $\Delta = (f(A), B)$	$O(1)$ (amortized)
6.	return best transformation f	$O(1)$

[1] We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{sr}}$

Output: $f \in \mathcal{F}_{\mathtt{sr}}$ with minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}(m, n)$	$O(n^6)$ (Edelsbrunner) [1]
2.	traverse all faces φ of $\mathcal{A}(m, n)$	$O(n^6)$ (DFS)
3.	find $(p, q) \in \varphi$	$O(1)$
4.	get $f = f(p, q)$ and compute $f(A)$	$O(1)$ (amortized)
5.	obtain $\Delta = (f(A), B)$	$O(1)$ (amortized)
6.	return best transformation f	$O(1)$
	overall time	$O(n^6)$

[1] We assume $m \approx n$.

Our Image Matching Algorithm

Input: Images A of size m and B of size n

Implicit: Transformation class $\mathcal{F}_{\mathtt{r}}$

Output: $A' = f(A)$ with $f \in \mathcal{F}_{\mathtt{r}}$ and minimum $\Delta = (f(A), B)$

1.	construct $\mathcal{A}_C(m,n)$	$O(n^3 \log n)$ (MergeSort)[2]
2.	traverse all faces φ of $\mathcal{A}_C(m,n)$	$O(n^3)$ (Linear Traversal)
3.	compute $f(A)$ for given φ	$O(1)$ (amortized)
4.	obtain $\Delta = (f(A), B)$	$O(1)$ (amortized)
5.	return A'	$O(1)$
	overall time	$O(n^3 \log n)$

[2] We assume $m \approx n$.

Recently raised Questions

1. How to resolve the $\log n$-discrepancy between $|\mathcal{A}_C(m,n)|$ and the $\mathcal{F}_{\mathtt{r}}$-algorithm's preprocessing time?
2. What is the "exact" cardinality of $\mathcal{A}(m,n)$?

Recently raised Questions

1. How to resolve the $\log n$-discrepancy between $|\mathcal{A}_C(m,n)|$ and the $\mathcal{F}_{\mathtt{r}}$-algorithm's preprocessing time?
2. What is the "exact" cardinality of $\mathcal{A}(m,n)$?

Recently raised Questions

1. How to resolve the $\log n$-discrepancy between $|\mathcal{A}_C(m, n)|$ and the $\mathcal{F}_{\mathtt{r}}$-algorithm's preprocessing time?
2. What is the "exact" cardinality of $\mathcal{A}(m, n)$?
   Same question: How fast does our algorithm run for $\mathcal{F}_{\mathtt{sr}}$?

New Results, partial Answers

Theorem

If $m \approx n$ then

1. *We can preprocess the data structure for $\mathcal{A}_C(m, n)$ in $O(n^3)$ and*
2. $|\mathcal{A}(m, n)| \in O(n^6) \cap \Omega(n^6/\log n)$.

New Results, partial Answers

Theorem

If $m \approx n$ then

1. *We can preprocess the data structure for $\mathcal{A}_C(m,n)$ in $O(n^3)$ and*
2. $|\mathcal{A}(m,n)| \in O(n^6) \cap \Omega(n^6 / \log n)$.

New Results, partial Answers

Theorem

If $m \approx n$ *then*

1. *We can preprocess the data structure for* $\mathcal{A}_C(m, n)$ *in* $O(n^3)$ *and*
2. $|\mathcal{A}(m, n)| \in O(n^6) \cap \Omega(n^6 / \log n)$.

Preprocessing $\mathcal{A}_C(m, n)$ efficiently

Basic Ideas:

1. Points and line segements are given by line intersections with C.
2. So, we will compute the intersections

$$p = \frac{i(k-0.5) \pm j\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

$$q = \frac{j(k-0.5) \mp i\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

   and sort them according to their linear occurence on C.
3. But we will now use RadixSort instead of MergeSort to get linear preprocessing time.
4. Problem: All (p, q) are irrational.
5. Hence, we have to show that $O(\log n)$ precision is enough.
6. Then we can approximate (p, q) by constantly many arithmetic operations of $O(\log n)$ precision.

Preprocessing $\mathcal{A}_C(m, n)$ efficiently

Basic Ideas:

1. Points and line segements are given by line intersections with C.
2. So, we will compute the intersections

$$p = \frac{i(k-0.5) \pm j\sqrt{i^2 + j^2 - (k-0.5)^2}}{i^2 + j^2}$$

$$q = \frac{j(k-0.5) \mp i\sqrt{i^2 + j^2 - (k-0.5)^2}}{i^2 + j^2}$$

   and sort them according to their linear occurence on C.
3. But we will now use RadixSort instead of MergeSort to get linear preprocessing time.
4. Problem: All (p, q) are irrational.
5. Hence, we have to show that $O(\log n)$ precision is enough.
6. Then we can approximate (p, q) by constantly many arithmetic operations of $O(\log n)$ precision.

Preprocessing $\mathcal{A}_C(m, n)$ efficiently

Basic Ideas:

1. Points and line segements are given by line intersections with C.
2. So, we will compute the intersections

$$p = \frac{i(k-0.5) \pm j\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

$$q = \frac{j(k-0.5) \mp i\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

   and sort them according to their linear occurence on C.

3. But we will now use RadixSort instead of MergeSort to get linear preprocessing time.
4. Problem: All (p, q) are irrational.
5. Hence, we have to show that $O(\log n)$ precision is enough.
6. Then we can approximate (p, q) by constantly many arithmetic operations of $O(\log n)$ precision.

Preprocessing $\mathcal{A}_C(m, n)$ efficiently

Basic Ideas:

1. Points and line segements are given by line intersections with C.
2. So, we will compute the intersections

$$p = \frac{i(k-0.5) \pm j\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

$$q = \frac{j(k-0.5) \mp i\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

   and sort them according to their linear occurence on C.

3. But we will now use RadixSort instead of MergeSort to get linear preprocessing time.
4. Problem: All (p, q) are irrational.
5. Hence, we have to show that $O(\log n)$ precision is enough.
6. Then we can approximate (p, q) by constantly many arithmetic operations of $O(\log n)$ precision.

Preprocessing $\mathcal{A}_C(m, n)$ efficiently

Basic Ideas:

1. Points and line segements are given by line intersections with C.
2. So, we will compute the intersections

$$p = \frac{i(k-0.5) \pm j\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

$$q = \frac{j(k-0.5) \mp i\sqrt{i^2+j^2-(k-0.5)^2}}{i^2+j^2}$$

   and sort them according to their linear occurence on C.
3. But we will now use RadixSort instead of MergeSort to get linear preprocessing time.
4. Problem: All (p, q) are irrational.
5. Hence, we have to show that $O(\log n)$ precision is enough.
6. Then we can approximate (p, q) by constantly many arithmetic operations of $O(\log n)$ precision.

Preprocessing $\mathcal{A}_C(m, n)$ efficiently

Basic Ideas:

1. Points and line segements are given by line intersections with C.
2. So, we will compute the intersections

$$p = \frac{i(k-0.5) \pm j\sqrt{i^2 + j^2 - (k-0.5)^2}}{i^2 + j^2}$$

$$q = \frac{j(k-0.5) \mp i\sqrt{i^2 + j^2 - (k-0.5)^2}}{i^2 + j^2}$$

   and sort them according to their linear occurence on C.
3. But we will now use RadixSort instead of MergeSort to get linear preprocessing time.
4. Problem: All (p, q) are irrational.
5. Hence, we have to show that $O(\log n)$ precision is enough.
6. Then we can approximate (p, q) by constantly many arithmetic operations of $O(\log n)$ precision.

Preprocessing $\mathcal{A}_C(m, n)$ efficiently

Basic Ideas:

1. Points and line segements are given by line intersections with C.
2. So, we will compute the intersections

$$p = \frac{i(k-0.5) \pm j\sqrt{i^2 + j^2 - (k-0.5)^2}}{i^2 + j^2}$$
$$q = \frac{j(k-0.5) \mp i\sqrt{i^2 + j^2 - (k-0.5)^2}}{i^2 + j^2}$$

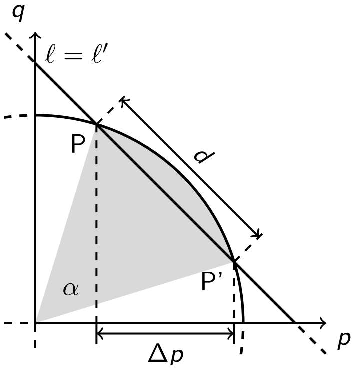
   and sort them according to their linear occurence on C.
3. But we will now use RadixSort instead of MergeSort to get linear preprocessing time.
4. Problem: All (p, q) are irrational.
5. Hence, we have to show that $O(\log n)$ precision is enough.
6. Then we can approximate (p, q) by constantly many arithmetic operations of $O(\log n)$ precision.

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?



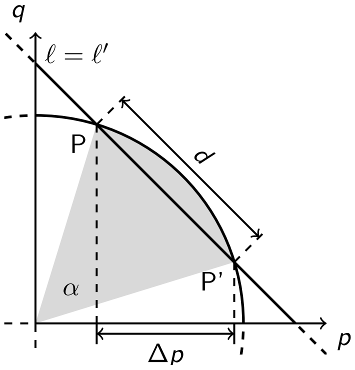
- $i = i'$, $j = j'$, $k = k'$.
- Since $4i^2 + 4j^2 \geq (2k-1)^2$:

$$d = \|P - P'\|$$
$$= \sqrt{\frac{4i^2 + 4j^2 - (2k-1)^2}{i^2 + j^2}}$$
$$\geq \sqrt{\frac{1}{2n^2}} \geq \frac{1}{\sqrt{2}n}.$$

- Hence, d is huge compared to $\Omega(n^{-5})$.

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?



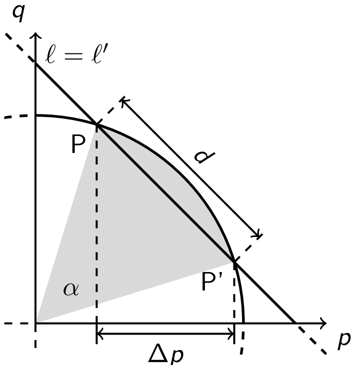
- $i = i'$, $j = j'$, $k = k'$.
- Since $4i^2 + 4j^2 \geq (2k-1)^2$:

$$d = \|P - P'\| = \sqrt{\frac{4i^2 + 4j^2 - (2k-1)^2}{i^2 + j^2}} \geq \sqrt{\frac{1}{2n^2}} \geq \frac{1}{\sqrt{2}n}.$$

- Hence, d is huge compared to $\Omega(n^{-5})$.

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?



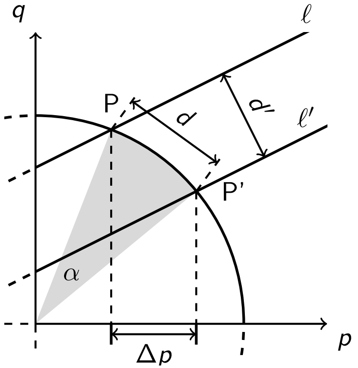
- $i = i'$, $j = j'$, $k = k'$.
- Since $4i^2 + 4j^2 \geq (2k-1)^2$:

$$d = \|P - P'\|$$
$$= \sqrt{\frac{4i^2 + 4j^2 - (2k-1)^2}{i^2 + j^2}}$$
$$\geq \sqrt{\frac{1}{2n^2}} \geq \frac{1}{\sqrt{2}n}.$$

- Hence, d is huge compared to $\Omega(n^{-5})$.

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?



- Point on ℓ': $\begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{2k'-1}{2j'} \end{pmatrix}$.
- Then

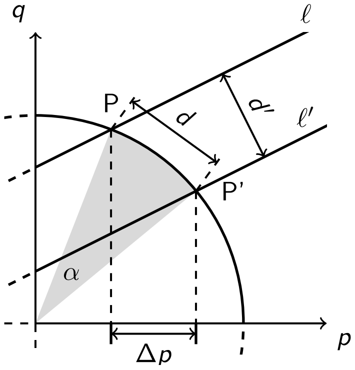
$$d \geq d' = \frac{|ip_0 + jq_0 - (k - 0.5)|}{\sqrt{i^2 + j^2}}$$

$$= \left| \frac{j(2k' - 1) - j'(2k - 1)}{2j'\sqrt{i^2 + j^2}} \right|$$

$$\geq \frac{1}{\sqrt{8}n^2}.$$

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?



- Point on ℓ': $\begin{pmatrix} p_0 \\ q_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \frac{2k'-1}{2j'} \end{pmatrix}$.
- Then

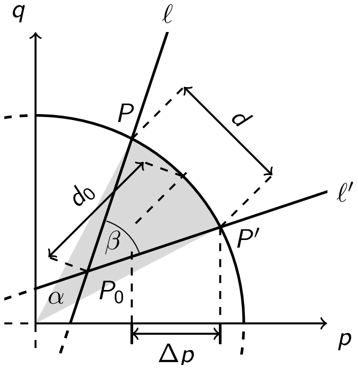
$$d \geq d' = \frac{|ip_0 + jq_0 - (k - 0.5)|}{\sqrt{i^2 + j^2}}$$

$$= \left| \frac{j(2k'-1) - j'(2k-1)}{2j'\sqrt{i^2 + j^2}} \right|$$

$$\geq \frac{1}{\sqrt{8}n^2}.$$

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

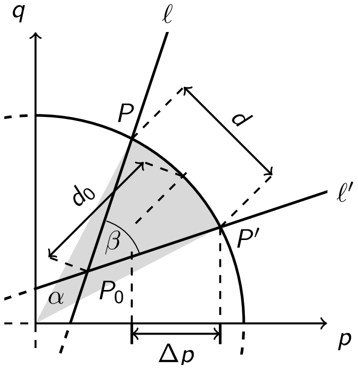


- d is found by distance from P_0 to C and angle β between ℓ and ℓ'.
- We show $d_0 \geq \frac{1}{34n^4}$ and $\beta \geq \frac{1}{n}$.
- $\beta \geq \frac{\pi}{2}$ leads to $d \geq \Omega(n^{-4})$.
- If $\beta < \frac{\pi}{2}$ then isosceles triangle is worst case:

$$d \geq d_0 \tan \frac{\beta}{2} \geq \frac{1}{34n^4} \tan \frac{1}{2n} \geq \frac{1}{68n^5}$$

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

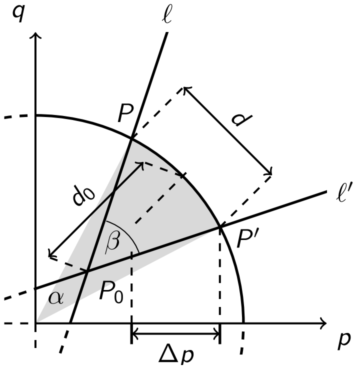


- d is found by distance from P_0 to C and angle β between ℓ and ℓ'.
- We show $d_0 \geq \frac{1}{34n^4}$ and $\beta \geq \frac{1}{n}$.
- $\beta \geq \frac{\pi}{2}$ leads to $d \geq \Omega(n^{-4})$.
- If $\beta < \frac{\pi}{2}$ then isosceles triangle is worst case:

$$d \geq d_0 \tan \frac{\beta}{2} \geq \frac{1}{34n^4} \tan \frac{1}{2n} \geq \frac{1}{68n^5}$$

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

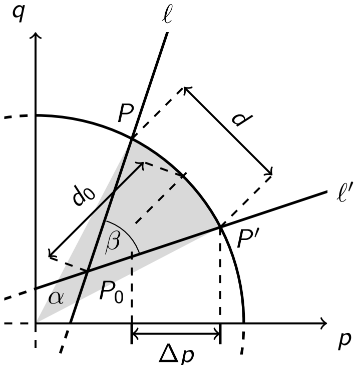


- d is found by distance from P_0 to C and angle β between ℓ and ℓ'.
- We show $d_0 \geq \frac{1}{34n^4}$ and $\beta \geq \frac{1}{n}$.
- $\beta \geq \frac{\pi}{2}$ leads to $d \geq \Omega(n^{-4})$.
- If $\beta < \frac{\pi}{2}$ then isosceles triangle is worst case:

$$d \geq d_0 \tan \frac{\beta}{2} \geq \frac{1}{34n^4} \tan \frac{1}{2n} \geq \frac{1}{68n^5}$$

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?



- d is found by distance from P_0 to C and angle β between ℓ and ℓ'.
- We show $d_0 \geq \frac{1}{34n^4}$ and $\beta \geq \frac{1}{n}$.
- $\beta \geq \frac{\pi}{2}$ leads to $d \geq \Omega(n^{-4})$.
- If $\beta < \frac{\pi}{2}$ then isosceles triangle is worst case:

$$d \geq d_0 \tan \frac{\beta}{2} \geq \frac{1}{34n^4} \tan \frac{1}{2n} \geq \frac{1}{68n^5}$$

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

- The lower bound on d gives that $\alpha \geq \frac{1}{68n^5}$.
- Now Δp can be estimated:

$$\Delta p = 1 - \cos\alpha \geq 1 - \cos\left(\frac{1}{68n^5}\right) \geq 2^{-14}n^{-10}.$$

- The order on p-coordinates gives the order of intersection on C.
- The p-coordinates can be approximated with $O(\log n)$ precision by a constant number of arithmetic operations.

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

- The lower bound on d gives that $\alpha \geq \frac{1}{68n^5}$.
- Now Δp can be estimated:

$$\Delta p = 1 - \cos\alpha \geq 1 - \cos\left(\frac{1}{68n^5}\right) \geq 2^{-14}n^{-10}.$$

- The order on p-coordinates gives the order of intersection on C.
- The p-coordinates can be approximated with $O(\log n)$ precision by a constant number of arithmetic operations.

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

- The lower bound on d gives that $\alpha \geq \frac{1}{68n^5}$.
- Now Δp can be estimated:

$$\Delta p = 1 - \cos\alpha \geq 1 - \cos\left(\frac{1}{68n^5}\right) \geq 2^{-14}n^{-10}.$$

- The order on p-coordinates gives the order of intersection on C.
- The p-coordinates can be approximated with $O(\log n)$ precision by a constant number of arithmetic operations.

The $O(\log n)$ Precision Bound

What ist the minimum distance between the intersections of C with two lines ℓ and ℓ'?

- The lower bound on d gives that $\alpha \geq \frac{1}{68n^5}$.
- Now Δp can be estimated:

$$\Delta p = 1 - \cos\alpha \geq 1 - \cos\left(\frac{1}{68n^5}\right) \geq 2^{-14}n^{-10}.$$

- The order on p-coordinates gives the order of intersection on C.
- The p-coordinates can be approximated with $O(\log n)$ precision by a constant number of arithmetic operations.

The Cardinality of $\mathcal{A}(m, n)$

Basic Ideas:

1. #points ≈ #line segments ≈ #regions
2. So, we will count line segments then.
3. All lines $h_{ijk} : ip + jq = k - 0.5$ are cut into segments by other lines.
4. Hence, count the number of intersection points on h_{ijk} to get the number of its segments:

$$p = \frac{j'(k - 0.5) - j(k' - 0.5)}{ij' - i'j} \quad \text{and} \quad q = \frac{i(k' - 0.5) - i'(k - 0.5)}{ij' - i'j}$$

5. Finally sum over all lines.

The Cardinality of $\mathcal{A}(m, n)$

Basic Ideas:

1. #points $\approx$ #line segments $\approx$ #regions
2. So, we will count line segments then.
3. All lines $h_{ijk} : ip + jq = k - 0.5$ are cut into segments by other lines.
4. Hence, count the number of intersection points on h_{ijk} to get the number of its segments:

$$p = \frac{j'(k - 0.5) - j(k' - 0.5)}{ij' - i'j} \quad \text{and} \quad q = \frac{i(k' - 0.5) - i'(k - 0.5)}{ij' - i'j}$$

5. Finally sum over all lines.

The Cardinality of $\mathcal{A}(m, n)$

Basic Ideas:

1. $\#$points $\approx$ $\#$line segments $\approx$ $\#$regions
2. So, we will count line segments then.
3. All lines $h_{ijk} : ip + jq = k - 0.5$ are cut into segments by other lines.
4. Hence, count the number of intersection points on h_{ijk} to get the number of its segments:

$$p = \frac{j'(k - 0.5) - j(k' - 0.5)}{ij' - i'j} \quad \text{and} \quad q = \frac{i(k' - 0.5) - i'(k - 0.5)}{ij' - i'j}$$

5. Finally sum over all lines.

The Cardinality of $\mathcal{A}(m, n)$

Basic Ideas:

1. #points $\approx$ #line segments $\approx$ #regions
2. So, we will count line segments then.
3. All lines $h_{ijk} : ip + jq = k - 0.5$ are cut into segments by other lines.
4. Hence, count the number of intersection points on h_{ijk} to get the number of its segments:

$$p = \frac{j'(k - 0.5) - j(k' - 0.5)}{ij' - i'j} \quad \text{and} \quad q = \frac{i(k' - 0.5) - i'(k - 0.5)}{ij' - i'j}$$

5. Finally sum over all lines.

The Cardinality of $\mathcal{A}(m, n)$

Basic Ideas:

1. #points ≈ #line segments ≈ #regions
2. So, we will count line segments then.
3. All lines $h_{ijk} : ip + jq = k - 0.5$ are cut into segments by other lines.
4. Hence, count the number of intersection points on h_{ijk} to get the number of its segments:

$$p = \frac{j'(k-0.5) - j(k'-0.5)}{ij' - i'j} \quad \text{and} \quad q = \frac{i(k'-0.5) - i'(k-0.5)}{ij' - i'j}$$

5. Finally sum over all lines.

The Cardinality of $\mathcal{A}(m, n)$

Basic Ideas:

1. #points ≈ #line segments ≈ #regions
2. So, we will count line segments then.
3. All lines $h_{ijk} : ip + jq = k - 0.5$ are cut into segments by other lines.
4. Hence, count the number of intersection points on h_{ijk} to get the number of its segments:

$$p = \frac{j'(k - 0.5) - j(k' - 0.5)}{ij' - i'j} \quad \text{and} \quad q = \frac{i(k' - 0.5) - i'(k - 0.5)}{ij' - i'j}$$

5. Finally sum over all lines.

The Cardinality of $\mathcal{A}(m, n)$

Make it easier:

- $H_a(n) = \{(i, j, k) \mid 0.5n \leq i, j \leq n, 1 \leq k \leq 0.01n, \gcd(i, j) = 1\}$
- $H_b(n) = \{(i, j, k) \mid -n \leq i, j \leq n, 1 \leq k \leq 0.01n\}$

Consider only segments obtained by cutting lines $h_{i(-j)k}$ with $h_{i'j'k'}$ where $(i, j, k) \in H_a(n)$ and $(i', j', k') \in H_b(n)$.

The Cardinality of $\mathcal{A}(m, n)$

This means: Count for each $(i, j, k) \in H_a(n)$ the number of $(i', j', k') \in H_b(n)$, which give a new fraction:

$$p = \frac{N}{D} = \frac{j'(k - 0.5) + j(k' - 0.5)}{ij' + i'j}$$

1. We have i and j coprime und thus use Bézout's Lemma.
2. Then D can be any number in $\{1, \ldots, j \cdot n\}$ by the choice of i', j'.
3. Particularly, any of the $\Omega(n^2/\log n)$ primes in $\{0.04n^2, \ldots, 0.5n^2\}$.
4. By k' we can still choose $0.01n$ many numerators with $N < 0.04n^2$.
5. That gives $\Omega(n^3/\log n)$ segments on $h_{i(-j)k}$.
6. Since $|H_a(n)| \in \Omega(n^3)$ we get $\Omega(n^6/\log n)$ segments.

The Cardinality of $\mathcal{A}(m, n)$

This means: Count for each $(i, j, k) \in H_a(n)$ the number of $(i', j', k') \in H_b(n)$, which give a new fraction:

$$p = \frac{N}{D} = \frac{j'(k - 0.5) + j(k' - 0.5)}{ij' + i'j}$$

1. We have i and j coprime und thus use Bézout's Lemma.
2. Then D can be any number in $\{1, \ldots, j \cdot n\}$ by the choice of i', j'.
3. Particularly, any of the $\Omega(n^2/\log n)$ primes in $\{0.04n^2, \ldots, 0.5n^2\}$.
4. By k' we can still choose $0.01n$ many numerators with $N < 0.04n^2$.
5. That gives $\Omega(n^3/\log n)$ segments on $h_{i(-j)k}$.
6. Since $|H_a(n)| \in \Omega(n^3)$ we get $\Omega(n^6/\log n)$ segments.

The Cardinality of $\mathcal{A}(m, n)$

This means: Count for each $(i, j, k) \in H_a(n)$ the number of $(i', j', k') \in H_b(n)$, which give a new fraction:

$$p = \frac{N}{D} = \frac{j'(k - 0.5) + j(k' - 0.5)}{ij' + i'j}$$

1. We have i and j coprime und thus use Bézout's Lemma.
2. Then D can be any number in $\{1, \ldots, j \cdot n\}$ by the choice of i', j'.
3. Particularly, any of the $\Omega(n^2/\log n)$ primes in $\{0.04n^2, \ldots, 0.5n^2\}$.
4. By k' we can still choose $0.01n$ many numerators with $N < 0.04n^2$.
5. That gives $\Omega(n^3/\log n)$ segments on $h_{i(-j)k}$.
6. Since $|H_a(n)| \in \Omega(n^3)$ we get $\Omega(n^6/\log n)$ segments.

The Cardinality of $\mathcal{A}(m, n)$

This means: Count for each $(i, j, k) \in H_a(n)$ the number of $(i', j', k') \in H_b(n)$, which give a new fraction:

$$p = \frac{N}{D} = \frac{j'(k - 0.5) + j(k' - 0.5)}{ij' + i'j}$$

1. We have i and j coprime und thus use Bézout's Lemma.
2. Then D can be any number in $\{1, \ldots, j \cdot n\}$ by the choice of i', j'.
3. Particularly, any of the $\Omega(n^2/\log n)$ primes in $\{0.04n^2, \ldots, 0.5n^2\}$.
4. By k' we can still choose $0.01n$ many numerators with $N < 0.04n^2$.
5. That gives $\Omega(n^3/\log n)$ segments on $h_{i(-j)k}$.
6. Since $|H_a(n)| \in \Omega(n^3)$ we get $\Omega(n^6/\log n)$ segments.

The Cardinality of $\mathcal{A}(m, n)$

This means: Count for each $(i, j, k) \in H_a(n)$ the number of $(i', j', k') \in H_b(n)$, which give a new fraction:

$$p = \frac{N}{D} = \frac{j'(k - 0.5) + j(k' - 0.5)}{ij' + i'j}$$

1. We have i and j coprime und thus use Bézout's Lemma.
2. Then D can be any number in $\{1, \ldots, j \cdot n\}$ by the choice of i', j'.
3. Particularly, any of the $\Omega(n^2/\log n)$ primes in $\{0.04n^2, \ldots, 0.5n^2\}$.
4. By k' we can still choose $0.01n$ many numerators with $N < 0.04n^2$.
5. That gives $\Omega(n^3/\log n)$ segments on $h_{i(-j)k}$.
6. Since $|H_a(n)| \in \Omega(n^3)$ we get $\Omega(n^6/\log n)$ segments.

The Cardinality of $\mathcal{A}(m, n)$

This means: Count for each $(i, j, k) \in H_a(n)$ the number of $(i', j', k') \in H_b(n)$, which give a new fraction:

$$p = \frac{N}{D} = \frac{j'(k - 0.5) + j(k' - 0.5)}{ij' + i'j}$$

1. We have i and j coprime und thus use Bézout's Lemma.
2. Then D can be any number in $\{1, \ldots, j \cdot n\}$ by the choice of i', j'.
3. Particularly, any of the $\Omega(n^2 / \log n)$ primes in $\{0.04n^2, \ldots, 0.5n^2\}$.
4. By k' we can still choose $0.01n$ many numerators with $N < 0.04n^2$.
5. That gives $\Omega(n^3 / \log n)$ segments on $h_{i(-j)k}$.
6. Since $|H_a(n)| \in \Omega(n^3)$ we get $\Omega(n^6 / \log n)$ segments.

The Cardinality of $\mathcal{A}(m, n)$

This means: Count for each $(i, j, k) \in H_a(n)$ the number of $(i', j', k') \in H_b(n)$, which give a new fraction:

$$p = \frac{N}{D} = \frac{j'(k - 0.5) + j(k' - 0.5)}{ij' + i'j}$$

1. We have i and j coprime und thus use Bézout's Lemma.
2. Then D can be any number in $\{1, \ldots, j \cdot n\}$ by the choice of i', j'.
3. Particularly, any of the $\Omega(n^2/\log n)$ primes in $\{0.04n^2, \ldots, 0.5n^2\}$.
4. By k' we can still choose $0.01n$ many numerators with $N < 0.04n^2$.
5. That gives $\Omega(n^3/\log n)$ segments on $h_{i(-j)k}$.
6. Since $|H_a(n)| \in \Omega(n^3)$ we get $\Omega(n^6/\log n)$ segments.

End of Talk

Thank you for your attention!