Project Introduction to Digital Imaging

Copy-move forgery detection based on Patchmatch

Haozhe SUN

Motivation

- acquire first experience with image processing
- acquire experience with manipulating pixels instead of studying Deep Learning model
- acquire experience with patch-based method

Image forgery detection pipeline

- Featuring: suitable features are associated with all pixels or with a limited set of key points
- Matching: for each pixel of interest, the best matching is located based on the associated features
- Post-processing: the displacement field is filtered and processed to detect actual copy-moved regions.

This paper mainly talks about the Matching step.

Image forgery detection pipeline

- Featuring: suitable features are associated with all pixels or with a limited set of key points
- Matching: for each pixel of interest, the best matching is located based on the associated features
- Post-processing: the displacement field is filtered and processed to detect actual copy-moved regions.

In this context:

- Featuring: RGB values or Zernike moments
- Matching: modified-PatchMatch algorithm
- Post-processing: filtering + Same Affine Transformation Selection (SATS)

PatchMatch algorithm

- Random initialization of Nearest Neighbor Field (NNF)
- For a certain number of iterations:
  - Propagation: the image is raster scanned top-down or left-to-right according to the parity of iteration.

$$nnf(z) = \arg \min_{\delta \in \{nnf(z), nnf(z^u), nnf(z^l)\}} Distance(P(z), P(z + \delta))$$

  - Random search: the image is raster scanned in the same way as before.

$$nnf(z) = \arg \min_{\delta \in \{f_0, f_1, f_2, \ldots\ldots\}} Distance(P(z), P(z + \delta))$$

$$f_i = nnf(z) + 2^{i-1} R_i$$

Hypothesis: true NNF is mostly regular.

Modified-PatchMatch algorithm

$$\{nnf(z),\quad nnf(z^u) + \Delta\, nnf(z^u),\quad nnf(z^l) + \Delta\, nnf(z^l)\}$$

where

$$\Delta\, nnf(z^u) = nnf(z^u) - nnf(z^{uu})$$

z^{uu} represents the pixel above z^u

- Random initialization of Nearest Neighbor Field (NNF)
- For a certain number of iterations:
  - Propagation: the image is raster scanned top-down or left-to-right according to the parity of iteration.

$$nnf(z) = \arg \min_{\delta \in \{nnf(z), nnf(z^u), nnf(z^l)\}} Distance(P(z), P(z+\delta))$$

  - Random search: the image is raster scanned in the same way as before.

$$nnf(z) = \arg \min_{\delta \in \{f_0, f_1, f_2, \ldots\}} Distance(P(z), P(z+\delta))$$

$$f_i = nnf(z) + 2^{i-1} R_i$$

Hypothesis: true NNF is mostly regular.

Discussion about the modification

- Basic PatchMatch: piece-wise constant NNF, zero order predictor
- Modified-PatchMatch: piece-wise linear NNF (rotations), first order predictor

$$\{nnf(z), \quad nnf(z^u), \quad nnf(z^l)\}$$

$$\{nnf(z), \quad nnf(z^u) + \Delta\, nnf(z^u), \quad nnf(z^l) + \Delta\, nnf(z^l)\}$$

where

$$\Delta\, nnf(z^u) = nnf(z^u) - nnf(z^{uu})$$

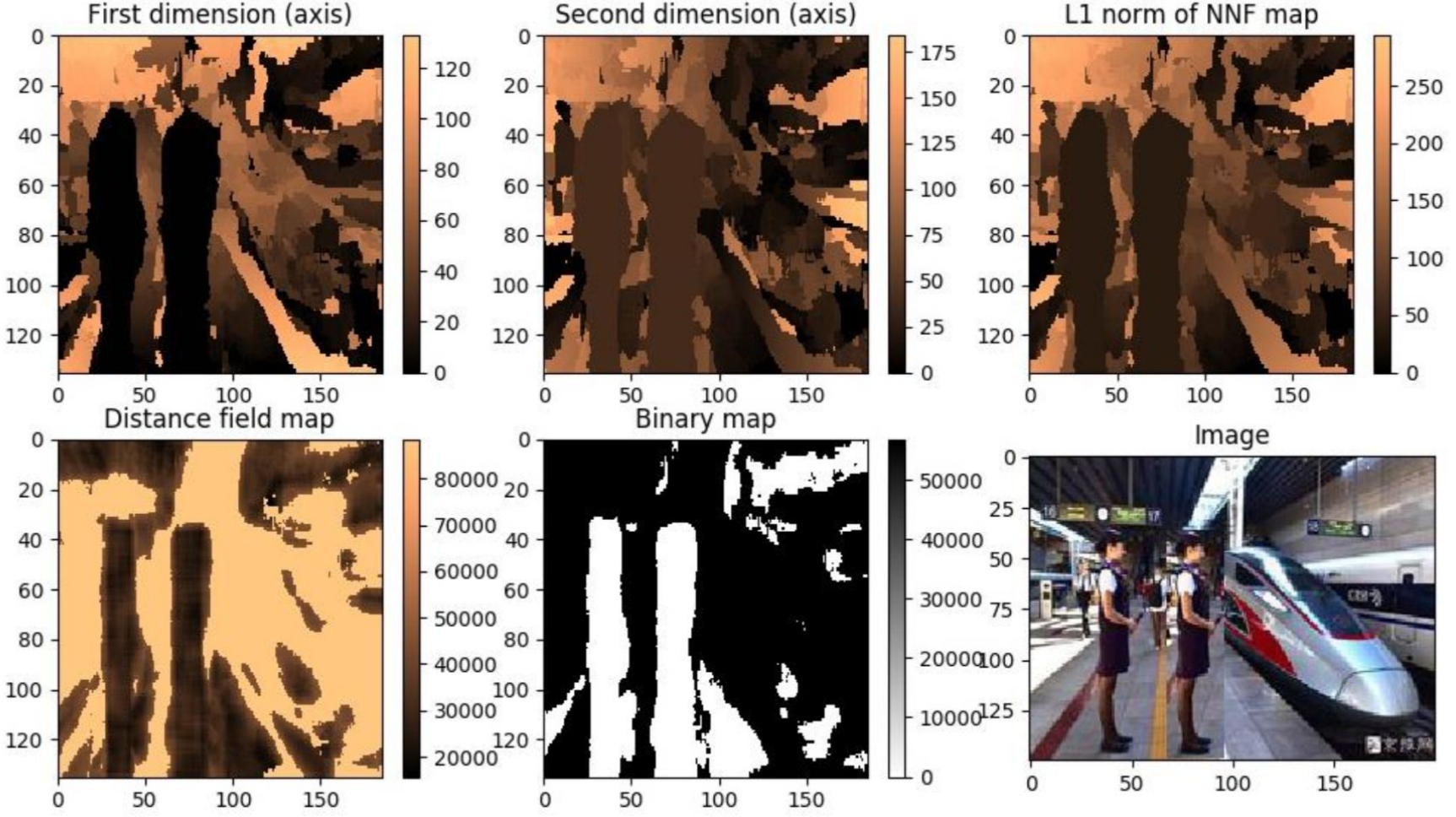
z^{uu} represents the pixel above z^u

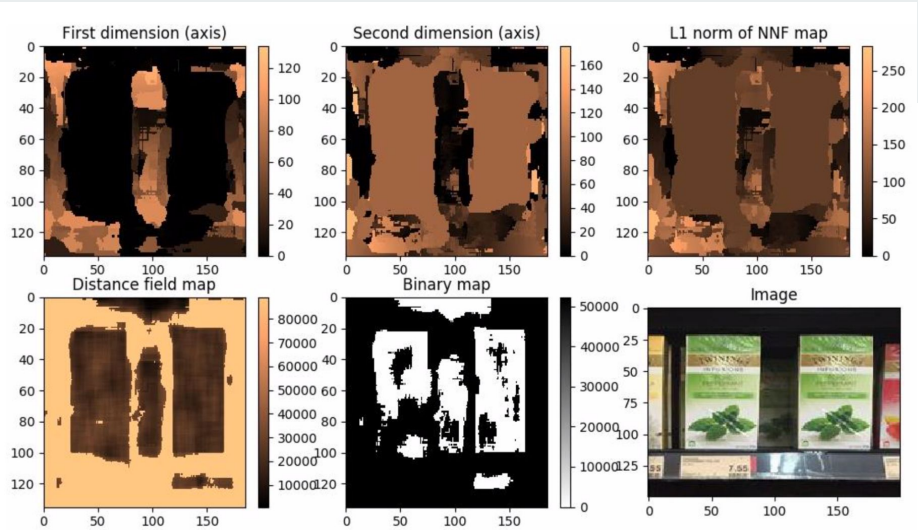
Featuring and post-processing

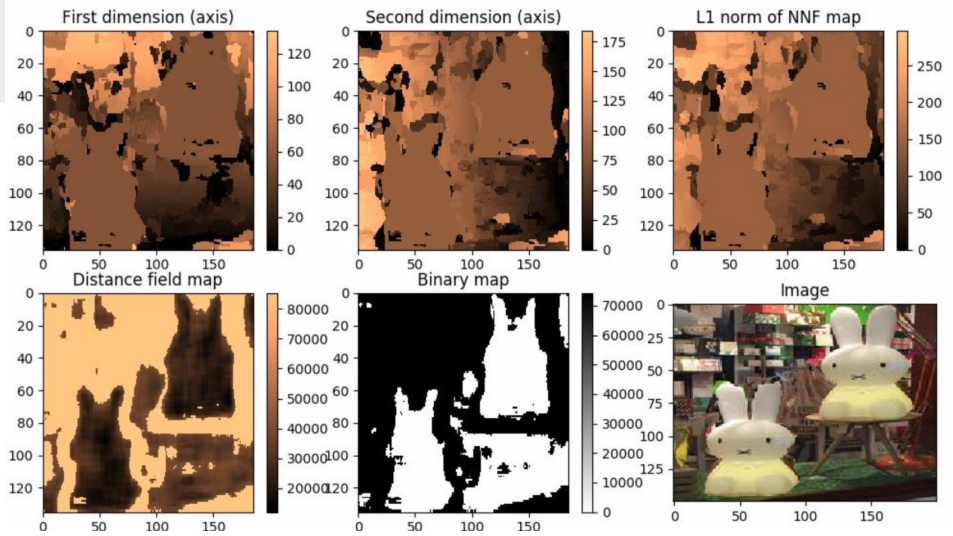
- In the paper:
  - Zernike moments (rotation invariance)
  - filtering + Same Affine Transformation Selection (SATS)
- In my implementation:
  - RGB values
  - filtering + binary map thresholding

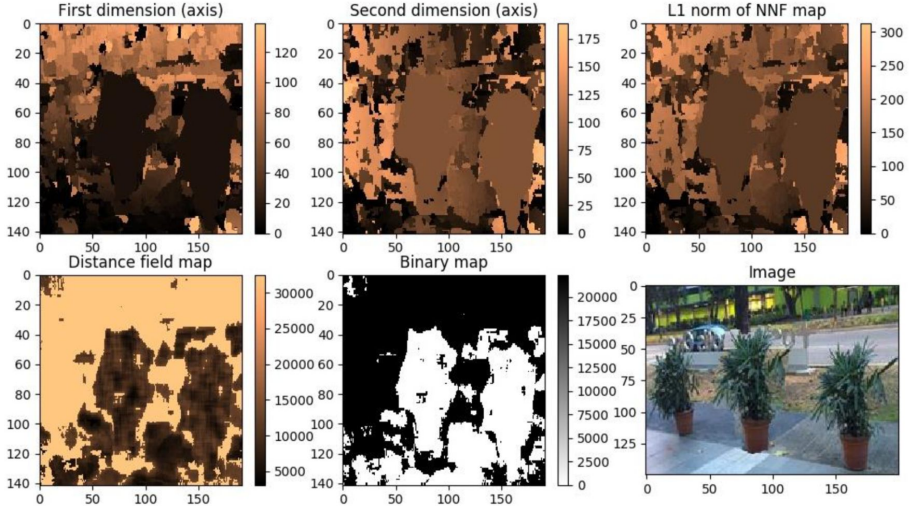
NNF offsets less than a certain threshold are filtered out.

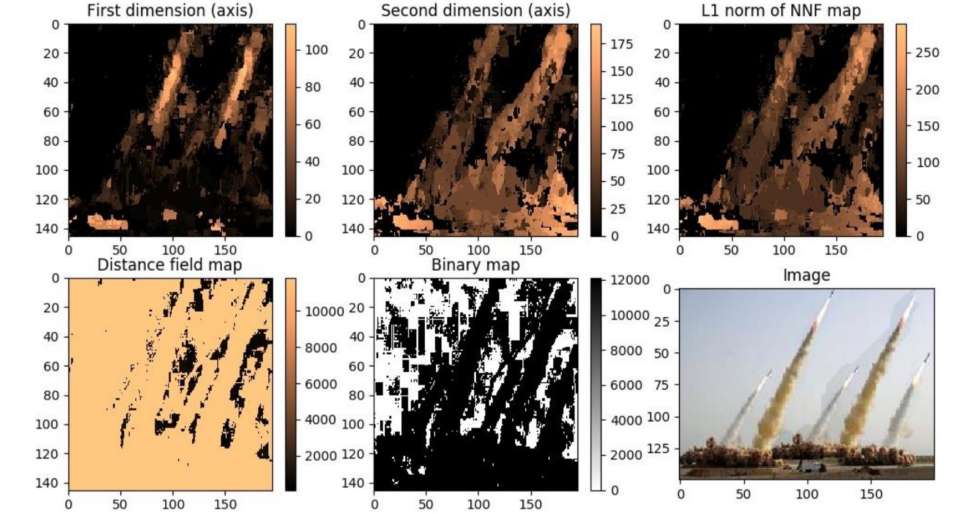
A binary map is generated by the normalized sum of L1 norm of NNF and the associated distance field map. (based on quantiles)

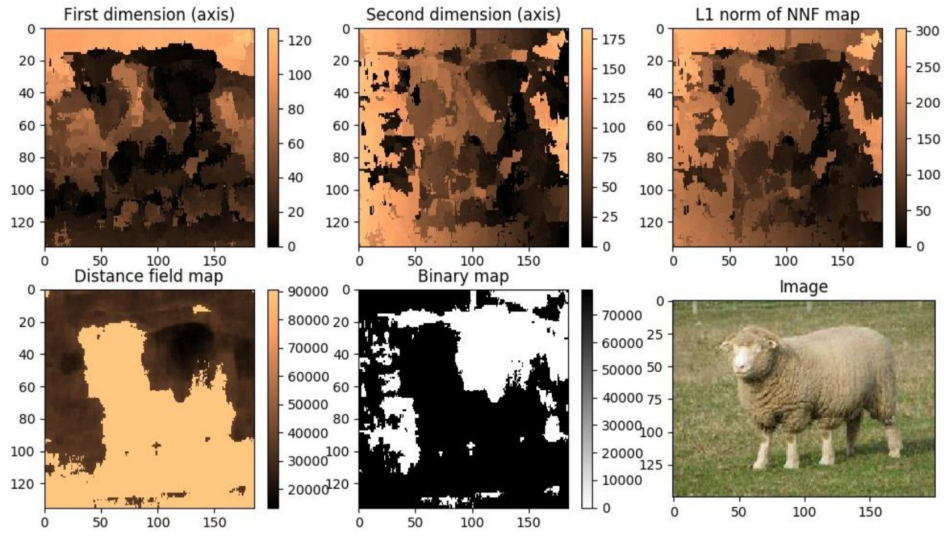

Nearest Neighbor Field map (absolute value)
First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image

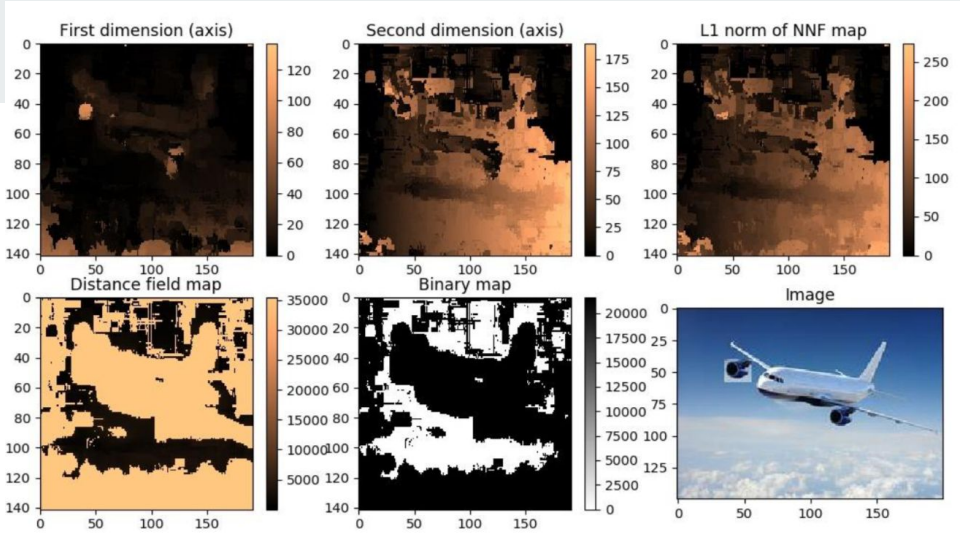

First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image

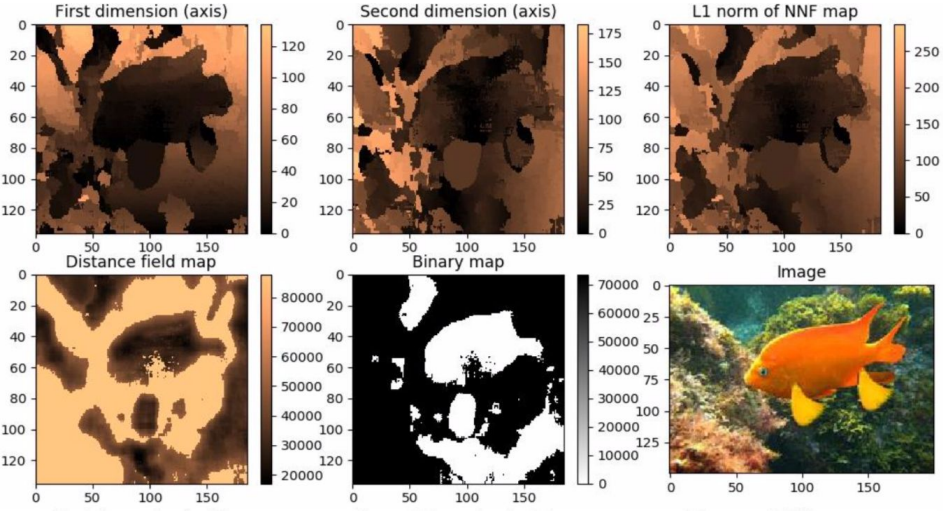

First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image

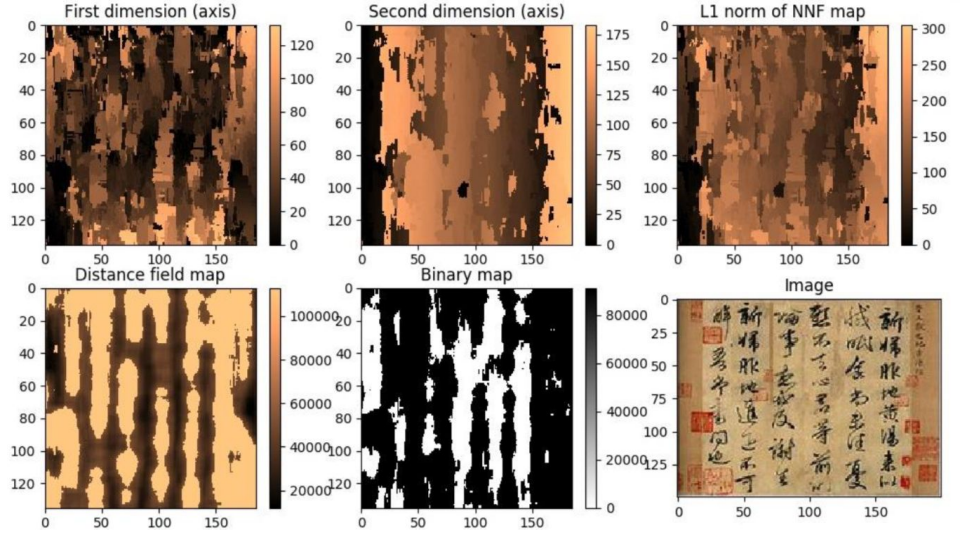

First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image


First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image


First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image


First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image


First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image


First dimension (axis)
Second dimension (axis)
L1 norm of NNF map
Distance field map
Binary map
Image

Thanks for listening