Digital Image Analysis project: Image Inpainting [in MATLAB]

(1) What is image inpainting?

- Modifying an image in a non-detectable form.
- Can be used to remove a particular object in an image (privacy protection).
- Biological Inpainting.
- Human blind spot.
- Difference between image denoising and image inpainting.

(2) Types of Image Inpainting:

- Texture synthesis based image inpainting: holes are filled by sampling and copying neighboring pixels (used for small area of inpainting).
- Exemplar and search based image inpainting.
- PDE based inpainting.
- Fast semi-automatic inpainting.
- Hybrid inpainting.
- PatchMatch Algorithm: https://www.youtube.com/watch?v=n3aoc36V8LM
- Image Restoration
  Url: http://www.ijarcsms.com/docs/paper/volume2/issue1/V2I1-0078.pdf
  Survey:

https://pdfs.semanticscholar.org/15cf/0642038acb2c8230ee5d6e8d8cf936cfce47.pdf

Relevant URLs to various research papers.

Image Inpainting

Image Inpainting
http://users.rsise.anu.edu.au/~junzhou/papers/C_ICPR_2010.pdf
Image Inpainting-2
http://www.math.zju.edu.cn/amjcu/B/201101/70-76.pdf (Paper not found!)
Image Inpainting-3
http://www.zju.edu.cn/jzus/downloadpdf.php?doi=10.1631/jzus.C0910182 (Paper not found!)
Image Inpainting-4
http://192.167.206.42/papers/carleatom11a/InpaintSanPetersburg_new.pdf (Paper not found!)
Image Inpainting-5
http://www.dgip.utfsm.cl/publico/Pub_aca/mat42.pdf (Paper not found!)

Image Inpainting-6
http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA540639&Location=U2&doc=GetTRDoc.pdf
Image Inpainting-7
http://gr.xjtu.edu.cn:8080/upload/PUB.3251.2/inpainting_TIP.pdf
Deformation Inpainting
http://www.mps.com.mx/visual/publications/sources/src-2010/Lamecker_cibt2010.pdf
Fast Image Inpainting Method
http://www.icpr2010.org/pdfs/icpr2010_ThBCT9.9.pdf
Blind Image Inpainting
http://www.math.nus.edu.sg/~matzuows/BlindInpainting.pdf
Inpainting Using Sparsity
http://arxiv.org/pdf/1011.5458
Stereoscopic image inpainting
http://hal.inria.fr/docs/00/62/25/91/PDF/inp.pdf
Study of image inpainting
http://www.iaeng.org/publication/IMECS2010/IMECS2010_pp1442-1445.pdf
Parameter-assistant inpainting
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.172.7719&rep=rep1&type=pdf
Mcalab
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.154.4285&rep=rep1&type=pdf

Reviews:

Review of different inpainting algorithms
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.303.5459&rep=rep1&type=pdf
Review on inpainting techniques
http://ijarcsse.com/Before_August_2017/docs/papers/Volume_5/2_February2015/V5I2-0204.pdf
Comparative research on image inpainting techniques
http://ieeexplore.ieee.org/document/6321859/

Digital inpainting Review

https://irjet.net/archives/V3/i1/IRJET-V3I1150.pdf

Links to github repositories.

1. High-Resolution Image Inpainting using Multi-Scale Neural Patch Synthesis

https://github.com/leehomyc/Faster-High-Res-Neural-Inpainting (complex!, and no idea about "Lua" language)

2. Semantic Image Inpainting using Deep Generative Models

https://github.com/moodoki/semantic_image_inpainting
(I think our main focus shall be on image processing, rather than machine learning)

3. Generative Image Inpainting with contextual Attention

https://github.com/JiahuiYu/generative_inpainting
(I don't know tensorflow!)

4. Image inpainting

https://github.com/fivemok/image-inpainting

5. Color-Image-Inpainting

https://github.com/chongyangtao/Color-Image-Inpainting
(*)(Working, but taking too much time to compile, and is quite complex.)

6. Image Inpainting using Coherency sensitive hashing.

https://github.com/PetterS/patch-inpainting
(*Complex!)
(*Needs a lot of changes)

7. Image Completion with Deep Learning in TensorFlow

https://github.com/bamos/dcgan-completion.tensorflow
(ML Intensive!)

8. Fast Digital Image inpainting

https://github.com/Mugichoko445/Fast-Digital-Image-Inpainting (*)

9.http://users.cecs.anu.edu.au/~junzhou/papers/C_ICPR_2010.pdf

ReferenceCode:

https://www.pantechsolutions.net/image-processing-projects/matlab-code-for-image-inpainting

---

https://in.mathworks.com/matlabcentral/fileexchange/55326-matlab-codes-for-the-image-inpainting-problem

Github repo on fast digital image inpainting:
https://github.com/Mugichoko445/Fast-Digital-Image-Inpainting
Command: g++ -std=c++0x Main.cpp -o output `pkg-config --cflags --libs opencv`

Workable github repos:

Matlab: #5
OpenCv and C++ (recommended): #8

Fast Digital Inpainting Paper: http://www.inf.ufrgs.br/~oliveira/pubs_files/inpainting.pdf

https://github.com/PetterS/patch-inpainting - Patch Inpainting (Matlab)

Research paper
http://users.cecs.anu.edu.au/~junzhou/papers/C_ICPR_2010.pdf

Fast image inpainting paper summary: http://www.inf.ufrgs.br/~oliveira/pubs_files/inpainting.pdf

- Used for reconstruction of small missing and damaged portions of images.
- Two to three orders of magnitude faster than general methods while producing comparable results.
- Useful in the restoration of photographs, vandalized images, and text removal.
- Regions to be inpainted must be locally small. The human visual system can tolerate some amount of blurring in areas not associated with high contrast edges.
- Ω: small area to be inpainted, ∂Ω: its boundary.
- Since Ω is assumed to be small, inpainting can be approximated by an isotropic difusion process that propagates information from ∂Ω into Ω.
- Simplest version of the algorithm:
    - Initialise Ω by clearing its color information, and repeatedly convolve the region to be inpainted with a diffusion kernel. The user can specify the number of iterations, and as the diffusion process is iterated, the inpainting progresses from ∂Ω into Ω.

- Image is convolved with a gaussian kernel. (considers contributions only from neighbor pixels). Mask is created by user intervention (hit and trial method).
- Diffusion barriers: boundaries for the diffusion process inside Ω. As the diffusion process reaches the barrier, the reached pixel has its color set, but the process stops.
- The simple diffusion-based inpainting algorithm produces blurred spots at the intersection between Ω and high contrast edges. By appropriately adding diffusion barriers, the user stops the diffusion process from mixing information from both sides of the mask.
- All general inpainting algorithms require a step of manual masking, and the time required to create a mask depends only on the available features and is independent of the inpainting algorithm used.

Conclusion:

- The results produced by this simple model are comparable to non-linear inpainting models, but 2-3 orders of magnitude faster, thus making it practical for interactive applications.
- Being able to create and refine Ω interactively can greatly improve the quality of the reconstruction.
- This algorithm is intended for filling in locally small areas only.

Possible uses:

(1) Special effects
(2) Scratch removal
(3) *Compression of images*

Summary of Image Inpainting Based on Local Optimisation - http://users.cecs.anu.edu.au/~junzhou/papers/C_ICPR_2010.pdf

Removing objects from an image or repairing damaged pictures by replacing the missing regions using the information in the rest of the scene.
Method: Builds on an exemplar-based perspective so as to improve the local consistency of the inpainted region which is done by maximising the local consistency with respect to abutting candidate patches.
The similarity computation generates weights based upon an edge prior and the structural differences between inpainting exemplar candidates.

- Select the pixel on the boundary of the missing area with the highest filling priority using exemplar method
- The details and sharpness of the image are preserved.
- The local consistency of the inpainted region is preserved by exploring the relationship between a set of candidate inpainting patches and the neighbourhood of the missing region.
- The optimal patches are recovered by maximising the local consistency with respect to the neighborhood candidates and the inpainting region boundary pixels.
- Extract a list of patches from locations in the rest of the image other than the missing area for the selected pixel.

- Extract patches for neighboring boundary pixels
- Compute the local consistency value so as to select the optimal supporting patch for the current boundary pixel as introduced in the above section.
- Inpainting from the pixel with highest priority may cause loss of detail or a suboptimal sequence over the entire inpainting region.
- The optimal patch should be consistent with those patches used to fill-in neighboring pixels in the inpainting region by minimising the difference between the two.
- Go back to 1 until the full region has been inpainted.

Result:

- Inpainted regions are in better qualitative accordance.
- We only extract image patches from areas that are not occluded by the objects to be removed, i.e. we only use full patches, and set the patch size to be 9 × 9 pixels.
- weights are assigned to the image patches in the neighbourhood according to their importance over a set of candidates, and thus, does not suffer from oversmoothing.
- This preserves detail and provides sharper results while maintaining local image consistency.

Conclusions:

- Inpainting method that preserves image detail, textures and sharp edged in the image by extending the exemplar-based method to a local consistency setting.
- We combine the local edge prior with the similarity of the candidate inpainting exemplars to the inpainting region boundary over a local neighborhood.
- The similarity computation generates weights for each candidate patch, which in turn contributes to the inpainting sequence decision through a list of iterates.