CIS 6930/4930 Deep Learning for Computer Graphics

Fall 2020 MWF Period 5 (11:45am-12:35 pm)

Final Project Proposals

Dr. Corey Toler-Franklin

Asst. Professor, Computer Science

Director: Graphics, Imaging & Light Measurement Lab

Computer & Information Science & Engineering, UF

Image Sources:
(background) Thalles Silva
(foregound) Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings, Bako et al.

Final Projects

- Demonstrate a practical understanding of topics covered
- Produce a creative application that applies DL to CG
- Experiment with implementing classic (current or new) algorithms

Final Projects

- Individual or in groups
- Groups of 3 recommended
- Use canvas to connect with team members

Final Projects

- Projects can be just about anything
- DL applications for CG fall into 2 general categories: analysis and synthesis as discussed in class
- Any platform, any development tools (though you may choose to use the frameworks you developed in Part I and Part II)
- Scope too small: add a couple of extra features to a Part II
- Scope too big: something with multiple major challenges in multiple areas

Proposal: Submit 1-2 Pages

- State overall Goal
- Explain expected outcome
- Summarize what you propose to do
- Provide an implementation plan
- List team members
- Explain how work will be distributed among team members
- Identify DL course topics (and graphics application)
- List 3rdparty API's or framework you will use and how
- List resources (research papers, websites, textbook chapters)
- List datasets you will use
- Include a timeline/schedule

What to avoid

- Re-inventing the wheel (not leveraging helpful APIs)
- Overly ambitious projects
- Super easy projects
- Note: This is a programing project. No user-based applications that do not require programing.

Final Project Proposal

I will evaluate each team proposal and provide constructive feedback for how to scope your work to be successful.

Final Project Proposal

- As always, site any external contributions

Course Project Part III: Evaluation Criterion

Mid-Point Eval, progress, code

Presentation - Background/Previous Work

Presentation – Oral

Presentation - Slide Content

Project - Technical Difficulty

Project - Approach/Creativity

Project - Write-up

Project - Results

Project - Source Code

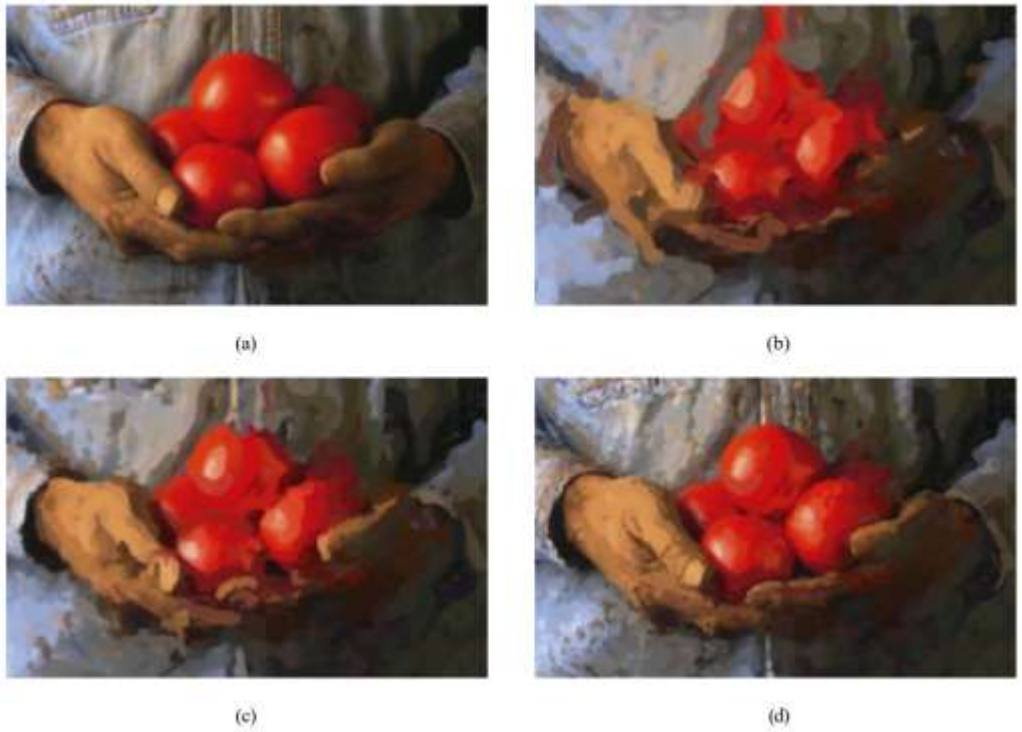
Final Project Proposals: Examples

- Re-implementations papers from course reading list
- Implementations of SIGGRAPH papers that incorporate DL
- You can modify, simplify or implement variants of the paper

- Add a deep learning component to a graphics paper.
- Careful: Papers must be non-trivial (especially those with online solutions!! Beware!!!!)

Painterly Rendering with Curved Brush Strokes of Multiple Sizes

Aaron Hertzmann
Media Research Laboratory
Department of Computer Science
New York University



Tiziano Portenier, Qiyang Hu, Attila Szabó, Siavash Arjomand Bigdeli, Paolo Favaro, and Matthias Zwicker. 2018 Faceshop: deep sketch-based face image editing. ACM Trans. Graph. 37, 4, Article 99 (August 2018), 13 pages. DOI:https://doi.org/10.1145/3197517.3201393

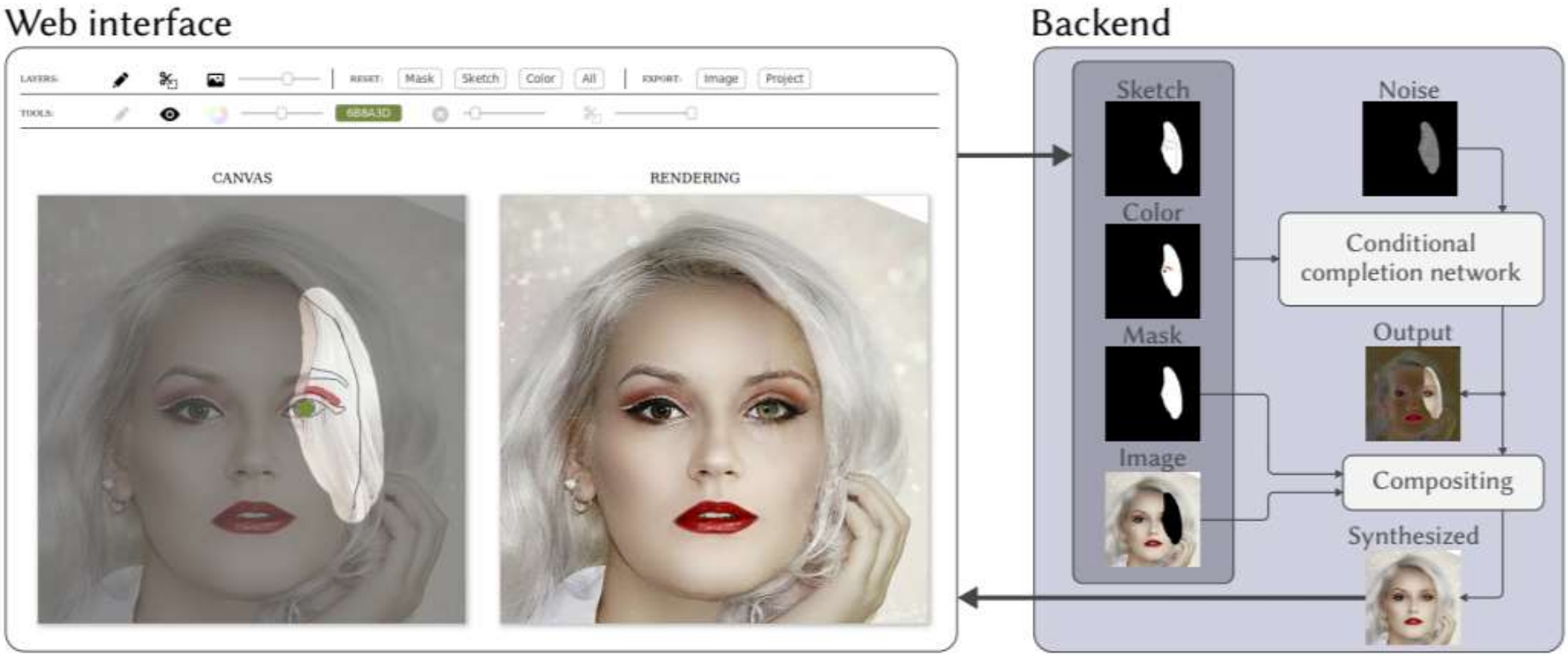


Fig. 2. System overview. The web-based interface allows users to specify masking regions, an edge sketch, and color strokes. The core of the backend is a conditional image completion network that takes the user input, the original image, and a noise layer, and reconstructs a full image. The reconstructed image is composited with the original by filling the masked region in the original with the reconstructed output, and returned. Photo from the public domain.

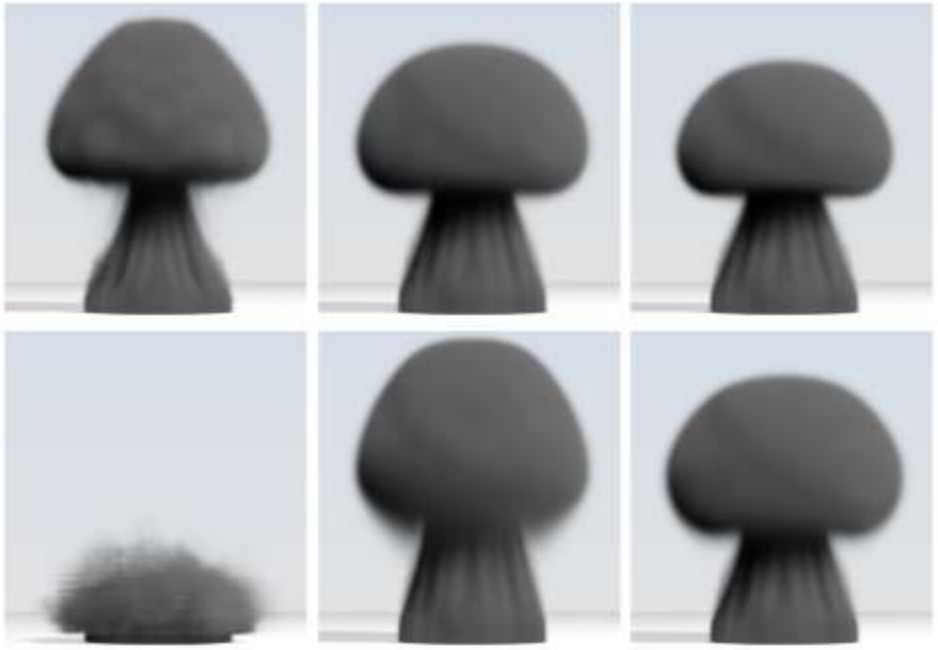
GAN Video http://www.cs.columbia.edu/~vondrick/tinyvideo/ Generating videos with scene dynamics Carl Vondrick, Hamed Pirsiavash, Antonio Torralba Publication:NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing SystemsDecember 2016 Pages 613–621

Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings
Steve Bako*, Thijs Vogels*, Brian McWilliams, Mark Meyer, Jan Novák,
Alex Harvill, Pradeep Sen, Tony DeRose, and Fabrice Rousselle
ACM Transactions on Graphics (Proceedings of SIGGRAPH 2017), vol. 36, no. 4

Accelerating Eulerian Fluid Simulation With Convolutional Networks

Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, Ken Perlin

https://arxiv.org/abs/1607.03597



- Neural network agents for playing other games not discussed in the lecture
- Computer generated art using CNNs and/or RNNs. Can you learn parameters for an "art generator" network using training, similar to the Gatys et al. paper discussed on Nov 1? Another alternative is to try to train an art generator to directly predict from a low-resolution input artistic image a higher output artistic image. Then this could be applied to repeatedly "upsample" and "add art" to arbitrary low-resolution images. Or you might investigate genetic algorithm techniques, e.g. given a population of programs with randomly sampled parameters, have a user select one that produces the most appealing artwork for its output image, and then sample more programs nearby the user's selection in weight space (with decreasing radius of the random sampling throughout the training process). See related paper on design galleries.

- Neural network music creation and/or visualization of music using CNN generators.
- Image classifiers for a topic of your interest: for example, classify species of birds, or dishes of food, or any topic of your interest. An easy way to create a custom classifier is to fine-tune starting with an ImageNet trained classifier. See for example, the model zoo for Torch for pretrained models. One challenge in this topic will be creating your custom dataset for supervised learning and using data augmentation with it.
- Image synthesis using GANs. It might be interesting to condition the GAN generation on a particular dataset such as shirts and try to create controls for e.g. shirt design. Or faces: previous papers have synthesized faces without using neural networks: it might be interesting to adapt this to deep learning approaches such as the Laplacian GAN approach, and it might be interesting to add controls for e.g. recoloring hair, glasses, etc.

- Learning view interpolation for videos. A video with a low framerate (e.g. 30 frames/sec) might not be useful for "slow motion" effects. So one might wish to learn to increase the framerate of the video by say a factor of 2 (increasing to e.g. 60 frames/sec) by inserting interpolated frames. One could train for example a convolutional/deconvolutional architecture similar to project 2 that predicts the missing interpolated frame given the previous and next frames (or the k previous and k next frames) as input. Supervised training datasets could be easily constructed from existing downloaded videos: suppose a video has a framerate of 30/frames/sec, then supervised data can be obtained for the problem of going from e.g. 15 frames/sec to 30/frames per sec.