Analysis of Block Matching Algorithms for Image Transformation

ADA Final Project: Questions 5 - 10

Alejandro Goicochea   Diego Linares   Ariana Villegas

Universidad de Ingeniería y Tecnología

July 23, 2020

Table of Contents

1. Recap of Block Matching
2. Decoding Images through Luma
   - What is Luma?
   - Transforming into a Matrix of bits
3. Our Approach
   - Base Process for Transformation
   - Different Algorithms
4. Examples
5. Conclusion

Table of Contents

1. Recap of Block Matching
2. Decoding Images through Luma
   - What is Luma?
   - Transforming into a Matrix of bits
3. Our Approach
   - Base Process for Transformation
   - Different Algorithms
4. Examples
5. Conclusion

Previous Presentation

In the previous presentation we designed **two algorithms** for block matching:

- Greedy / Naive Algorithm
- Memoized Algorithm (now improved to DP version).

For this last part we are including a third algorithm, a DP with **better weight**.

We are gonna see how each of this performs for image transformation.

Greedy Algorithm

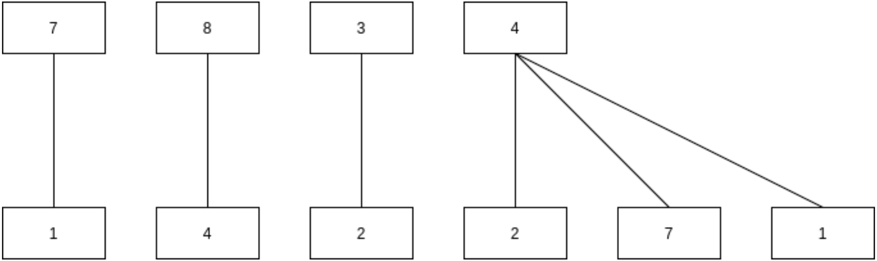


Figure: Greedy Algorithm

Recursion

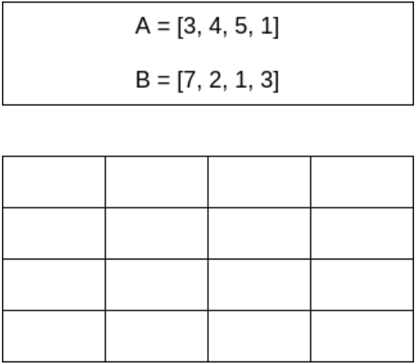
$$OPT(i,j) = \begin{cases} \frac{X_i}{Y_j} & i = 1 \quad \wedge \quad j = 1 \\ \frac{X_i}{\sum_{p=1}^{j} Y_p} & i = 1 \quad \wedge \quad j > 1 \\ \frac{\sum_{p=1}^{i} X_p}{Y_j} & i > 1 \quad \wedge \quad j = 1 \\ min(M_g(i,j), M_d(i,j)) & i > 1 \quad \wedge \quad j > 1 \end{cases}$$

$$M_g(i,j) = min_{k=1}^{i-1}\{\frac{\sum_{p=k+1}^{i} X_p}{Y_j} + OPT(k, j-1)\}$$

$$M_d(i,j) = min_{k=1}^{j-1}\{\frac{X_i}{\sum_{p=k+1}^{j} Y_p} + OPT(i-1, k)\}$$

Figure: Recursion

Dynamic Programming Algorithm

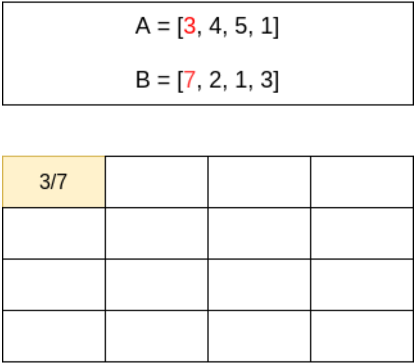


A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

Figure: Dynamic Programming Algorithm - Initial

Dynamic Programming Algorithm



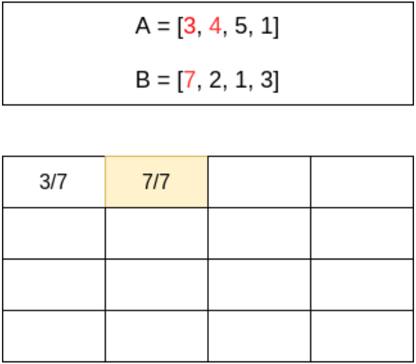
A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7			

Figure: Dynamic Programming Algorithm - Step 1

Dynamic Programming Algorithm



A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7		

Figure: Dynamic Programming Algorithm - Step 1

Dynamic Programming Algorithm

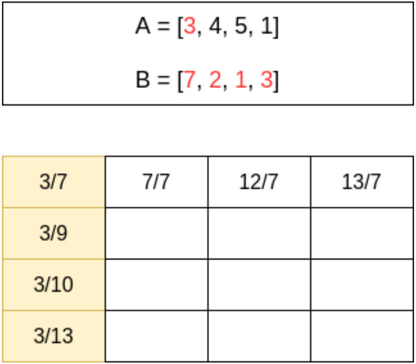
A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7	12/7	13/7

Figure: Dynamic Programming Algorithm - Step 1

Dynamic Programming Algorithm



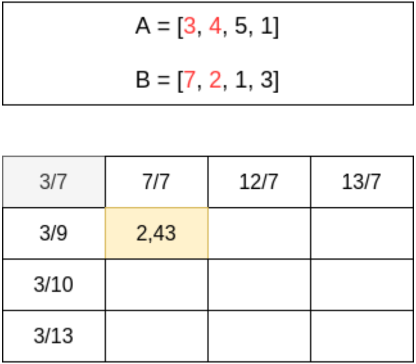
A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7	12/7	13/7
3/9			
3/10			
3/13			

Figure: Dynamic Programming Algorithm - Step 2

Dynamic Programming Algorithm



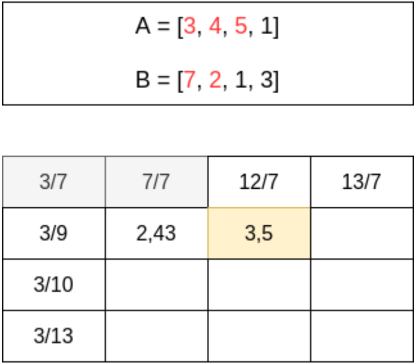
A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7	12/7	13/7
3/9	2,43		
3/10			
3/13			

Figure: Dynamic Programming Algorithm - Step 3

Dynamic Programming Algorithm



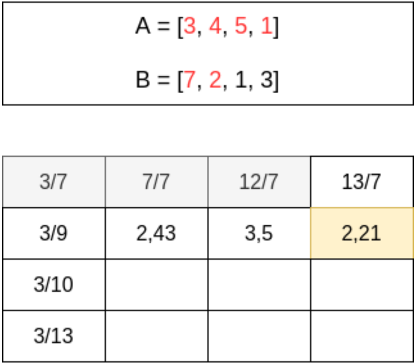
A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7	12/7	13/7
3/9	2,43	3,5	
3/10			
3/13			

Figure: Dynamic Programming Algorithm - Step 3

Dynamic Programming Algorithm



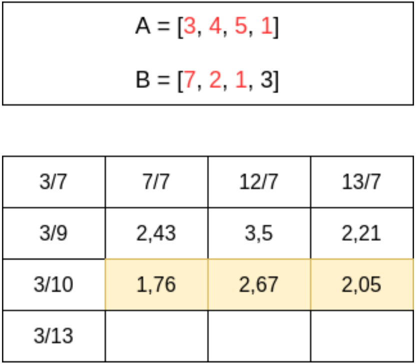
A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7	12/7	13/7
3/9	2,43	3,5	2,21
3/10			
3/13			

Figure: Dynamic Programming Algorithm - Step 3

Dynamic Programming Algorithm



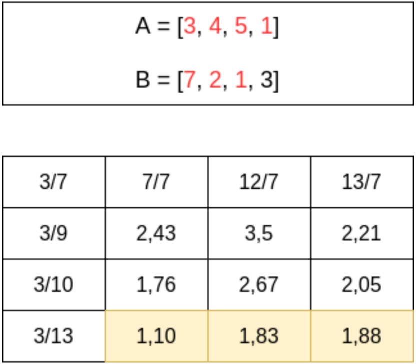
A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7	12/7	13/7
3/9	2,43	3,5	2,21
3/10	1,76	2,67	2,05
3/13			

Figure: Dynamic Programming Algorithm - Step 4

Dynamic Programming Algorithm



A = [3, 4, 5, 1]

B = [7, 2, 1, 3]

3/7	7/7	12/7	13/7
3/9	2,43	3,5	2,21
3/10	1,76	2,67	2,05
3/13	1,10	1,83	1,88

Figure: Dynamic Programming Algorithm - Step 5

Table of Contents

1. Recap of Block Matching

2. Decoding Images through Luma
   - What is Luma?
   - Transforming into a Matrix of bits

3. Our Approach
   - Base Process for Transformation
   - Different Algorithms

4. Examples

5. Conclusion

Table of Contents

1. Recap of Block Matching

2. Decoding Images through Luma
   - What is Luma?
   - Transforming into a Matrix of bits

3. Our Approach
   - Base Process for Transformation
   - Different Algorithms

4. Examples

5. Conclusion

Table of Contents

1. Recap of Block Matching
2. Decoding Images through Luma
   - What is Luma?
   - Transforming into a Matrix of bits
3. Our Approach
   - Base Process for Transformation
   - Different Algorithms
4. Examples
5. Conclusion

Table of Contents

1. Recap of Block Matching
2. Decoding Images through Luma
   - What is Luma?
   - Transforming into a Matrix of bits
3. Our Approach
   - Base Process for Transformation
   - Different Algorithms
4. Examples
5. Conclusion