UE COMPVISIO Bibliographic restitution

Topic B : PatchMatch

Franck XU, Nampoina RAVELOMANANA, Selman SEZGIN

February 2023

Table of Contents

Image editing tools

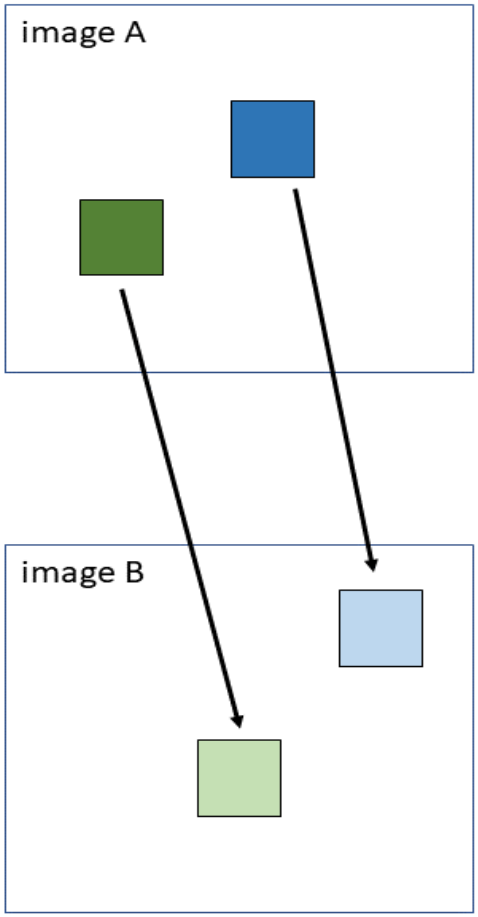
- Goal: Develop algorithms which can be applied to perform image or video editing tools.
- Examples of such tools: image retargeting, image completion, image reshuffling etc.

(a) Original images (b) Masked images (c) Inpainted images

Figure: Example of image completion.

Table of Contents

The Nearest-Neighbor Field (NNF)



- Problem: For every patch in **A** find the most similar patch in **B** called its *nearest-neighbor*.
- Using of a fixed patch distance metric L_p to compute patch similarity.
- All these mappings form the *nearest-neighbor field*.
- **Challenge: How to compute this field ?**

Key observations

Three key observations to approximate the NNF:

1. Low dimensionality: Patches in **B** are searched in a 2D-space called *offsets space*.
2. Natural structure of images: Adjacent pixels are not independent.
3. Law of large numbers: Assigning the patches randomly lead to many good guesses if the images are large.

These observations lead to the PatchMatch algorithm.

The PatchMatch algorithm (1/2)

- Main steps:
  - Random initialization: each patch in **A** is assigned randomly to a patch in **B**.
  - Iterations of *propagation* followed by *random search*.
- If P_j and S_j denote propagation and random search operations at patch $j \in \{1, \ldots, n\}$ then the sequence of operations $P_1, S_1, \ldots, P_n, S_n$ corresponds to one iteration.
- Convergence to the exact NNF in the limit.
- But only few iterations to get a patch matches approximation.

The PatchMatch algorithm (2/2)

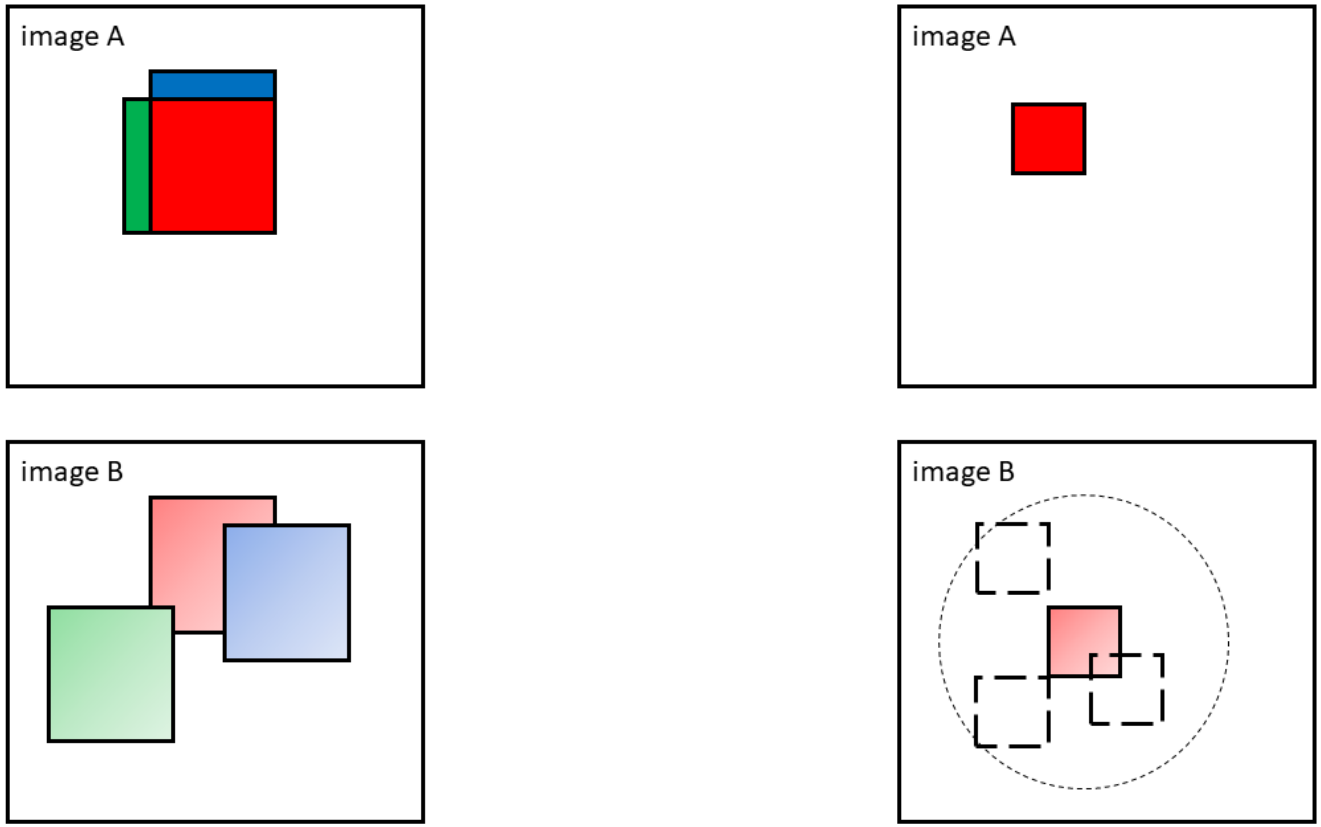
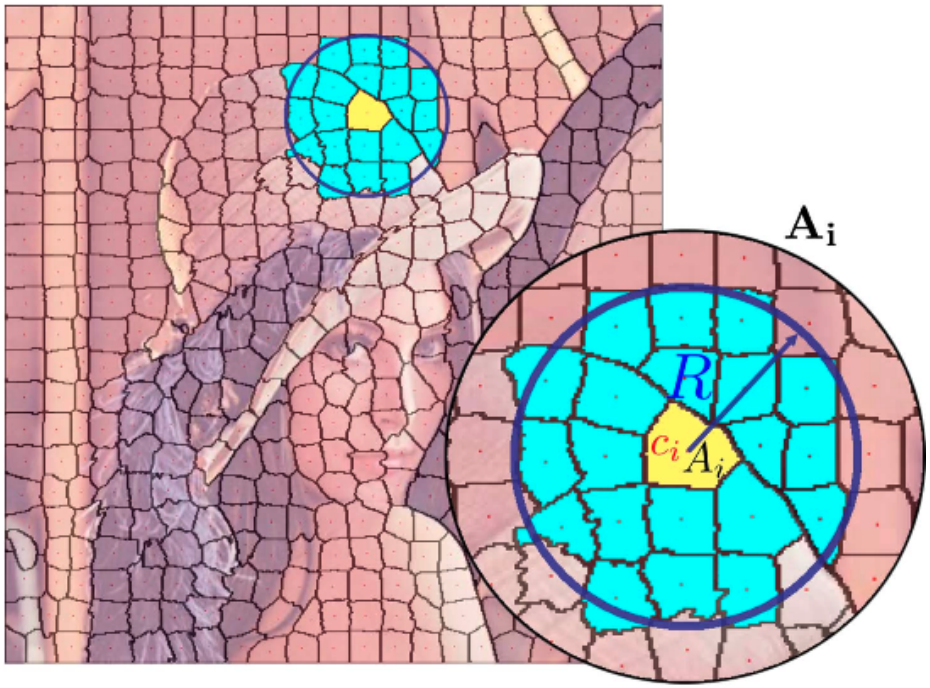


Figure: Illustration of the *propagation* and *random search* operations applied at each patch, at each iteration. Propagation on the left figure, random search on the right one.

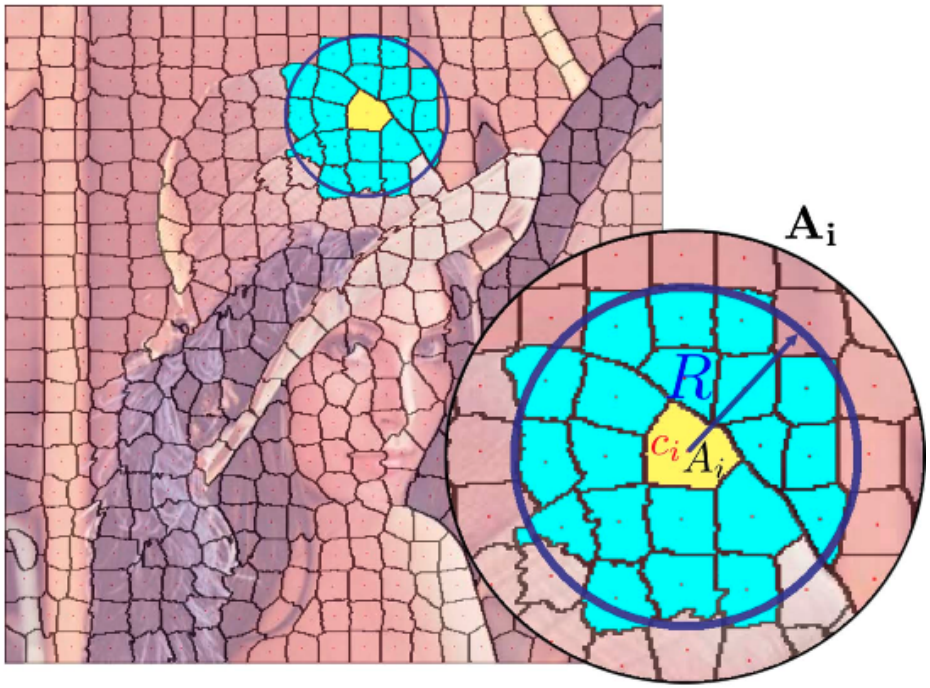
Table of Contents

SuperPixel and SuperPatch



- SuperPixel : subset of homogeneous pixels
- Superpixel decomposition $A = \{a_i\}_{i=1,\dots,N}$ where superpixel a_i is located at "center" c_i
- SuperPatch $\mathbf{A_i}$: set of superpixels within a fixed radius R centered on a_i

SuperPixel and SuperPatch



- SuperPixel : subset of homogeneous pixels
- Superpixel decomposition $A = \{a_i\}_{i=1,\dots,N}$ where superpixel a_i is located at "center" c_i
- SuperPatch $\mathbf{A_i}$: set of superpixels within a fixed radius R centered on a_i

$\Rightarrow$ can we apply PatchMatch on superpixels ?

Superpatch Comparison Framework

How to compare two different superpatches $\mathbf{A_i}$ and $\mathbf{B_j}$?

Some difficulties :

- number of superpixels of each superpatch may be different
- superpatches' shape may also be different
- one-to-many overlapping superpixels

Superpatch Comparison Framework

How to compare two different superpatches $\mathbf{A_i}$ and $\mathbf{B_j}$?

Some difficulties :

- number of superpixels of each superpatch may be different
- superpatches' shape may also be different
- one-to-many overlapping superpixels

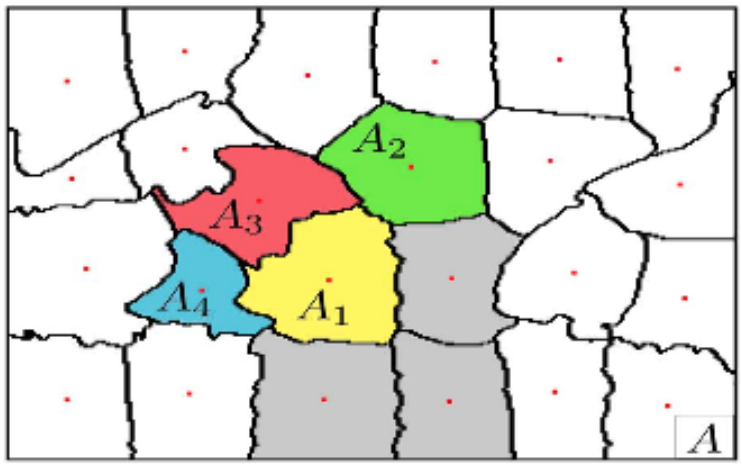
$\Rightarrow$ Need to define a specific distance on superpatch space :

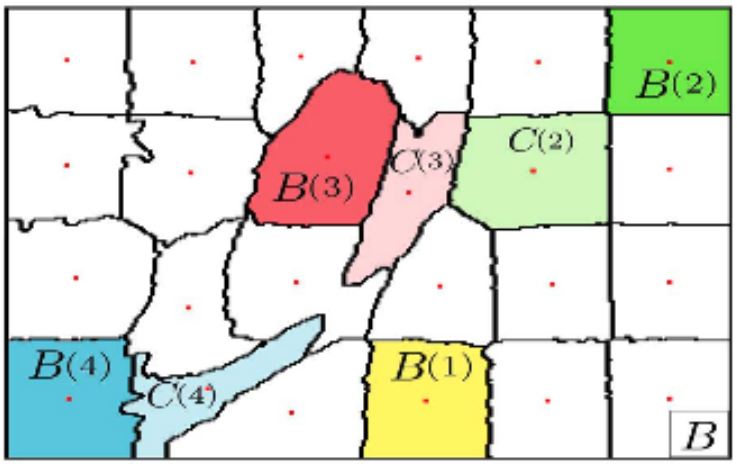
$$D(\mathbf{A_i}, \mathbf{B_j}) = \frac{\sum_{\mathbf{A_i}} \sum_{\mathbf{B_j}} w(a_i, b_j) d(a_i, b_j)}{\sum_{\mathbf{A_i}} \sum_{\mathbf{B_j}} w(a_i, b_j)}$$

SuperPatchMatch Algorithm (SPM)

SPM on two different images A and B :

- Pre-processing : compute SuperPixel decomposition
- Initialization : for each a_i assign randomly a superpixel $b_{j(i)}$ from B
- Iterative process :
  - Propagation : improve initial NNF by considering adjacent superpixel of A



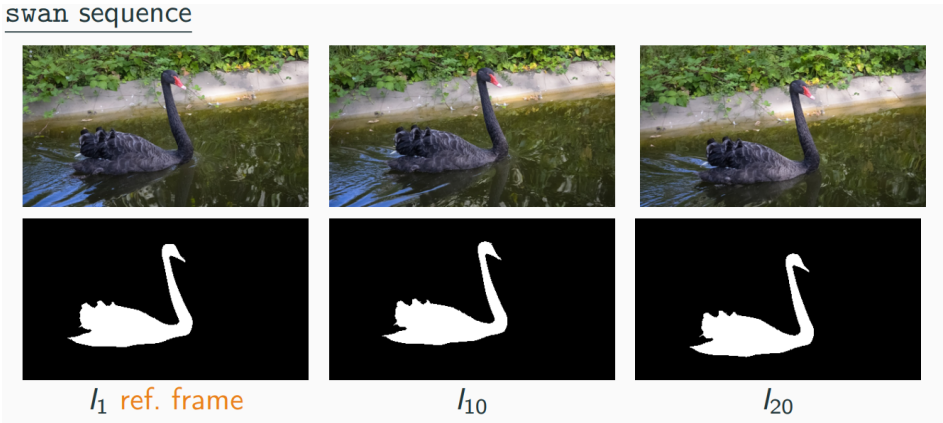


  - Random Search : Random sampling of superpixels in B around the best match

Table of Contents

Goal of the project

- To propagate the location of an object of interest in the whole video sequence
- Training data: video sequences containing, with a reference image and its binary mask (to define the region of interest), and rest of frames and also its corresponding ground truth masks (only used for evaluation purpose)

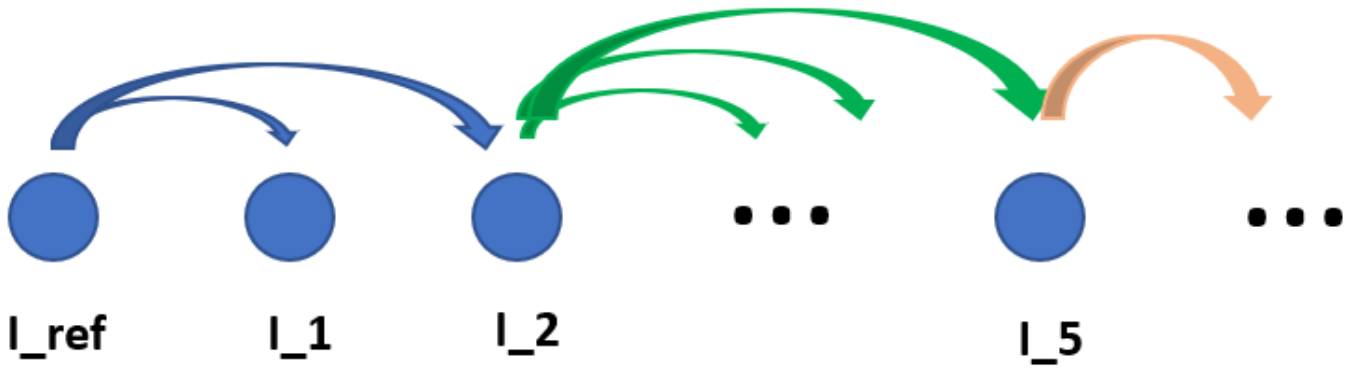


Application of patchmatch algorithm

- Get the set of patches that interest us on the reference image, by using the corresponding binary mask
- Apply patchmatch on the whole image and project patches of interest on next frames
- Two options:
  - Direct integration: patchmatch on reference image I_0 and each image I_n
  - Sequential integration: patchmatch on image I_{n-1} and image I_n
- Advantages
  - Direct integration can handle temporary occlusions
  - Sequential integration provide best results on large deformations

Application of patchmatch algorithm

- Combine both approaches

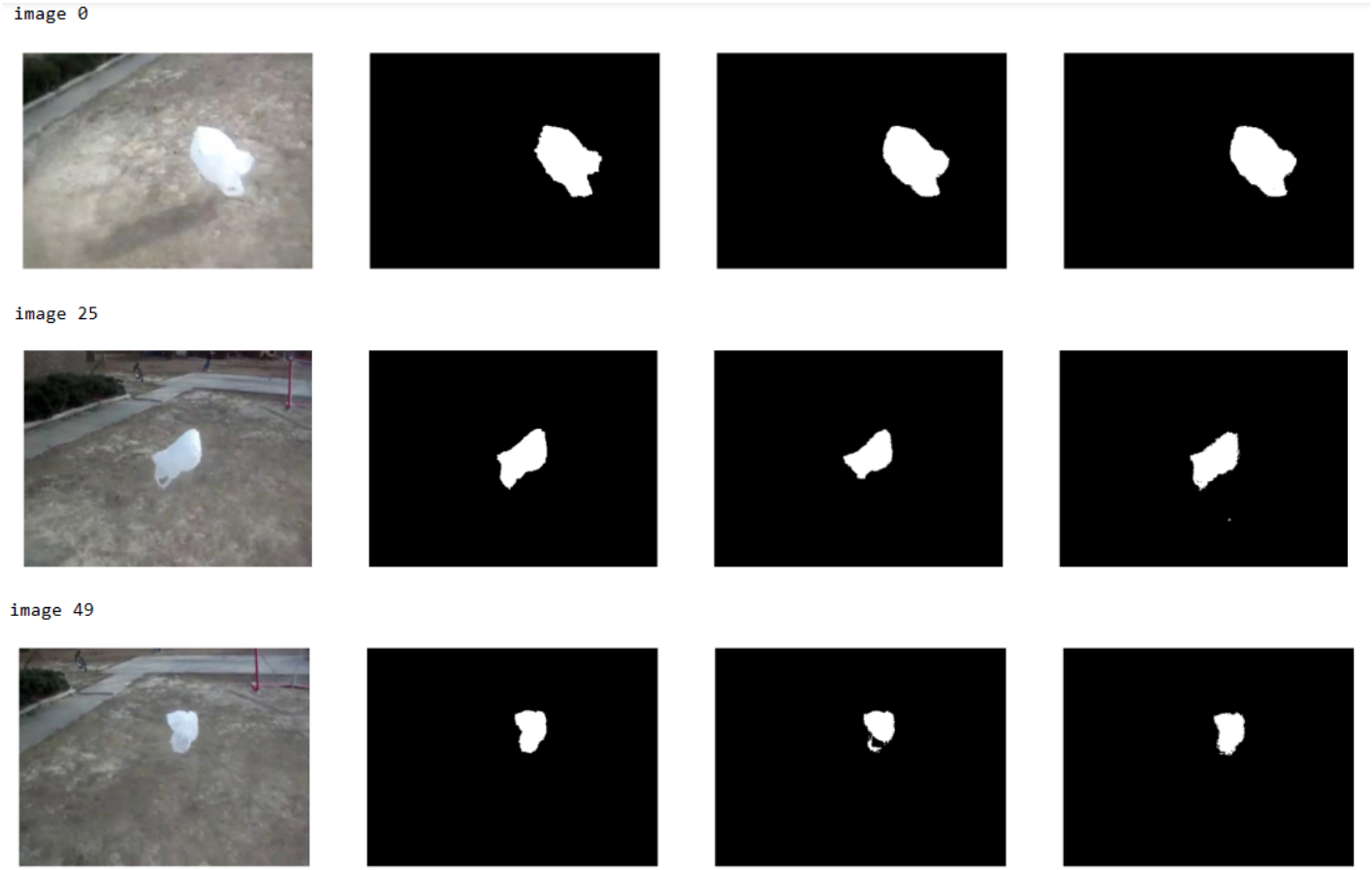


- Best stepsize ?

Table of Contents

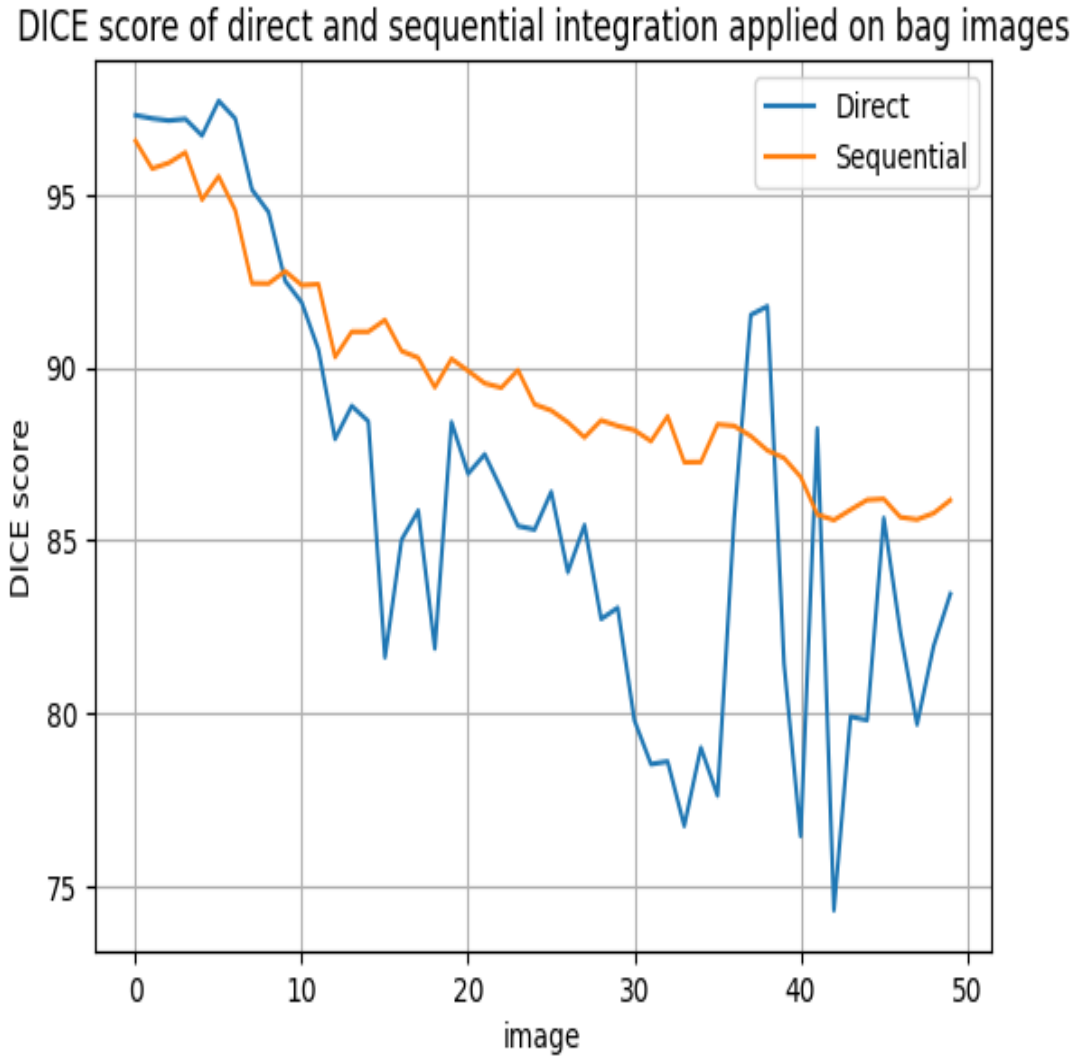
Results of patchmatch

Some visualizations



Image, groundtruth mask, direct mask, sequential mask

Results of patchmatch



For bag video

Results of patchmatch

- Sequential's dice score is more stable, because, in these bag sequences, the bag undergoes many deformations. Sequential integration is better at handling these deformations, because it performs an image-to-image tracking.

Table of Contents

Further thoughts

- First idea: Start with PatchMatch before going to SuperPatchMatch because of implementation simplicity.
- Performance comparison between PatchMatch and SuperPatchMatch.
- Problem if the size of the object becomes larger at the next frame.
- Should PatchMatch (or SuperPatchMatch) be applied to each patch of the image or only to the patches that belong to the reference mask ?