

Transparency

Gradient Mask Creation:

- In the MaskCreator class, the method `create_mask()` generates a smooth transparency gradient using a smoothstep function. This gradient defines how transparent or opaque each part of the image will be within the masked region.
- The `smoothstep()` function provides a non-linear gradient to create a smooth transition from fully transparent to fully opaque (or vice versa), enhancing visual smoothness.

Transparency Factor:

- The `transparency_factor` parameter is applied to the gradient. This factor scales the gradient values, affecting the overall transparency. A lower `transparency_factor` makes the gradient more transparent, while a factor closer to 1 maintains higher opacity.

Alpha Blending:

- After generating the `alpha_gradient` (adjusted by `transparency_factor`), the `alpha_blending()` method applies it.
- The method defines a region of interest (ROI) on one side of the image (left or right) and blends the region with a black background using the `alpha_gradient` as a mask.
- `alpha` values from `self.__alpha_gradient` determines how much of the ROI pixels and the black background are combined:
- $$\text{blended} = (\text{alpha} * \text{roi} + (1 - \text{alpha}) * \text{black_background})$$
- In this calculation:
 - ◆ `alpha * roi` applies the transparency gradient to the image.
 - ◆ `(1 - alpha) * black_background` ensures a gradual blend with the black background.

Transparency Side Control:

- The transparency gradient can be applied from either left-to-right or right-to-left, controlled by the `image_side` parameter.
- Depending on the `image_side`, the method either keeps the gradient as-is (right side transparent) or reverses it (left side transparent).

So, the transparency effect combines the smoothstep gradient, scaled by `transparency_factor`, with alpha blending to control the transparency on the chosen side.

