

# Neighbourhood-Based Inpainting

Vicknesh Selvam  
Faculty Advisor: Todd Wittman  
3/21/08, UCLA

## Neighbourhood-based Spatial inpainting

- The similarity between two neighbourhoods is given by the Gaussian weighted sum of squared distance (SSD)
- Nearest Neighbour
  - Takes the pixel with the best matching neighbourhood

$$d(N(x), N(y)) = \sum_{\substack{\hat{x} \in N(x) \\ \hat{y} \in N(y)}} G_{\sigma}(\hat{x})(f(\hat{x}) - f(\hat{y}))^2$$

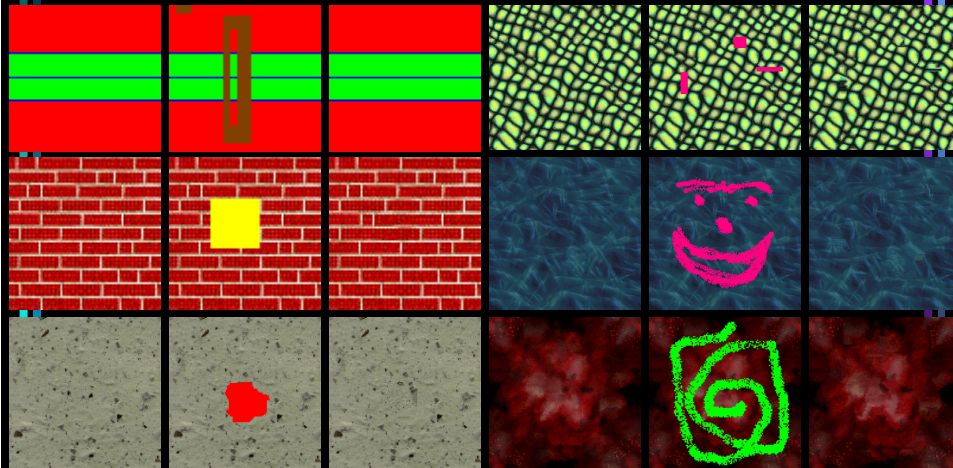
## Expansion to Color Images.

- In Black/White, only a single  $m$  by  $n$  ( $m, n$  odd) window used.
- In Color, however, since its split into 3 grayscale images, we take the SSD of the color image as the sum of the SSDs at each frame.

## Problems with expanding to color

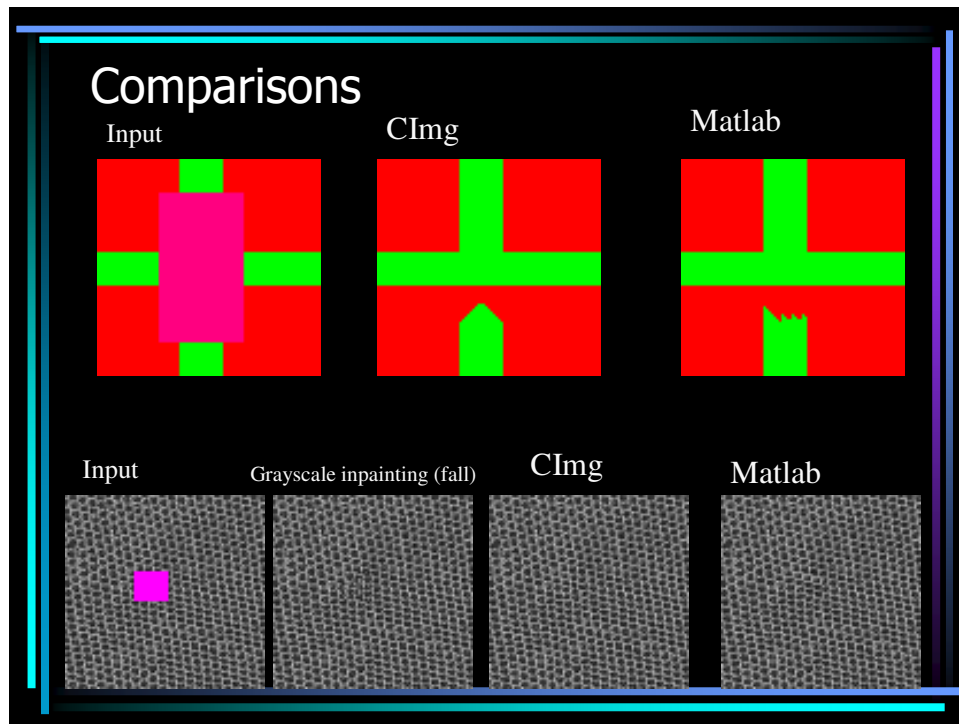
- In grayscale, windows with any number of known common pixels are used for comparisons. In grayscale this does not affect the image considerably.
- In color, however, using windows with few common pixels can lead to huge errors, because the amount of information available is tripled.
- Expanding NL-means to color was also attempted, but the averaging of pixel values yielded very homogenous values which was not a reasonable approximation of the inpainted region.

## Examples



## Differences between Matlab and CImg

- Nested For loops of pixel lookups in Cimg seem to be considerably slower than the code that's optimized in Matlab
- Overall, the same image run through matlab is 15-20x faster than CImg.
- Also, Matlab code inpaints pixels with most available neighbors first, which is theoretically a better order of inpainting than raster.



## Conclusions

- Matlab is much faster for our purposes, and should be used for future work.
- Order in which pixels are inpainted makes a difference: reason why even a simple image like a cross isn't inpainted
- Future work: Criminisi-Perez Algorithm
  - Inpaints lines in the image(isophotes) first