



SEAM CARVING

Kompetenslunch by Hampus Londögård @ AFRY

MALMÖ IT KOMPETENS

IT South at AFRY
The blog

Posts Competence Tipsrundan Categories Tags About Q

Welcome to IT South @ AFRYs internal blog!
Here you'll find everything from blog-posts related to competence or other interesting things to our newsletter ([Tipsrundan](#)).

Enjoy!

If you bump into issues: Contact info available in [about](#).

Recent Posts

[Streamlit - Unifying Backend & Frontend for MVPs](#)

📅 April 20, 2021 · 🕒 1 minute read

Streamlit - A recommendation of the quickest MVP tool out there unifying backend & frontend in one piece!

Competence Groups

Machine Learning

Algorithms

Bokcirkeln

Front-End

Computer Security

 **Tipsrundan**

Your biweekly newsletter of 'Tips' from **IT South@AFRY** with ❤️

👉 Tipsboxen ([Email](#), [GitHub](#) & [Slack](#)) - All tips are appreciated! 🙏

BLOGGEN

Contributors needed!

KOMPETENSGRUPPER

Join a group
Create a group

ANNAT

Presentationer
Tipsrundan-tips

TABLE OF CONTENTS

- 01 SEAM CARVING
What is it and why do I care?
- 02 ALGORITHM OVERVIEW
A “bird-view” of the algorithm
- 03 ALGORITHM IMPLEMENTATION
Implementing the algorithm
on the JVM with BoofCV
- 04 RECAP
Briefly repeat everything and
sharing sources



IMAGES

Matrixes of pixels

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	25	100	254	205	254	170	103	100	96	0	0	0
2	0	0	0	84	100	254	253	254	106	213	203	253	253	253	0	0	0	0
3	0	91	244	249	253	100	95	10	8	4	10	104	253	253	253	0	0	0
4	0	107	253	253	253	100	0	0	0	0	0	100	253	253	100	0	0	0
5	0	3	20	20	10	0	0	0	0	0	0	41	224	253	249	74	0	0
6	0	0	0	0	0	0	0	0	0	0	0	205	205	249	25	0	0	0
7	0	0	0	0	0	0	0	14	101	103	253	249	249	249	0	0	0	0
8	0	0	0	0	0	0	10	200	253	253	253	101	101	101	0	0	0	0
9	0	0	0	0	0	0	10	249	249	253	253	253	252	253	101	2	0	0
10	0	0	0	0	0	0	0	43	96	96	200	253	253	253	253	107	10	0

00 BACKGROUND



IMAGES

Matrixes

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	29	150	195	254	255	254	176	193	150	98	0	0	0
2	0	0	0	48	166	224	253	253	234	198	253	253	253	253	233	0	0	0
3	0	93	244	240	253	187	48	10	8	4	10	194	253	253	233	0	0	0
4	0	107	253	253	230	48	0	0	0	0	0	192	253	253	156	0	0	0
5	0	3	20	20	15	0	0	0	0	0	43	224	253	245	74	0	0	0
6	0	0	0	0	0	0	0	0	0	0	240	253	245	126	0	0	0	0
7	0	0	0	0	0	0	0	14	101	223	253	248	124	0	0	0	0	0
8	0	0	0	0	0	11	160	239	253	253	253	187	30	0	0	0	0	0
9	0	0	0	0	0	16	248	250	253	253	253	253	232	213	111	2	0	0
10	0	0	0	0	0	0	0	43	98	98	208	253	253	253	253	187	22	0



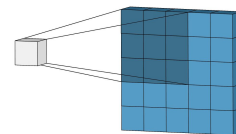
IMAGES

Matrixes of pixels

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	25	50	254	205	254	170	193	100	96	0	0	0
2	0	0	0	84	104	255	255	234	186	213	203	253	253	233	0	0	0	0
3	0	91	244	249	253	184	95	10	8	4	37	104	255	253	233	0	0	0
4	0	107	253	253	225	95	0	0	0	0	0	100	255	253	184	0	0	0
5	0	3	20	20	16	0	0	0	0	0	0	43	224	255	249	74	0	0
6	0	0	0	0	0	0	0	0	0	0	0	205	205	245	25	0	0	0
7	0	0	0	0	0	0	0	14	111	223	253	246	246	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	43	96	96	206	253	253	253	197	25	0	0



KERNEL



Better GIF

00 BACKGROUND



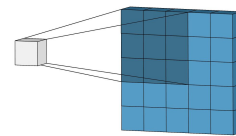
IMAGES

Matrixes of pixels

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	25	50	100	254	255	254	190	100	50	25	0	0	0
2	0	0	0	88	100	255	254	190	100	255	254	190	100	255	254	190	100	0
3	0	95	244	249	253	190	100	100	100	100	100	100	100	100	100	100	100	0
4	0	107	253	253	255	100	0	0	0	0	0	0	0	0	0	0	0	0
5	0	3	30	30	10	0	0	0	0	0	0	41	224	255	249	100	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	205	205	249	100	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	14	100	223	249	100	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



KERNEL



Better GIF

OO BACKGROUND



GRADIENT

Gradient of colors = change
Find Edges this way

≈ A derivative



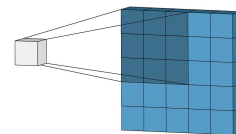
IMAGES

Matrixes of pixels

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	25	50	100	254	255	254	170	102	50	0	0	0	0
2	0	0	0	84	168	254	255	254	166	253	203	253	253	253	0	0	0	0
3	0	91	241	249	253	188	165	102	8	4	32	104	253	253	253	0	0	0
4	0	107	253	253	255	165	0	0	0	0	0	100	253	253	166	0	0	0
5	0	3	20	20	16	0	0	0	0	0	0	42	224	253	249	74	0	0
6	0	0	0	0	0	0	0	0	0	0	0	205	205	245	25	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	6	14	111	223	246	0	0
8	0	0	0	0	0	0	16	246	254	253	253	181	36	0	0	0	0	0
9	0	0	0	0	0	0	0	249	249	253	253	253	253	252	253	161	2	0
10	0	0	0	0	0	0	0	0	43	84	84	206	253	253	253	197	25	0



KERNEL



Better GIF

OO BACKGROUND



GRADIENT

Gradient of colors = change
Find Edges this way

≈ A derivative



DYNAMIC PROGRAMMING

Break down into subproblem
Subproblem solves the large one

Think: Puzzle

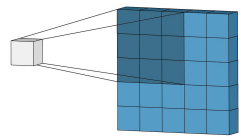


find edges this way

≈ A derivative



KERNEL



Better GIF



AMIC PROGRAMMING

break down into subproblem
Subproblem solves the large one

Think: Puzzle

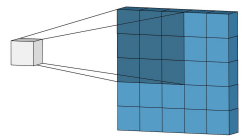


find Edges this way

≈ A derivative



KERNEL



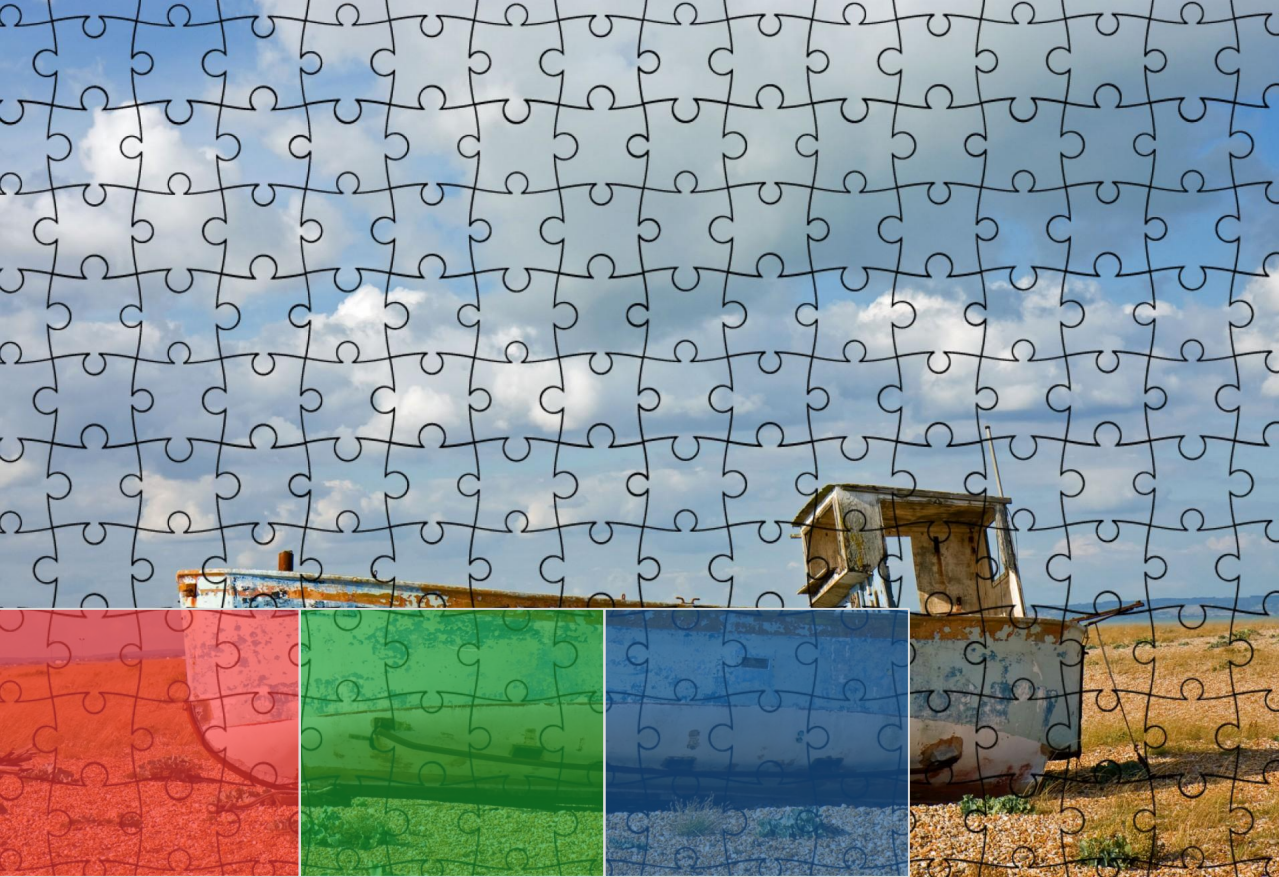
Better GIF



AMIC PROGRAMMING

break down into subproblem
Subproblem solves the large one

Think: Puzzle

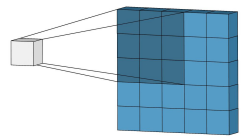


find Edges this way

≈ A derivative



KERNEL



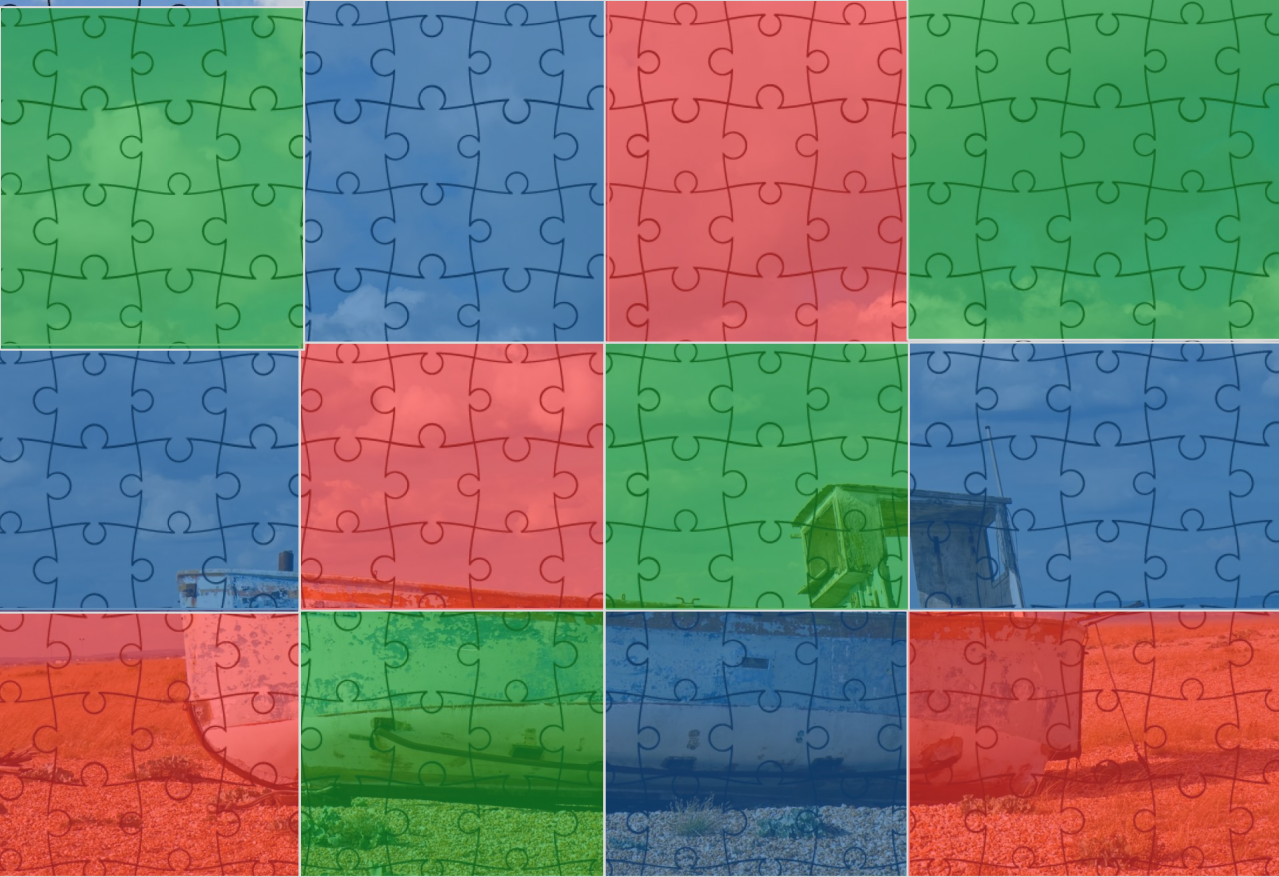
Better GIF



DYNAMIC PROGRAMMING

break down into subproblem
Subproblem solves the large one

Think: Puzzle

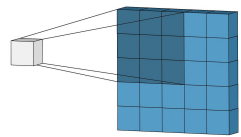


find Edges this way

≈ A derivative



KERNEL



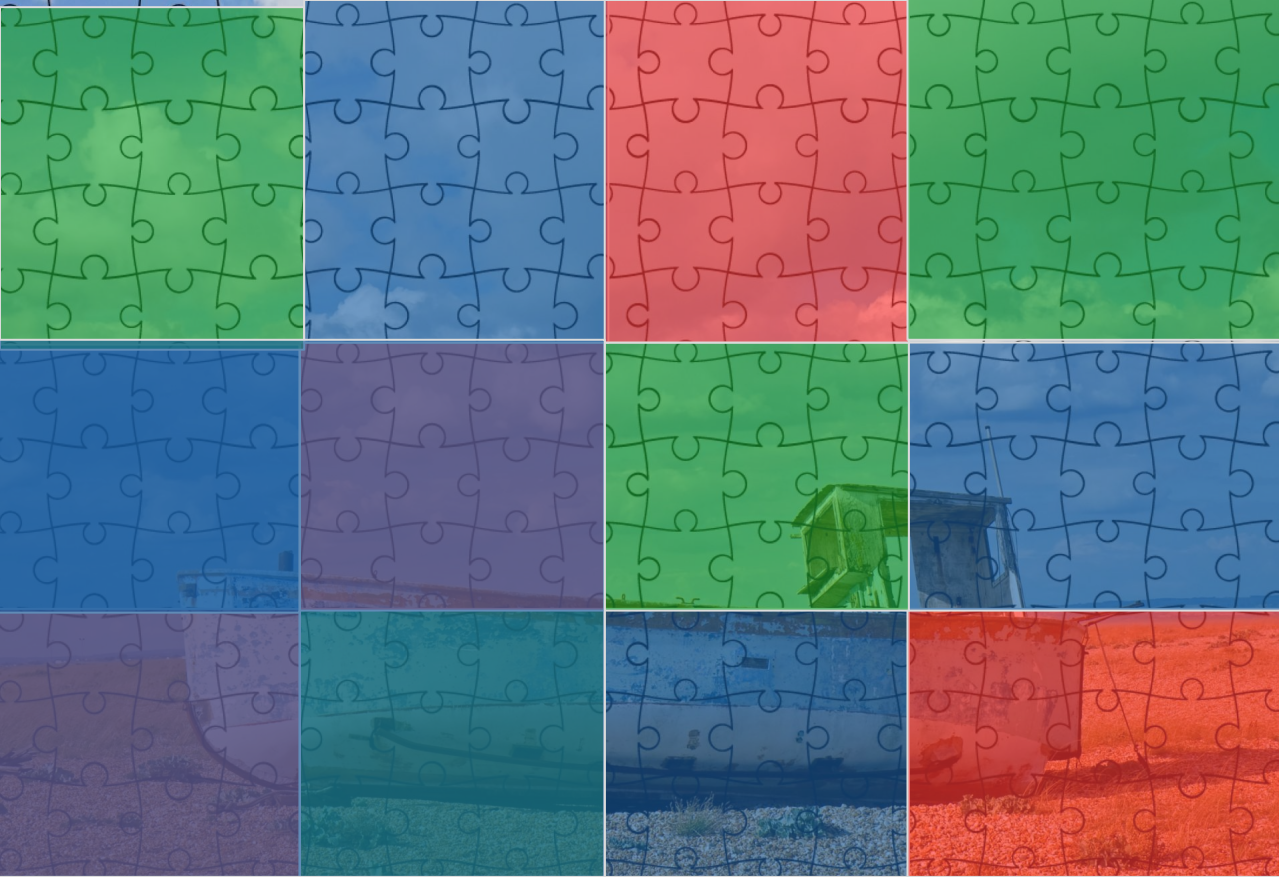
Better GIF



AMIC PROGRAMMING

break down into subproblem
Subproblem solves the large one

Think: Puzzle

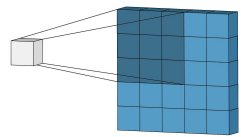


find Edges this way

≈ A derivative



KERNEL



Better GIF



AMIC PROGRAMMING

break down into subproblem
Subproblem solves the large one

Think: Puzzle



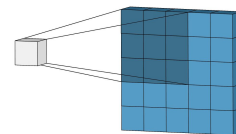
IMAGES

Matrixes of pixels

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	25	50	100	254	255	254	170	102	50	0	0	0	0
2	0	0	0	84	168	254	255	254	166	253	203	253	253	253	0	0	0	0
3	0	91	241	249	253	188	95	50	8	4	57	104	253	253	253	0	0	0
4	0	107	253	253	255	165	0	0	0	0	0	100	253	253	166	0	0	0
5	0	3	20	20	16	0	0	0	0	0	0	41	224	253	249	74	0	0
6	0	0	0	0	0	0	0	0	0	0	0	205	205	245	25	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	6	14	111	223	246	0	0
8	0	0	0	0	0	0	16	246	254	253	253	181	96	0	0	0	0	0
9	0	0	0	0	0	0	16	246	249	253	253	203	232	213	161	7	0	0
10	0	0	0	0	0	0	9	43	96	96	206	253	253	253	197	25	0	0



KERNEL



Better GIF

00 BACKGROUND



GRADIENT

Gradient of colors = change
Find Edges this way

≈ A derivative



DYNAMIC PROGRAMMING

Break down into subproblem
Subproblem solves the large one

Think: Puzzle



01

SEAM CARVING

What the actual f#!k is *Seam Carving*?



SEAM CARVING

Remove “empty space” out of an image to reduce either *width* or *height*.





SEAM CARVING

Remove “empty space” out of an image to reduce either *width* or *height*.

...or in Photoshop: *Content Aware Scaling*





SEAM CARVING

Remove “empty space” out of an image to reduce either *width* or *height*.

...or in Photoshop: *Content Aware Scaling™*





SEAM CARVING

Remove “empty space” out of an image to reduce either *width* or *height*.

...or in Photoshop: *Content Aware Scaling™*



GIF



02 ALGORITHM

An overview



STEPS



IMAGE

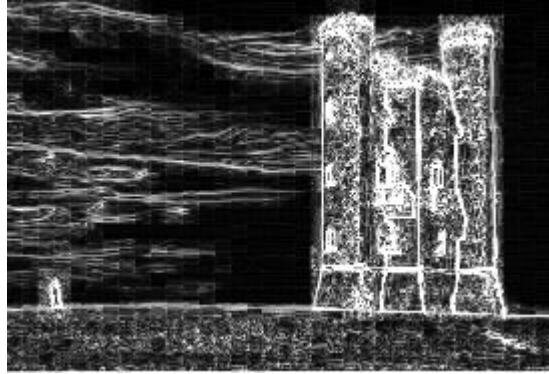
Have an image where you wish
to remove “empty space”

STEPS



IMAGE

Have an image where you wish to remove “empty space”



FIND EDGES

Calculate what’s called “intensity”

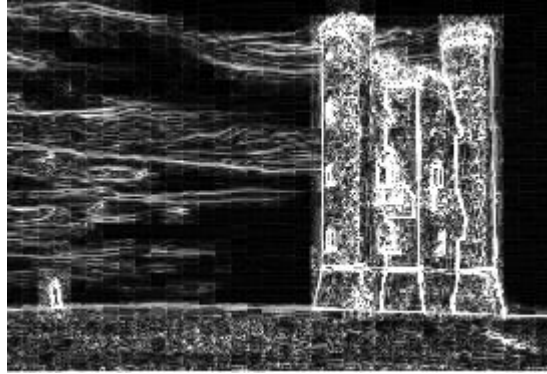
(e.g. Sobelfilter)

STEPS



IMAGE

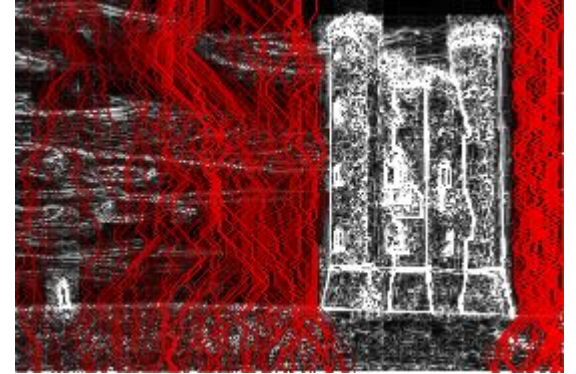
Have an image where you wish to remove “empty space”



FIND EDGES

Calculate what’s called “intensity”

(e.g. Sobelfilter)



FIND PATH(S)

And remove that line of pixels

03 IMPLEMENTATION

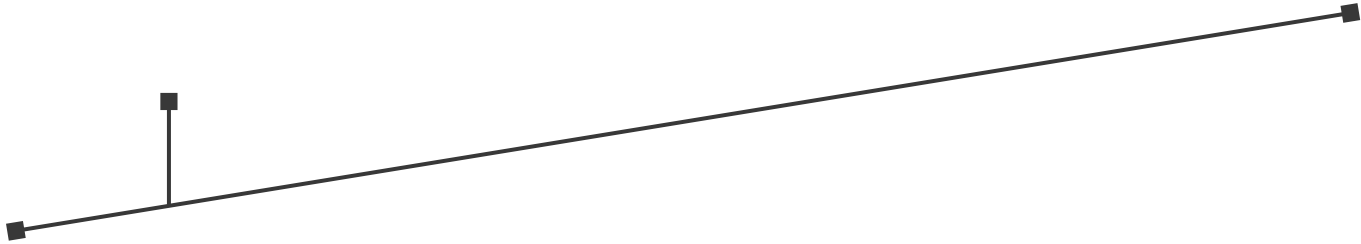
Let's jump into live-coding 🥳

04 RECAP

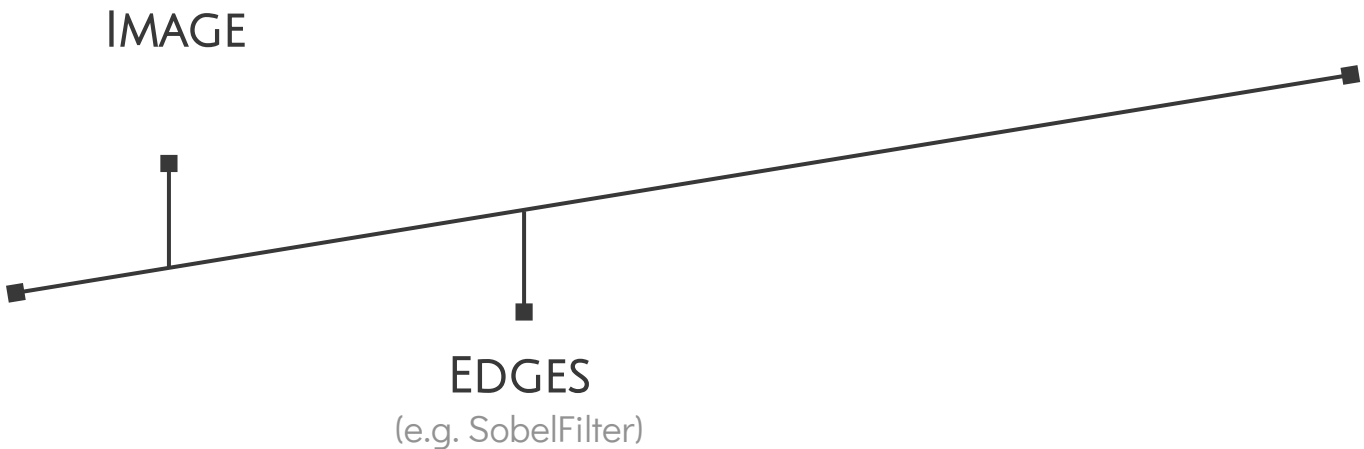
Let's go through the algorithm again

ALGORITHM

IMAGE



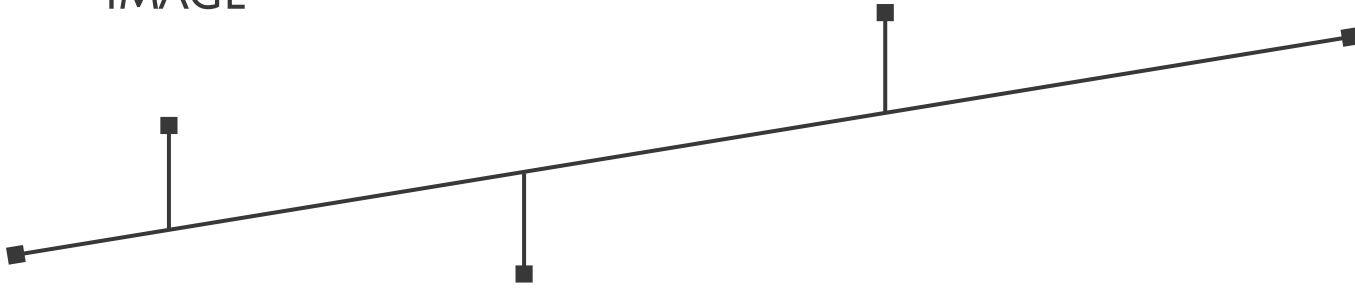
ALGORITHM



ALGORITHM

ENERGY MATRIX

IMAGE



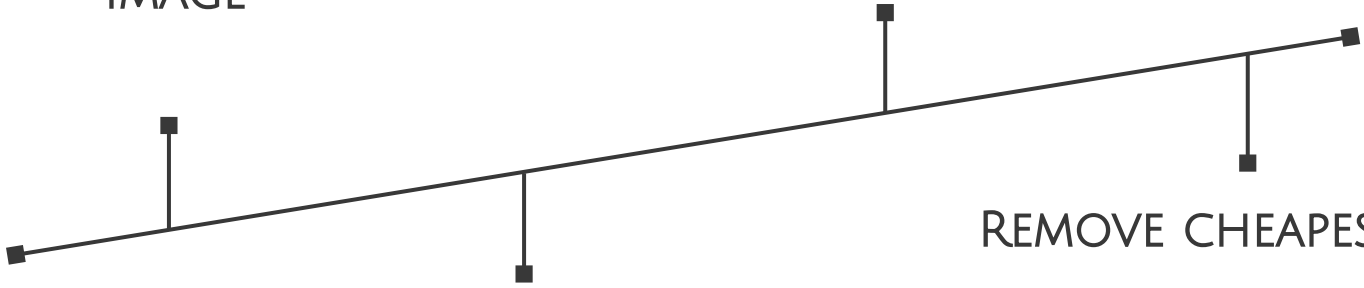
EDGES

(e.g. SobelFilter)

ALGORITHM

ENERGY MATRIX

IMAGE



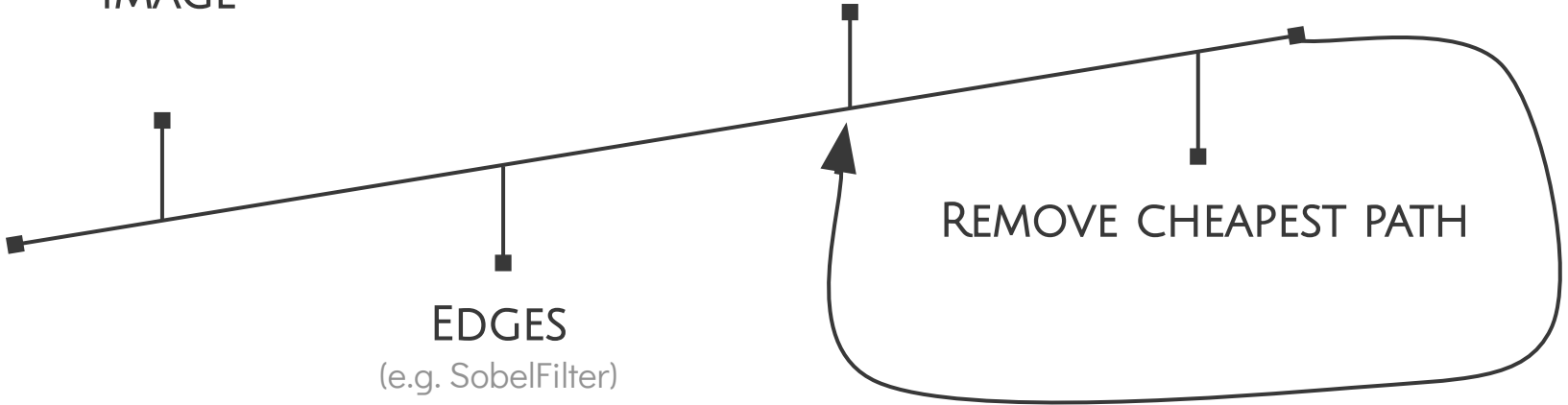
EDGES
(e.g. SobelFilter)

REMOVE CHEAPEST PATH

ALGORITHM

ENERGY MATRIX

IMAGE



EDGES
(e.g. SobelFilter)

REMOVE CHEAPEST PATH

LOOP TILL HAPPY

THANKS

Do you have any question?

hampus.londogard@afry.com

0733 673 179

afry.com

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, infographics & images by **Freepik**

Please keep this slide for attribution