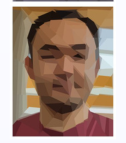
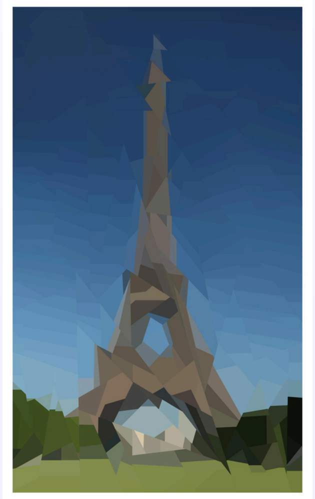
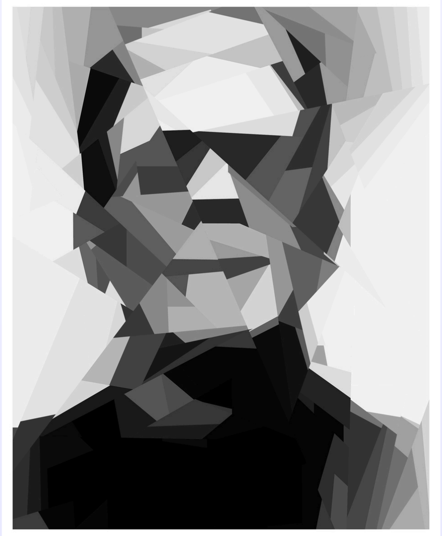
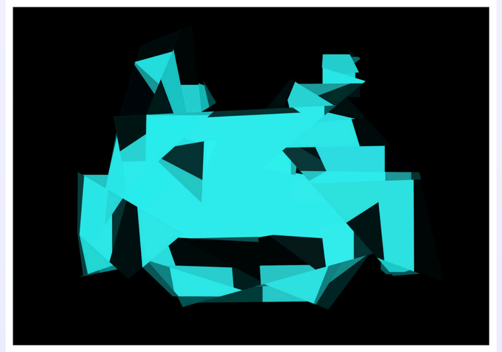
OBLIQUE DECISION TREES AS AN IMAGE MODEL FOR CUBIST IMAGE RESTYLING


Edric Chan — Great Oak High School, Temecula, CA, USA & DVC Diablo Valley College, CA, USA


Magzhan Gabidolla — Dept. Computer Science & Engineering, University of California, Merced, CA, USA


Miguel Á. Carreira-Perpiñán — Dept. Computer Science & Engineering, University of California, Merced, CA, USA




ABSTRACT

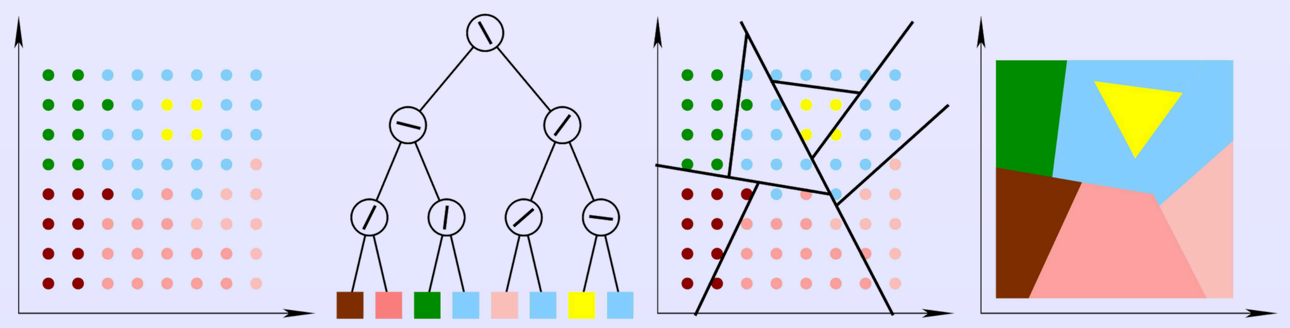
We propose the use of oblique decision trees and forests, originally a supervised machine learning approach, as a model of a single image. This results in a hierarchical partition into constant-color polygons ("pixels") that best approximates the source image. The tree or forest can be trained using a recent algorithm that applies alternating optimization over the nodes of a tree and the trees of a forest. In this paper, we use this for aesthetic effects, such as restyling an image to look like a cubist painting or a stained-glass window. We illustrate this with multiple examples and compare it with the results achieved by Generative AI and Neural Style Transfer.

INTRODUCTION

- We propose to use oblique trees and forests as a model for a single image.
- Traditional decision trees in machine learning are a weak image model:
  - Axis-aligned: limited to horizontal and vertical splits.
  - Suboptimal training using the CART algorithm (no objective function).
  - In random or boosted forests, trees are combined suboptimally as well.
- Oblique trees are a much stronger image model:
  - Splits can have any orientation.
  - We can train them to optimize the squared pixel error using the Tree Alternating Optimization (TAO) algorithm (single tree) or Forest Alternating Optimization (FAO) algorithm (multiple trees).
  - This produces smaller trees or forests with low error and a much better match to the image.
- This image model can be used for segmentation, compression and other tasks.
- Here we focus on image processing for aesthetic effects, which have applicability in software packages such as PhotoShop or for game and film production effects.
- Our approach (possibly in conjunction with other design tools such as Adobe Illustrator) could be used for vector art: company logos on business cards, posters and billboards, as well as for printing on clothing, bags, stickers or even tattoos.

OBLIQUE TREES AS *HIERARCHICAL POXELS* IMAGE MODELS

- We consider a source image as a function from 2D to color space, sampled over a grid (the pixel locations).
- An oblique tree approximates the source image by a hierarchical piecewise-constant function:
  - Each piece is a convex, colored polygon that we call poxel (for polygon pixel).
  - Its fidelity can be controlled by the model size (tree depth Δ and, with forests, number of trees *T*).
- Each oblique tree is a binary tree of depth Δ having trainable parameters in each node:
  - Decision node: linear split.
  - Leaf: constant color.
- Estimating the parameters of the tree or forest from the source image is a least squares regression problem. We solve it with TAO or FAO.
- The output image represented by the tree or forest:
  - Is a vector graphics object rather than a bitmap.
  - Is strikingly reminiscent of cubist paintings such as those of Picasso and other artists.
  - Other effects are also possible, such as the style of stained-glass windows.
- This can be seen as a new stroke-based rendering technique that, given a source image, uses hierarchical pixels as variable-shape "strokes" to minimize the image error subject to a maximum number of strokes.

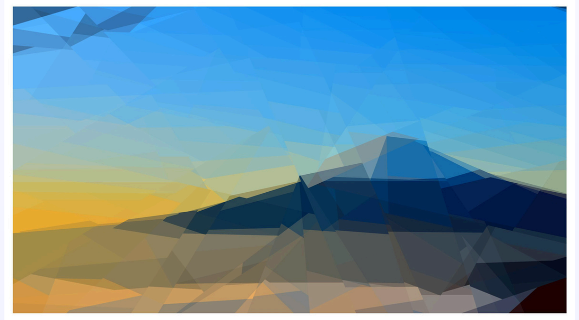


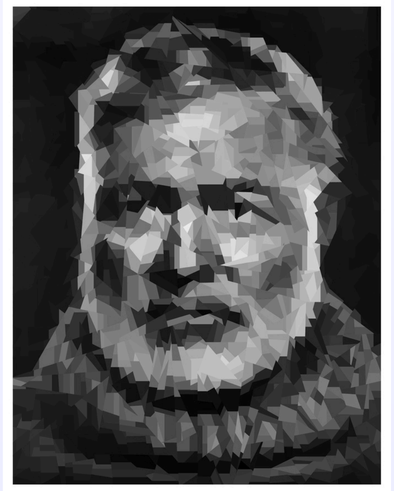
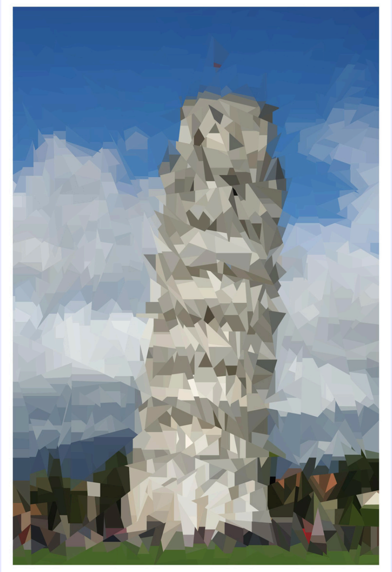
TRAINING TREES (TAO) AND FORESTS (FAO)

- At each iteration, TAO updates the parameters of each node of the tree given all other nodes are fixed:
  - For a decision node, this requires solving a binary linear classification.
  - For a leaf, this requires computing the average color of the pixels in the leaf.
  - Nodes at the same depth can be updated in parallel.
- At each iteration of FAO:
  - We update each tree given all other trees are fixed, using TAO.
  - We update the parameters of all the leaves of all trees at once, solving a linear system.
- For both TAO and FAO, the error decreases after each iteration until convergence (to a local optimum).
- The initial tree is a random median tree: we recursively split 50/50 with a random split.
- The runtime depends on the tree/forest size. Good results can be obtained in around a second in a laptop.

USER PARAMETERS & CONTROL: EFFECTS

- Tree depth Δ and number of trees *T*: size of the model:
  - Few poxels: cartoon-like.
  - The bigger the model, the more photorealistic the result.
  - Using multiple trees gives an appearance of "broken glass".
- Seed of the pseudorandom number generator for the initial tree: adds some randomness to the result (as in rotoscopic animation with video).
- Number of iterations.
- Line width of the poxel boundaries: can produce an effect reminiscent of stained-glass windows.
- Other effects: apply only to parts of the image via a mask, etc.

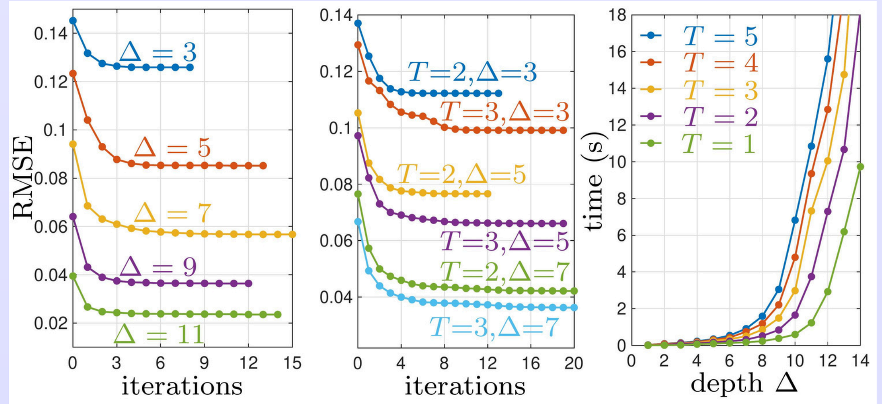


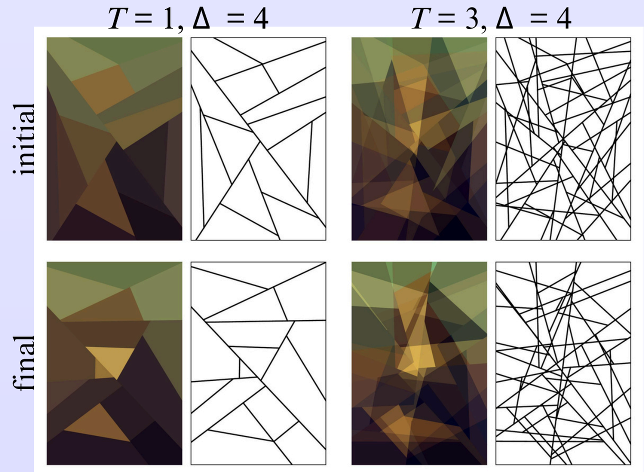




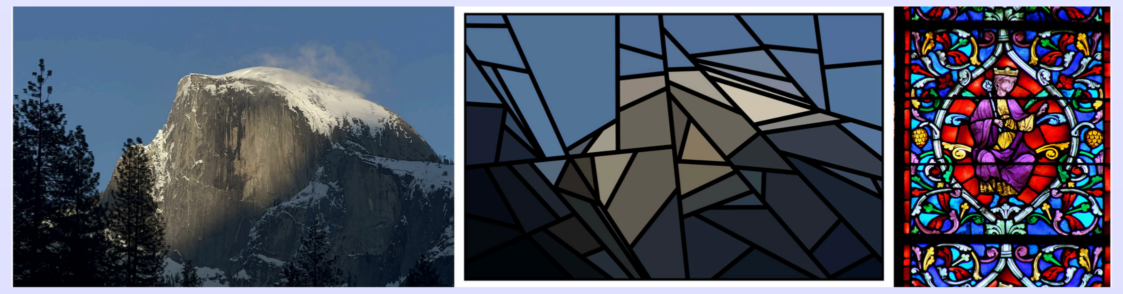
Real paintings: *Mona Lisa* (Leonardo da Vinci), *Starry Night* (Van Gogh) and a cubist painting, *Les Demoiselles d'Avignon* (Picasso).



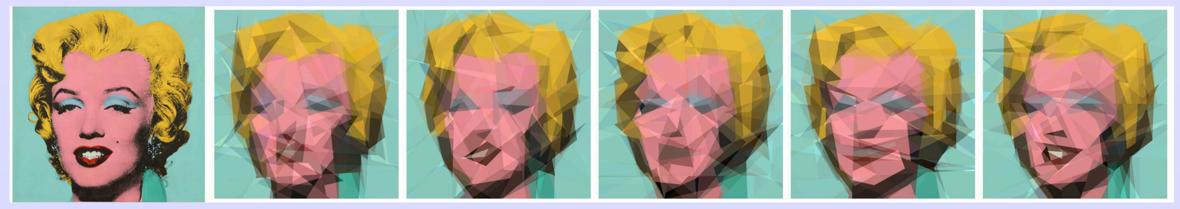
RMSE over TAO/FAO iterations and training time per iteration for *Mona Lisa*.

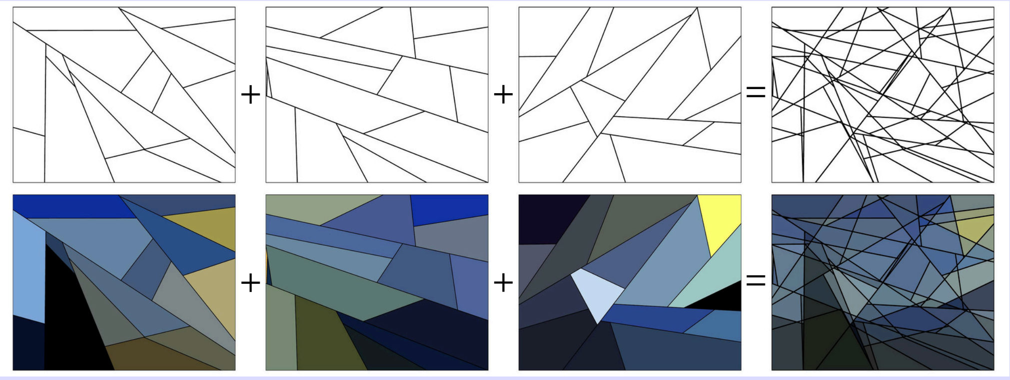


Initial & final result (after TAO/FAO converge) for *Mona Lisa*, as poxels and boundaries.


Stained-glass window effect (Δ = 6 and *T* = 1) for a source photograph.

Cubist tree and forest results using different depth Δ and number of trees *T* for a source photograph.


Cubist tree images using different seeds (for Δ = 6, *T* = 3).



Starry Night using a forest of *T* = 3 trees of depth Δ = 4: individual trees.