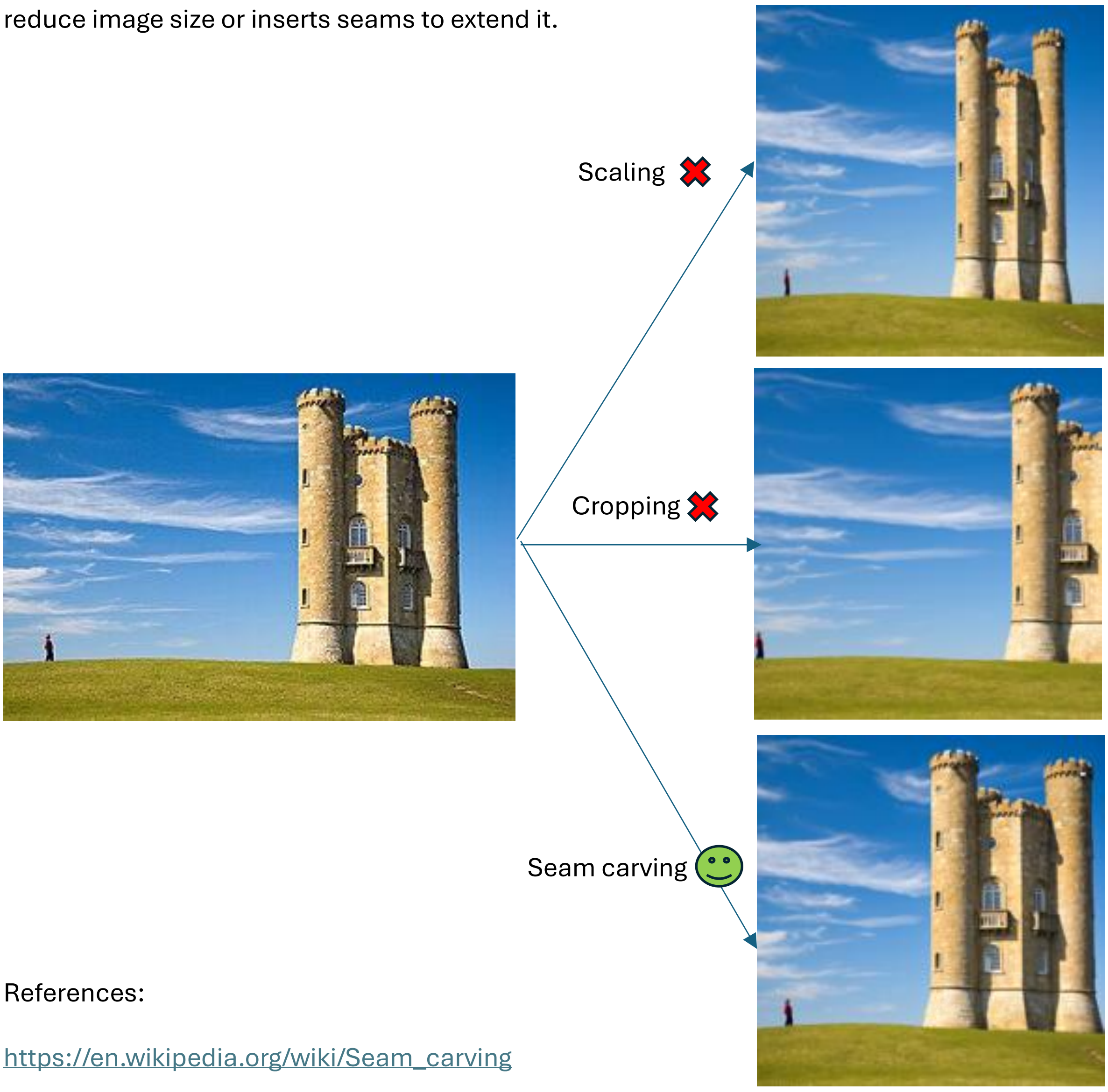
Assignment 2 (100 marks + 10 bonus)

Image Compression by Seam Carving

Seam carving (or liquid rescaling) is an algorithm for content-aware image resizing, developed by Shai Avidan and Ariel Shamir. It functions by establishing a number of seams (paths of least importance) in an image and automatically removes seams to reduce image size or inserts seams to extend it.



References:

https://en.wikipedia.org/wiki/Seam_carving

https://dl.acm.org/doi/10.1145/1275808.1276390

https://avikdas.com/2019/05/14/real-world-dynamic-programming-seam-carving.html

You can choose to use any programming languages, you can use third-party libraries, including image libraries like OpenCV for image processing (reading/writing images, computing gradient magnitude or Sobel filter) and math libraries for math computations. However, you cannot use existing functions of content-aware image resizing in the image libraries.

1. **Implement Seam carving using Dynamic programming** (for both vertical and horizontal seams). The main steps are:
   a) **Calculate the Energy Map:** For each pixel in the image, compute its energy value, which represents the disruption measure. The energy is typically calculated based on the gradient (the difference in intensity) between adjacent pixels, or by using filters like the Sobel filter to detect edges and transitions.
   b) **Use Dynamic Programming to Find the Minimum Seam:** Apply dynamic programming to find the seam (both vertically and horizontally) with the minimum cumulative energy. This seam represents the path through the image with the least disruption, as measured by the sum of the energy values along the seam.
   c) **Remove the Seam and Repeat:** After identifying the minimum energy seam, remove it from the image. Then recalculate the energy map, and repeat the process until the image is resized to the desired dimensions.
2. **Question**:
   a) Note that, this is to help you start, your final implementation does not need to follow this question.
      Given a color picture consisting of an *m x n* array, *img[1…m,1… n]*, of pixels, where each pixel specifies a triple of red, green, and blue (RGB) intensities. Specifically, we wish to remove one pixel from each of the m rows, so that the whole picture becomes one pixel narrower. To avoid disturbing visual effects, however, we require that the pixels removed in two adjacent rows be in the same or adjacent columns; the pixels removed form a "seam" from the top row to the bottom row where successive pixels in the seam are adjacent vertically or diagonally.
      i. Show that the number of such possible seams grows at least exponentially in m, assuming that *n > 1*.
      ii. Suppose now that along with each pixel *img[i, j]*, we have calculated a real valued energy or disruption measure *e[i,j]*, indicating how disruptive or energy effective it would be to remove pixel *img[i,j]*. Intuitively, the lower a pixel's disruption measure (energy), the more similar the pixel is to its neighbors. Suppose further that we define the disruption measure of a seam to be the sum of the disruption measures of its pixels. In other words, how to relate to subproblems?
   b) The gradient can be computed using various filters, such as the Sobel filter. Convolution operations are typically used for this purpose. Discuss whether applying the Divide and Conquer method could improve the efficiency of this

computation.

3. **Implement using Greedy Algorithm**. As an alternative to dynamic programming, implement the seam carving using a Greedy Algorithm. In this approach, select the seam by making a local, row-by-row decision, choosing the minimum-energy pixel at each step, without considering the entire image. Does the Greedy Algorithm provide results that are comparable to dynamic programming in terms of visual quality?
4. **Analysis and discussion**
   a) Answer the questions (Item 2).
   b) Time and space complexity.
   c) Comparison with Greedy Algorithm.
   d) Any improvement, e.g., do we need to maintain the entire DP Table?
5. **Bonus**: up to 10 marks will be given. The bonus will be evaluated based on the technical complexity and quality of the implementation:
   a) Step by step visualization, with all the seams visualized.
   b) User interaction, enable user interactively edit image sizes.
   c) Implement using graph-cuts and compare with the Dynamic Programming approach.
   d) CI/CD
6. **Presentation**
   a) Each person has a maximum of 2 minutes, with a total team time limit of 12 minutes.
   b) Focus on presenting the key content.
   c) Demonstrate your app and show that your program runs successfully.
   d) Record your presentation, upload it to an online platform like YouTube, and include the link in a video.txt file.
7. **Deliverables & Submission Deadline**
   a) XSite Dropbox Assignment 2 Report Only: Report,
   b) XSite Dropbox Assignment 2 Sources and Other Materials
      i. Code (please provide instructions on how to build and run your code)
      ii. AI Usage Declaration, please specify how AI tools were utilized (e.g., during design, implementation, or report writing) and provide the prompts used
      iii. Link to Presentation Video
      iv. Other materials like test input/outputs
   c) Deadline: Week 12, Fri, 21 Nov, 23:59