Texture Analysis and Synthesis for Video Coding

Maoz Muzikansky
Nimrod Natan

Supervised by Yair Moshe
What Is A Texture?

- **Texture**-the definition (one of many): Some visual pattern on an infinite 2-D plane which, at some scale, has a stationary distribution

- Textures regions usually contain strong high spatial frequencies
  - Takes many bits to code
  - Human Visual System is less sensitive to high spatial frequencies

- For example:
Texture Synthesis (our task): The general idea

- We use the small patch to rebuild the texture in the output image.
Texture Synthesis – Quilting
(Efros and Freeman, 2001)

The General Idea: square blocks from the input texture are patched together to synthesize a new texture.

First Step:
Go through the image to be synthesized in raster scan order in steps of one block (minus the overlap).

Second Step:
For every location, search in the input texture a set of blocks that satisfy the overlap constraints (above and left) within some error tolerance. Randomly pick one such block.

Third Step:
Find the minimum cost path along this surface using Dynamic programming and make that the boundary of the new block. Paste the block onto the texture. Repeat.
Minimal Error Boundary
Using Dynamic Programming

overlapping blocks

vertical boundary

overlap error

min. error boundary
Some Examples:
The General Idea:

- Patch regions from a sample image or video are copied to the output and then stitched together along optimal seams to generate a new output.

- Images are represented as graphs and seams between patches are made using Min-Cut algorithm.

- In contrast to other techniques, the size of the patch is not chosen a-priori, but instead a graph cut technique is used to determine the optimal patch region for any given offset between the input and output texture.
In the article were suggested 3 algorithms for patch selection:

1. **Random placement.**
2. **Entire patch matching.**
3. **Sub patch matching** - isn’t in the project scope because of time constraints and complex implantation in Matlab.
**Random Placement:** Place the input randomly on the output.
**Entire Patch Matching:**

Searching for translations of the input image that match well with the currently synthesized output. The new patch location is picked stochastically - Better location has better chance to be chosen.

Costfunction for placement (SSD):

\[ C(t) = \frac{1}{|A_t|} \sum_{p \in A_t} |I(p) - O(p + t)|^2 \]

- \( C(t) \) is the cost at translation \( t \) of the input.
- \( I \) and \( O \) are the input and output images respectively.
- \( A_t \) is the overlap region.
- \( t \) is for translation.

The probability function goes like:
Examples:
Conclusions

- Quilting generally gives good results for all kinds of textures in short run-time. Not robust as Graphcut.

- Random placement is the fastest synthesis method, and gives very good results for random textures but not for repeated ones.

- Entire patch matching gives the best results for all kinds of textures, but in very long run-time (can be improved as suggested in the article)