Project: Foliage Detection

Iuliia Pliushch, Wolfgang Stammer, Dennis Soukup

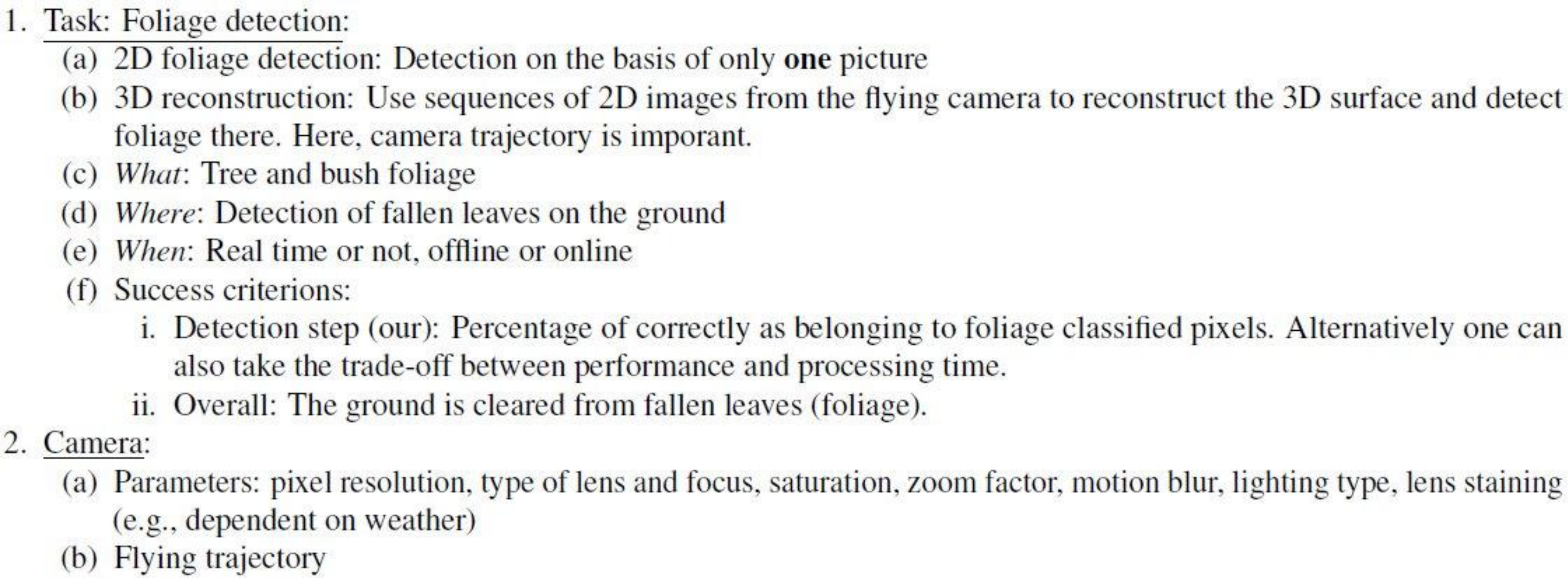
Outline

- Introduction
- Context model
- Blender realization
- Algorithmic pipeline
  - Ground detection
  - Color transformation
  - Edge detection
- Results
- Discussion

Introduction

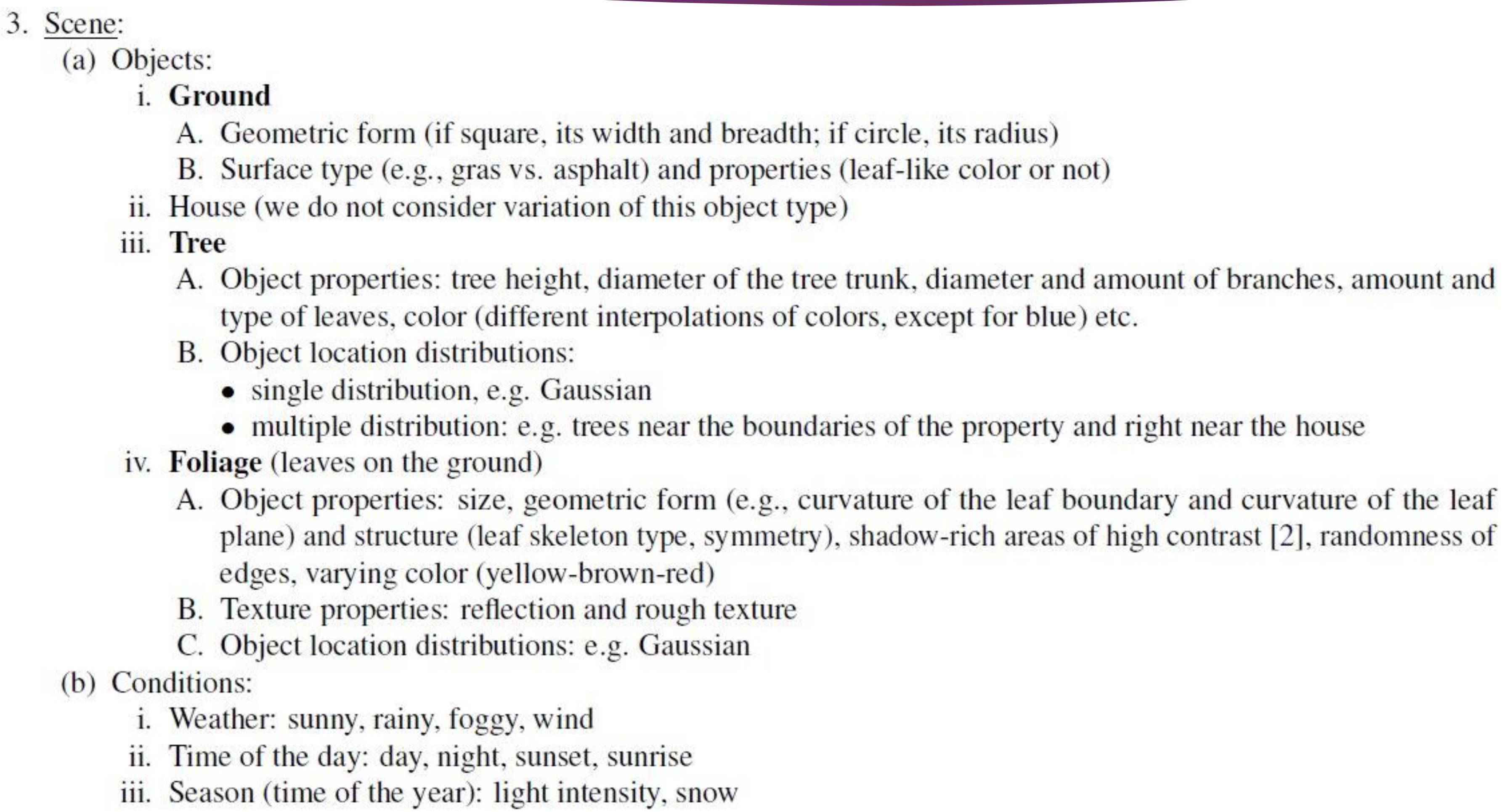
- Project task: foliage detection on single images
- Procedure:
  - Formulating the context model (CM)
  - Implementing the CM in Blender
  - Searching for invariants in CM (low entropy)
  - Designing an algorithmic pipeline
  - Evaluating results

Context model 1/6



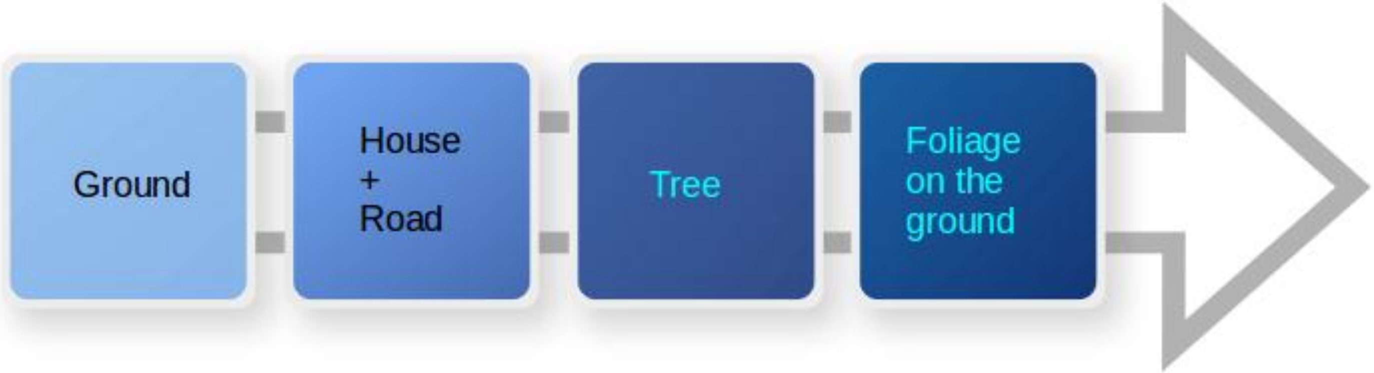
1. Task: Foliage detection:
   (a) 2D foliage detection: Detection on the basis of only **one** picture
   (b) 3D reconstruction: Use sequences of 2D images from the flying camera to reconstruct the 3D surface and detect foliage there. Here, camera trajectory is imporant.
   (c) *What*: Tree and bush foliage
   (d) *Where*: Detection of fallen leaves on the ground
   (e) *When*: Real time or not, offline or online
   (f) Success criterions:
      i. Detection step (our): Percentage of correctly as belonging to foliage classified pixels. Alternatively one can also take the trade-off between performance and processing time.
      ii. Overall: The ground is cleared from fallen leaves (foliage).
2. Camera:
   (a) Parameters: pixel resolution, type of lens and focus, saturation, zoom factor, motion blur, lighting type, lens staining (e.g., dependent on weather)
   (b) Flying trajectory

Context model 2/6

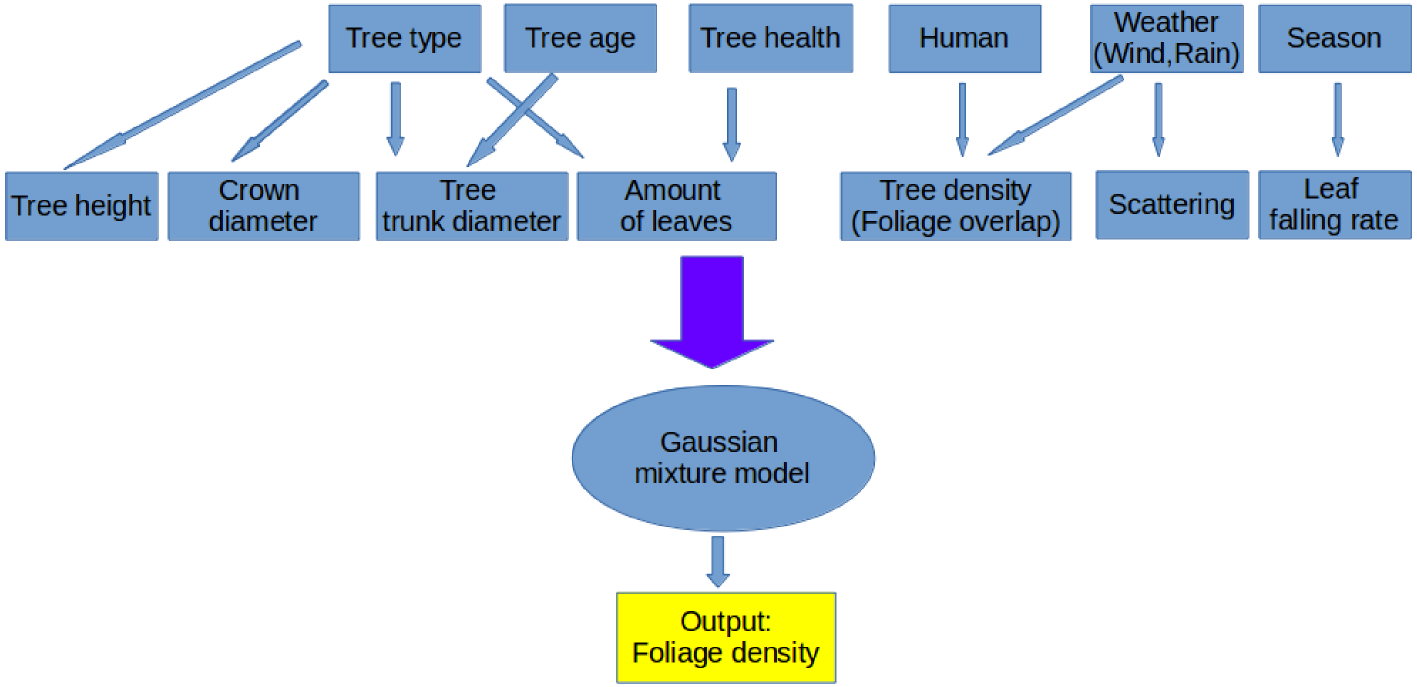


3. Scene:
    - (a) Objects:
        - i. **Ground**
            - A. Geometric form (if square, its width and breadth; if circle, its radius)
            - B. Surface type (e.g., gras vs. asphalt) and properties (leaf-like color or not)
        - ii. House (we do not consider variation of this object type)
        - iii. **Tree**
            - A. Object properties: tree height, diameter of the tree trunk, diameter and amount of branches, amount and type of leaves, color (different interpolations of colors, except for blue) etc.
            - B. Object location distributions:
                - single distribution, e.g. Gaussian
                - multiple distribution: e.g. trees near the boundaries of the property and right near the house
        - iv. **Foliage** (leaves on the ground)
            - A. Object properties: size, geometric form (e.g., curvature of the leaf boundary and curvature of the leaf plane) and structure (leaf skeleton type, symmetry), shadow-rich areas of high contrast [2], randomness of edges, varying color (yellow-brown-red)
            - B. Texture properties: reflection and rough texture
            - C. Object location distributions: e.g. Gaussian
    - (b) Conditions:
        - i. Weather: sunny, rainy, foggy, wind
        - ii. Time of the day: day, night, sunset, sunrise
        - iii. Season (time of the year): light intensity, snow

Context model 3/6



Context model 4/6



Tree type

Tree age

Tree health

Human

Weather (Wind,Rain)

Season

Tree height

Crown diameter

Tree trunk diameter

Amount of leaves

Tree density (Foliage overlap)

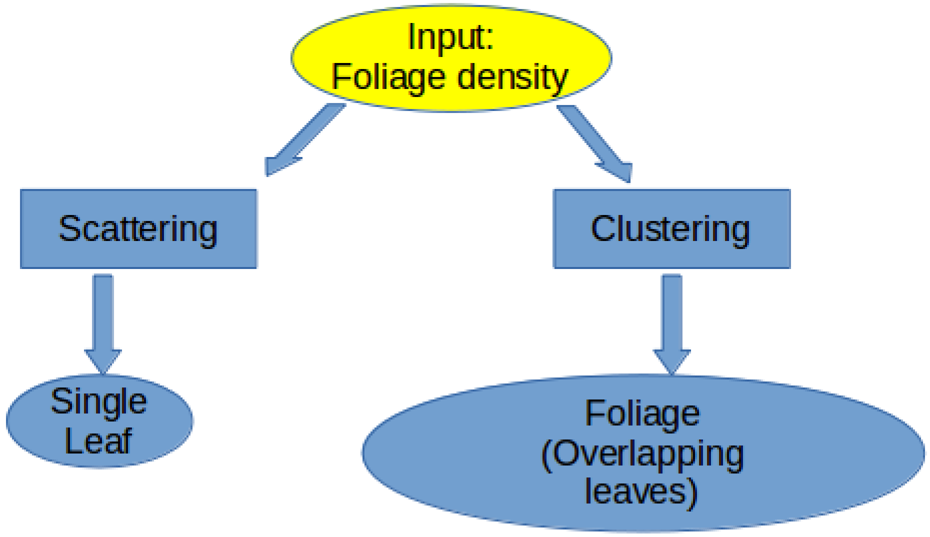
Scattering

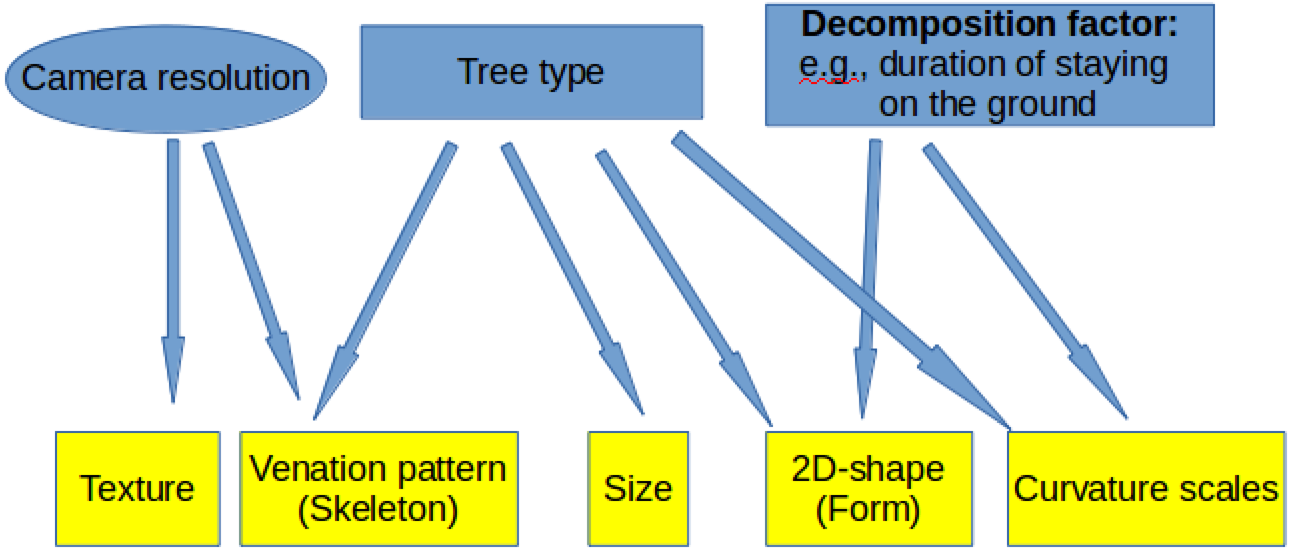
Leaf falling rate

Gaussian mixture model

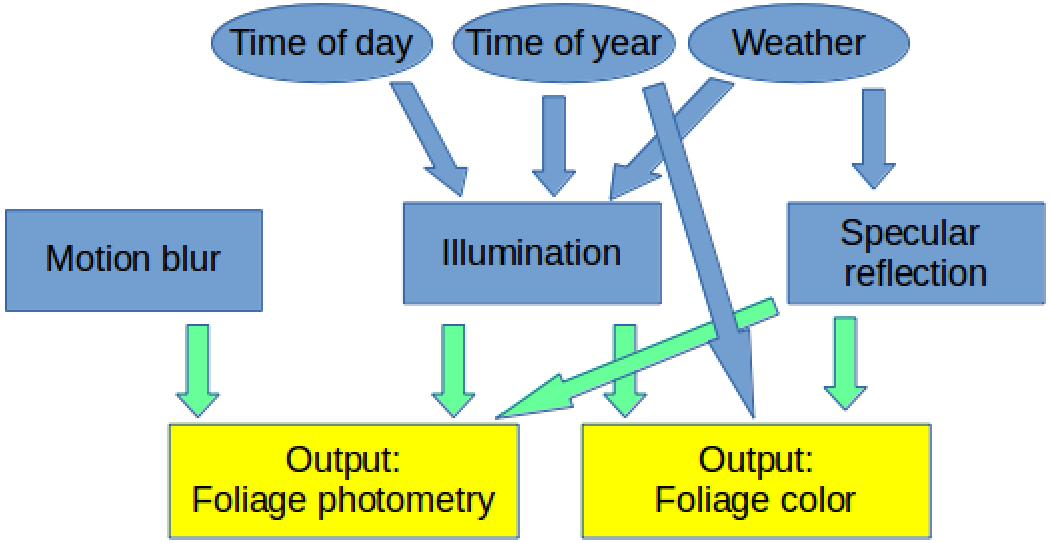
Output: Foliage density

Context model 5/6





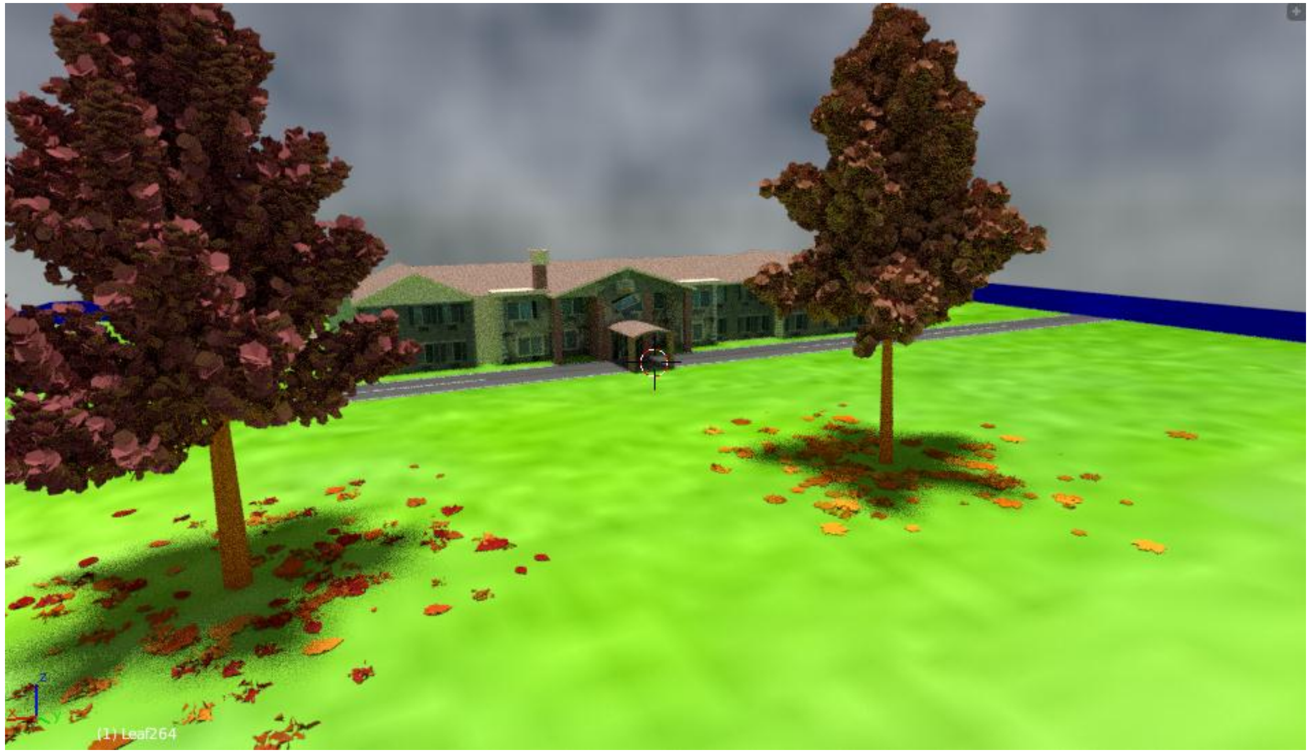
Context model 6/6

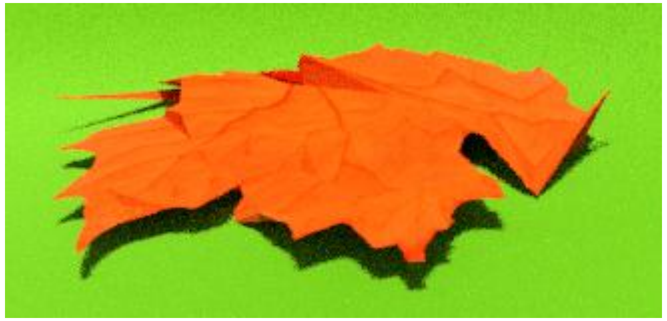


Blender realization 1/2

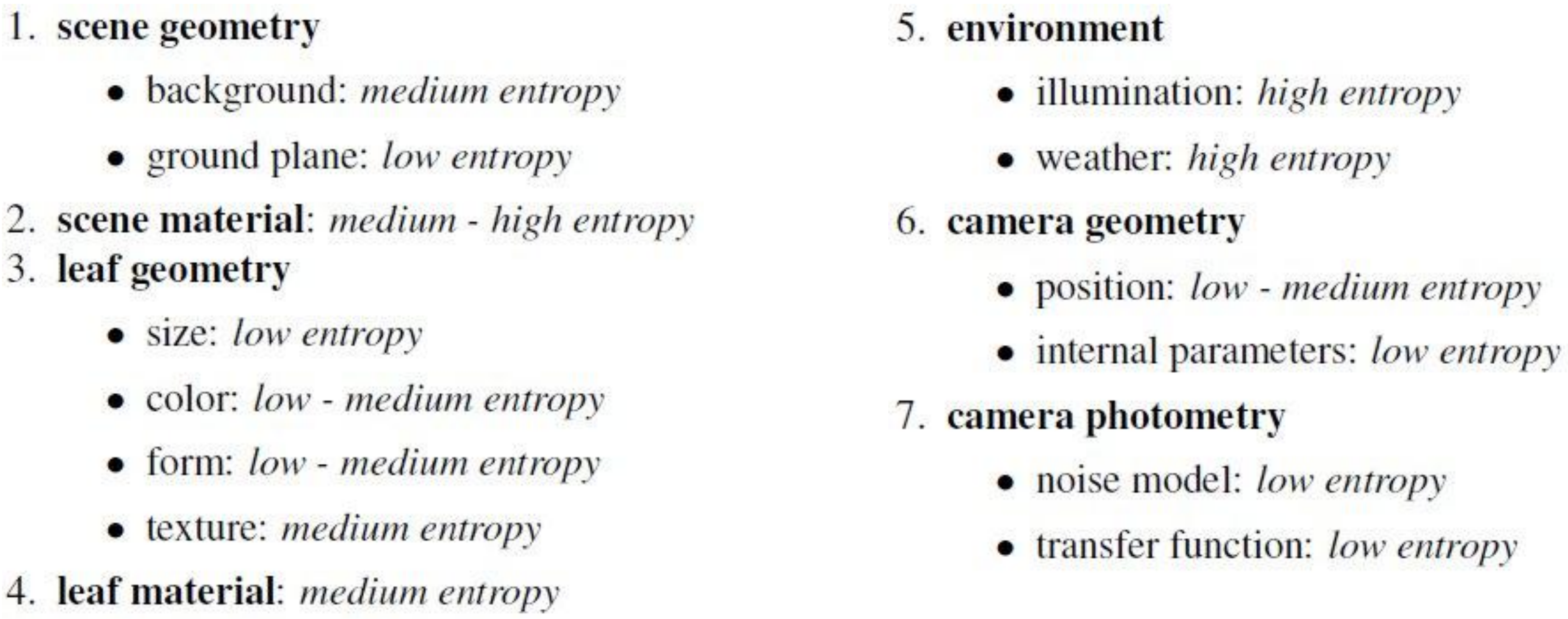
Objects	Geometry	Texture	Color
Ground	Plane	Different textures	Illumination parameters
Tree	- Tree trunk diameter - Tree height - Number of branches - Number and color of fallen leaves	No texture	Trunk: brown Leaves: adjusted to ground foliage
Foliage	Gaussian distribution around trees	5 different predefined textures	High amount of yellow and red

Blender realization 2/2



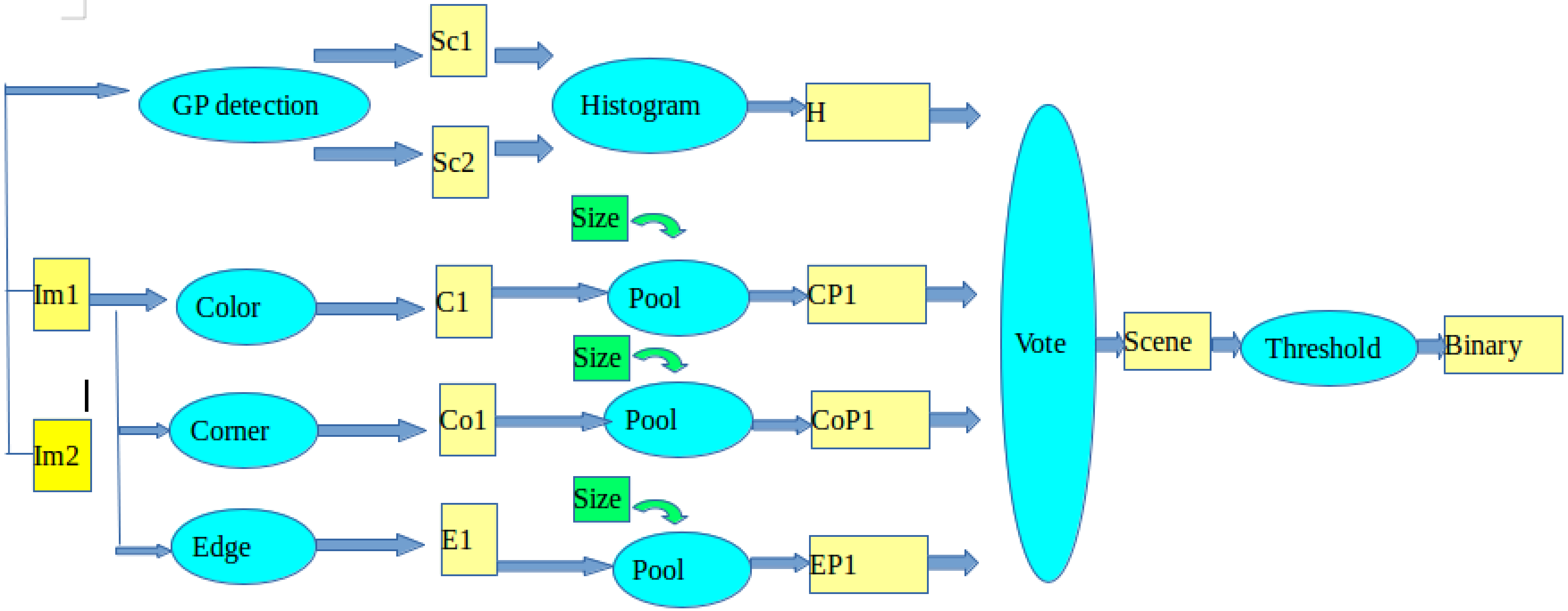


Algorithmic pipeline 1/2

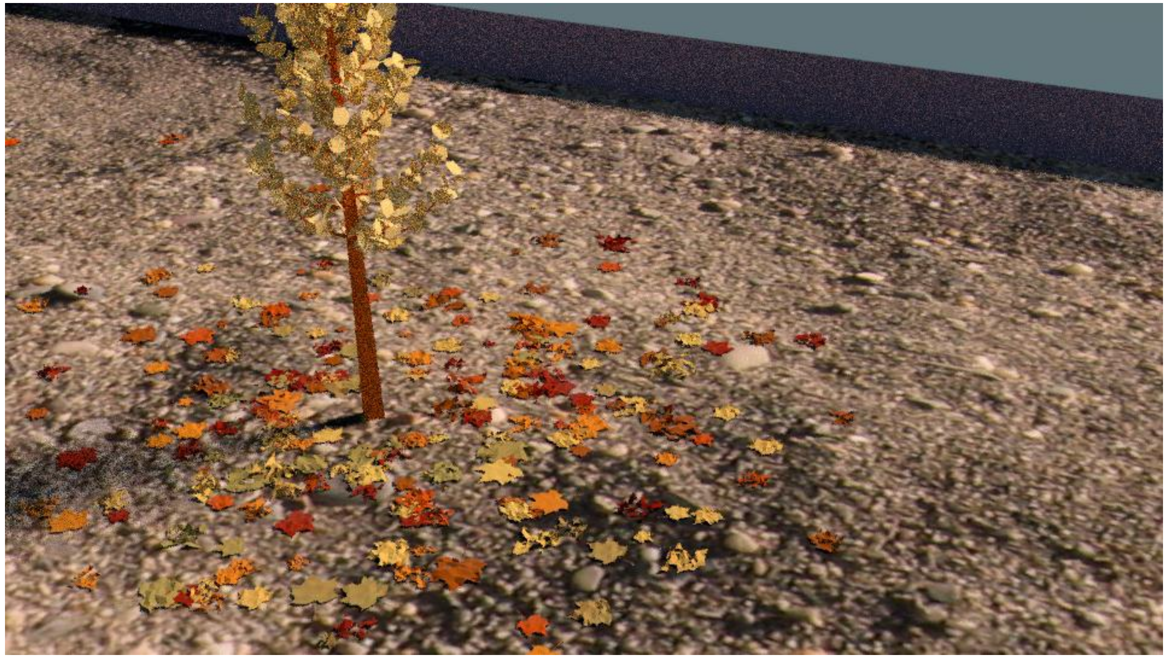


1. **scene geometry**
   - background: *medium entropy*
   - ground plane: *low entropy*
2. **scene material**: *medium - high entropy*
3. **leaf geometry**
   - size: *low entropy*
   - color: *low - medium entropy*
   - form: *low - medium entropy*
   - texture: *medium entropy*
4. **leaf material**: *medium entropy*
5. **environment**
   - illumination: *high entropy*
   - weather: *high entropy*
6. **camera geometry**
   - position: *low - medium entropy*
   - internal parameters: *low entropy*
7. **camera photometry**
   - noise model: *low entropy*
   - transfer function: *low entropy*

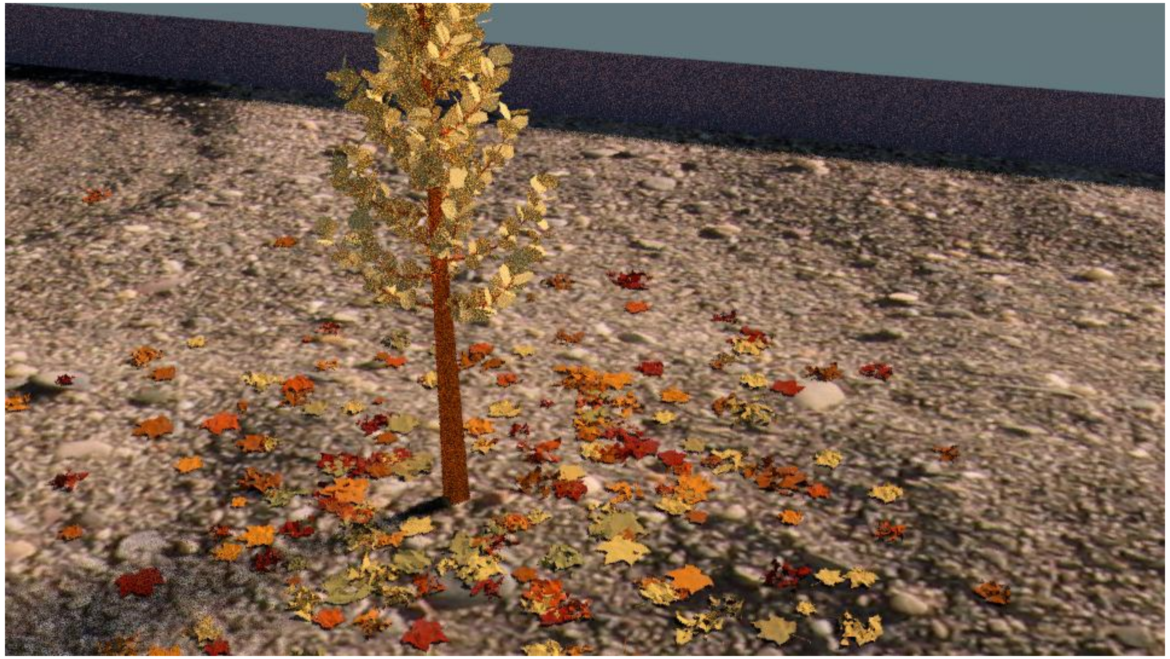
Algorithmic pipeline 2/2



Ground detection 1/4

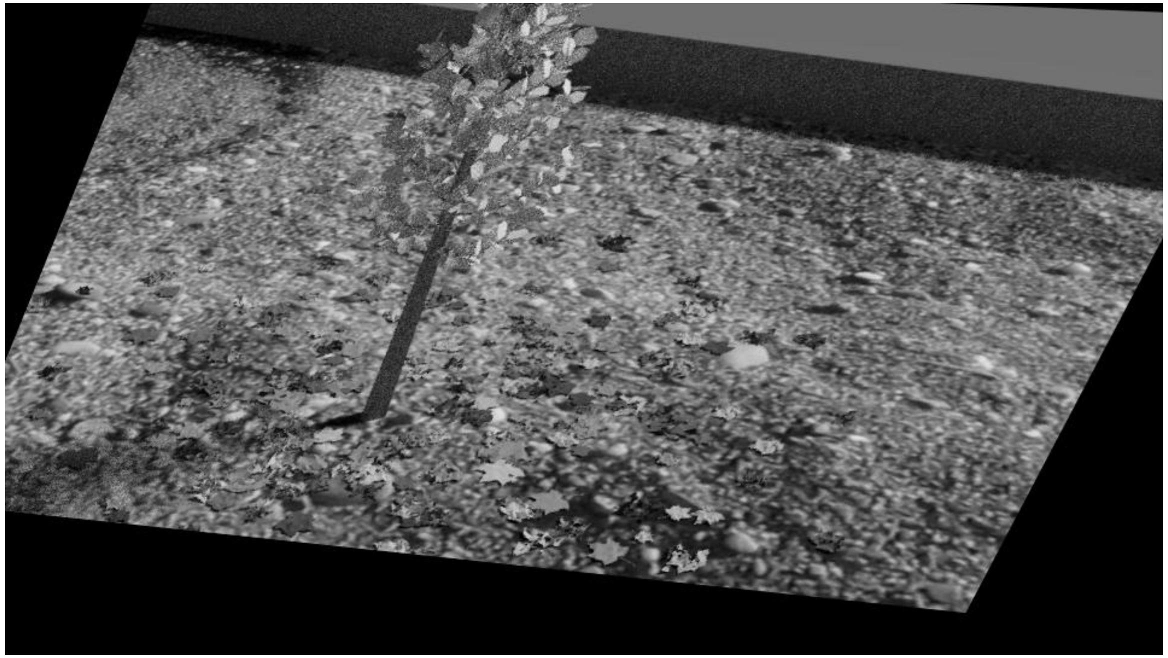


Destination image

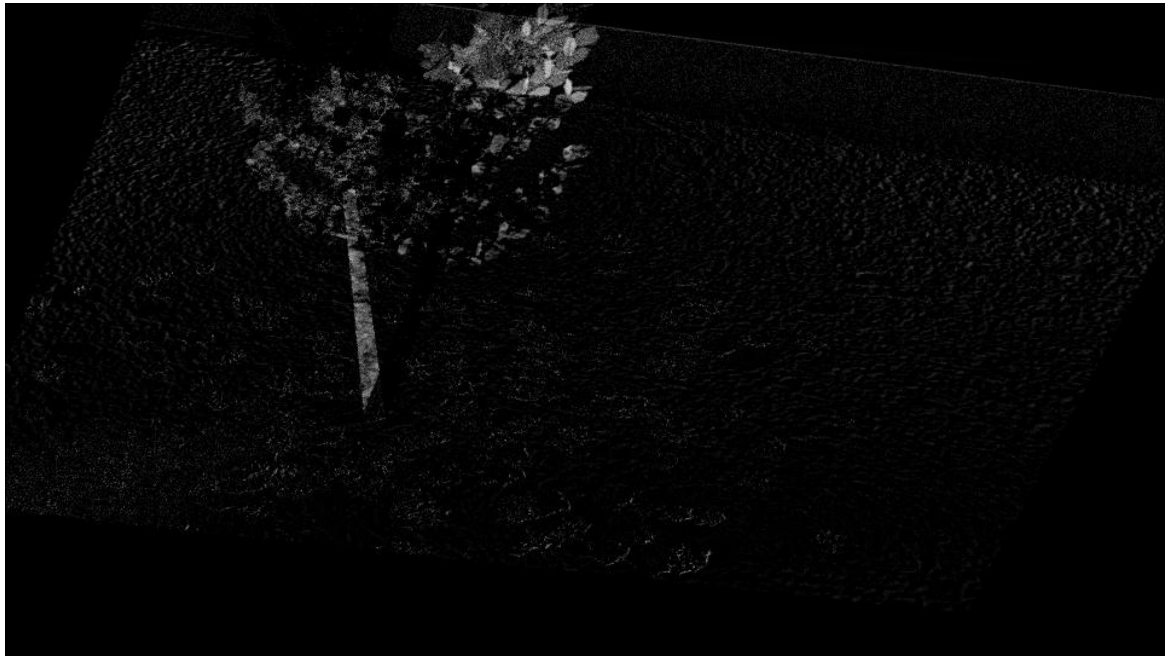


Source image

Ground detection 2/4

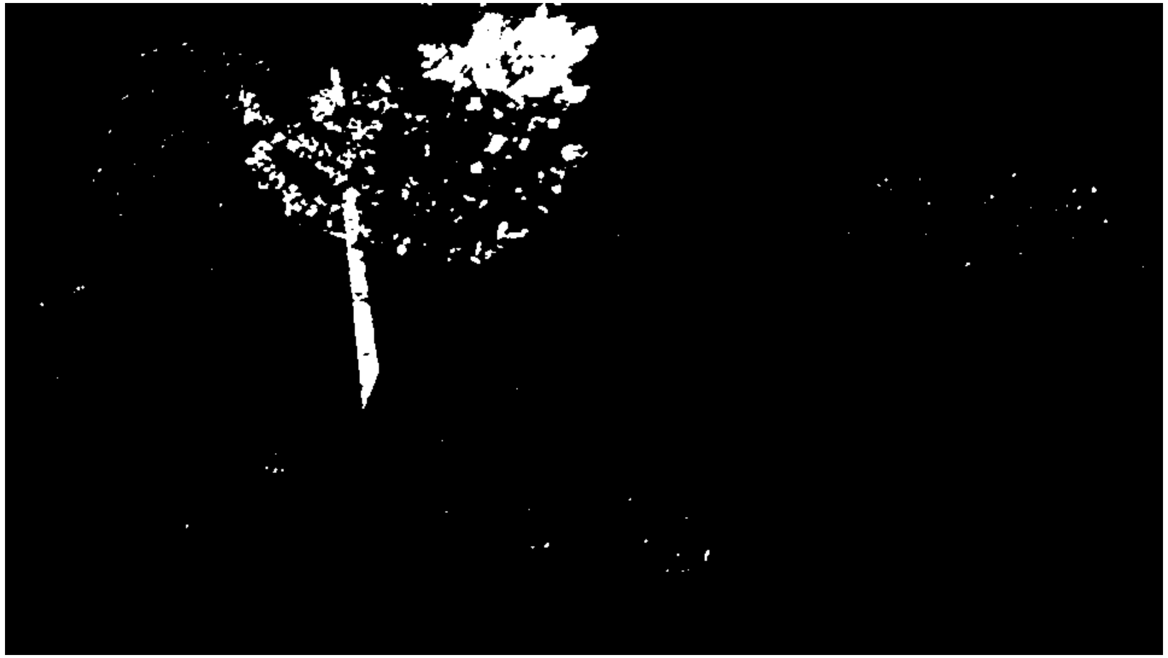


Warped image

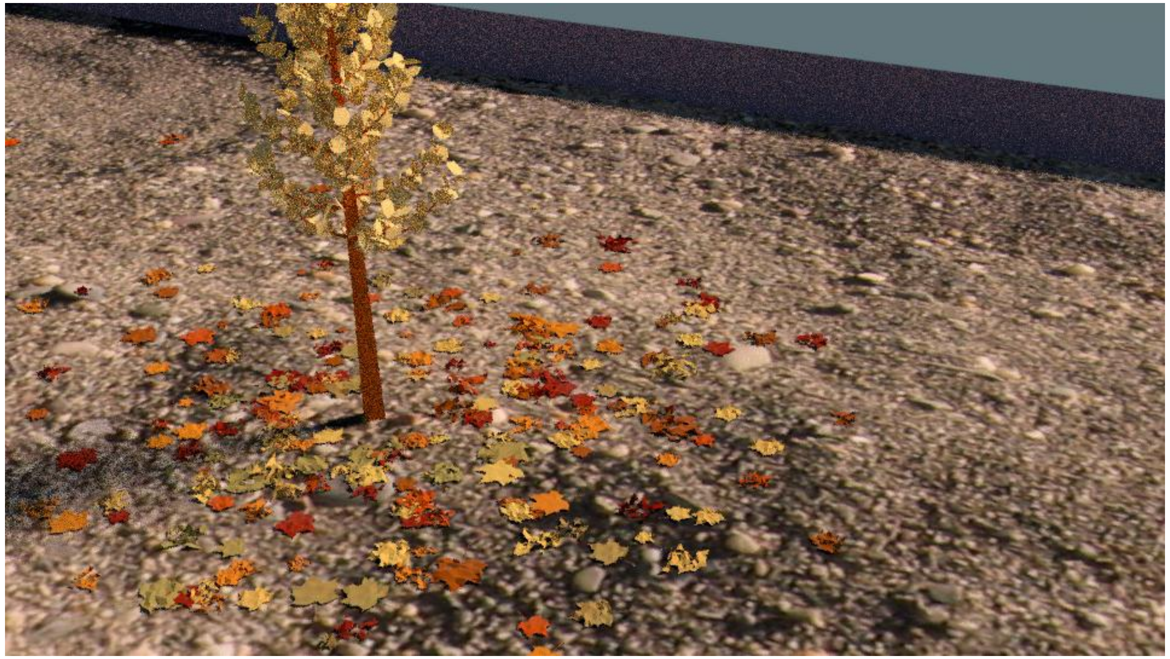


Subtracted image

Ground detection 3/4

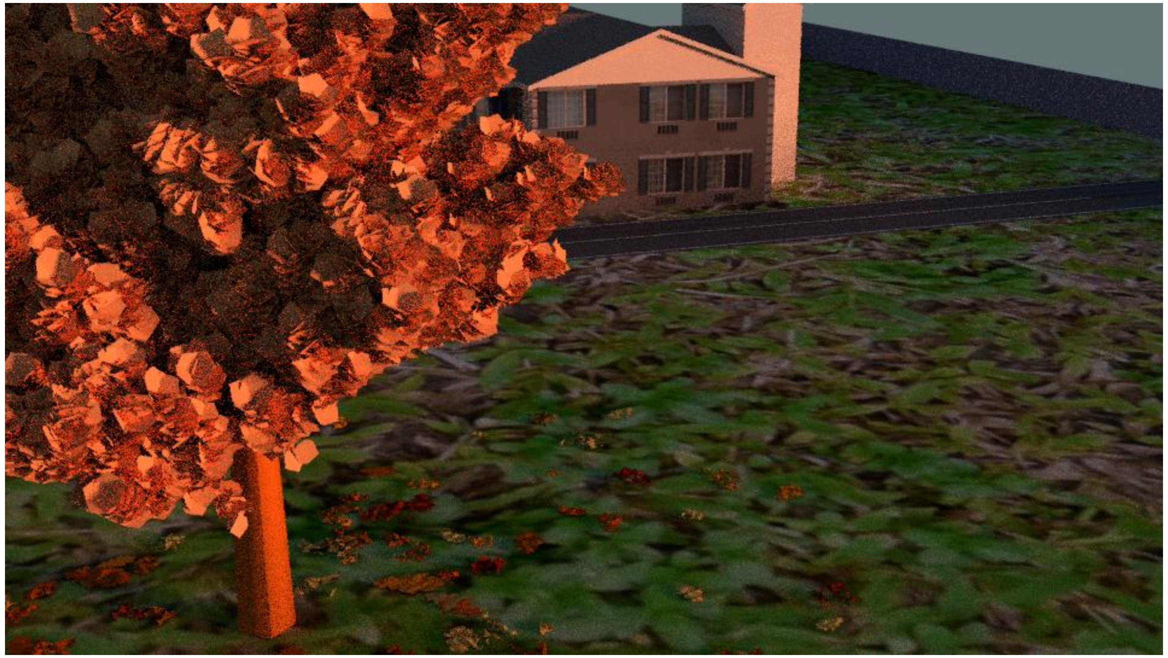


Otsu‘s threshold (T = 35)

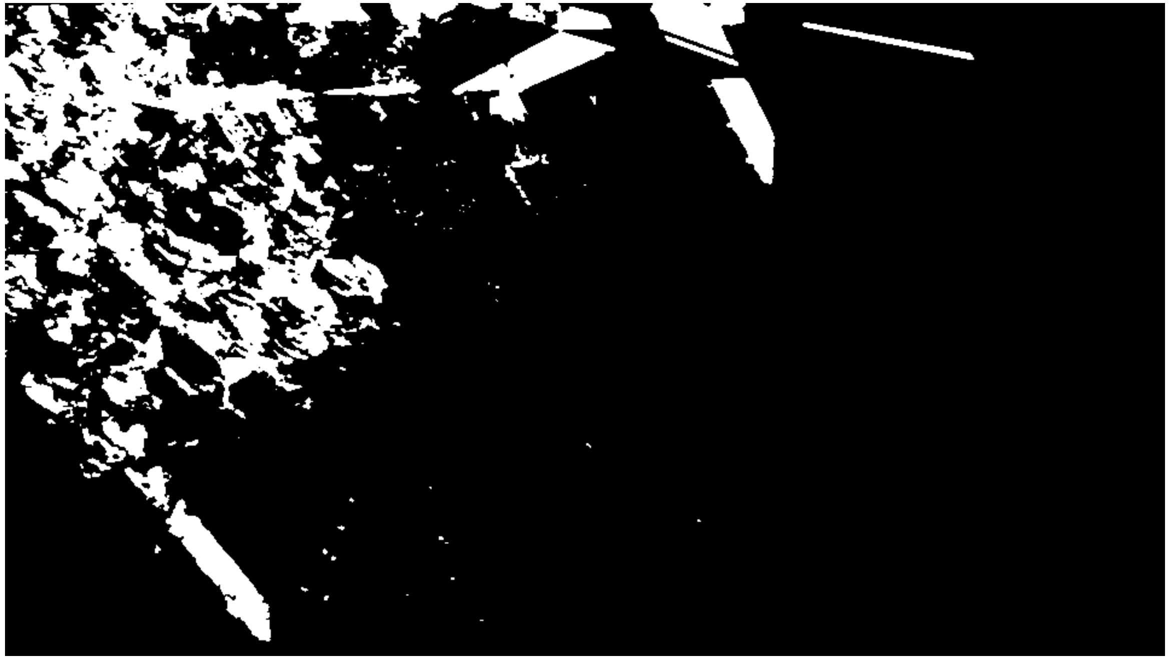


Reference: Destination image

Ground detection 4/4

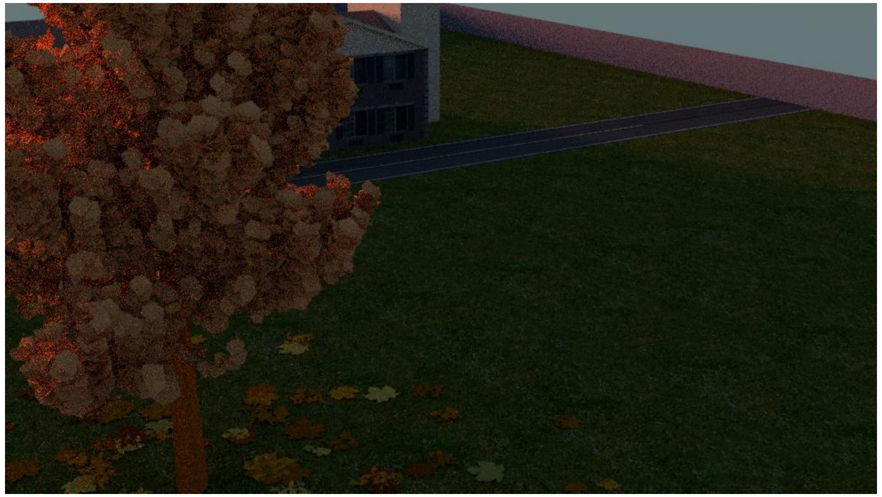


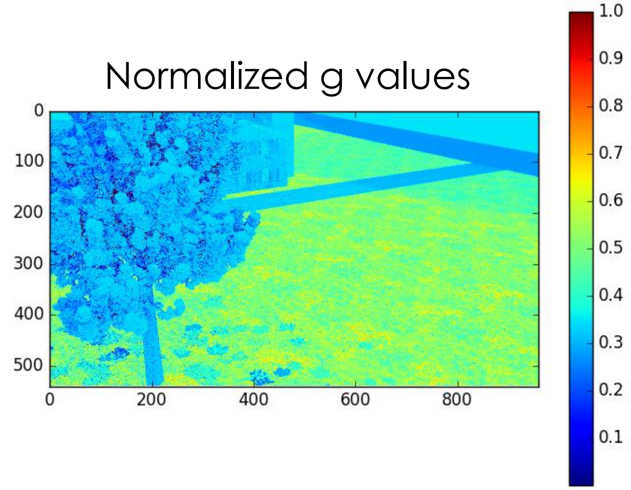
Destination image

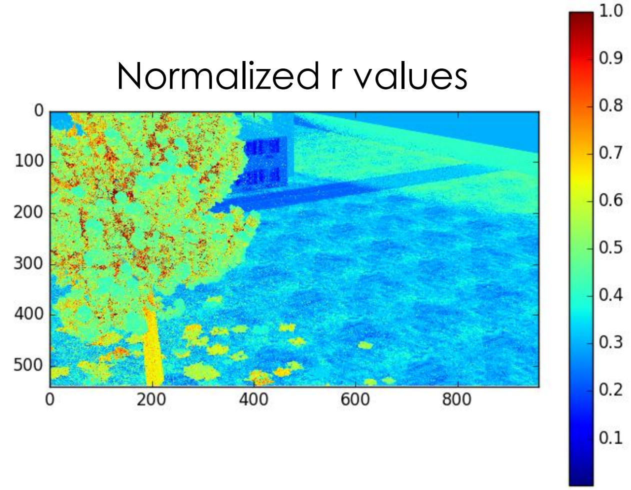


Otsu's threshold (T = 38)

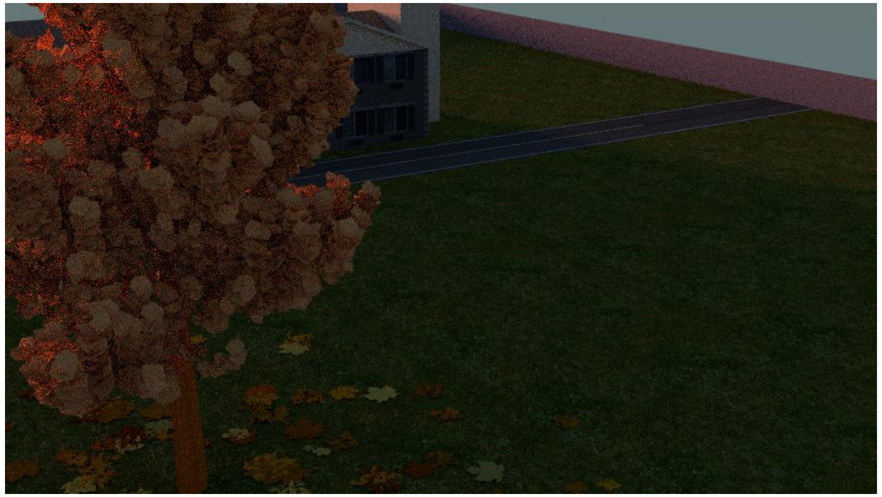
Color transformation 1/3

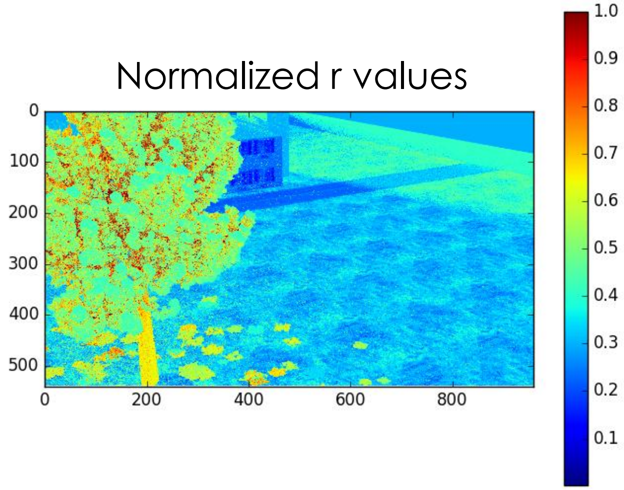


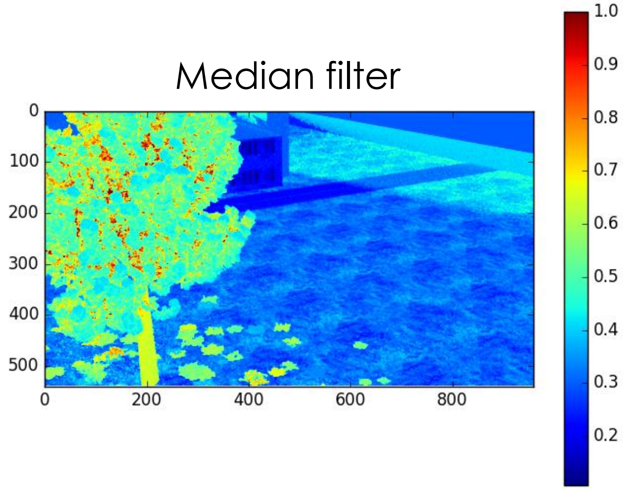




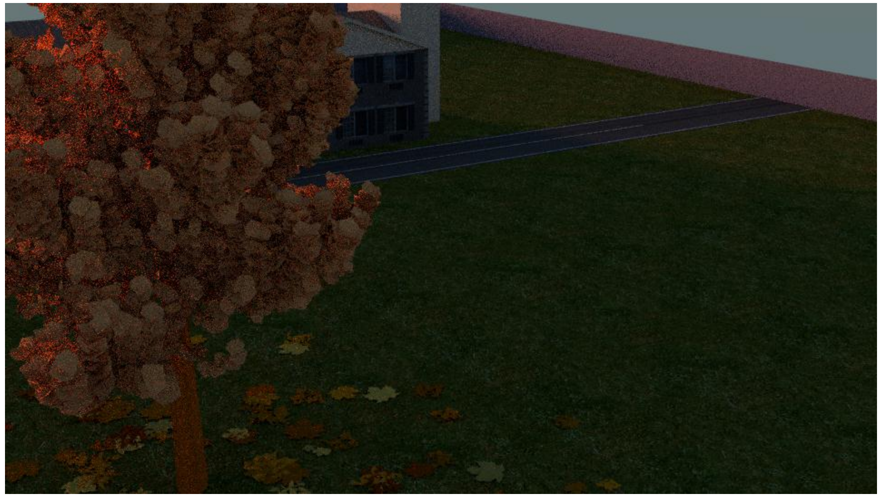
Color transformation 2/3

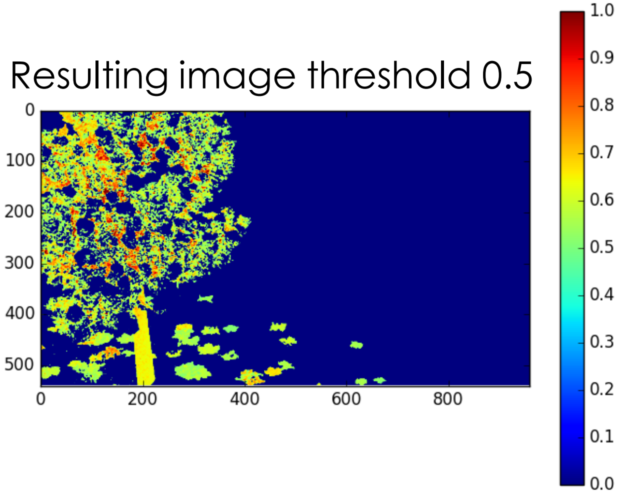




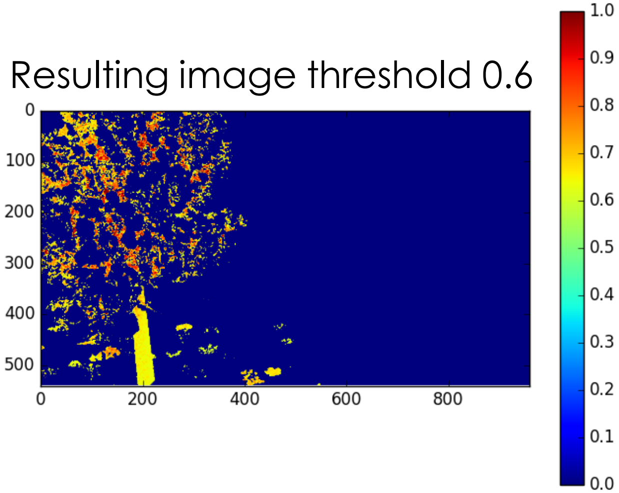


Color transformation 3/3



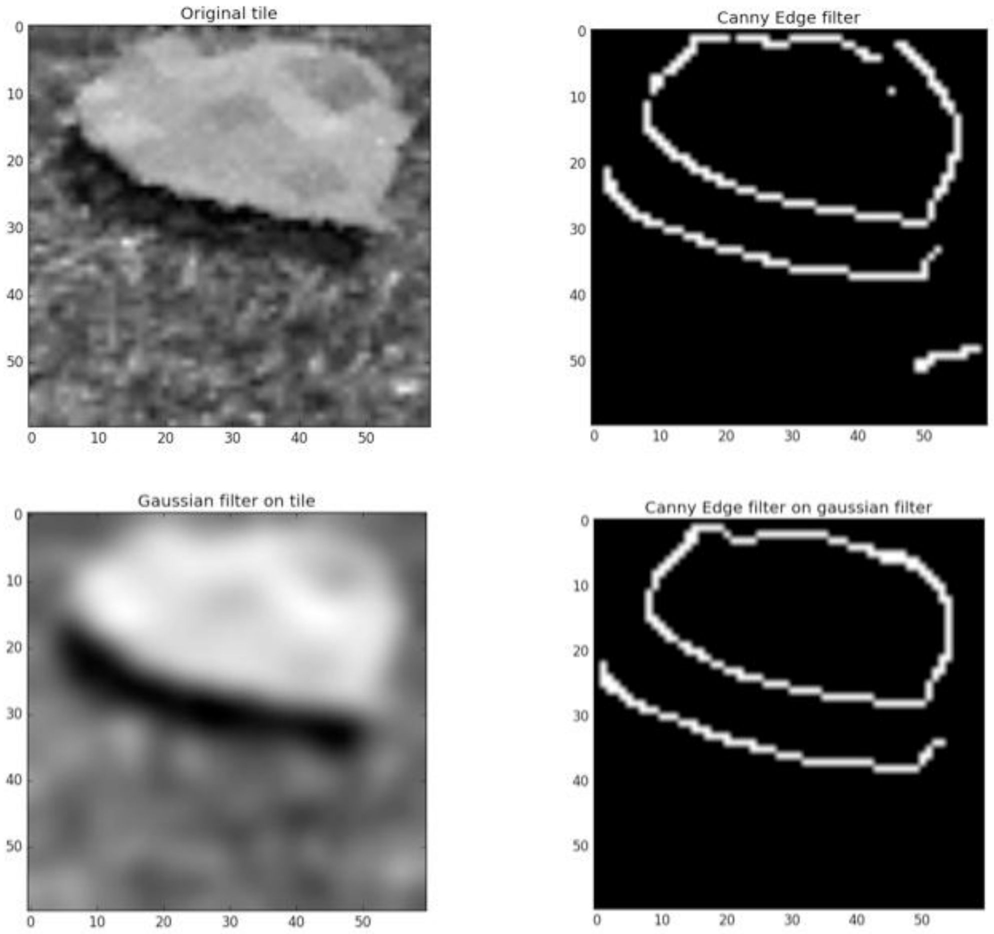


Resulting image threshold 0.5

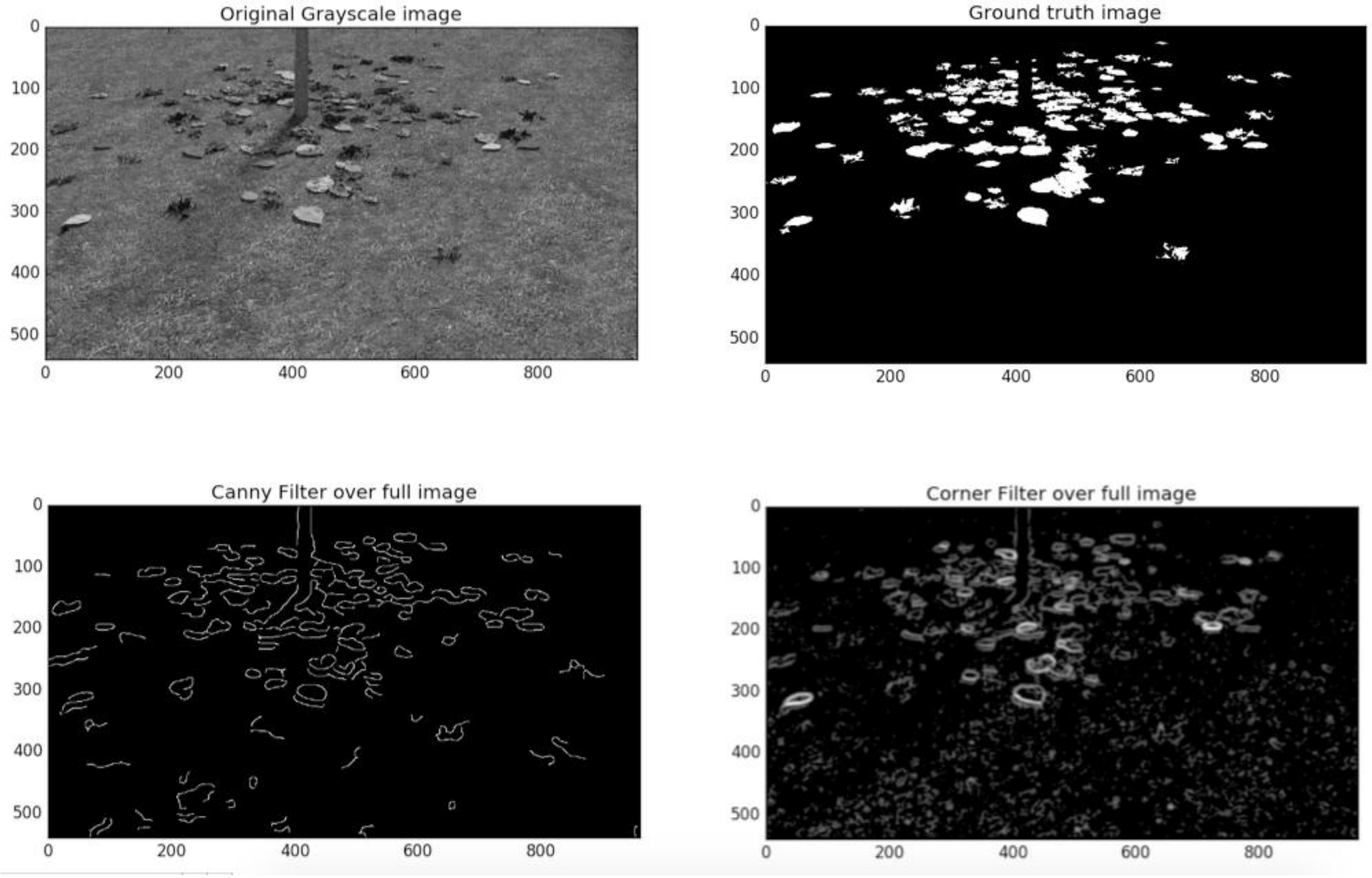


Resulting image threshold 0.6

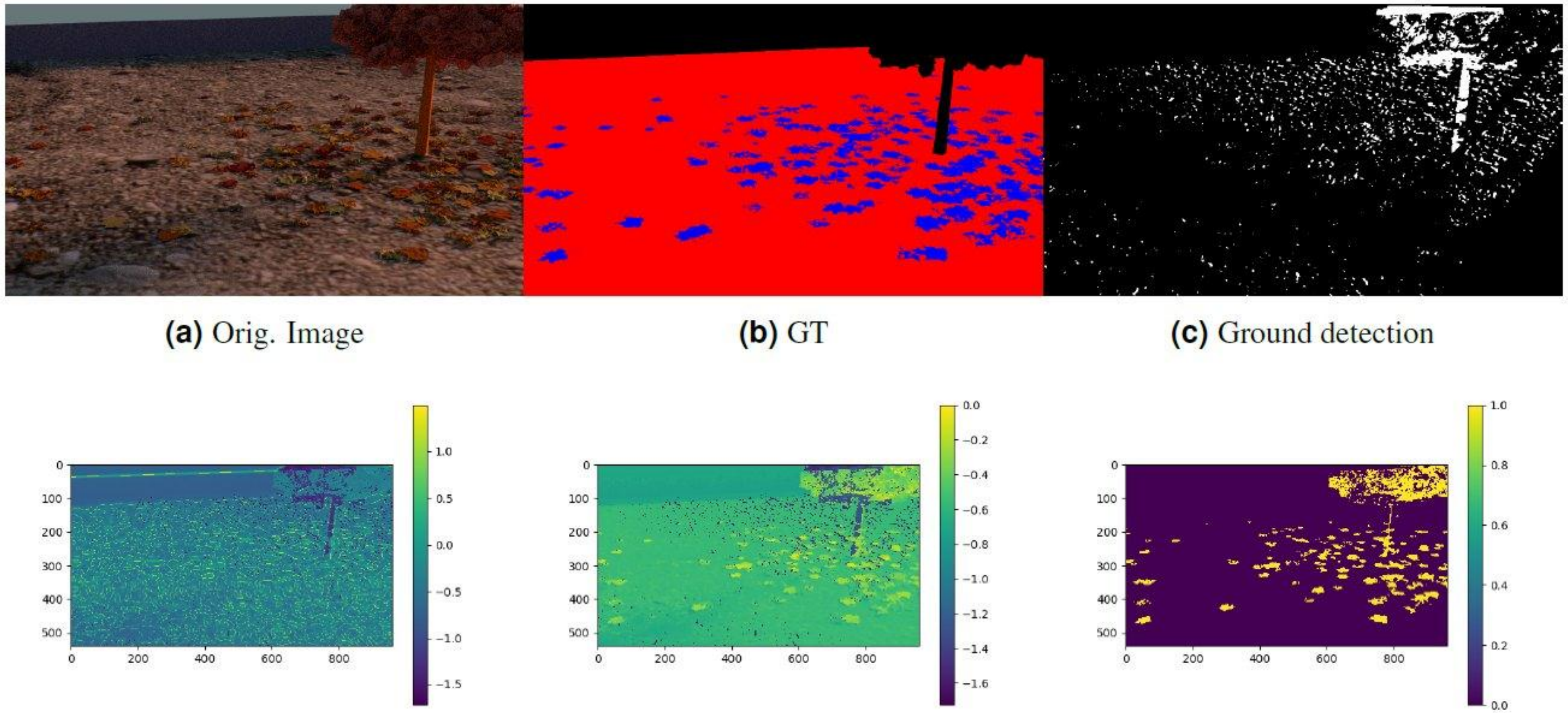
Edge detection 1/2



Edge detection 2/2



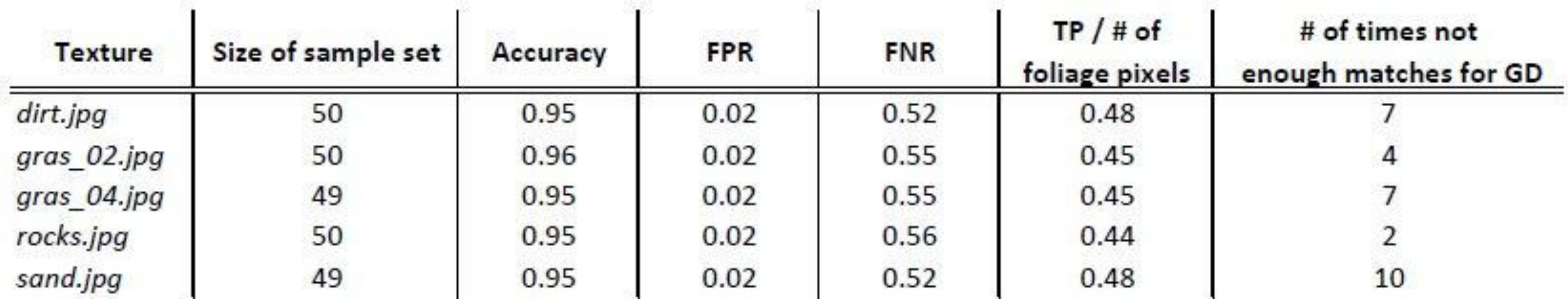
Results 1/2



(a) Orig. Image **(b)** GT **(c)** Ground detection

(d) inverted GD + color + edges + corner **(e)** inverted GD + color **(f)** thresholded, thr=-0.4

Results 2/2



Texture	Size of sample set	Accuracy	FPR	FNR	TP / # of foliage pixels	# of times not enough matches for GD
dirt.jpg	50	0.95	0.02	0.52	0.48	7
gras_02.jpg	50	0.96	0.02	0.55	0.45	4
gras_04.jpg	49	0.95	0.02	0.55	0.45	7
rocks.jpg	50	0.95	0.02	0.56	0.44	2
sand.jpg	49	0.95	0.02	0.52	0.48	10

dirt.jpg

gras_02.jpg

gras_04.jpg

rocks.jpg

sand.jpg

Discussion

- Ground detection (GD):
  - GD after color transformation
  - Imprecise warping
  - GD on both images and comparing them
- Color transformation:
  - Difficulty of fixing threshold
  - Handling of uncertainty not implemented yet
- Edge detection:
  - Entropy higher than expected
  - Contour filling not implemented yet

Thank you for your attention!