Georgia Institute of Technology

Computational Photography

Dr. Irfan Essa
Professor
School of Interactive Computing

Study the basics of computation and its impact on the entire workflow of photography, from capturing, manipulating and collaborating on, and sharing photographs.

Georgia Institute of Technology

Dr. Irfan Essa
Professor
School of Interactive Computing

Digital Images: Merging and Blending Images

Different methods for Combing Multiple Images to Generate a Novel Image

Lesson Objectives

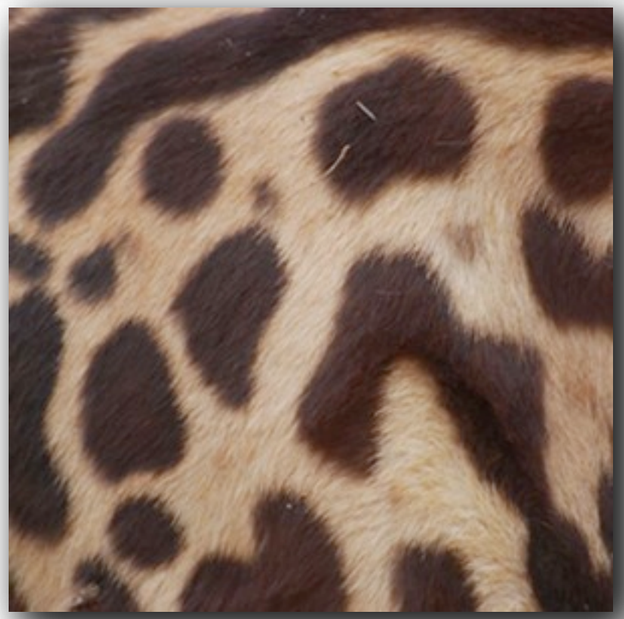
- ★ Describe in your own words one (1) way to merge two images.
- ★ Recall the two (2) issues caused by not being able to determine the window used for merging images
- ★ Describe in your own words the two (2) advantages of a using the Fourier Domain.

REVIEW: Combine, Merge, Blend Images

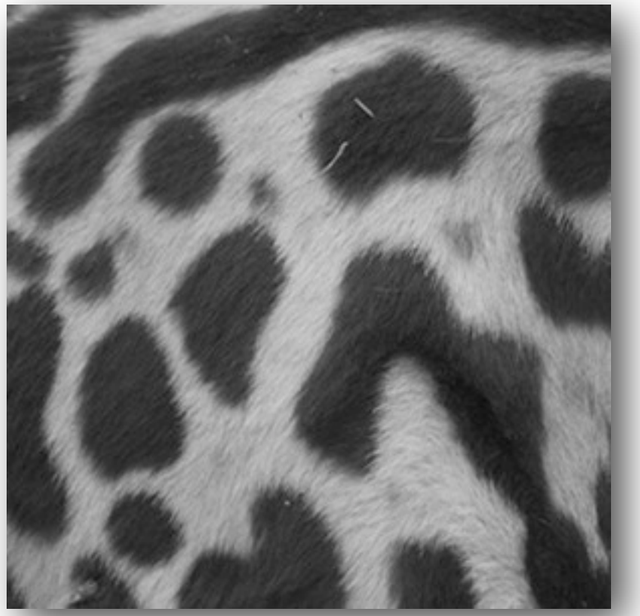
REVIEW: Combine, Merge, Blend Images

REVIEW: Combine, Merge, Blend Images

REVIEW: Combine, Merge, Blend Images

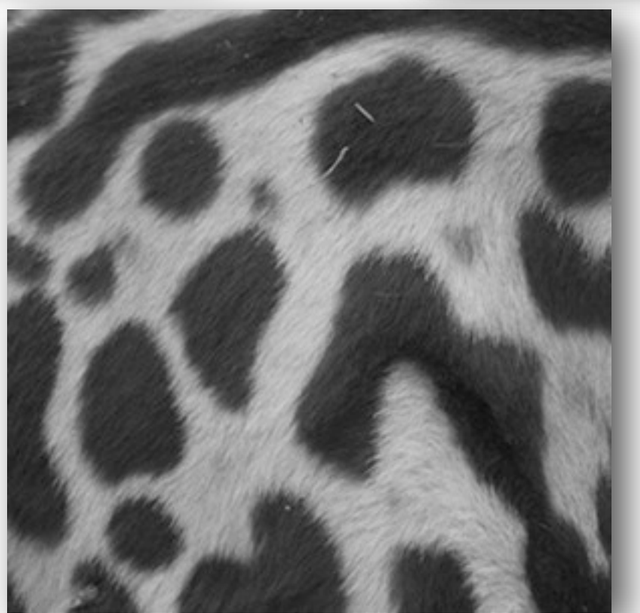


Merging Two Images

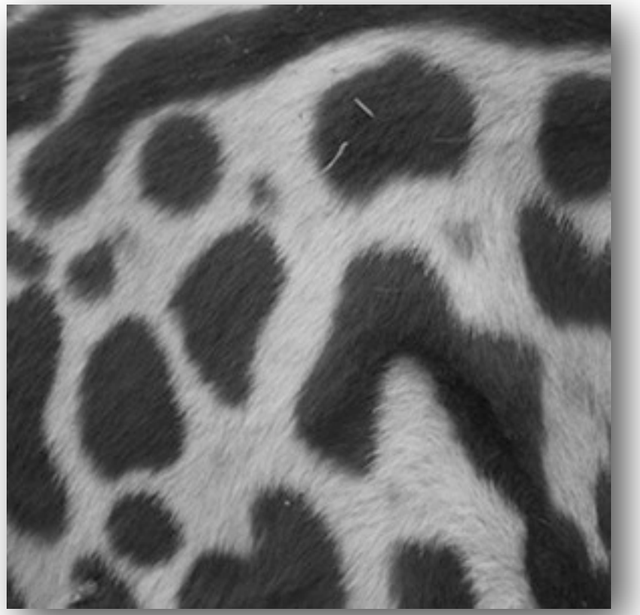


Merging Two Images

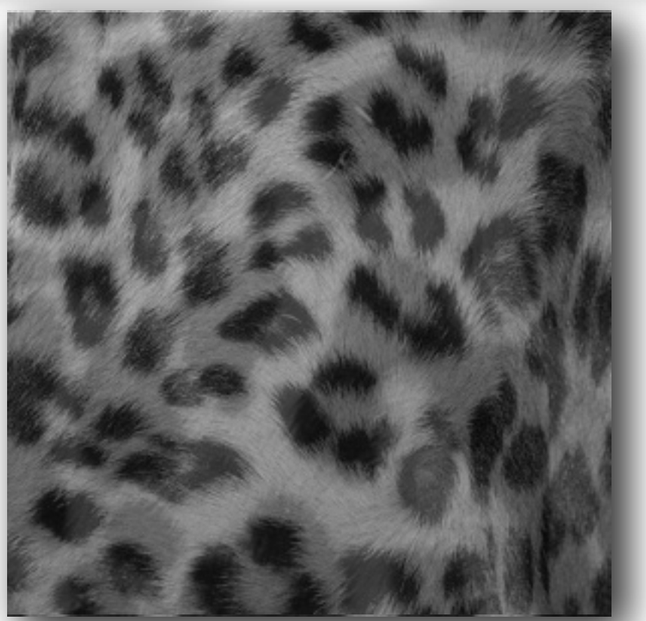
Merging Two Images



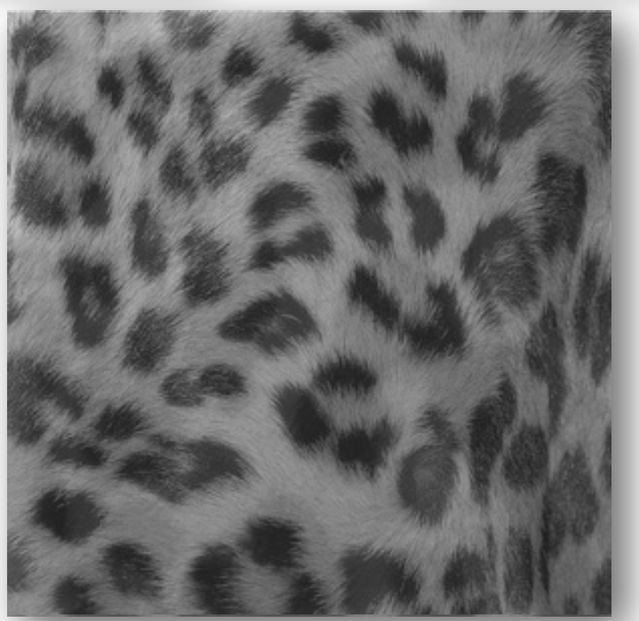
Merging Two Images



Merging Two Images



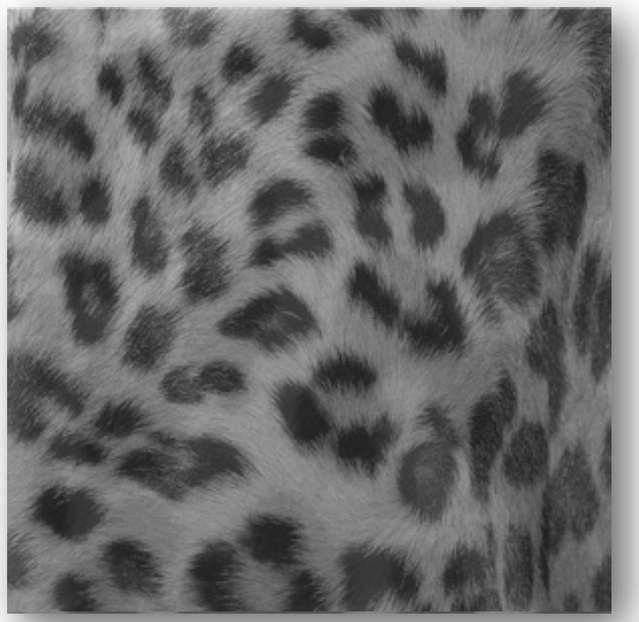
Merging Two Images



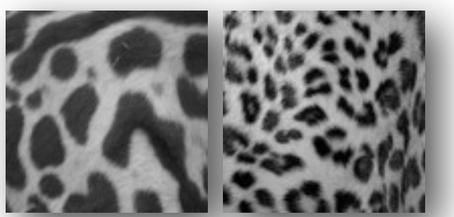
Merging Two Images

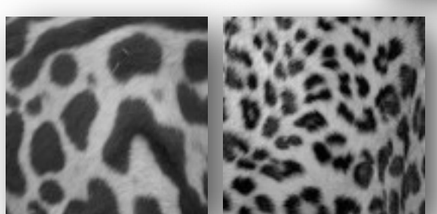
6

Merging Two Images

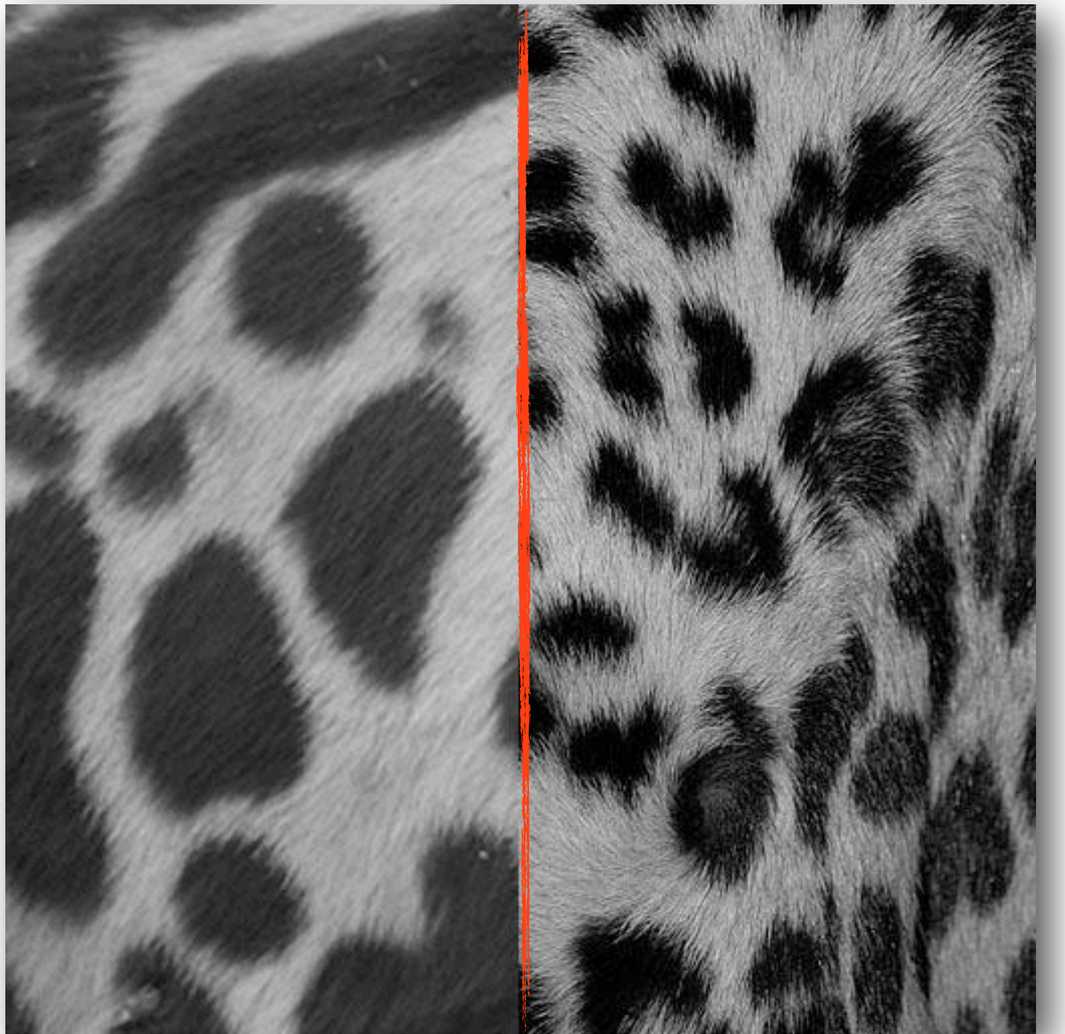


Merging Two Images

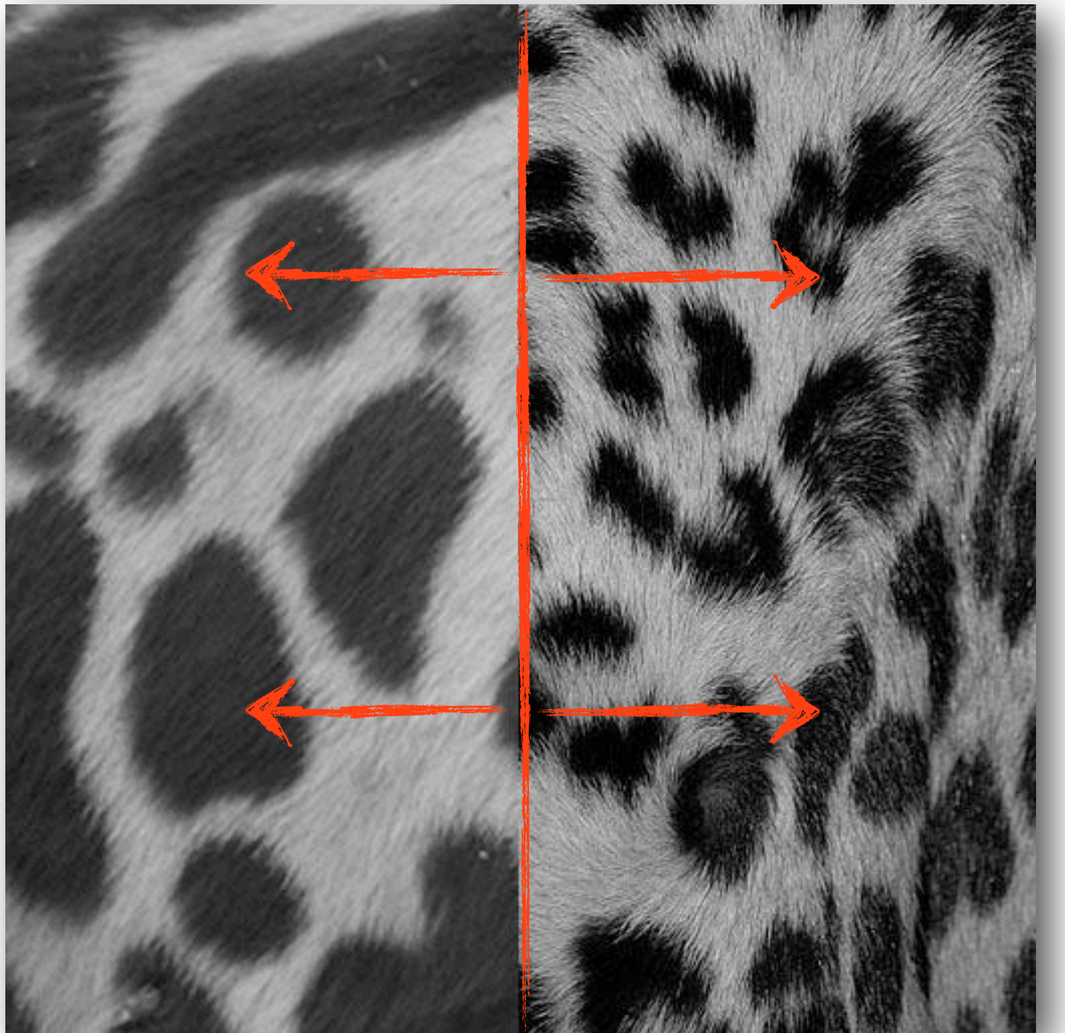




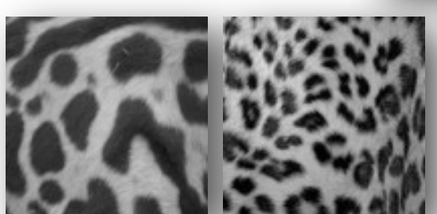
Merging Two Images



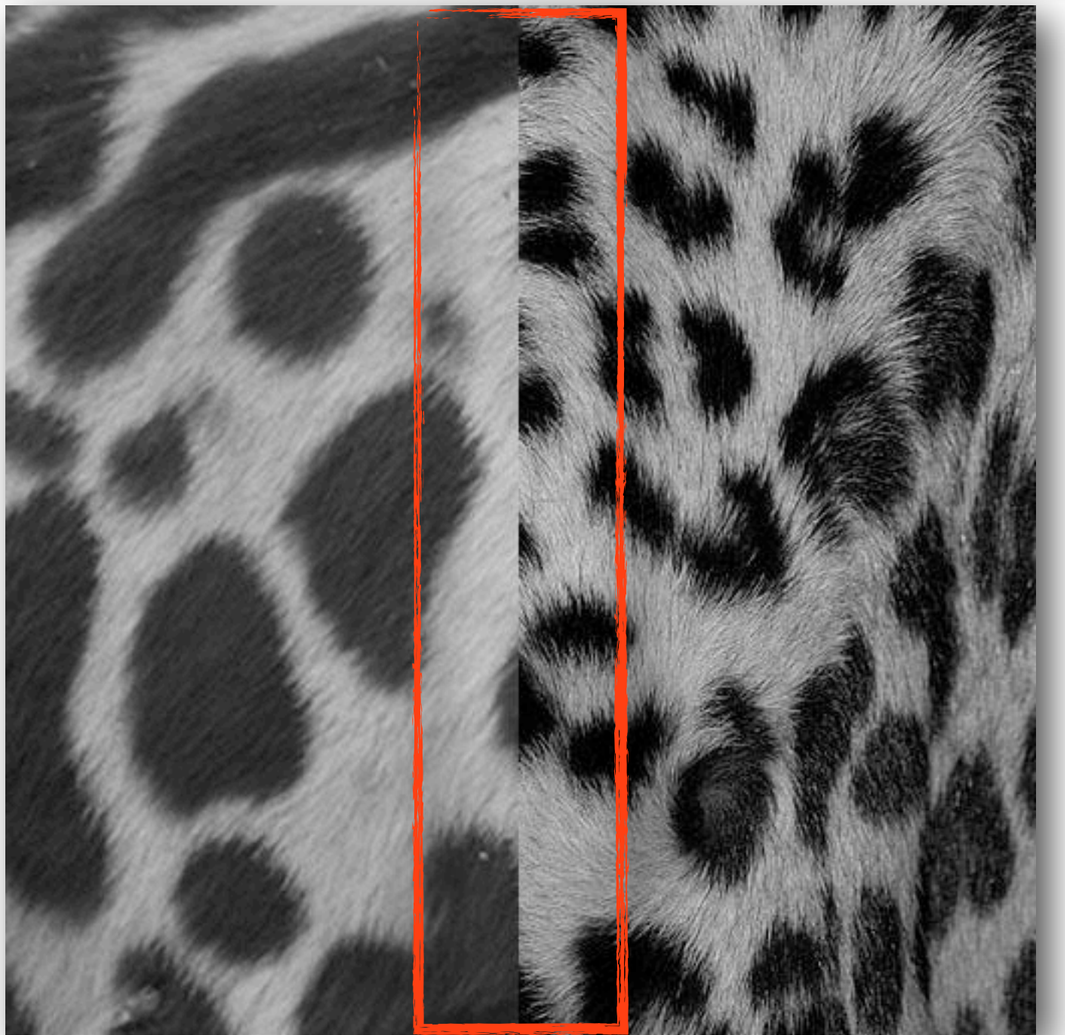
Merging Two Images



Merging Two Images



Merging Two Images



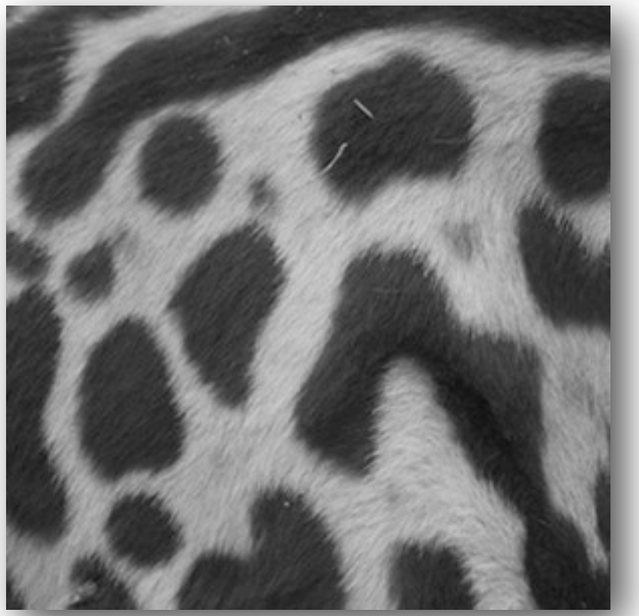
Merging Two Images

I_l

I_r

I_t

Cross-Fading Two Images



I_l

1.0

1.0

0.0

I_r

1.0

1.0

0.0

I_t

Cross-Fading Two Images

I_l

I_r

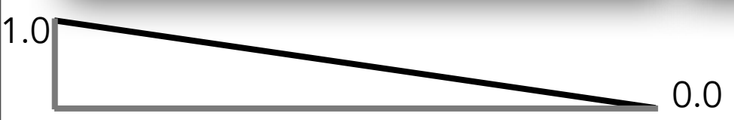
I_t

Cross-Fading Two Images

I_l

I_r

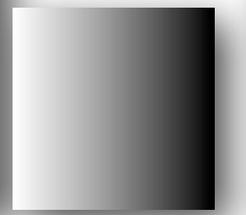
I_t



Cross-Fading Two Images

I_l

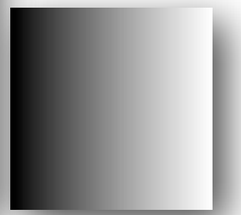
+



I_r

=

I_t



Cross-Fading Two Images

I_l

+

I_r

=

I_t

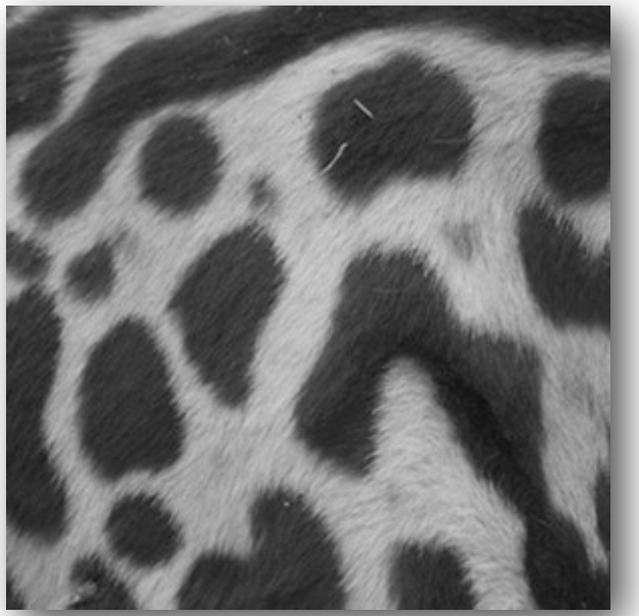
Cross-Fading Two Images

I_l

I_r

I_t

Cross-Fading Two Images



I_l

1.0

0.0

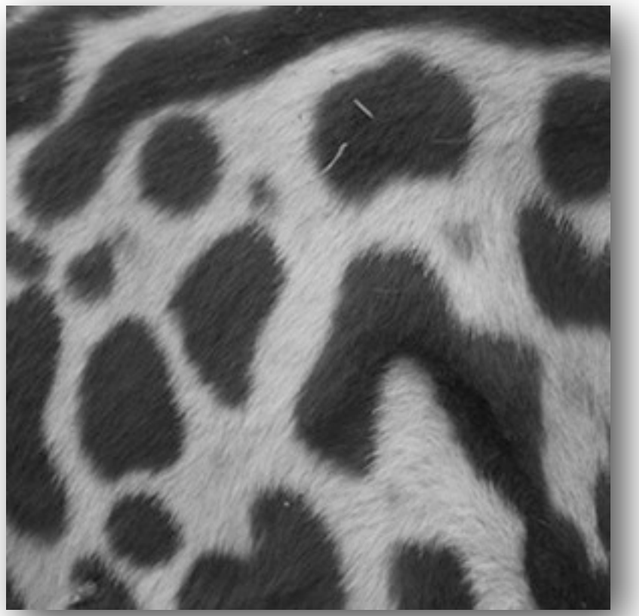
I_r

1.0

0.0

I_t

Cross-Fading Two Images

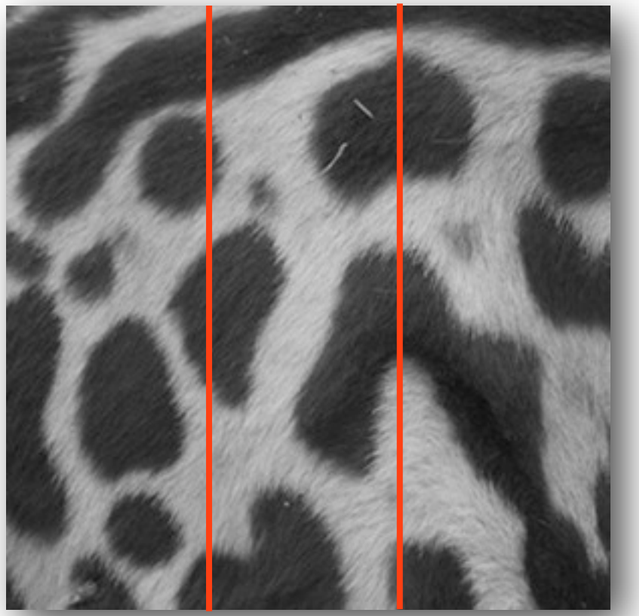


I_l

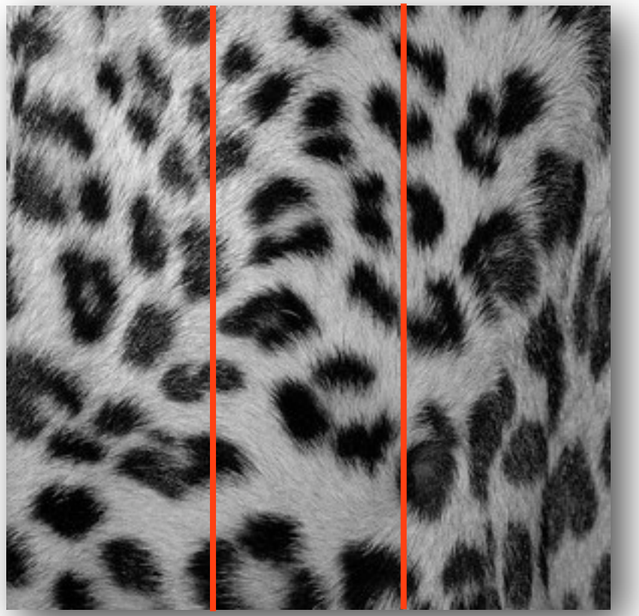
I_r

I_t

Cross-Fading Two Images



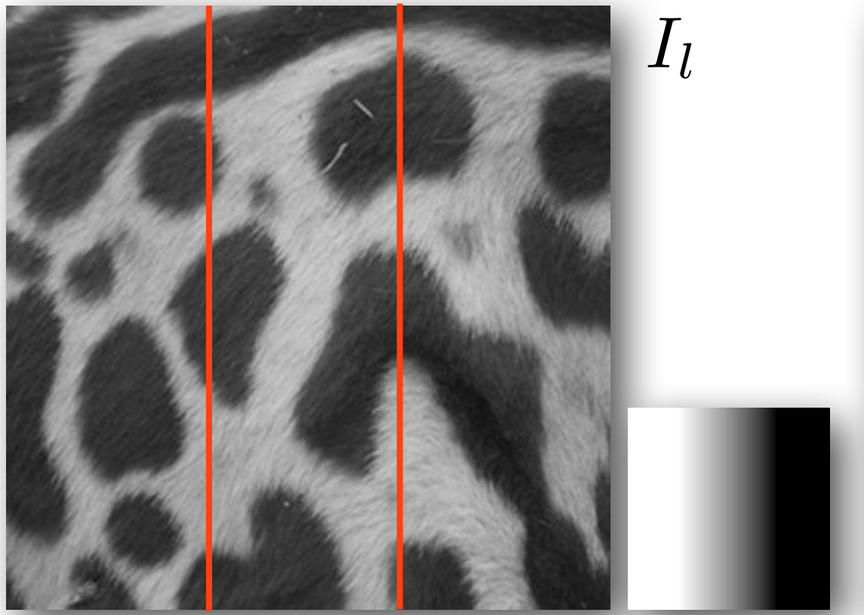
I_l



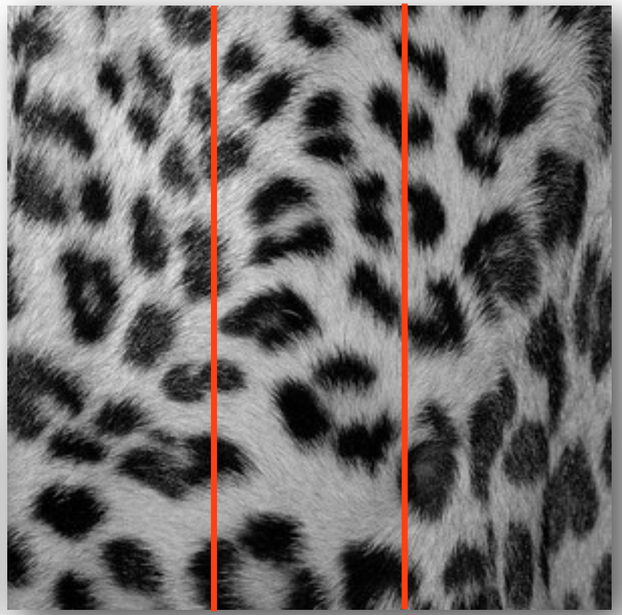
I_r

I_t

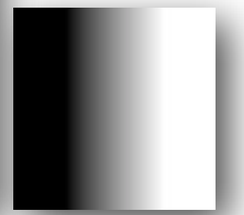
Cross-Fading Two Images



I_l

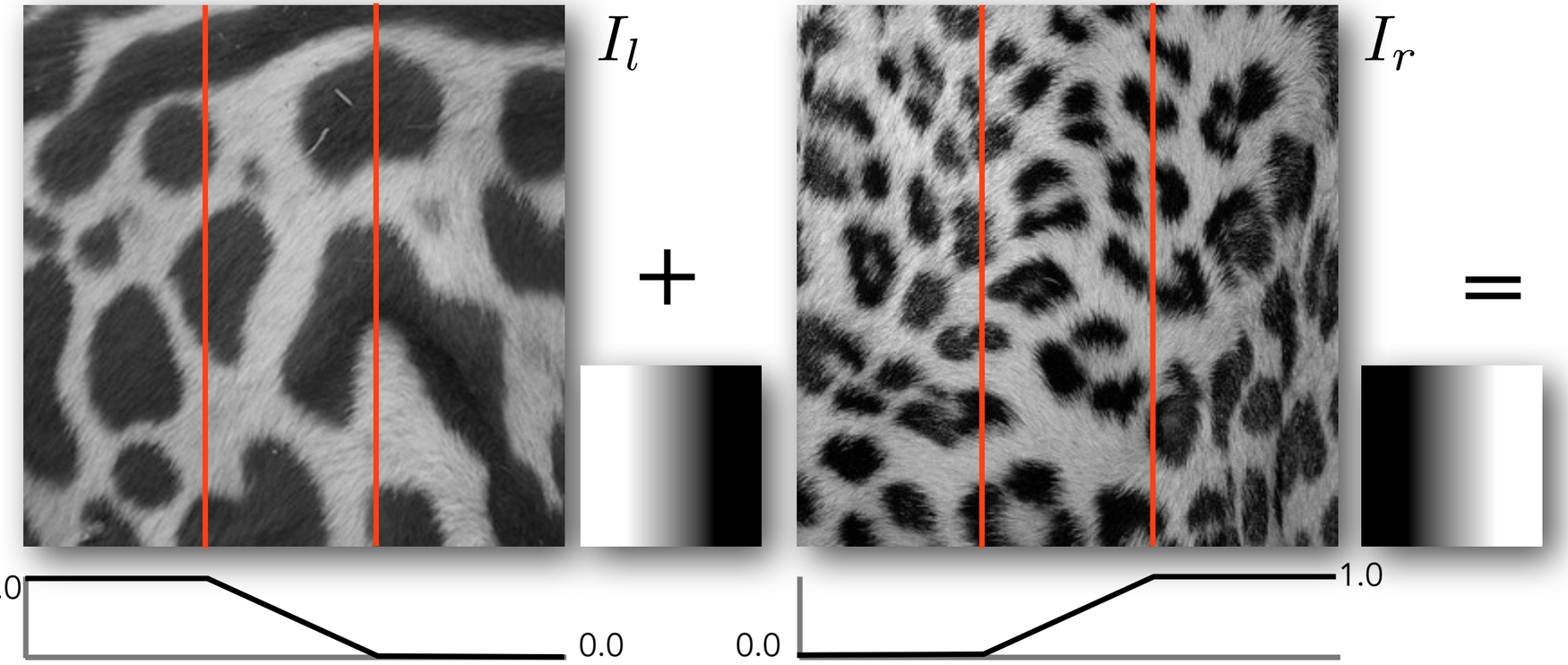


I_r

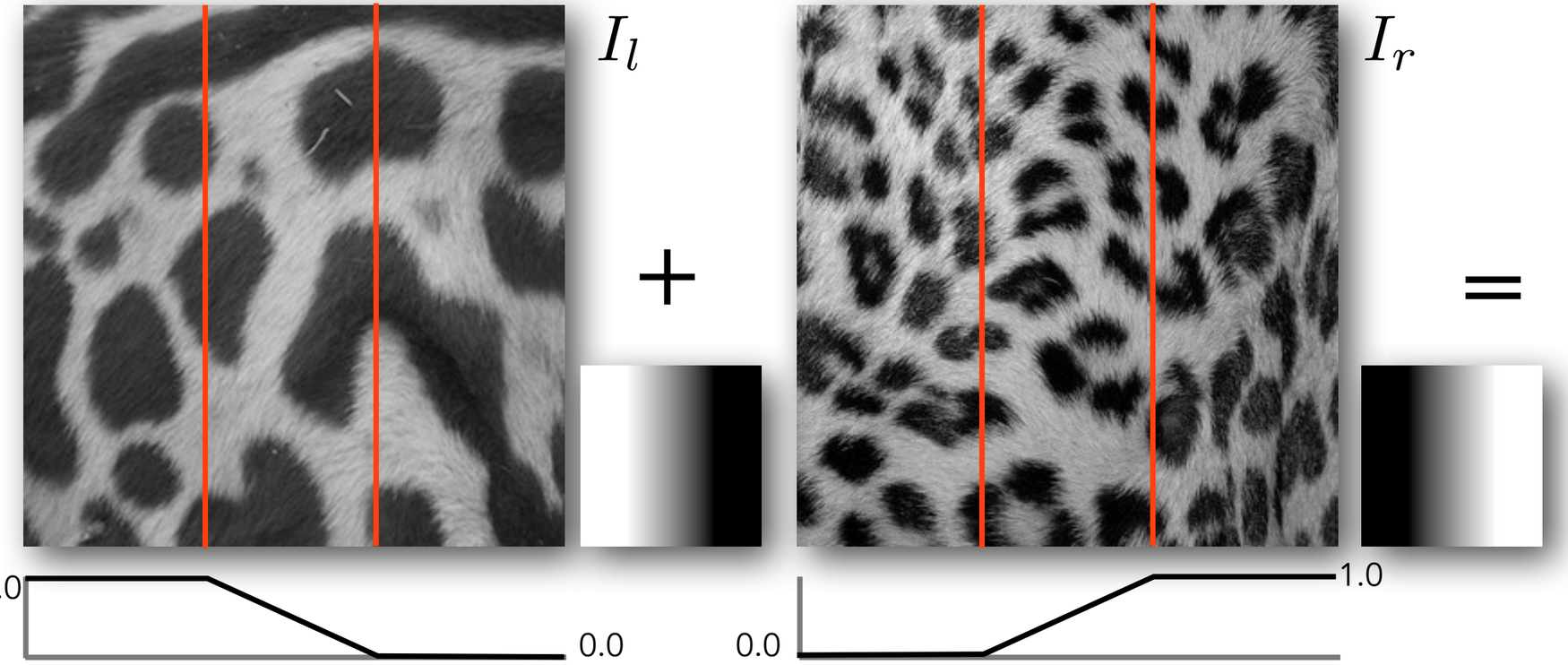


I_t

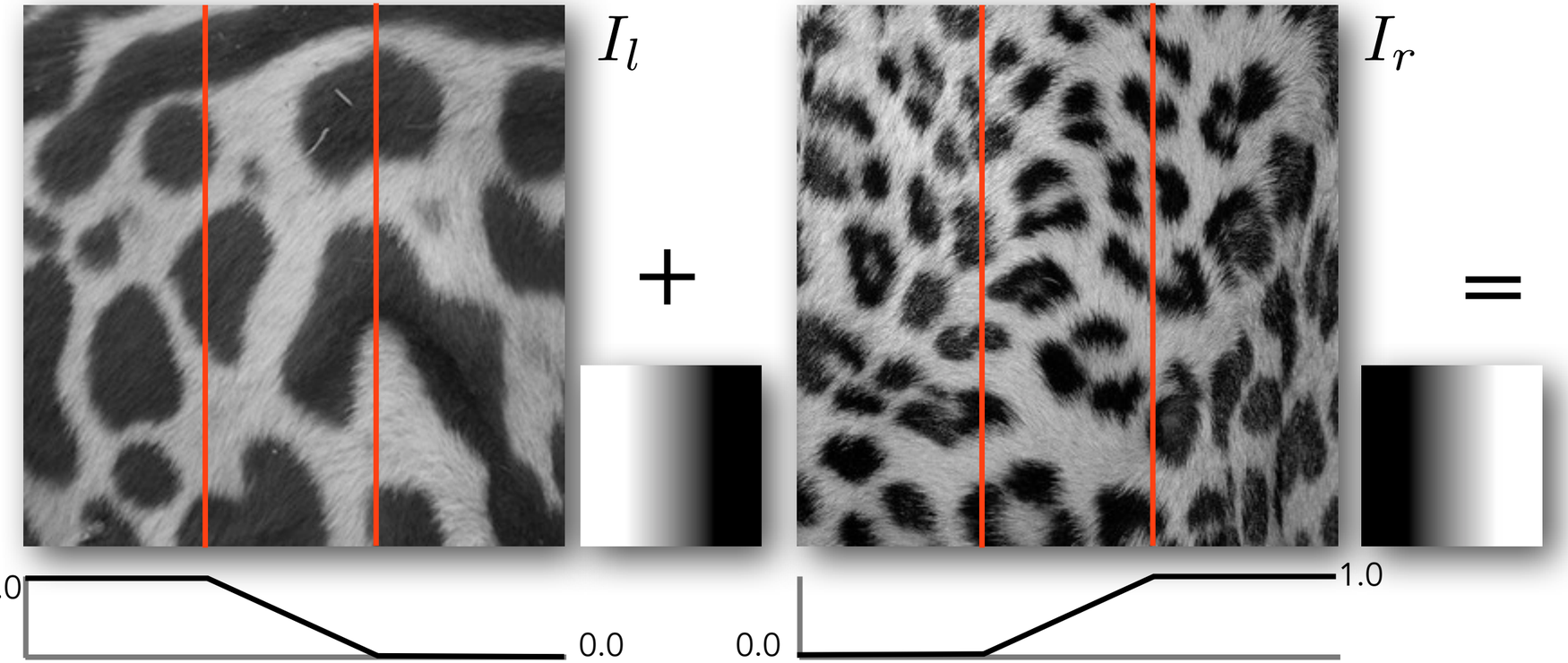
Cross-Fading Two Images



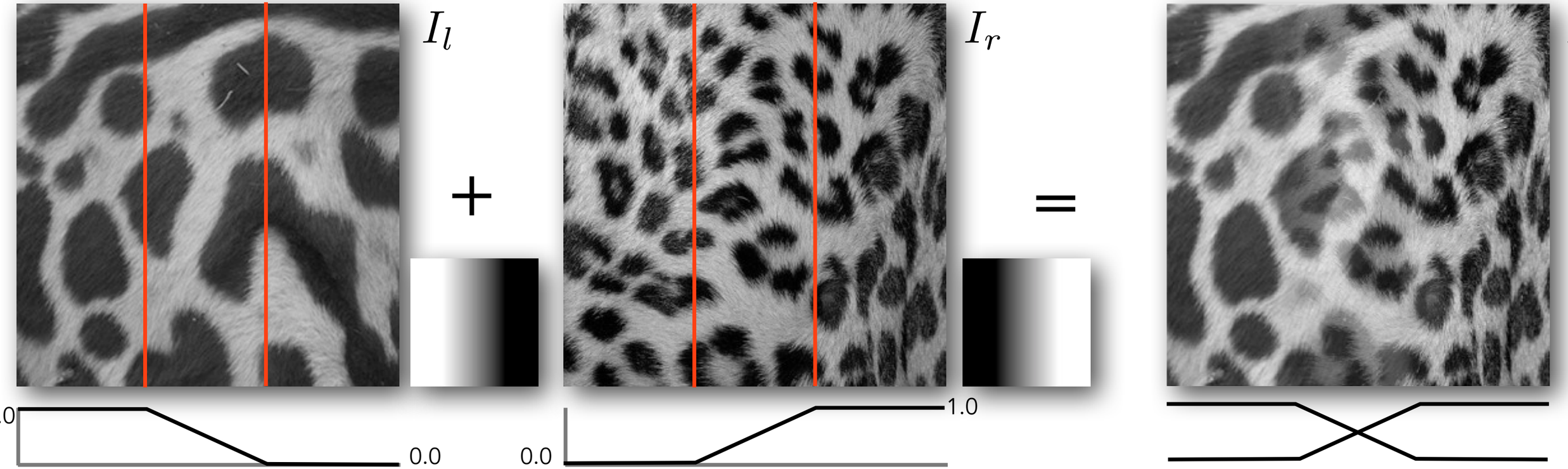
Cross-Fading Two Images



Cross-Fading Two Images



Cross-Fading Two Images



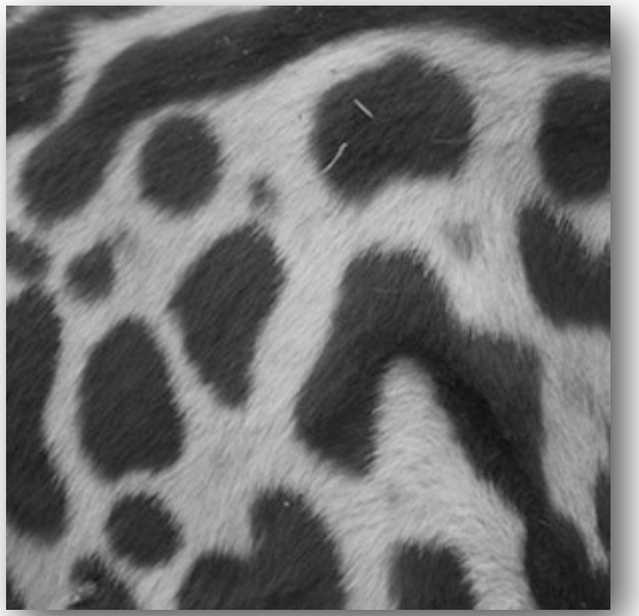
Cross-Fading Two Images

I_l

I_r

I_t

Cross-Fading Two Images



I_l

1.0

0.0

I_r

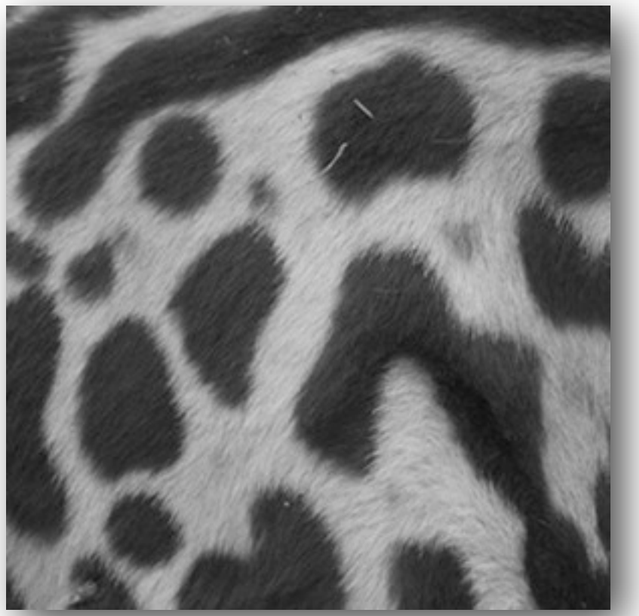
1.0

0.0

I_t

Cross-Fading Two Images

10

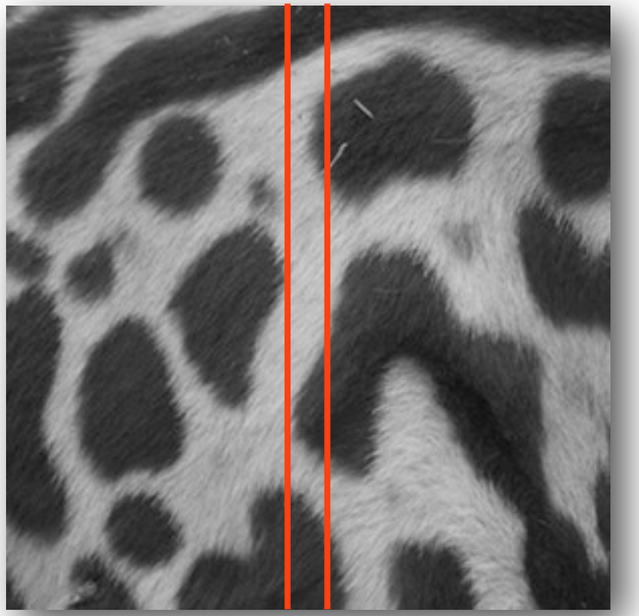


I_l

I_r

I_t

Cross-Fading Two Images



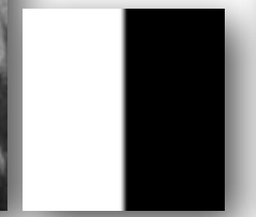
I_l

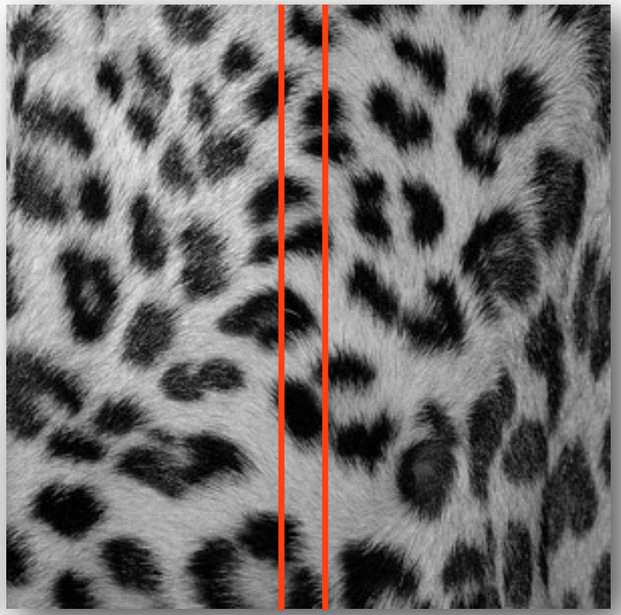
I_r

I_t

Cross-Fading Two Images

I_l

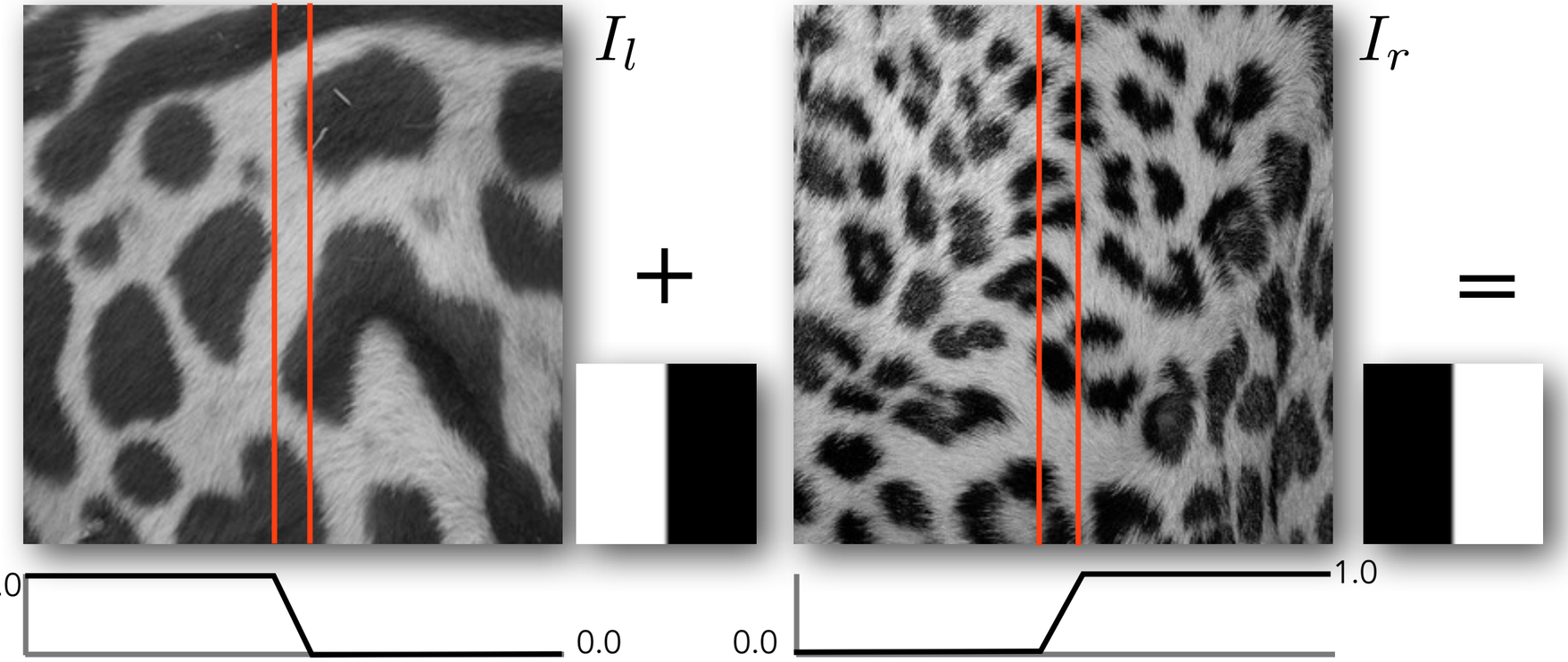




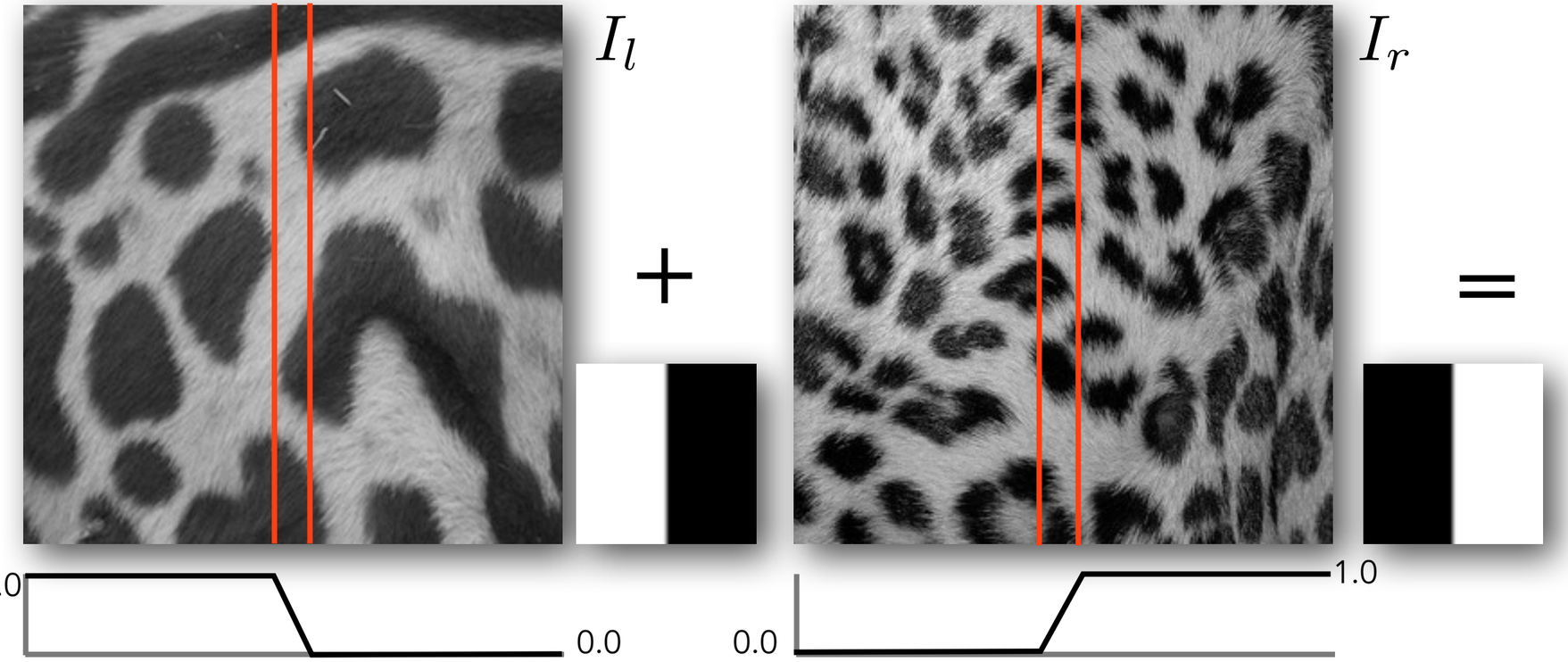
I_r

I_t

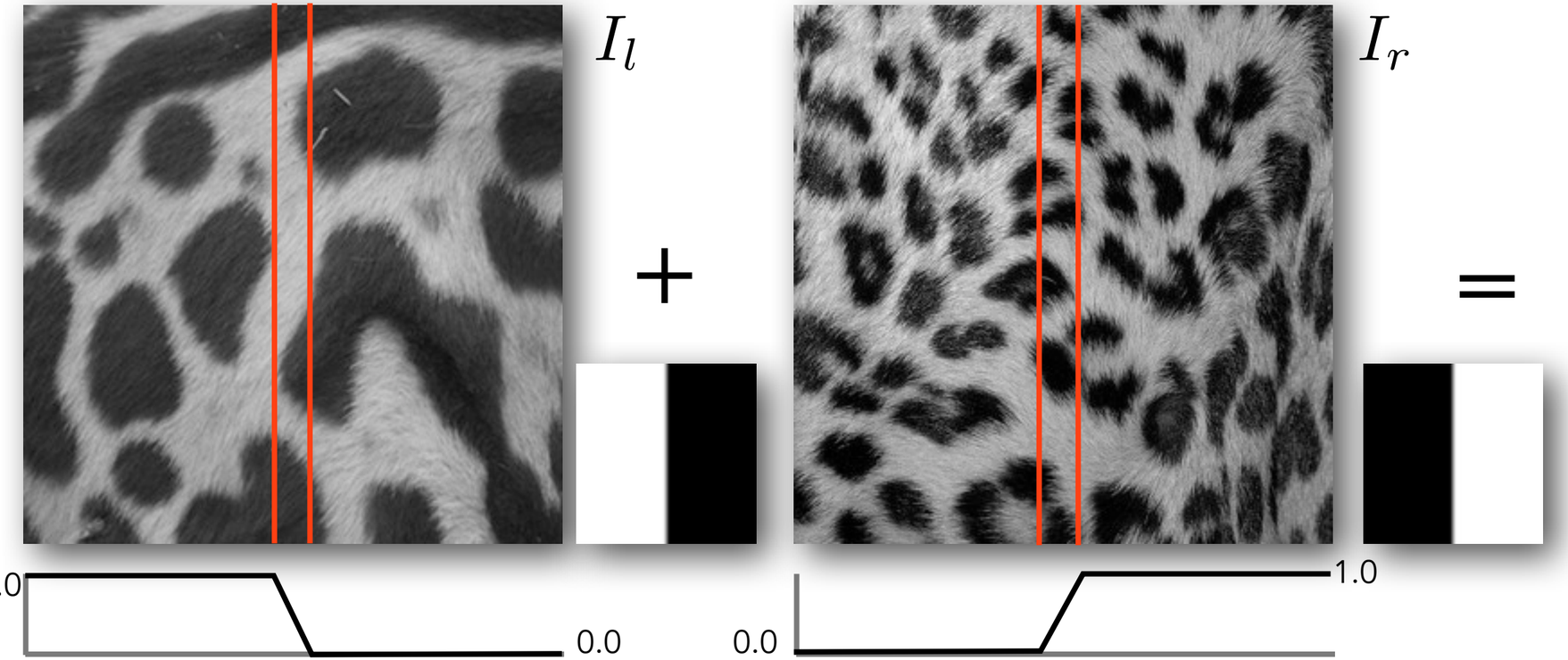
Cross-Fading Two Images



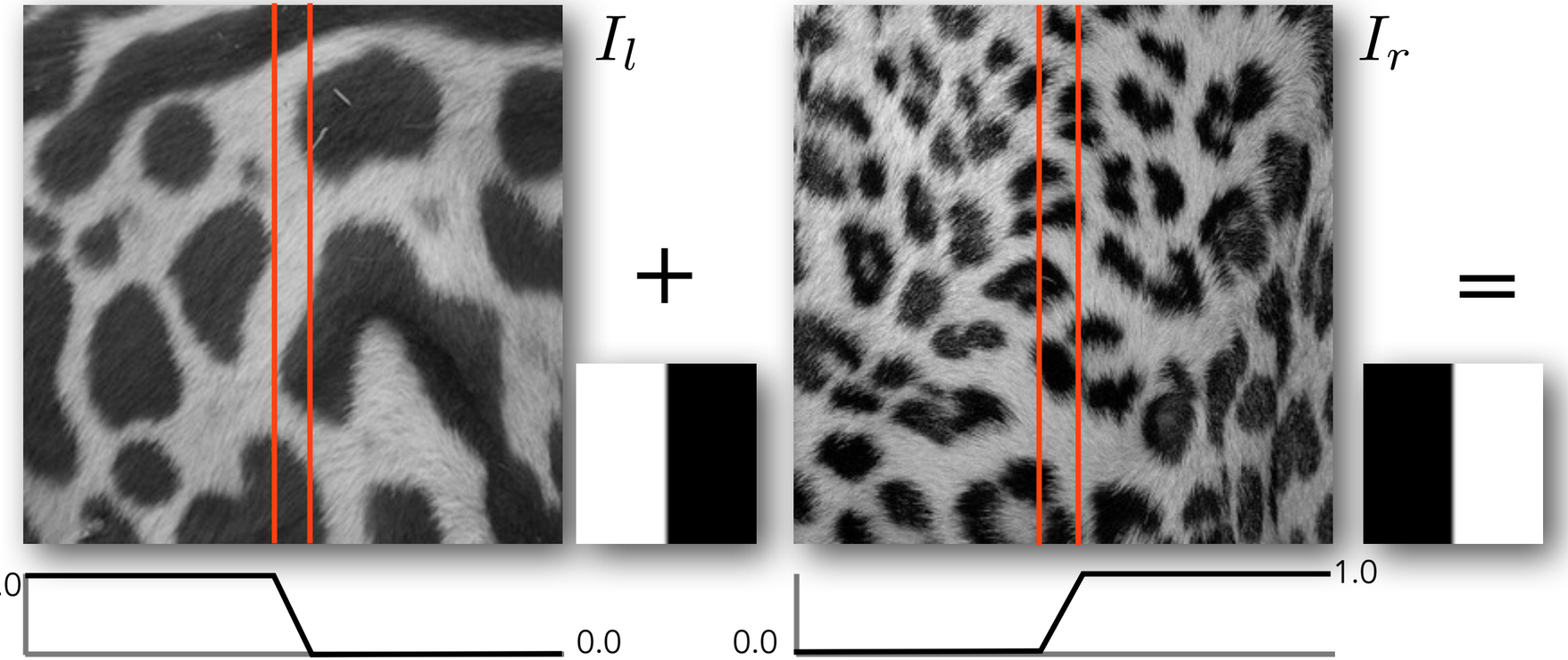
Cross-Fading Two Images



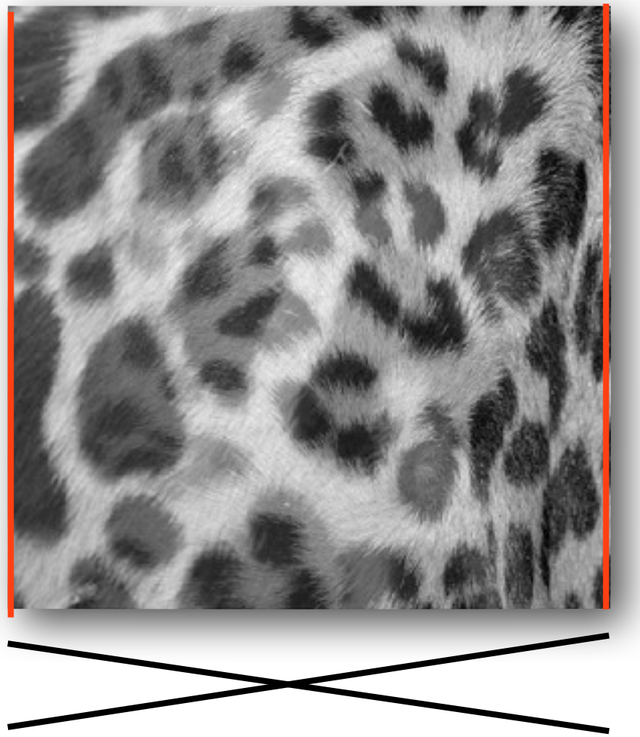
Cross-Fading Two Images

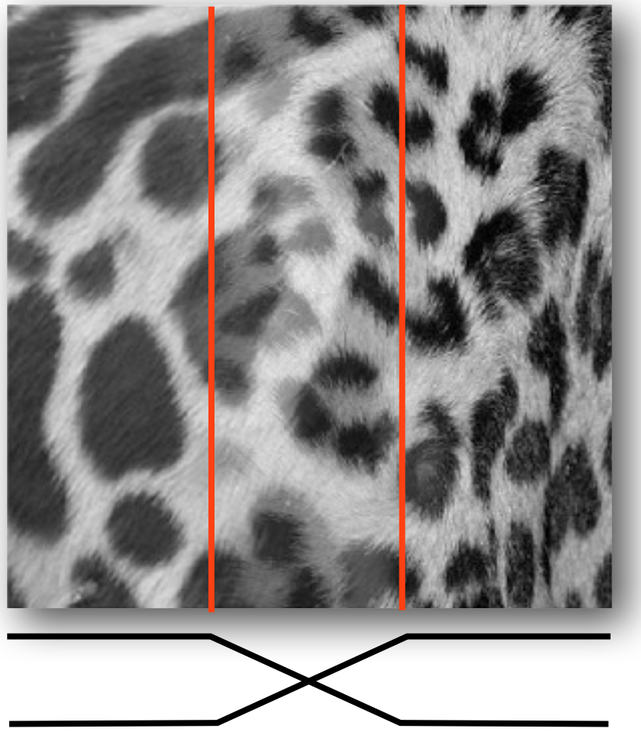


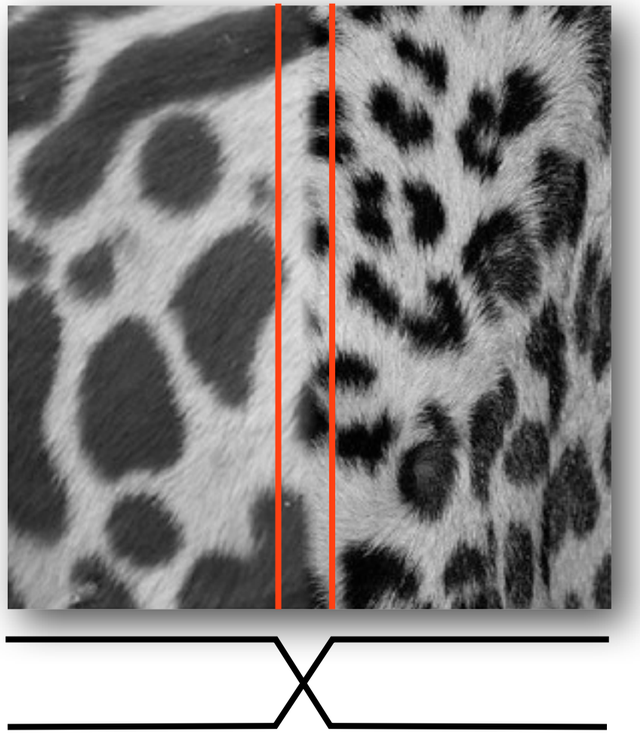
Cross-Fading Two Images



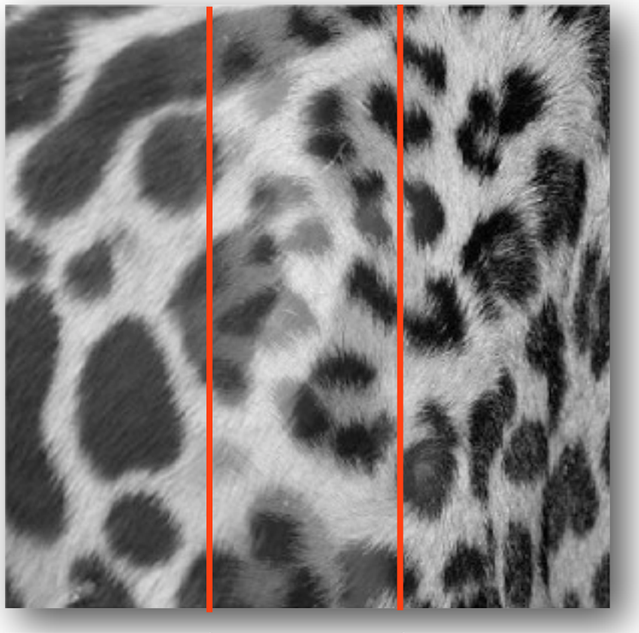
Cross-Fading Two Images

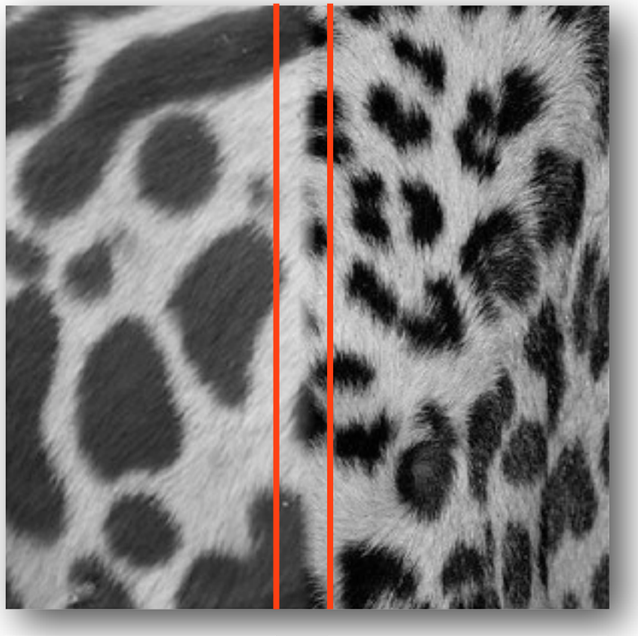






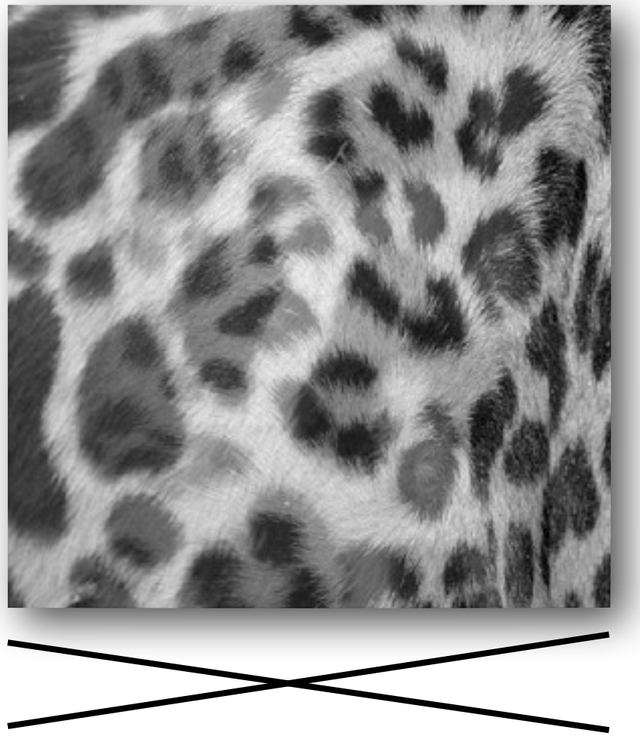
Cross-Fading Window Size

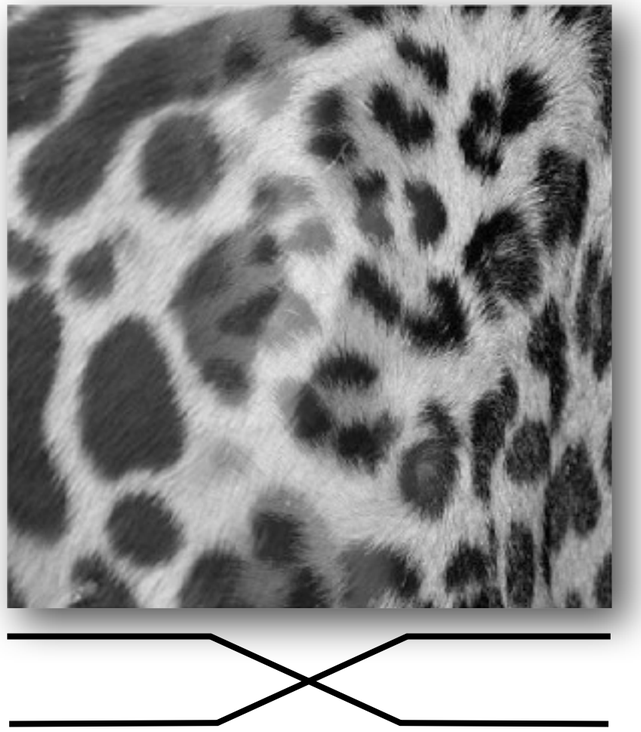


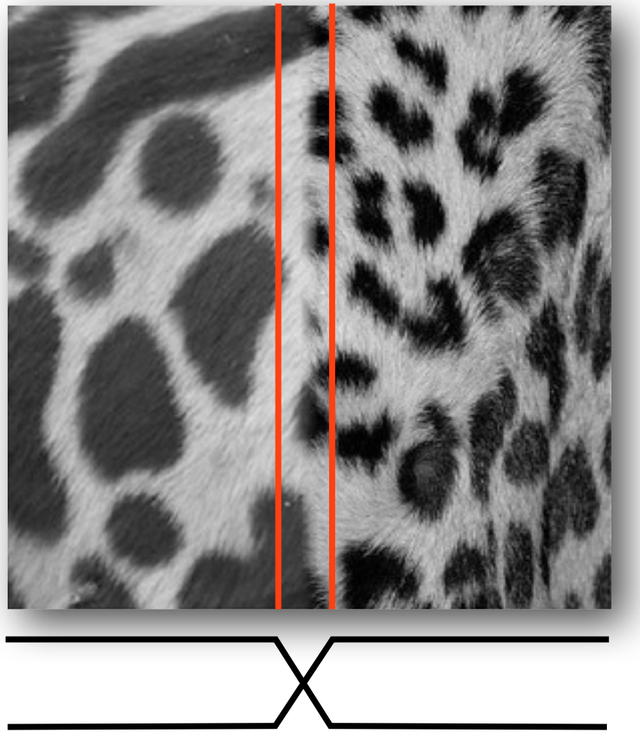


Cross-Fading Window Size

11







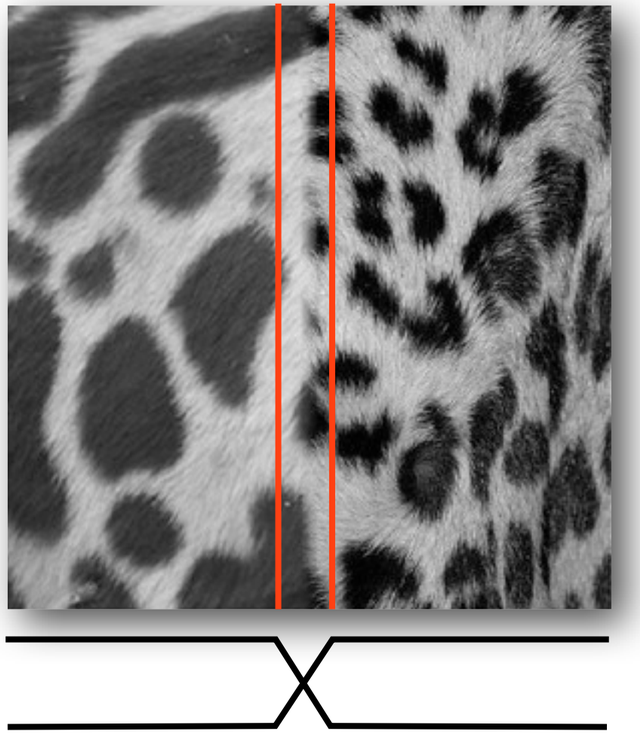
Cross-Fading Window Size

11

Cross-Fading Window Size

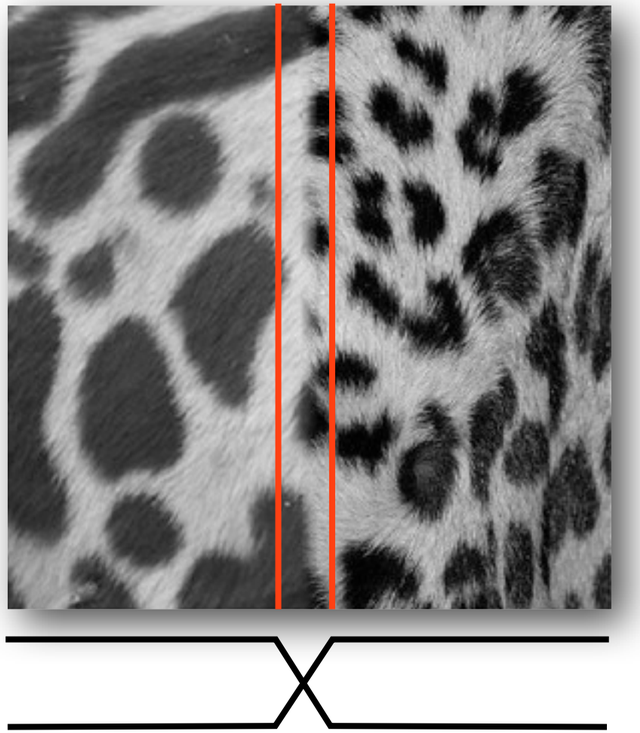
11

Factors for Optimal Window Size



12

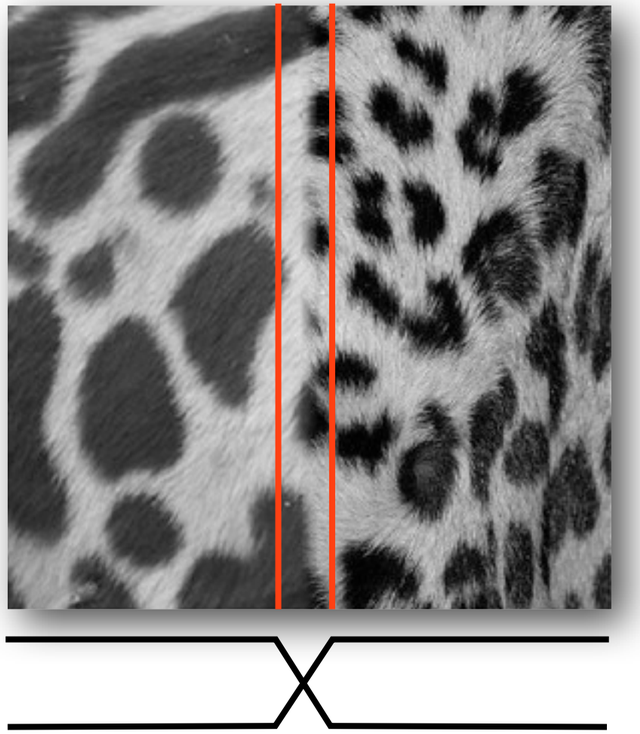
Factors for Optimal Window Size



- ★ To avoid seams: Window = size of largest prominent “feature”

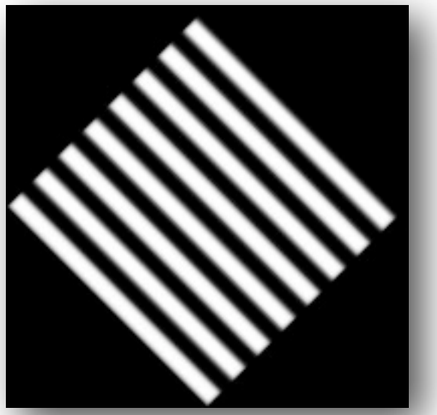
12

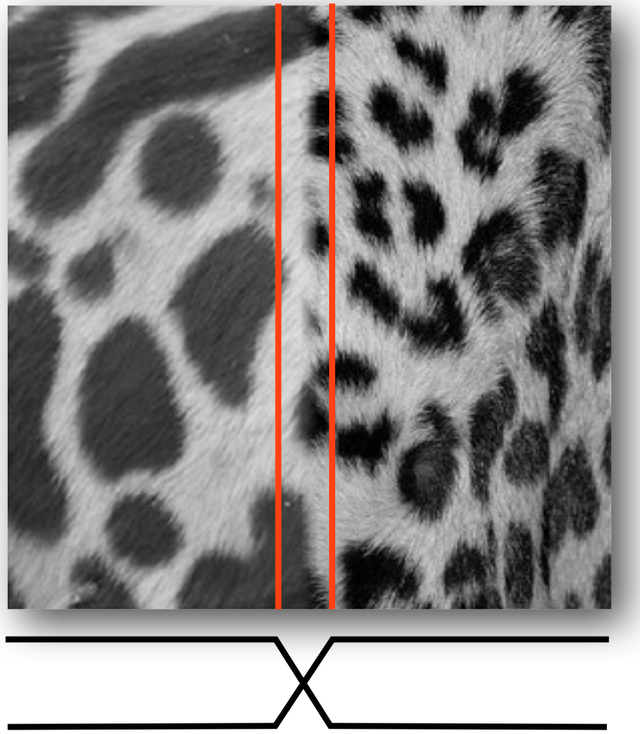
Factors for Optimal Window Size



- ★ To avoid seams: Window = size of largest prominent "feature"
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent "feature"

Factors for Optimal Window Size

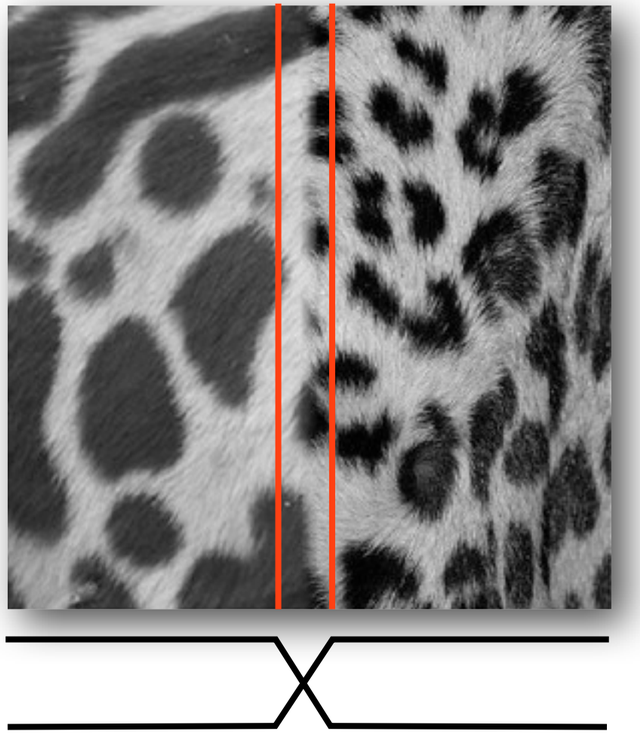




- ★ To avoid seams: Window = size of largest prominent "feature"
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent "feature"

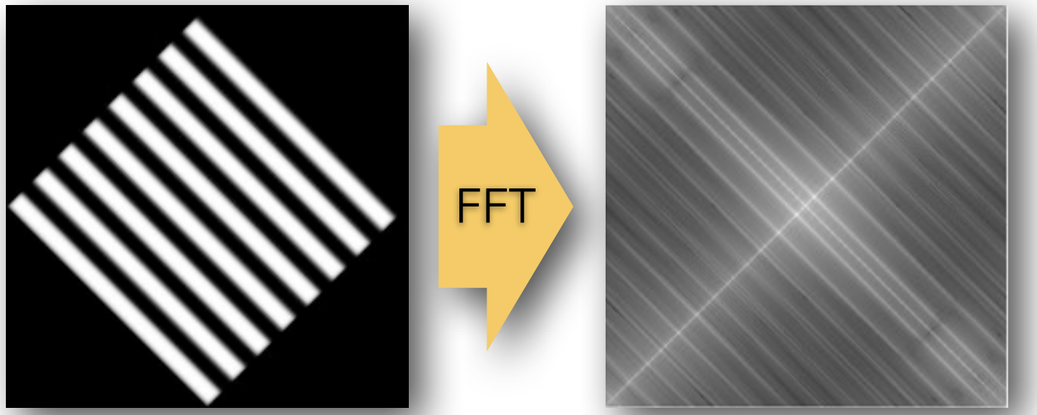
12

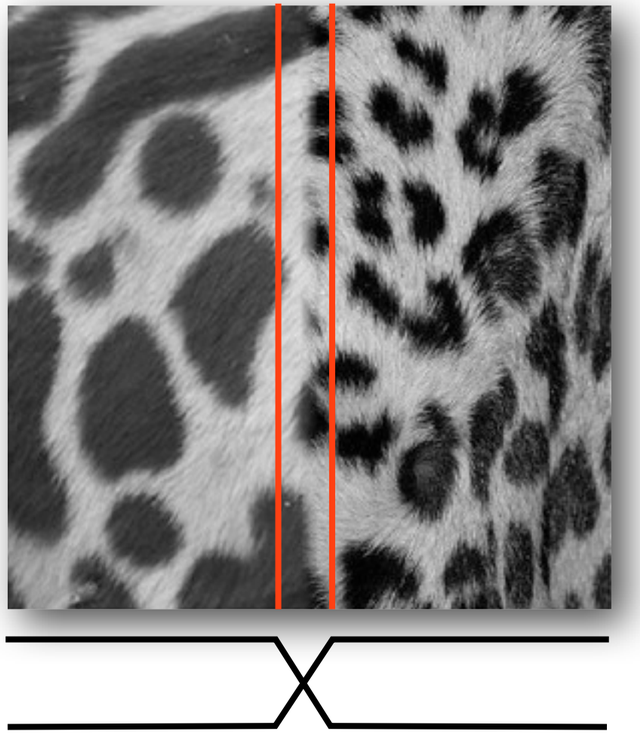
Factors for Optimal Window Size



- ★ To avoid seams: Window = size of largest prominent "feature"
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent "feature"

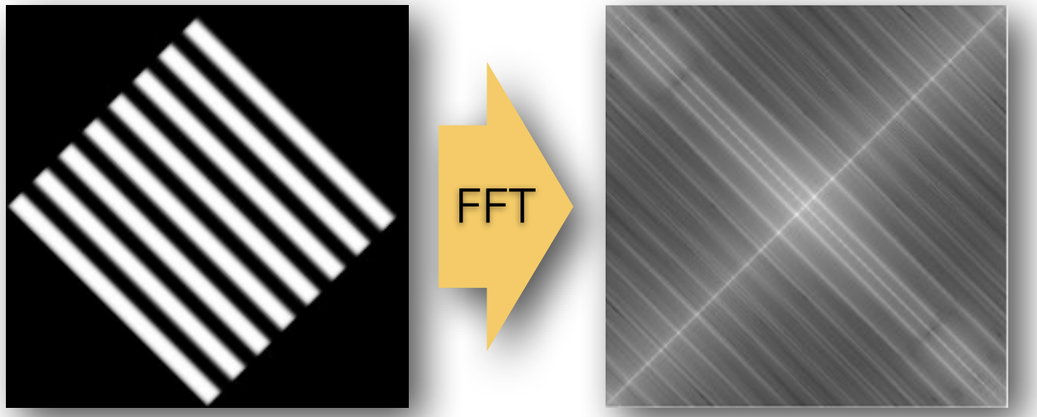
Factors for Optimal Window Size

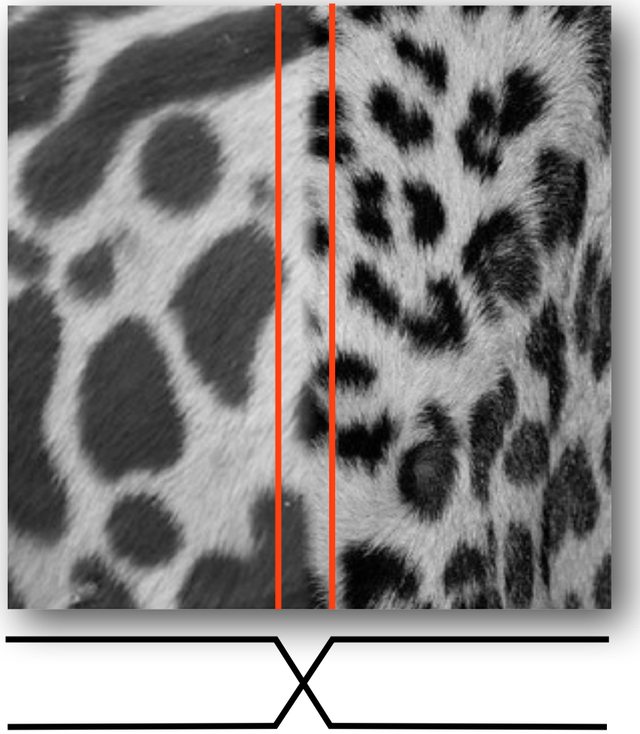




- ★ To avoid seams: Window = size of largest prominent “feature”
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent “feature”

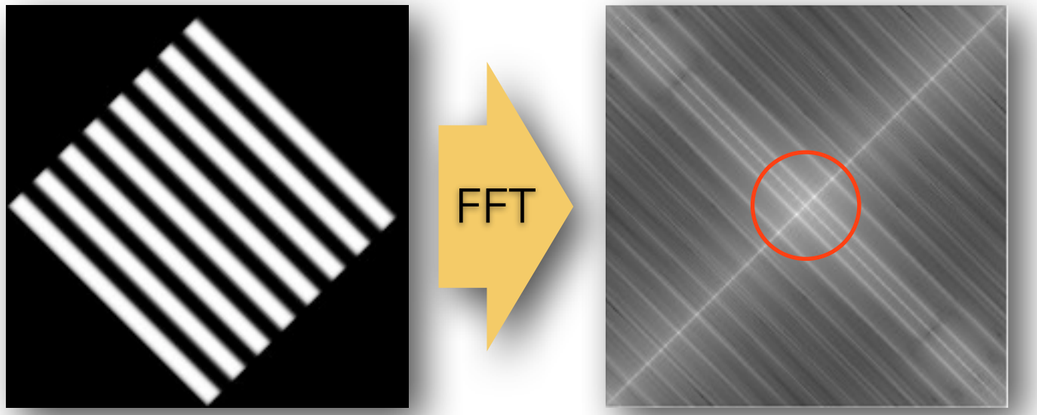
Factors for Optimal Window Size

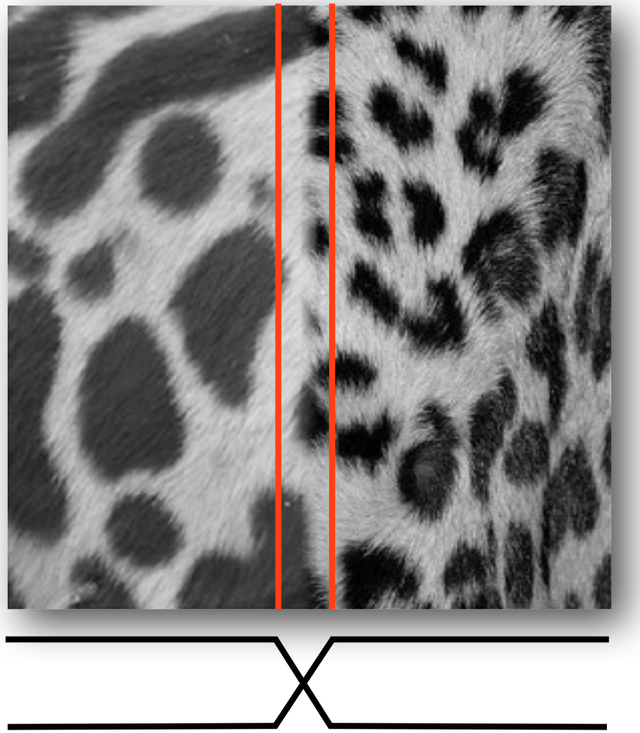




- ★ To avoid seams: Window = size of largest prominent "feature"
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent "feature"
- ★ Use Fourier domain
  - Largest frequency ≤ 2 × size of smallest frequency
  - Image frequency content should occupy one "octave" (power of two)

Factors for Optimal Window Size

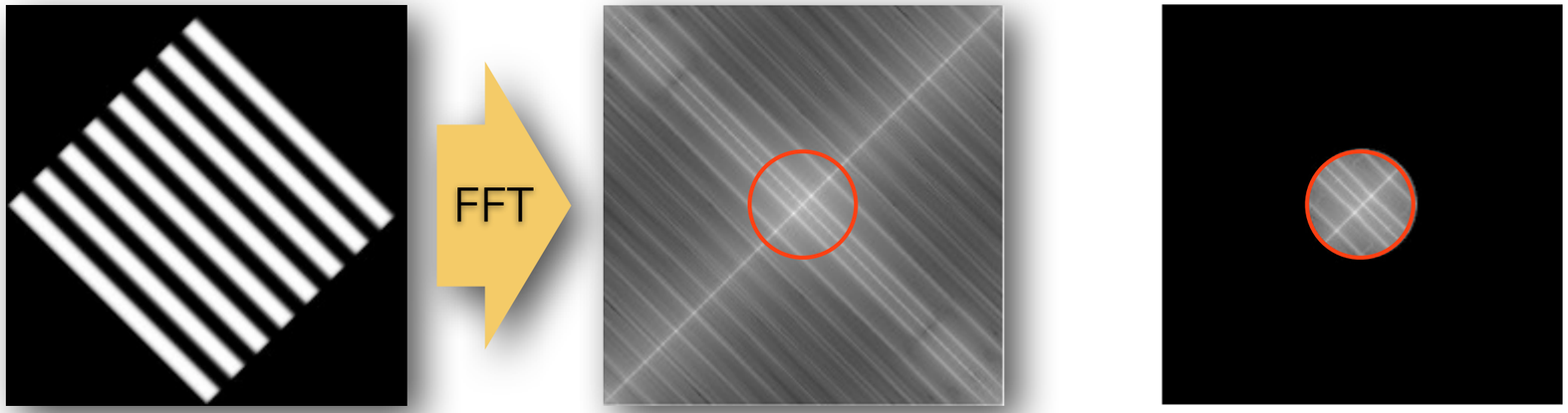


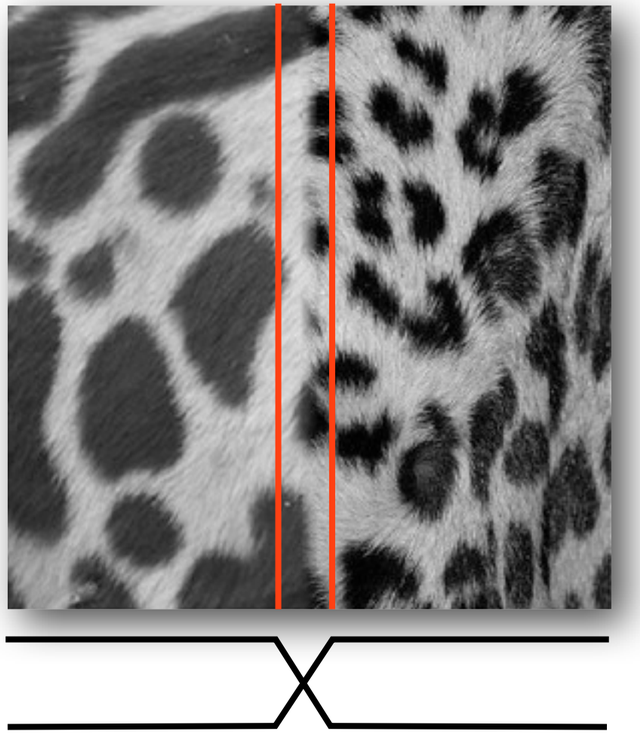


- ★ To avoid seams: Window = size of largest prominent "feature"
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent "feature"
- ★ Use Fourier domain
  - Largest frequency ≤ 2 × size of smallest frequency
  - Image frequency content should occupy one "octave" (power of two)

12

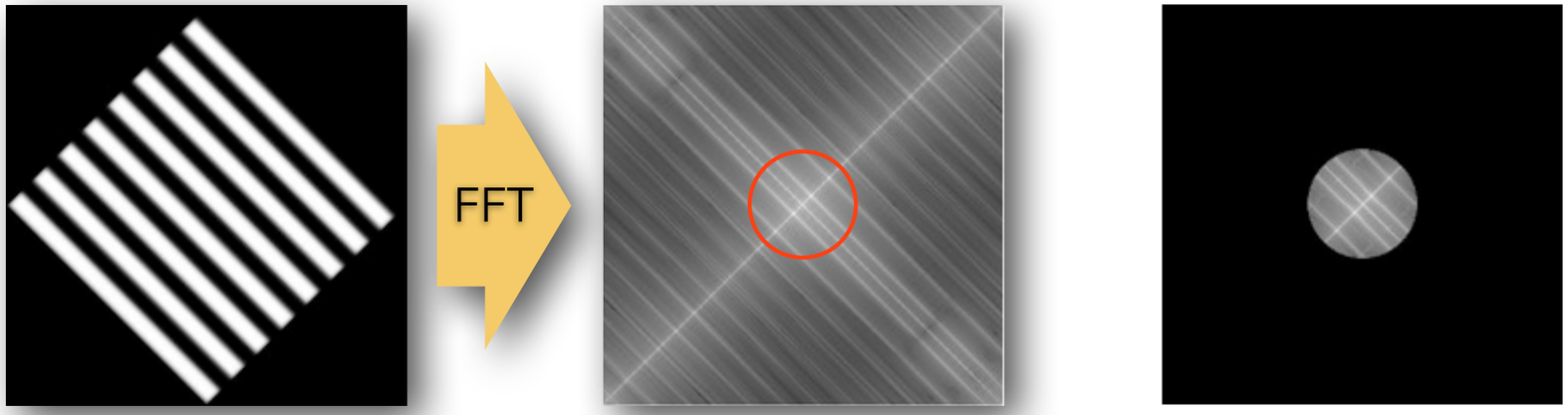
Factors for Optimal Window Size

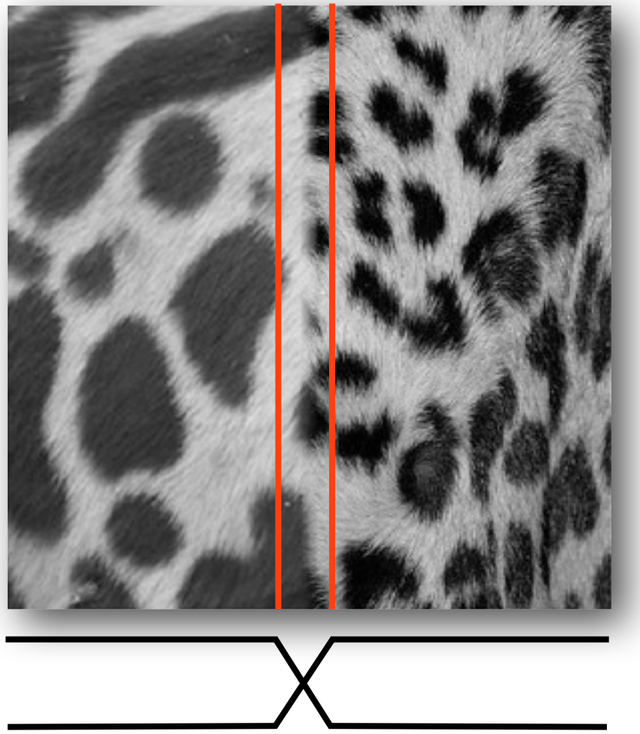




- ★ To avoid seams: Window = size of largest prominent "feature"
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent "feature"
- ★ Use Fourier domain
  - Largest frequency ≤ 2 × size of smallest frequency
  - Image frequency content should occupy one "octave" (power of two)

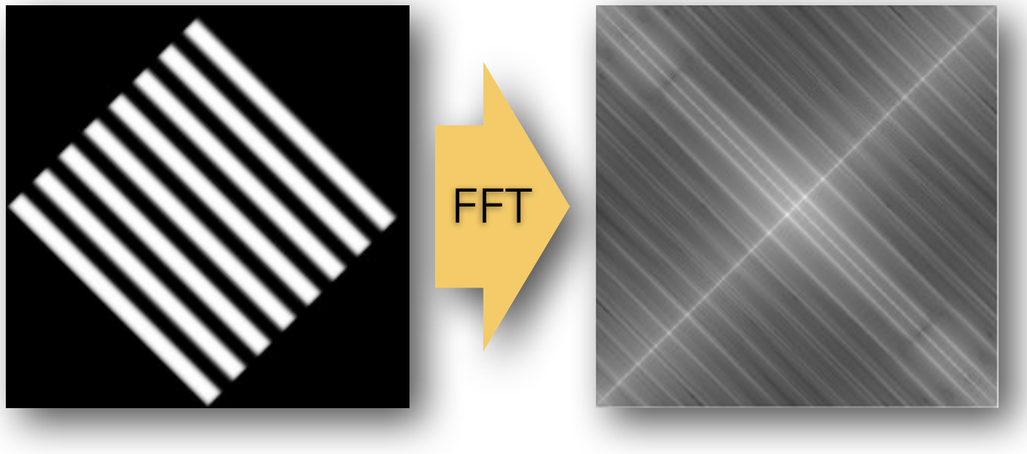
Factors for Optimal Window Size



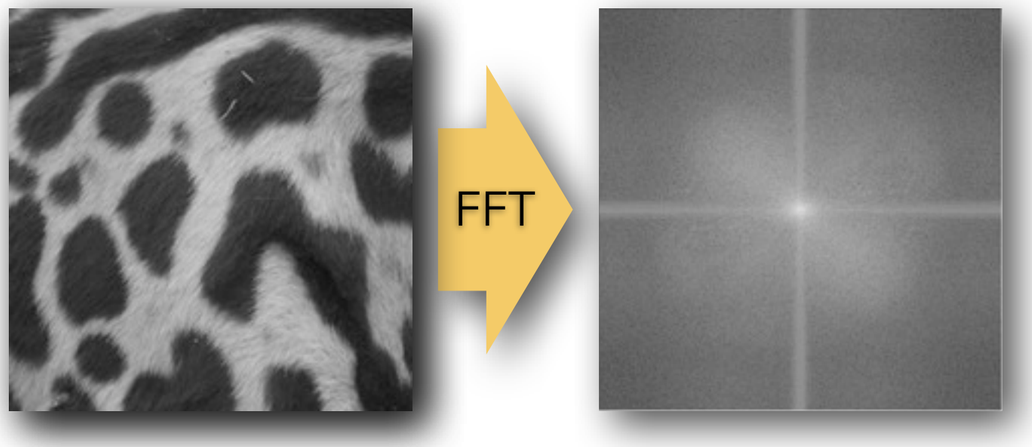


- ★ To avoid seams: Window = size of largest prominent "feature"
- ★ To avoid ghosting: Window ≤ 2×size of smallest prominent "feature"
- ★ Use Fourier domain
  - Largest frequency ≤ 2 × size of smallest frequency
  - Image frequency content should occupy one "octave" (power of two)

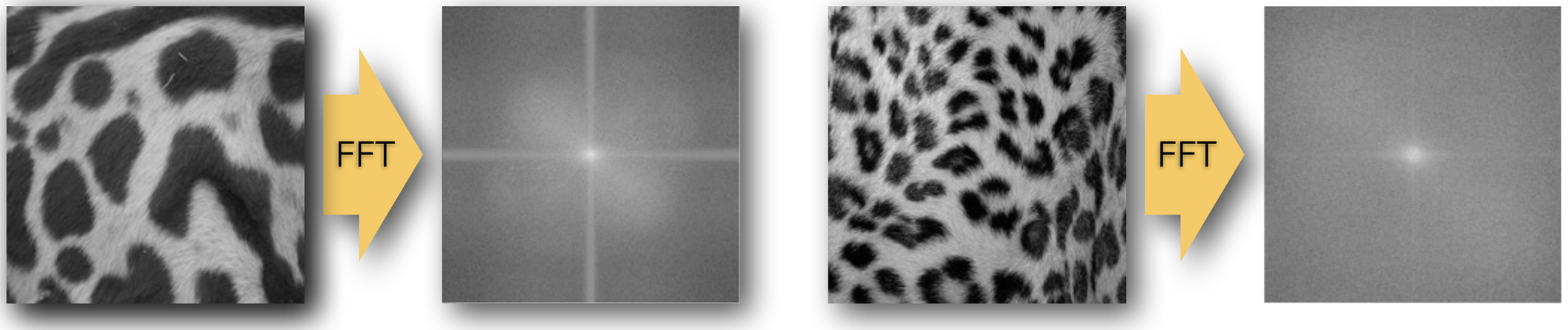
Frequency Spread needs to be Modeled



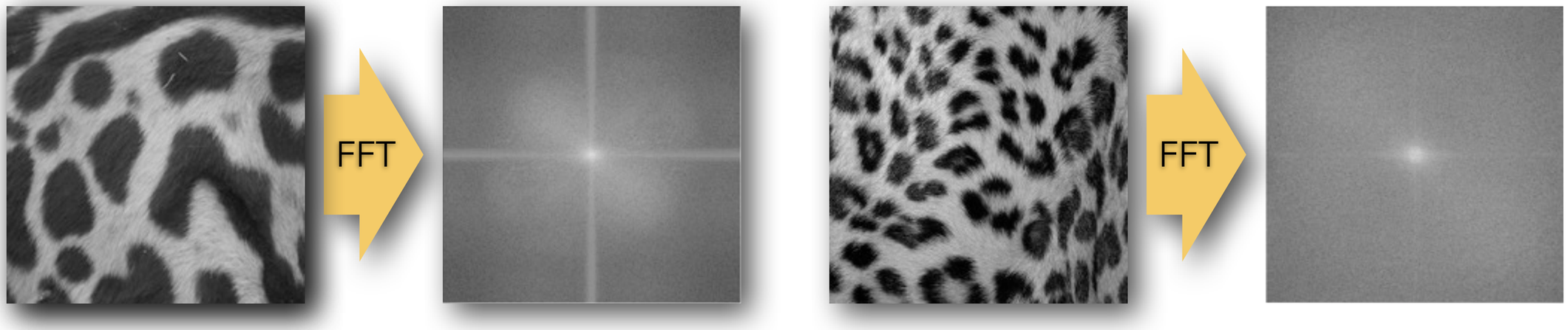
Frequency Spread needs to be Modeled



Frequency Spread needs to be Modeled

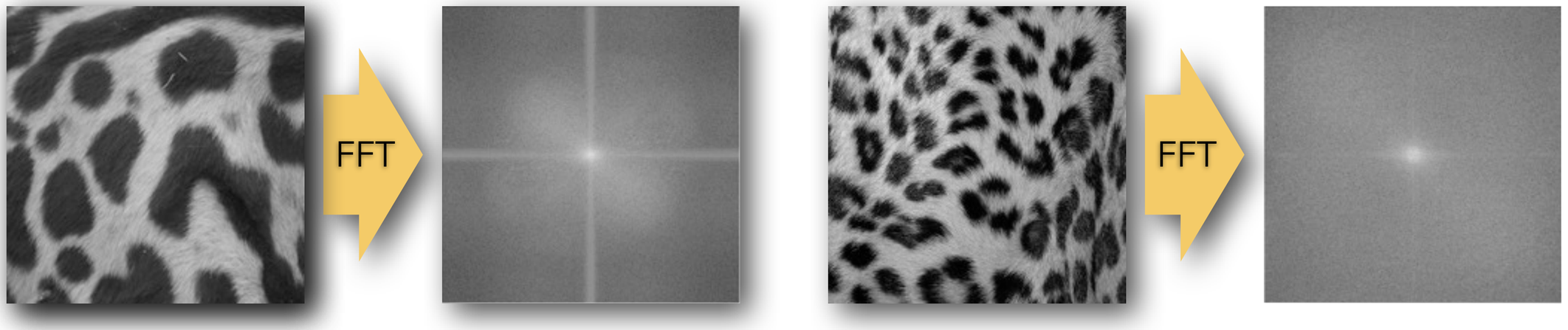


Frequency Spread needs to be Modeled



★ Compute: $FFT(I_l) \Rightarrow F_l, \quad FFT(I_r) \Rightarrow F_r$

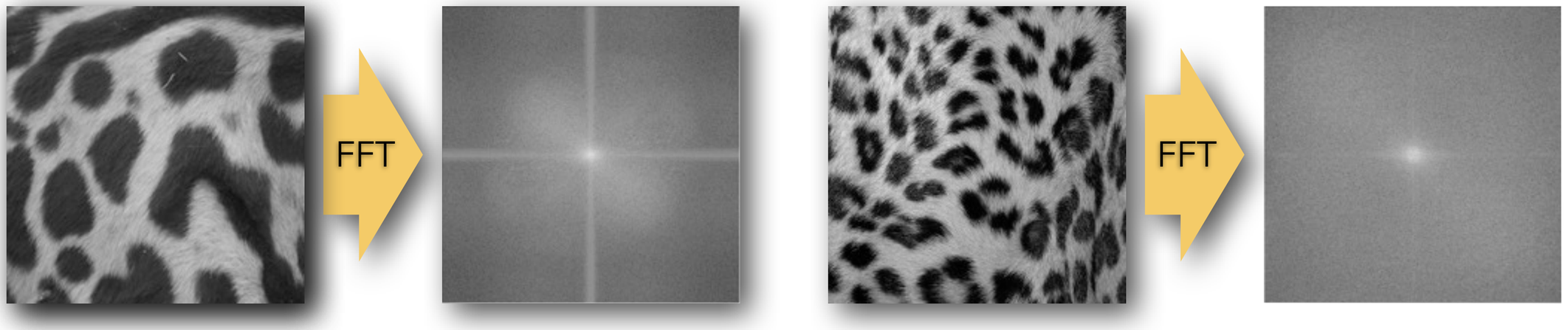
Frequency Spread needs to be Modeled



- Compute: $FFT(I_l) \Rightarrow F_l, \quad FFT(I_r) \Rightarrow F_r$
- Decompose Fourier image into octaves (bands)

$$F_l = F_l^1 + F_l^2 + F_l^3 + \ldots, \quad F_r = F_r^1 + F_r^2 + F_r^3 + \ldots$$

Frequency Spread needs to be Modeled

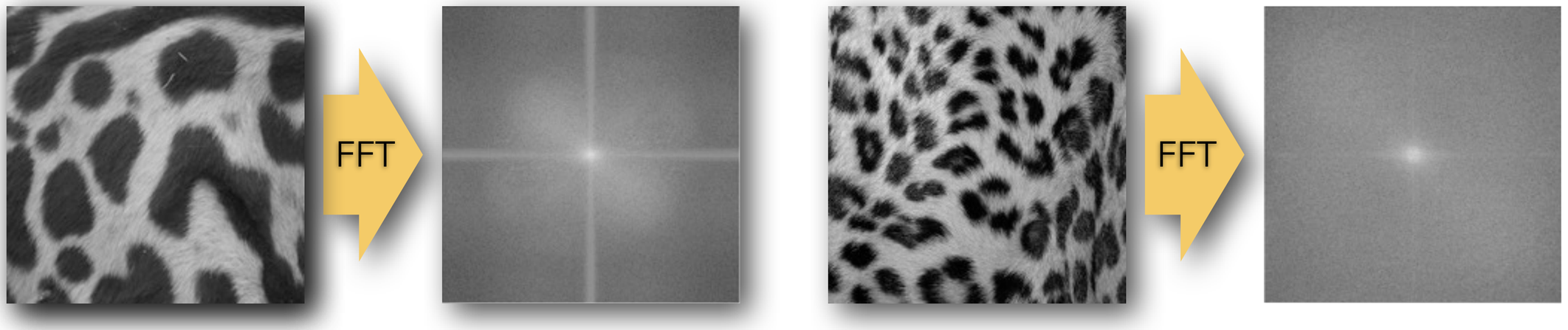


- Compute: $FFT(I_l) \Rightarrow F_l, \quad FFT(I_r) \Rightarrow F_r$
- Decompose Fourier image into octaves (bands)

$$F_l = F_l^1 + F_l^2 + F_l^3 + \ldots, \quad F_r = F_r^1 + F_r^2 + F_r^3 + \ldots$$

- "Feather" corresponding octaves of: $F_l \quad F_r$

Frequency Spread needs to be Modeled

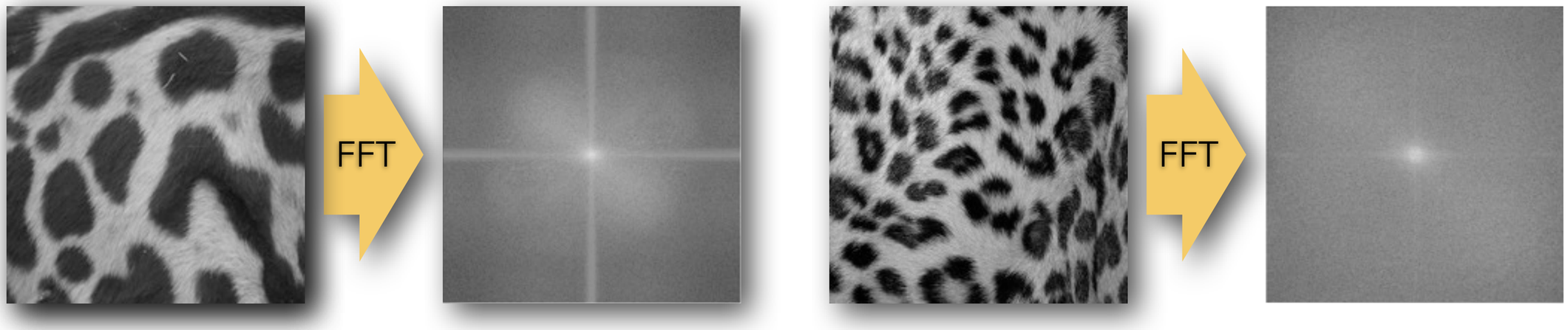


- Compute: $FFT(I_l) \Rightarrow F_l, \quad FFT(I_r) \Rightarrow F_r$
- Decompose Fourier image into octaves (bands)

$$F_l = F_l^1 + F_l^2 + F_l^3 + \ldots, \quad F_r = F_r^1 + F_r^2 + F_r^3 + \ldots$$

- "Feather" corresponding octaves of: $F_l \quad F_r$
- Compute inverse FFT and feather in spatial domain

Frequency Spread needs to be Modeled

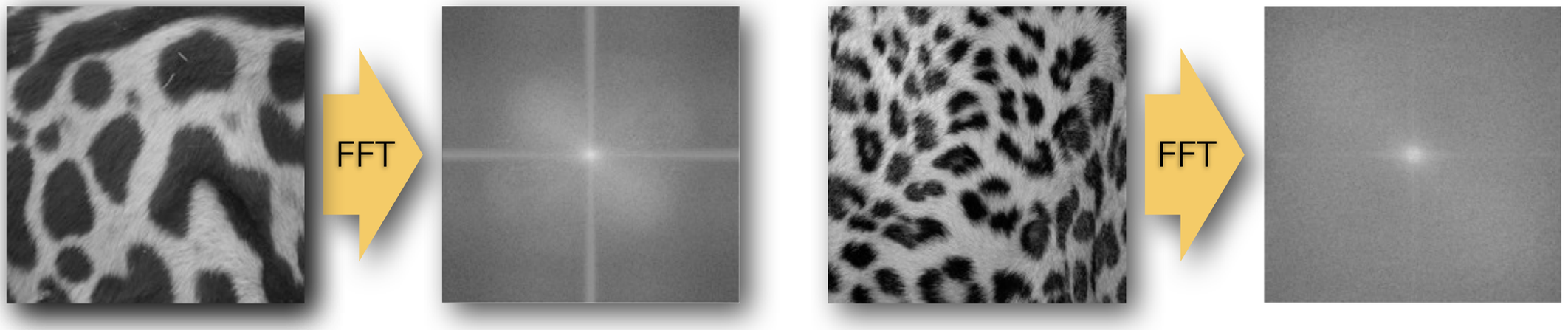


- Compute: $FFT(I_l) \Rightarrow F_l, \quad FFT(I_r) \Rightarrow F_r$
- Decompose Fourier image into octaves (bands)

$$F_l = F_l^1 + F_l^2 + F_l^3 + \ldots, \quad F_r = F_r^1 + F_r^2 + F_r^3 + \ldots$$

- "Feather" corresponding octaves of: $F_l \quad F_r$
- Compute inverse FFT and feather in spatial domain
- Sum feathered octave images in frequency domain

Frequency Spread needs to be Modeled



- Compute: $FFT(I_l) \Rightarrow F_l, \quad FFT(I_r) \Rightarrow F_r$
- Decompose Fourier image into octaves (bands)

$$F_l = F_l^1 + F_l^2 + F_l^3 + \ldots, \quad F_r = F_r^1 + F_r^2 + F_r^3 + \ldots$$

- "Feather" corresponding octaves of: $F_l \quad F_r$
- Compute inverse FFT and feather in spatial domain
- Sum feathered octave images in frequency domain
- Burt and Adelson (1983)

Feathering

- Feathering is achieved by “blurring” the edges before applying the blend operations.
- Blurring makes the blend, smoother.

14

Feathering

- Feathering is achieved by “blurring” the edges before applying the blend operations.
- Blurring makes the blend, smoother.

14






Feathering

- Feathering is achieved by “blurring” the edges before applying the blend operations.
- Blurring makes the blend, smoother.

14

15

Summary

- ★ Introduced the concept of merging two images.
- ★ Presented the two (2) issues caused by not being able to determine the window used for merging images
- ★ Presented the two (2) advantages of a using the Fourier Domain.

16

Next Class

★ Merging and Blending of Images: Use of Pyramids

17

Credits

- For more information, see
  - Szeliski (2010) Computer Vision: Algorithms and Applications, Springer.
  - Burt and Adelson (1983) “The Laplacian Pyramid as a Compact Image Code”, In IEEE Transactions on Communications, 31 (4). p 532-540. 1983 (DOI)
  - Burt and Adelson (1983) “A multiresolution spline with application to image mosaics”. In ACM Transactions on Graphics, 2 (4). 1983 (DOI)
- Some concepts in slides motivated by similar slides by A. Efros and J. Hays.
- Some images retrieved from
  - http://commons.wikimedia.org/.
  - List will be available on website.
  - by Irfan Essa

www.flickr.com/photos/neneonline/231886965/

Georgia Institute of Technology

Computational Photography

Dr. Irfan Essa
Professor
School of Interactive Computing

Study the basics of computation and its impact on the entire workflow of photography, from capturing, manipulating and collaborating on, and sharing photographs.