Seam Carving for Content-Aware Image Resizing

Shai Avidan
Mitsubishi Electric Research Labs
Ariel Shamir
The Interdisciplinary Center & MERL

The Operator

- approach to
  - content-aware resizing is to remove pixels in a judicious manner.
- how to chose the pixels to be removed?
- goal is to
  - remove unnoticeable pixels that blend with their surroundings.

(a) Original (b) Crop (c) Column (d) Seam (e) Pixel (f) Optimal

Figure 2: Results of 5 different strategies for reducing the width of an image. (a) the original image and its e_1 energy function, (b) best cropping, (c) removing columns with minimal energy, (d) seam removal, (e) removal of the pixel with the least amount of energy in each row, and finally, (f) global removal of pixels with the lowest energy, regardless of their position. Figure 3 shows the energy preservation of each strategy.

The Operator

- Given an energy function, let **I** be an n × m image

$$e_1(\mathbf{I}) = |\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|$$

- define a vertical seam

$$\mathbf{s^x} = \{s_i^x\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n, \text{ s.t. } \forall i, |x(i) - x(i-1)| \leq 1,$$

- where ***x*** is a mapping ***x*** : [1, . . . ,n] $\longrightarrow$ [1, . . . ,m].
- a vertical seam
  - is an 8-connected path of pixels in the image from top to bottom,
  - containing one, and only one, pixel in each row of the image
- The pixels of the path of seam s (e.g. vertical seam {si}) will therefore be

$$\mathbf{I_s} = \{\mathbf{I}(s_i)\}_{i=1}^n = \{\mathbf{I}(x(i), i)\}_{i=1}^n$$

The Operator

- define a horizontal seam

$$\mathbf{s^y} = \{s_j^y\}_{j=1}^m = \{(j, y(j))\}_{j=1}^m, \text{ s.t. } \forall j |y(j) - y(j-1)| \leq 1$$

- y is a mapping
  - y : [1, . . . ,m] → [1, . . . ,n]

- Removing
  - the pixels of a seam from an image has only a local effect:
  - all the pixels of the image are shifted left (or up) to compensate for the missing path.

- one can replace
  - the constraint $|x(i)-x(i-1)| \leqq 1$ with $|x(i)-x(i-1)| \leqq k$,
  - and get either a simple column (or row) for k = 0, a piecewise connected
  - Or even completely disconnected set of pixels for any value $1 \leqq k \leqq m$.

The Operator

- Given an energy function e, we can define the cost of a seam as $E(\mathbf{s}) = E(\mathbf{I_s}) = \sum_{i=1}^{n} e(\mathbf{I}(s_i))$

- look for the optimal seam s* that minimizes this seam cost :

$$s^* = \min_{\mathbf{s}} E(\mathbf{s}) = \min_{\mathbf{s}} \sum_{i=1}^{n} e(\mathbf{I}(s_i))$$

- The optimal seam can be found using dynamic programming.

The Operator

- The first step is
  - to traverse the image from the second row to the last row
  - and compute the cumulative minimum energy M for all possible connected seams for each entry (i, j):

$$M(i,j) = e(i,j) + \min(M(i-1,j-1), M(i-1,j), M(i-1,j+1))$$

- At the end of this process,
  - the minimum value of the last row in M will indicate the end of the minimal connected vertical seam.
- in the second step
  - backtrack from this minimum entry on M to find the path of the optimal seam .

Content from:

Mr. Visanu Upatumphan
Mr. Yingpong Vimugtipant
Miss. Sompak Boonwong

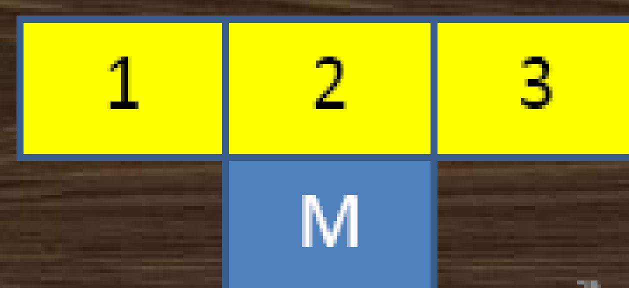
FIND THE BEST SEAM

Find M values: values prepared to find a seam

- Start with Second row
- Follow M equation for the individual

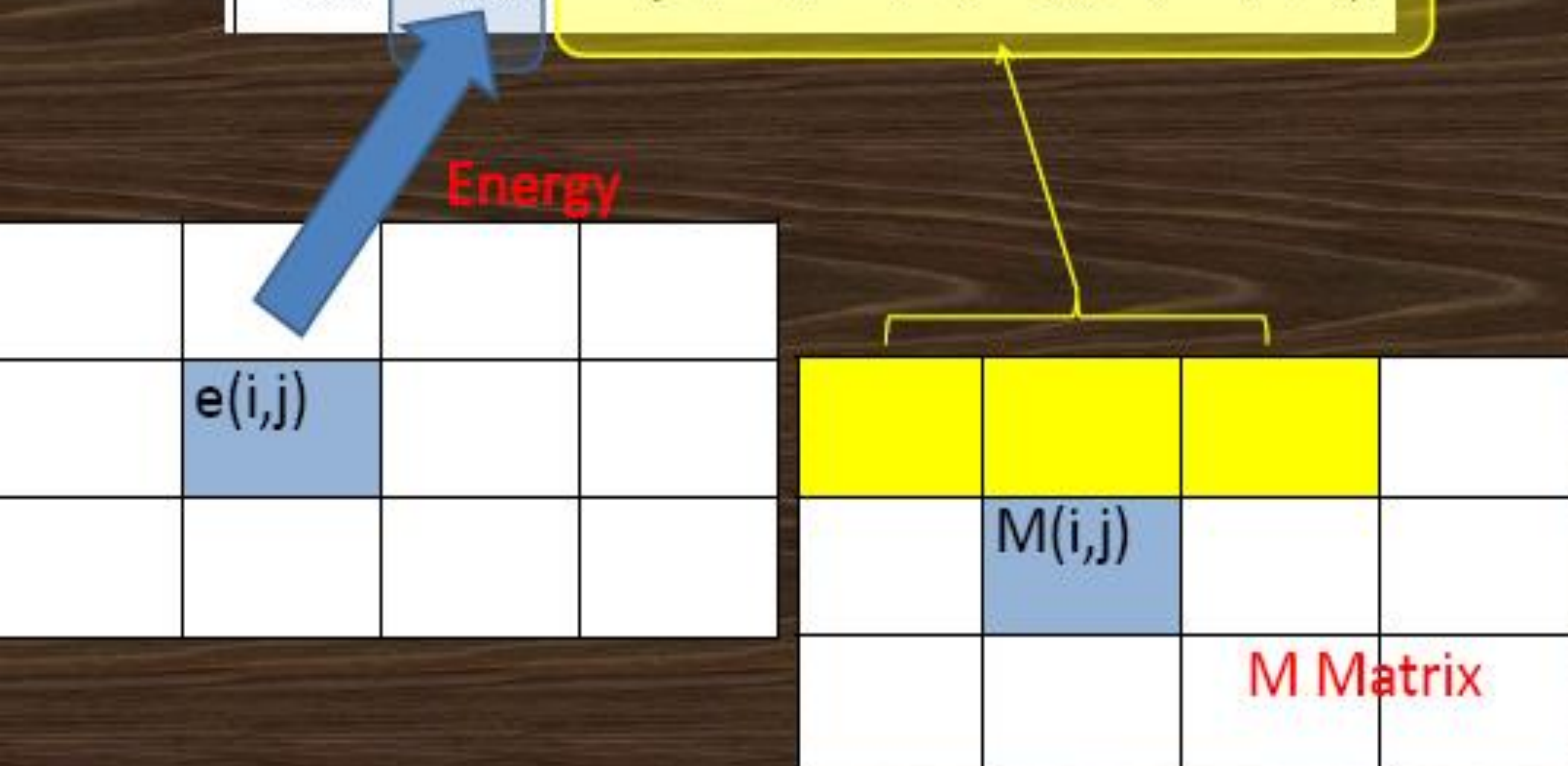
$$M(i,j) = e(i,j) + \min\big(M(i-1,j-1), M(i-1,j), M(i-1,j+1)\big)$$

- Keep track the minimal M values



COMPUTING M VALUES

$$M(i,j) = e(i,j) + \min\big(M(i-1,j-1), M(i-1,j), M(i-1,j+1)\big)$$



FIND THE BEST SEAM

- Example: energy matrix

5.5	16	22.5	13	16
6.5	8	11.5	20	6.5
2.5	9.5	7	7.5	4
5.5	3.5	5.5	10	4.5
6.5	3.5	9.5	9	0.5

FIND THE BEST SEAM

To find M, Start with the second row. In the first row we can use energy

5.5	16	22.5	13	16
6.5	8	11.5	20	6.5
2.5	9.5	7	7.5	4
5.5	3.5	5.5	10	4.5
6.5	3.5	9.5	9	0.5

FIND THE BEST SEAM

Find M value of this pixel

5.5	16	22.5	13	16
6.5	8	11.5	20	6.5
2.5	9.5	7	7.5	4
5.5	3.5	5.5	10	4.5
6.5	3.5	9.5	9	0.5

FIND THE BEST SEAM

Use 2 values to compare the minimum one

Keep track

5.5	16	22.5	13	16
12				

6.5+5.5 = 12

0	0	0
2		

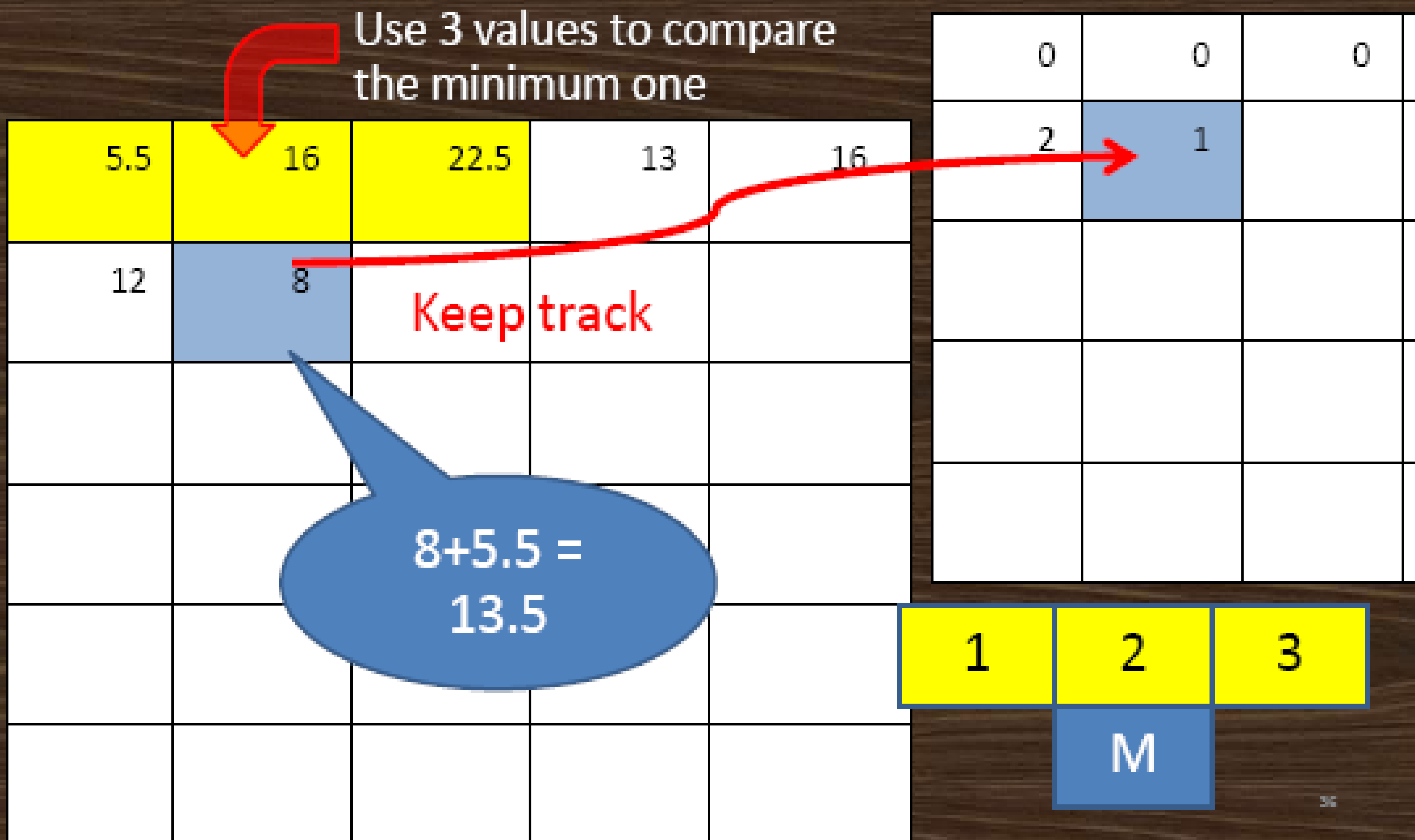
1	2	3
	M	

FIND THE BEST SEAM

Find M value of this pixel

5.5	16	22.5	13	16
6.5	8	11.5	20	6.5
2.5	9.5	7	7.5	4
5.5	3.5	5.5	10	4.5
6.5	3.5	9.5	9	0.5

FIND THE BEST SEAM



FIND THE BEST SEAM

5.5	16	22.5	13	16
12	13.5	11.5		

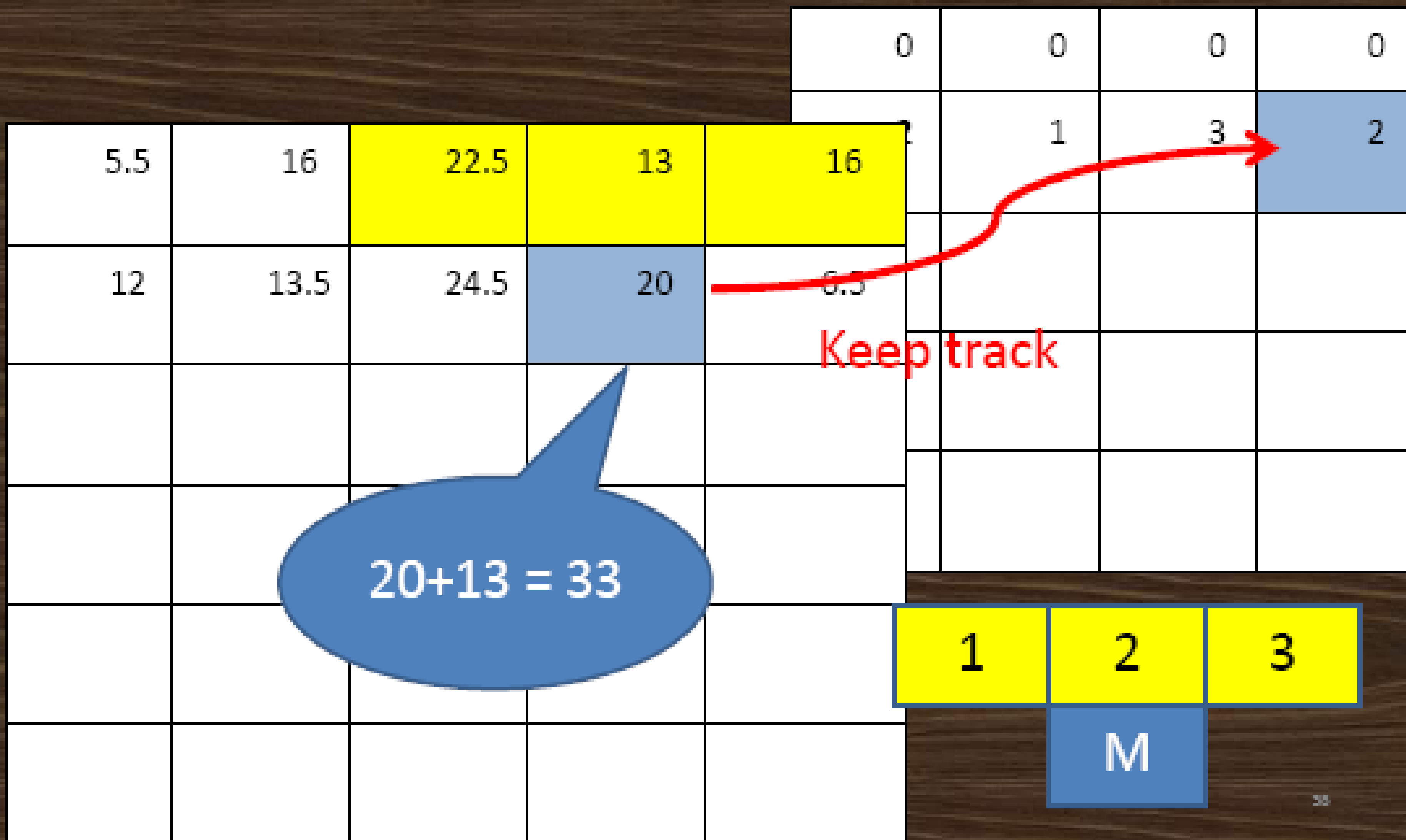
11.5+13 = 24.5

Keep track

0	0	0
2	1	3

1	2	3
	M	

FIND THE BEST SEAM



FIND THE BEST SEAM

0	0	0	0	0
		3	2	1

5.5	16	22.5	13	16
12	13.5	11.5	33	6.5

Keep track

6.5+13 = 19.5

FIND THE BEST SEAM

- Matrix of M

5.5	16	22.5	13	16
12	13.5	24.5	33	19.5
14.5	21.5	20.5	27	23.5
20	18	26	30.5	28
24.5	21.5	27.5	35	28.5

FIND THE BEST SEAM

- Matrix of Keeping track

0	0	0	0	0
2	1	3	2	1
2	1	1	3	2
2	1	2	1	2
3	2	1	1	2

BACKTRACK

– This process will produce the seam

- Find the least M value of the last row
- Follow the track that we keep from the previous process

BACKTRACK

- Example

0	0	0	0	0
2	1	3	2	1
2	1	1	3	2
2	1	2	1	2
	2	1	1	2

Backtrack

5.5	16	22.5	13	16
12	13.5	24.5	33	19.5
14.5	21.5	20.5	27	23.5
20	18	26	30.5	28
24.5	21.5	27.5	35	28.5

1	2	3
	M	

BACKTRACK

0	0	0	0	0
2	1	3	2	1
2	1	1	3	
2	1	2	1	
3	2	1	1	

Least M at the last row

5.5	16	22.5	13	16
12	13.5	24.5	33	19.5
14.5	21.5	20.5	27	23.5
20	18	26	30.5	28
24.5	21.5	27.5	35	28.5

1	2	3
	M	

BACKTRACK

1	2	3
	M	

0	0	0	0	0
2	1	3	2	1
2	1	1	3	
2	1	2	1	
3	2	1	1	

5.5	16	22.5	13	16
12	13.5	24.5	33	19.5
14.5	21.5	20.5	27	23.5
20	18	26	30.5	28
24.5	21.5	27.5	35	28.5

BACKTRACK

0	0	0	0	0
2	1	3	2	1
2	1	1	3	2
2	1	2	1	
3	2	1	1	

1	2	3
	M	

5.5	16	22.5	13	16
12	13.5	24.5	33	19.5
14.5	21.5	20.5	27	23.5
20	18	26	30.5	28
24.5	21.5	27.5	35	28.5

BACKTRACK

0	0	0	0	0
2	1	3	2	1
2	1	1	3	
2	1	2	1	
3	2	1	1	

1	2	3
	M	

5.5	16	22.5	13	16
12	13.5	24.5	33	19.5
14.5	21.5	20.5	27	23.5
20	18	26	30.5	28
24.5	21.5	27.5	35	28.5

BACKTRACK

1	2	3
	M	

0	0	0	0	0
2	1	3	2	1
2	1	1	3	2
2	1	2	1	
3	2	1	1	

5.5	16	22.5	13	16
12	13.5	24.5	33	19.5
14.5	21.5	20.5	27	23.5
20	18	26	30.5	28
24.5	21.5	27.5	35	28.5

DELETE SEAM

In order to reduce the image size, we can do by delete Seam:

- remove that seam directly
- shift the rest to fill the blanks

DELETE SEAM

- Example

5.5	16	22.5	13	16
12	13.5	24.5	33	
14.5	21.5	20.5	27	
20	18	26	30.5	
24.5	21.5	27.5	35	

16	22.5	13	16
13.5	24.5	33	19.5
21.5	20.5	27	23.5
20	26	30.5	28
24.5	27.5	35	28.5

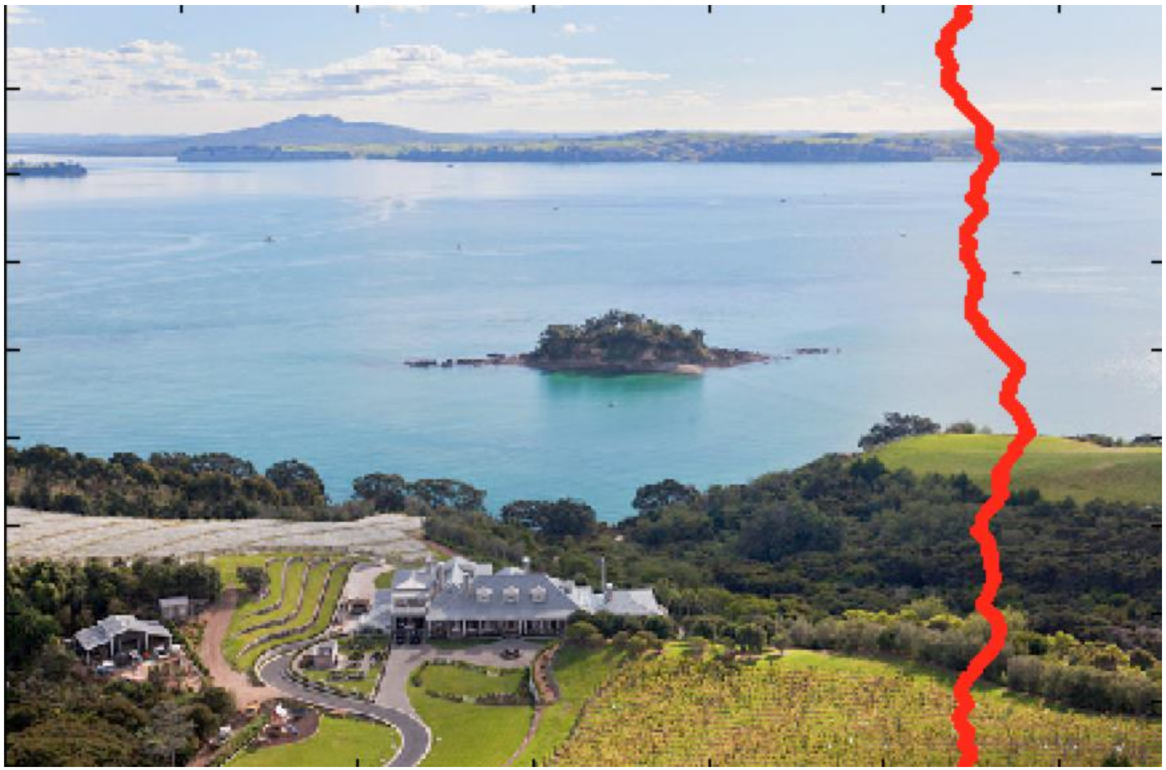
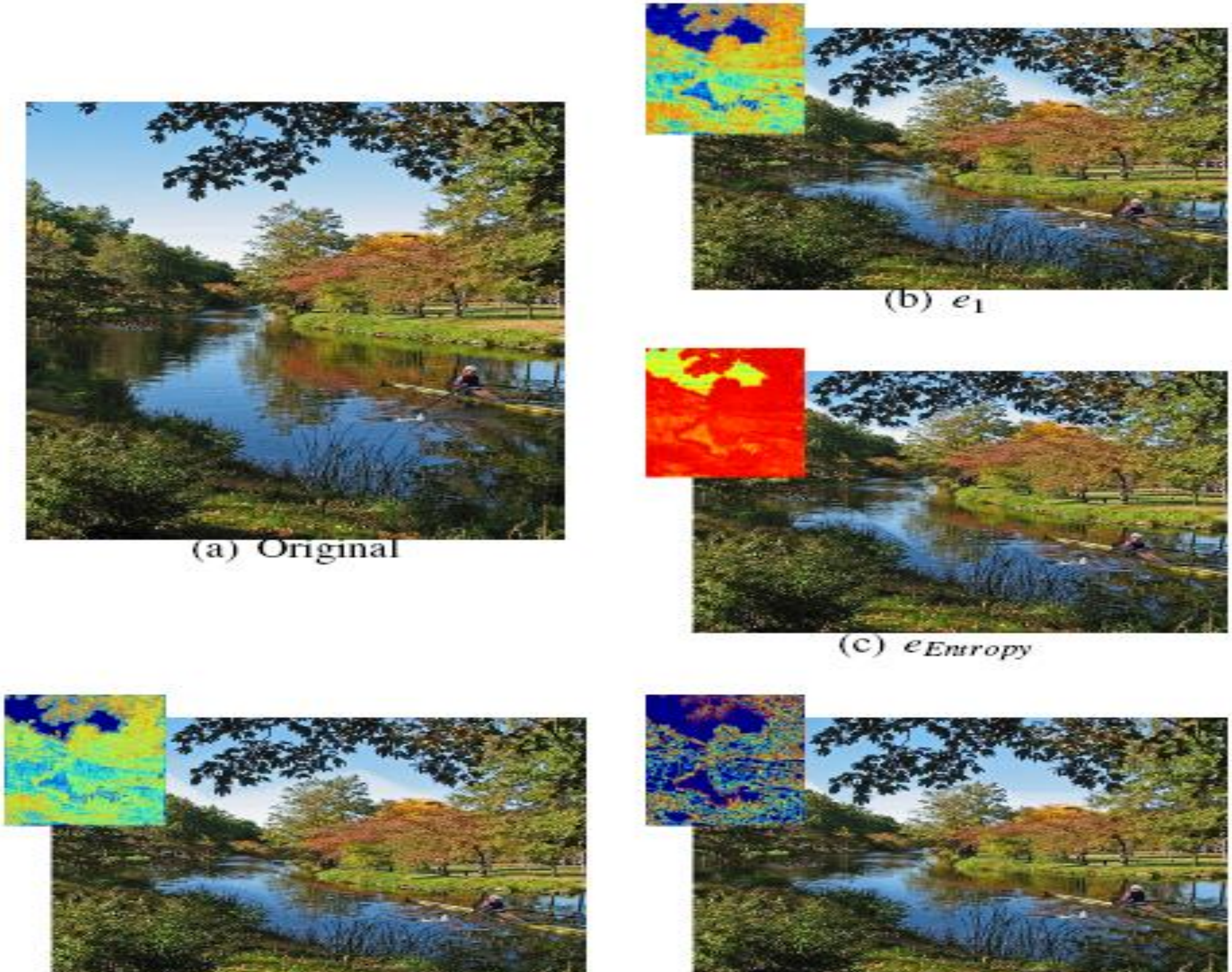


Image Energy Functions

- The entropy energy
  - Computes the entropy over a 9×9 window and adds it to e1.
- The segmentation method
  - first segments the image [Christoudias et al. 2002]
  - and then applies the e1 error norm on the results,
  - effectively leaving only the edges between segments.
- eHoG is defined as follows:

$$e_{HoG}(\mathbf{I}) = \frac{|\frac{\partial}{\partial x}\mathbf{I}| + |\frac{\partial}{\partial y}\mathbf{I}|}{\max\left(HoG(\mathbf{I}(x,y))\right)},$$

- where HoG(I(x,y))
  - is taken to be a **histogram of oriented gradients** at every pixel [Dalal and Triggs 2005].
  - use an 8-bin histogram computed over a 11×11 window around the pixel.
- found either e1 or eHoG to work quite well.



(a) Original

(b) e_1

(c) $e_{Entropy}$

(d) e_{HoG}

(e) Segmentation and L_1

Figure 4: Comparing different energy functions for content aware resizing.

Discrete Image Resizing

- Aspect Ratio Change
  - given image I from $n \times m$ to $n \times m'$
  - where $m - m' = c$
  - be achieved simply by
    - successively removing c vertical seams from I. (Figure 5).
  - can also be achieved by
    - increasing the number of column (Figure 6).
    - The added value of such an approach is that
      - » it does not remove any information from the image.

Figure 5: Comparing aspect ratio change. From left to right in the bottom: the image resized using seam removals, scaling and cropping.

Figure 6: Aspect ratio change of pictures of the Janapese master Utagawa Hiroshige, by seam insertion.

Retargeting with Optimal Seams-Order

- Image retargeting
  - generalizes aspect ratio
    - change from one dimension to two dimensions
  - such that an image I of size $n \times m$
    - will be retargeted to size $n' \times m'$ and,
    - assume that $m' < m$ and $n' < n$
- what is the correct order of seam carving?
  - remove vertical seams first?
  - horizontal seams first?
  - or alternate between the two?

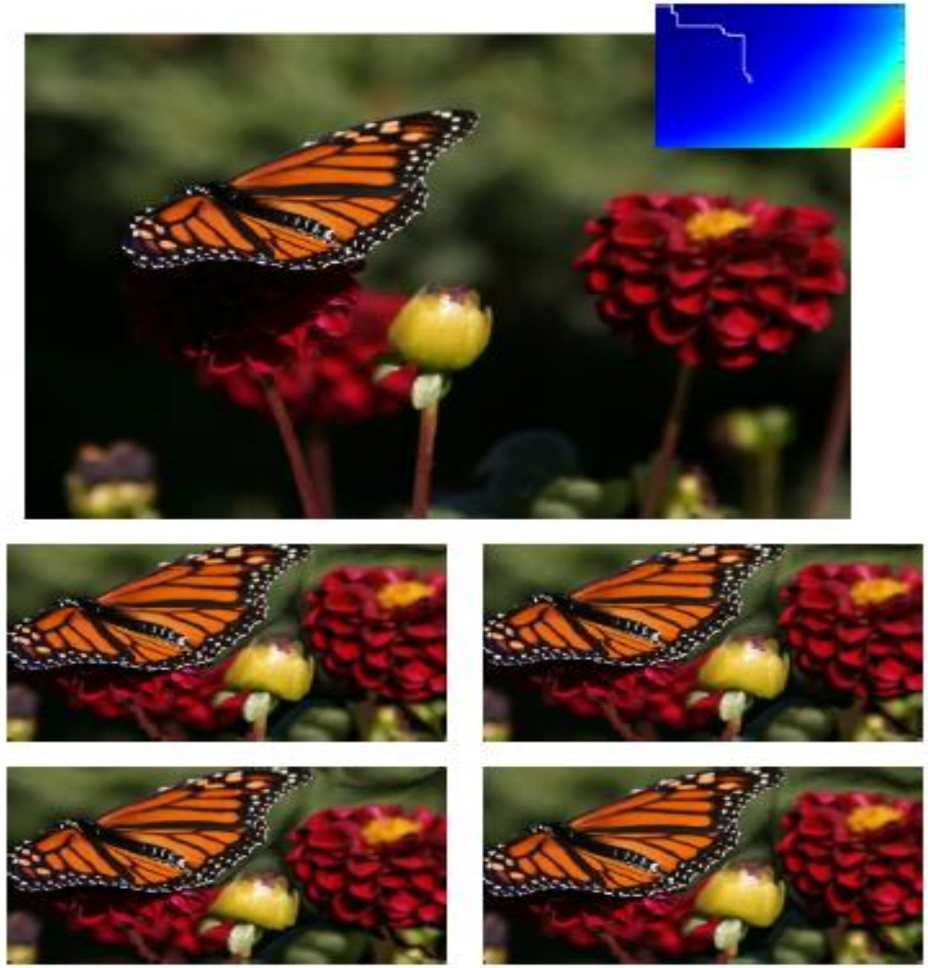


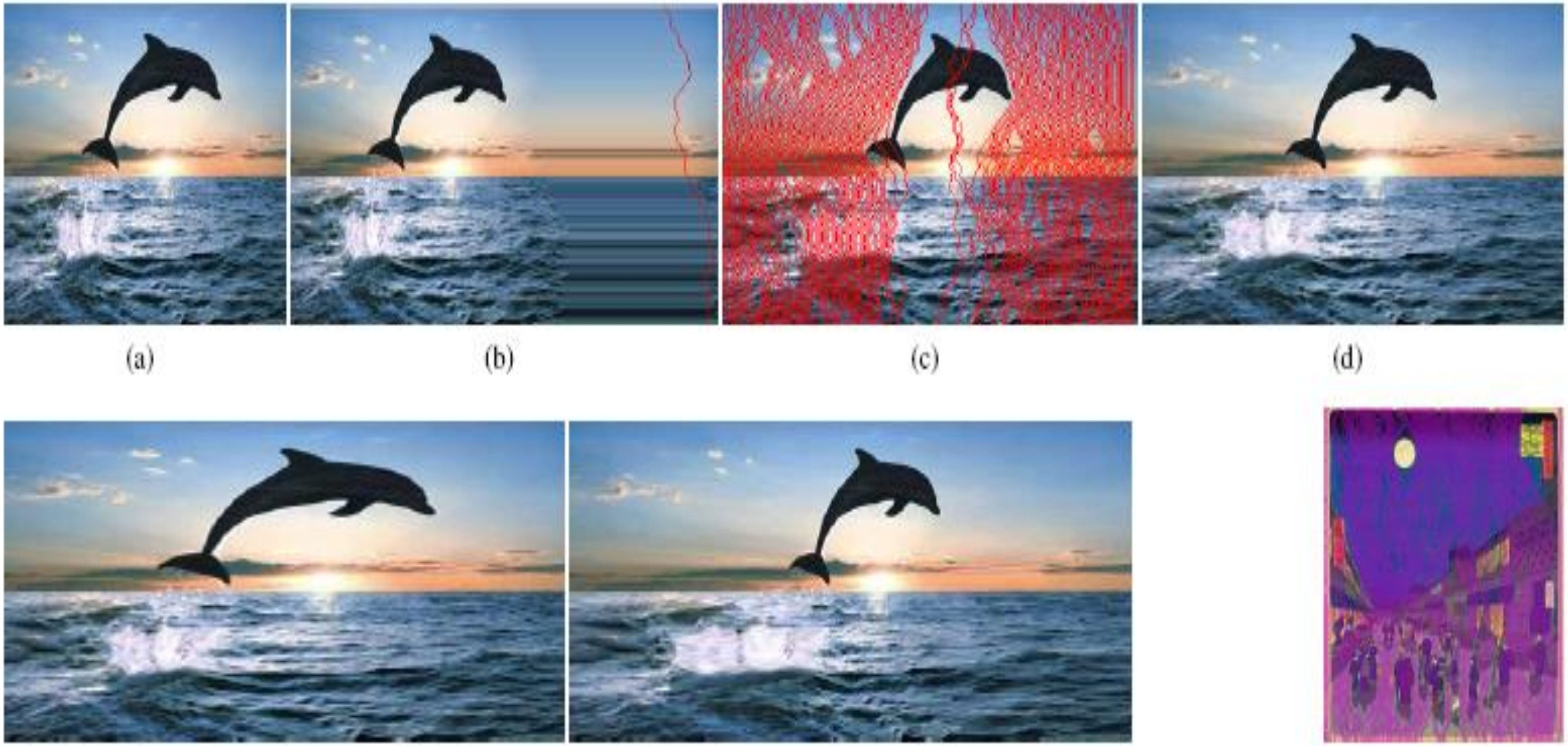
Figure 7: Optimal order retargeting: On the top is the original image and its transport map **T**. Given a target size, we follow the optimal path (white path on **T**) to obtain the retargeted image (top row, right). For comparison we show retargeting results by alternating between horizontal and vertical seam removal (top row, left), removing vertical seams first (bottom row, left), and removing horizontal seams first (bottom row, right)

Image Enlarging

- denote $I^{(t)}$ as
  - the smaller image created after t seam have been removed from I.
- denote $I^{(-1)}$ as
  - the larger image created after 1 seam have been enlarged from I
  - compute the optimal vertical (horizontal) seam s on I
  - and duplicate the pixels of s by averaging them with their left and right neighbors (top and bottom in the horizontal case).
- denote $I^{(-k)}$ as
  - enlarge an image by k,
  - find the first k seams for removal,
  - and duplicate them in order to arrive at $I^{(-k)}$

Image Enlarging

- To continue in content-aware fashion for excessive image enlarging (for instance, greater than 50%),
  - break the process into several steps.
  - Each step does not enlarge the size of the image in more than a fraction of its size from the previous step,
  - essentially guarding the important content from being stretched.
  - Nevertheless, extreme enlarging of an image would most probably produce noticeable artifacts (Figure 8 (f)).



(a) (b) (c) (d)

(e) (f) (g)

Figure 8: Seam insertion: finding and inserting the optimum seam on an enlarged image will most likely insert the same seam again and again as in (b). Inserting the seams in order of removal (c) achieves the desired 50% enlargement (d). Using two steps of seam insertions of 50% in (f) achieves better results than scaling (e). In (g), the seams inserted to expand figure 6 are shown.

Content Amplification

- **Content Amplification**
  - amplify the content of the image while preserving its size
    - be achieved by combining seam carving and scaling.
  - first
    - use standard scaling to enlarge the image and
    - only then apply seam carving on the larger image to carve the image back to its original size (see Figure 9).
  - Note that the pixels removed are in effect sub-pixels of the original image.

Content Amplification

Figure 9: Content amplification. On the right: a combination of seam carving and scaling amplifies the content of the original image (left).

Object Removal

- use a simple user interface for object removal.
  - The user marks the target object to be removed
  - and then seams are removed from the image
  - until all marked pixels are gone.
- The system can automatically
  - calculate the smaller of the vertical or horizontal diameters (in pixels) of the target removal region
  - and perform vertical or horizontal removals accordingly (Figure 11).
- to regain the original size of the image,
  - seam insertion could be employed on the resulting (smaller) image (see Figure 12).

Figure 11: Simple object removal: the user marks a region for removal (green), and possibly a region to protect (red), on the original image (see inset in left image). On the right image, consecutive vertical seam were removed until no 'green' pixels were left.

Figure 12: Object removal: find the missing shoe! (original image is top left). In this example, in addition to removing the object (one shoe), the image was enlarged back to its original size. Note that this example would be difficult to accomplish using in-painting or texture synthesis.

Limitations

- this method
  - does not work automatically on all images.
  - can be corrected by adding higher level cues, either manual or automatic. Figure 14, Figure 15
- Other times,
  - not even high level information can solve the problem.
- two major factors that limit this seam carving approach.
  - The first
    - is the amount of content in an image.
    - If the image is too condensed,
    - it does not contain ‘less important’ areas,
    - then any type of content-aware resizing strategy will not succeed.
  - The second type of limitation
    - is the layout of the image content.
    - In certain types of images, albeit not being condensed,the content is laid out in a manner that prevents the seams to bypass important parts (Figure 16).

Figure 14: Retargeting the left image with e_1 alone (middle), and with a face detector (right).

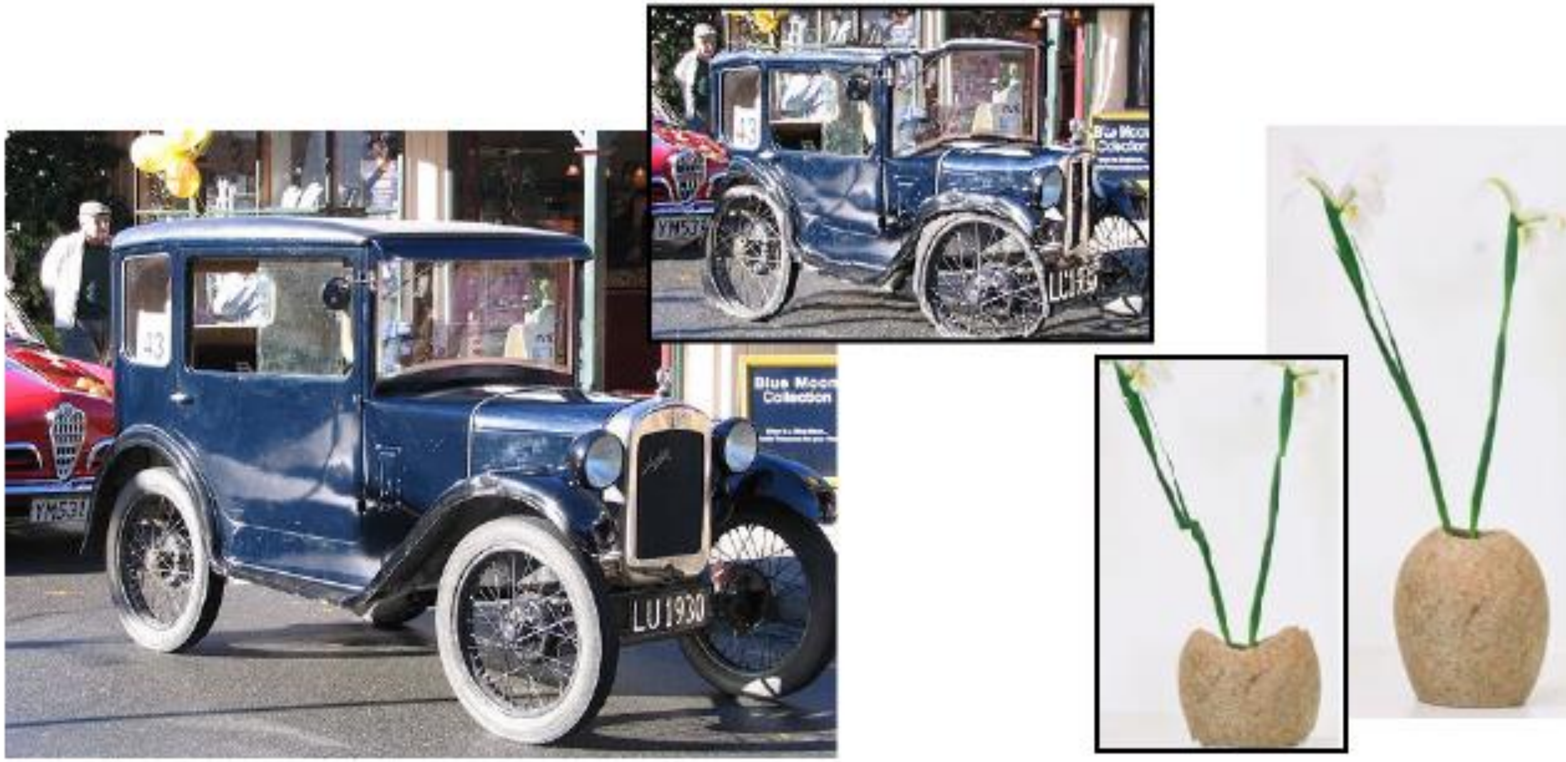


Figure 16: Examples when resizing using seams fails: images that are too condensed (left) or where the content layout prevents seams to bypass important parts (right). In such cases the best strategy would be to use scaling.