

# Artistic Thresholding

Jie Xu

Craig S. Kaplan

David R. Cheriton School of Computer Science  
University of Waterloo

Presentation in NPAR2008  
June 10, 2008



- 1 Background of Black-and-White Image Synthesis
- 2 Approach
  - Overview of our system
  - Constraints
  - Optimization Method
- 3 Results and Discussion

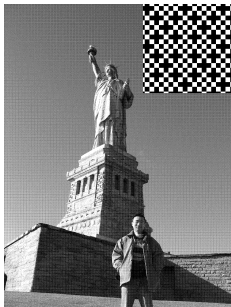
# Color vs. Monochrome

- Technological limitation: most devices started from monochrome.
- Color pictures are too real for some applications.
- Less is more.



# Black-and-White Image Generation

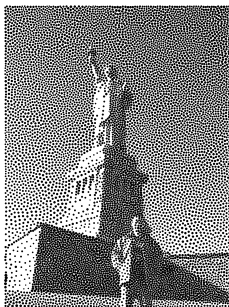
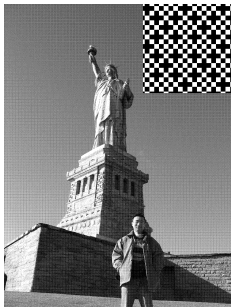
- Halftoning: ordered dithering, error diffusion
- Stippling
- Pen-and-ink illustration, mazes and line drawing





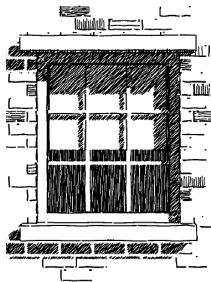
# Black-and-White Image Generation

- Halftoning: ordered dithering, error diffusion
- Stippling
- Pen-and-ink illustration, mazes and line drawing

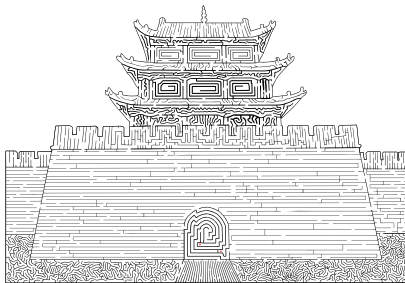


# Black-and-White Image Generation

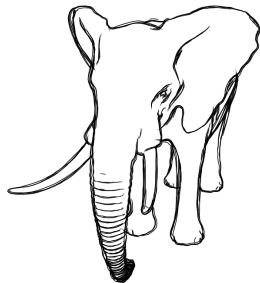
- Halftoning: ordered dithering, error diffusion
- Stippling
- Pen-and-ink illustration, mazes and line drawing



Winkenbach &  
Salesin 1994



Xu & Kaplan 2007



Goodwin et al. 2007

# Wholetoning

- The “simplest” coloring scheme
- High contrast
- Wide applications: graffiti art, comics, movies



Banksy



Sin City



Renaissance

- Dark regions should be black while bright regions should be white.

# Thresholding

- Naïve thresholding
- Adaptive thresholding
- Difference-of-Gaussians technique



thresholding



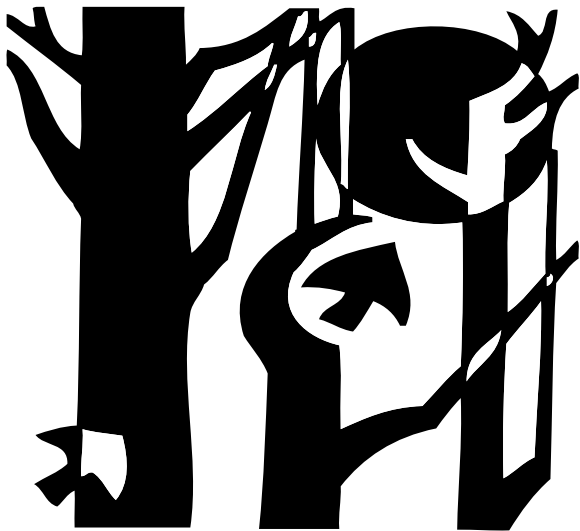
adaptive thresholding



DoG thresholding

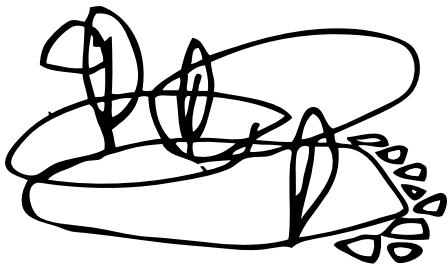
- Dark regions should be black while bright regions should be white.
- Adjacent things should have the same color: spatial locality, high-level coherence.

- Dark regions should be black while bright regions should be white.
- Adjacent things should have the same color: spatial locality, high-level coherence.
- Opposite colors should be used to depict edges.





- Dark regions should be black while bright regions should be white.
- Adjacent things should have the same color: spatial locality, high-level coherence.
- Opposite colors should be used to depict edges.
- Control relative amounts of black and white.



- Dark regions should be black while bright regions should be white.
- Adjacent things should have the same color: spatial locality, high-level coherence.
- Opposite colors should be used to depict edges.
- Control relative amounts of black and white.
- Distinguish high-level features.



# Motivation of Artistic Thresholding

- Discover the fundamental features of black-and-white illustration.
- Define a general framework for black-and-white image creation.
- Provide parameters to control style.

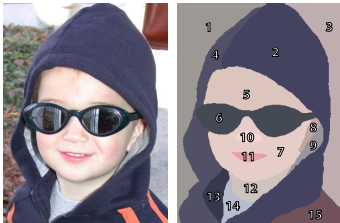
# Algorithm

- Segment source image.
- Create region adjacency graph.
- Optimize black-and-white coloring under constraints.



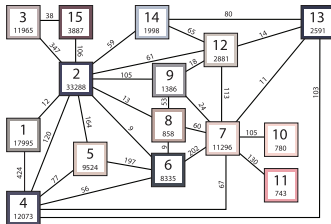
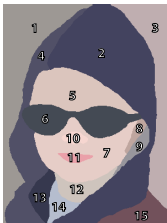
# Algorithm

- Segment source image.
- Create region adjacency graph.
- Optimize black-and-white coloring under constraints.



# Algorithm

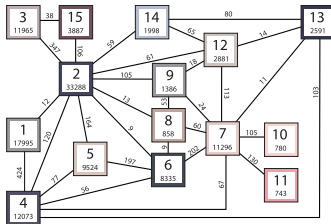
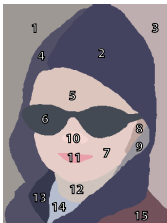
- Segment source image.
- Create region adjacency graph.
- Optimize black-and-white coloring under constraints.





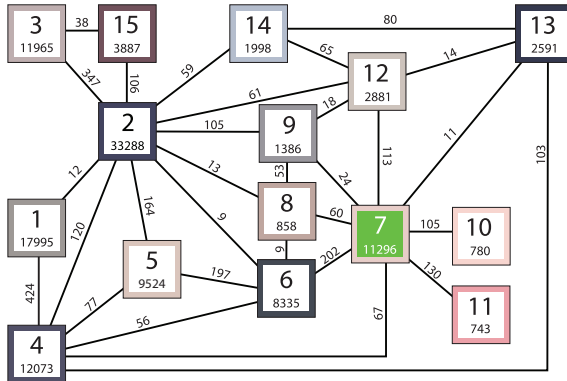
# Algorithm

- Segment source image.
- Create region adjacency graph.
- Optimize black-and-white coloring under constraints.



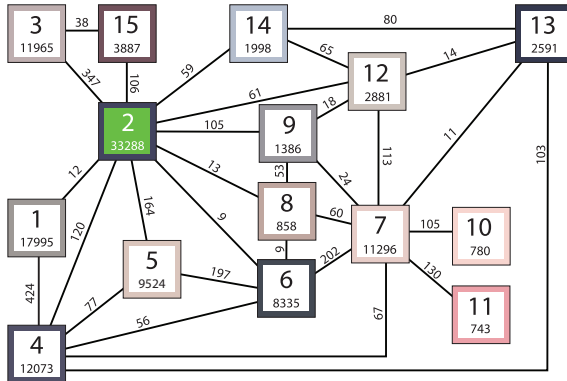
# Graph Construction

- Each vertex corresponds to a segment.
- Two adjacent segments are connected by an edge.
- Some properties are stored in vertices (color, area, boundary) and edges (the length of shared borders).



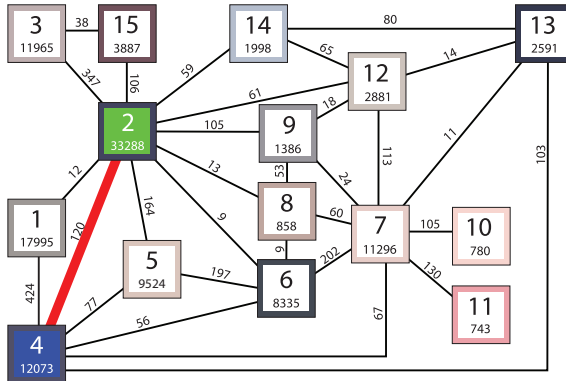
# Graph Construction

- Each vertex corresponds to a segment.
- Two adjacent segments are connected by an edge.
- Some properties are stored in vertices (color, area, boundary) and edges (the length of shared borders).



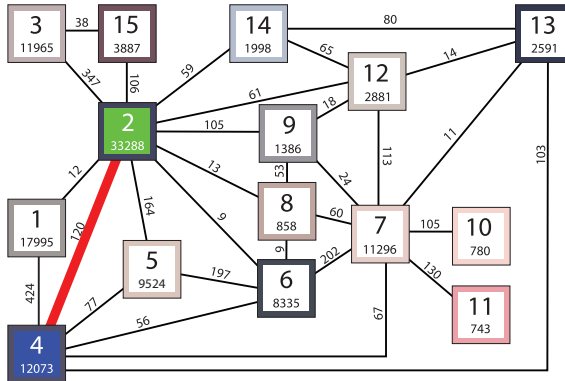
# Graph Construction

- Each vertex corresponds to a segment.
- Two adjacent segments are connected by an edge.
- Some properties are stored in vertices (color, area, boundary) and edges (the length of shared borders).



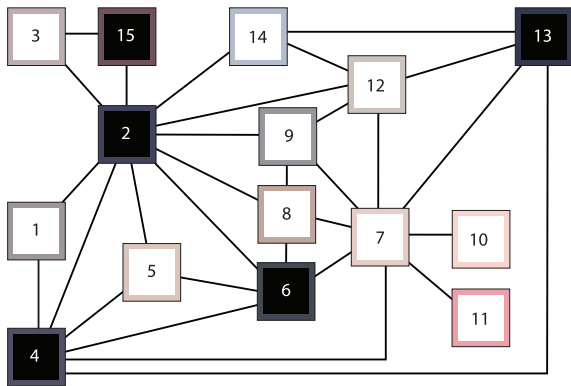
# Graph Construction

- Each vertex corresponds to a segment.
- Two adjacent segments are connected by an edge.
- Some properties are stored in vertices (color, area, boundary) and edges (the length of shared borders).



# Color Matching

- $C_{col}$  measures the color intensity difference.



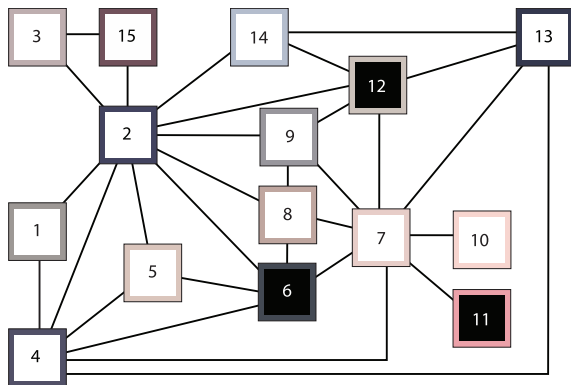
# Color Matching

- $C_{col}$  measures the color intensity difference.



# Area Matching

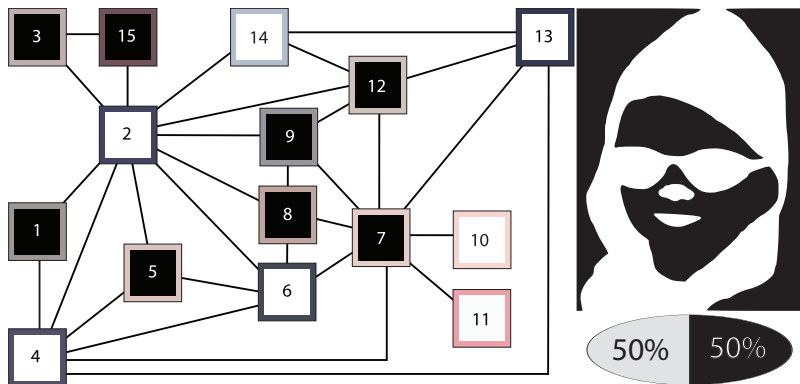
- $C_{area}$  controls the proportion of black in the total image.





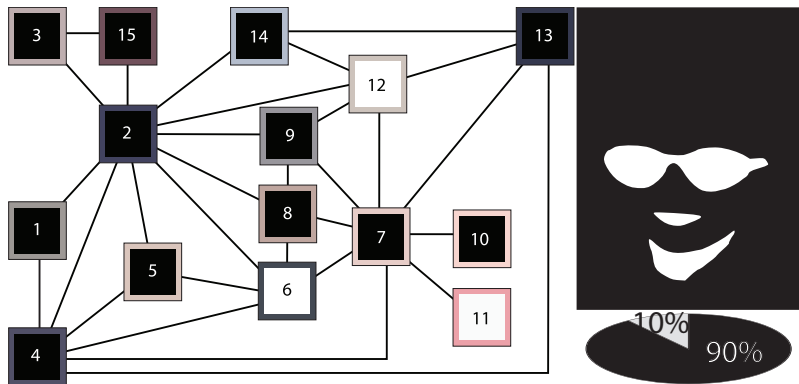
# Area Matching

- $C_{area}$  controls the proportion of black in the total image.



# Area Matching

- $C_{area}$  controls the proportion of black in the total image.

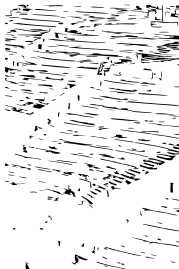


# Area Matching

- $C_{area}$  controls the proportion of black in the total image.



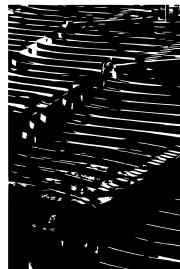
source



10%



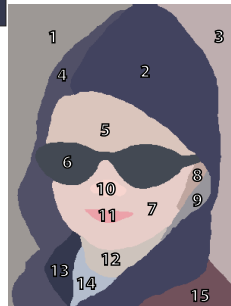
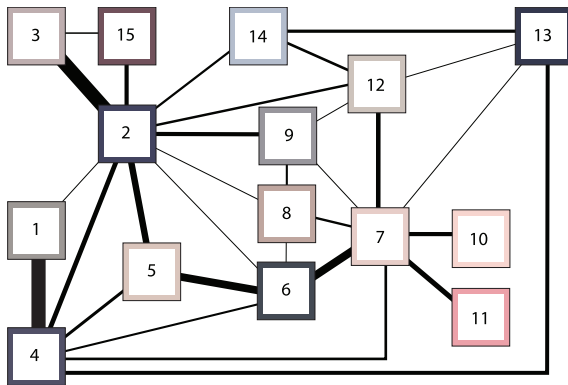
50%



90%

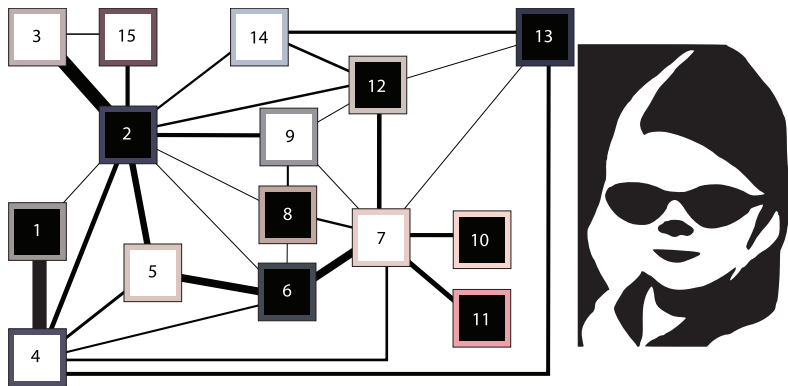
# Boundary Contrast

- Measure the impact of boundary contrast.
- $C_{alike}$  aims at highlighting boundaries.
- $C_{opp}$  aims at eliminating boundaries.



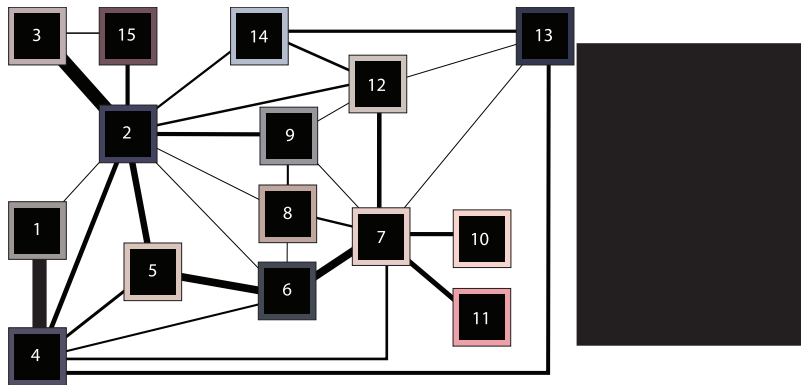
# Boundary Contrast

- Measure the impact of boundary contrast.
- $C_{alike}$  aims at highlighting boundaries.
- $C_{opp}$  aims at eliminating boundaries.



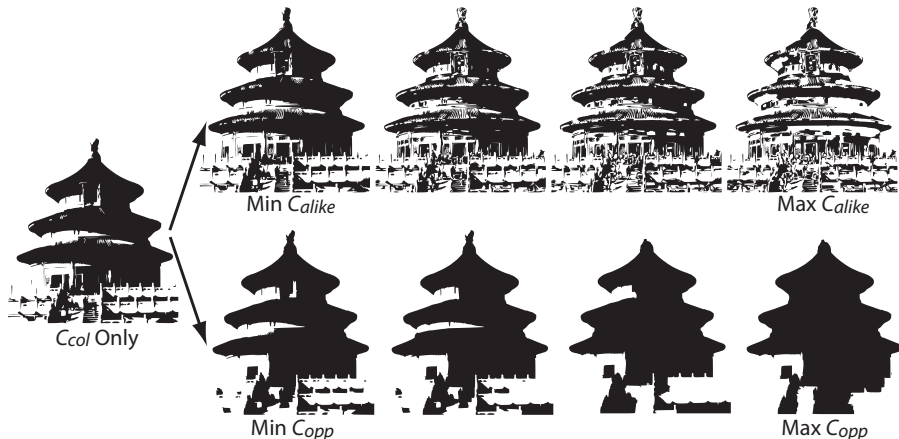
# Boundary Contrast

- Measure the impact of boundary contrast.
- $C_{alike}$  aims at highlighting boundaries.
- $C_{opp}$  aims at eliminating boundaries.



# Boundary Contrast

- Measure the impact of boundary contrast.
- $C_{alike}$  aims at highlighting boundaries.
- $C_{opp}$  aims at eliminating boundaries.



# Boundary Contrast

- Measure the impact of boundary contrast.
- $C_{alike}$  aims at highlighting boundaries.
- $C_{opp}$  aims at eliminating boundaries.

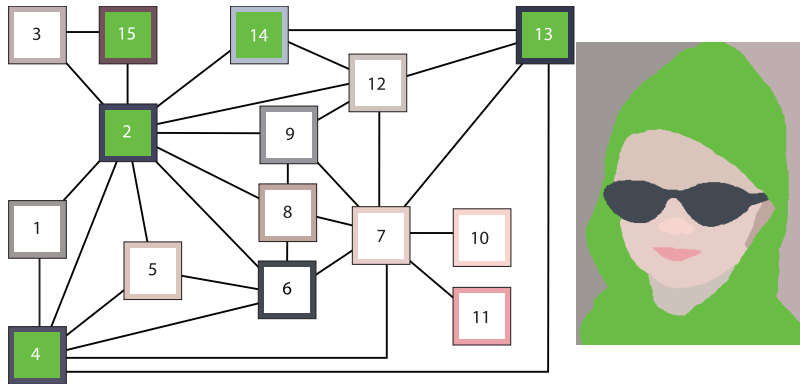


$$C_{col} + C_{alike} + C_{opp}$$



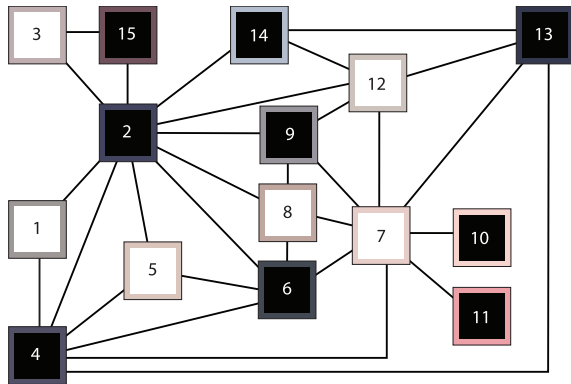
# Feature Homogeneity

- User can provide high-level features.
- $C_{group}$  tries to achieve a uniform color in each feature.



# Feature Homogeneity

- User can provide high-level features.
- $C_{group}$  tries to achieve a uniform color in each feature.

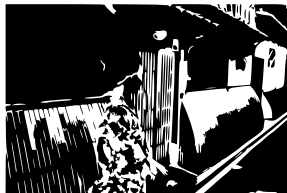


# Feature Homogeneity

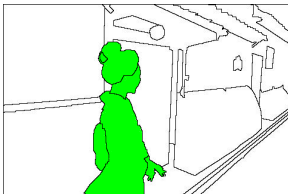
- User can provide high-level features.
- $C_{group}$  tries to achieve a uniform color in each feature.



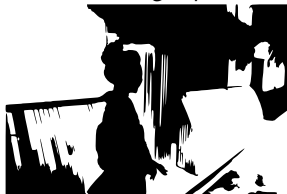
Source



No  $C_{group}$



User-supplied high-level features



With  $C_{group}$

- Compute the total cost as a weighted sum of all constraints.
- Each cost is normalized to  $[0, 1]$ .

$$C_{total} = \frac{w_{col} C_{col} + w_{area} C_{area} + w_{alike} C_{alike} + w_{opp} C_{opp} + w_{group} C_{group}}{w_{col} + w_{area} + w_{alike} + w_{opp} + w_{group}}$$

- Compute local optima of random subgraphs.
- Use a simulated annealing framework.
- Apply morphological operators as postprocessing.
- Vectorize final result.

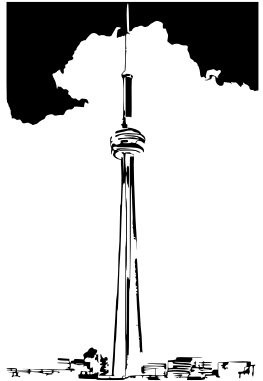
# Comparison



bilateral + adaptive

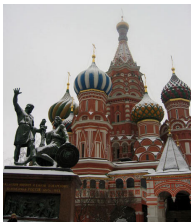


bilateral + threshold



artistic thresholding

# Results



# Results





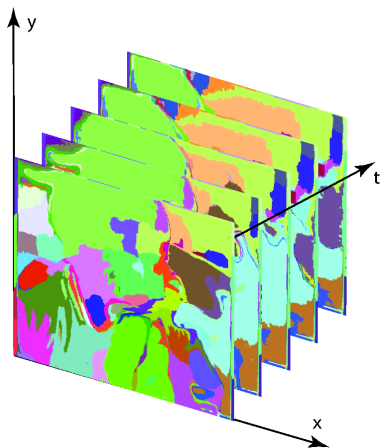
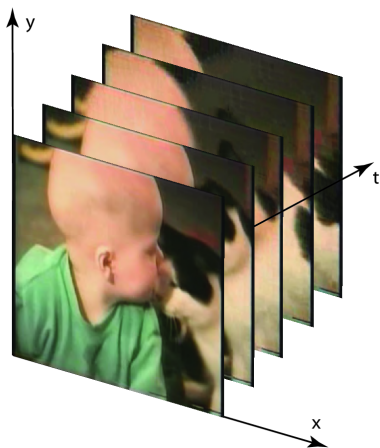
# Results



# Application: Papercutting



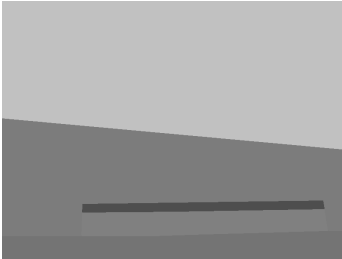
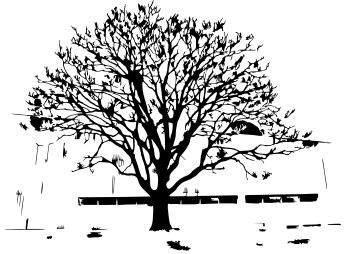
# Extension: Animation



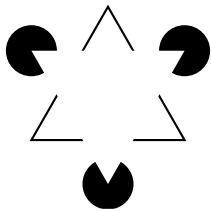
# Future Work: XOR Composition



# Future Work: XOR Composition



# Extension: Illusory Contours



- Present a general framework for black-and-white image synthesis.
- User can set constraints interactively.
- Support various styles.

# Questions?

