Instance Segmentation of Microscopy Cells Using Marker-Based Watershed

Your Name (Student ID: XXXXXXX)
Sabanci University, CS419 Digital Image and Video Analysis
Email: your.email@sabanciuniv.edu

***Abstract*—This report presents a non-deep-learning instance segmentation pipeline for microscopy cell images. The goal is to output a grayscale label image where each cell is assigned a unique integer value and background is set to zero. We adopt a marker-based watershed transform with distance-transform-driven markers to handle touching cells. We also analyze the effect of key parameters (thresholding, morphology, marker extraction) on segmentation performance measured by pixel-level F1 and IoU.**

***Index Terms*—segmentation, watershed, distance transform, microscopy, instance labeling**

I. Task Definition

Given a microscopy color image, the required output is a grayscale image of the same size in which each cell has a different scalar label (1..N) and the background is 0. The method must be implemented from scratch and must rely on at least one segmentation technique treated in class, without using deep learning.

II. Method Selection and Rationale

A. Chosen Method

We choose **marker-based watershed** as the core segmentation approach, combined with the **distance transform** to resolve *touching objects*. The watershed transform is a morphological segmentation tool that floods a topographic surface from minima. When applied directly on gradient images it tends to over-segment due to many local minima; marker-based watershed addresses this by constraining the flooding sources to user-defined markers, making the number of markers correspond to the number of regions.

B. Why Marker-Based Watershed Fits Cell Images

Microscopy cell images often contain: (i) non-uniform intensity, (ii) small debris/noise, (iii) clusters of touching cells. The distance transform turns a binary foreground mask into a smooth surface that peaks at cell centers, providing reliable internal markers. Watershed then places boundaries along high-gradient ridges between markers, which is well-suited for separating adjacent cells.

C. Why Other Methods Were Not Chosen

We considered the main categories covered in the course segmentation lecture:

- **Histogram/thresholding-only methods:** Fast, but extremely sensitive to illumination and threshold choice, and do not separate touching cells reliably.
- **Clustering (k-means, mean-shift):** k-means requires selecting K (often unknown), and both approaches can ignore spatial continuity unless coordinates are added; they can also merge nearby cells with similar colors.
- **Region growing/splitting:** Requires seed selection and similarity criteria; results can be unstable under noise and intensity variation, and splitting (quadtree) can be blocky.
- **Graph partitioning (normalized cuts):** Robust but computationally heavy and requires decisions about stopping criteria and graph construction.

Given the dataset difficulty and the need to separate touching cells, marker-based watershed provides a strong accuracy/computation trade-off.

III. Pipeline

Our end-to-end pipeline is summarized below.

A. Automatic Image Type Detection

To handle diverse microscopy image types (e.g., bright cells on dark background vs. dark cells on bright background), we implement an automatic invert detection mechanism. The algorithm analyzes the histogram of the feature channel after preprocessing:

1) Compute the intensity histogram of the feature image.
2) Calculate the ratio of dark pixels (intensity < 85) and bright pixels (intensity > 170).
3) If dark pixels dominate ($> 30\%$ of total and exceed bright pixels), the image likely has a dark background with bright cells, so we set the invert parameter to false.
4) Otherwise, we assume bright background with dark cells and set the invert parameter to true (default).

This automatic detection eliminates the need for manual parameter tuning when processing mixed datasets and ensures robust performance across different microscopy imaging conditions.

TABLE I
KEY PARAMETERS OF THE WATERSHED PIPELINE.

Parameter	Effect / trade-off
Feature channel	Drives separability of cells vs background
Auto invert detection	Automatically determines if cells are brighter/darker than background
Invert threshold	Manual override if auto-detection fails
CLAHE (clip, tile)	Enhances local contrast; too high amplifies noise
Blur kernel	Reduces noise; too high removes thin boundaries
Threshold method	Otsu (global) vs adaptive (uneven illumination)
Morph kernel, open/close iters	Removes noise / fills holes; too aggressive erodes cells
Distance threshold α	Controls number/size of internal markers
Background dilation iters	Controls unknown region size
Min area	Removes spurious tiny segments

TABLE II
EXAMPLE RESULTS OVER 10 IMAGES (FILL AFTER RUNNING).

Image	F1 (%)	IoU
img_01	–	–
img_02	–	–
⋮	⋮	⋮
Mean	–	–

B. Preprocessing and Foreground Mask

1) Convert image to an informative single-channel feature map (default: grayscale; optional: LAB/HSV channels).
2) Contrast enhancement using CLAHE (optional).
3) Denoising with Gaussian blur.
4) **Automatic invert detection** (if enabled): Analyze histogram to determine whether cells are brighter or darker than background.
5) Foreground/background separation via Otsu thresholding (or adaptive thresholding for uneven illumination), using the automatically detected or manually specified invert setting.
6) Morphological opening to remove small noise; optional closing to fill small holes.

C. Marker Extraction via Distance Transform

1) Compute the distance transform on the binary foreground mask.
2) Compute *sure foreground* by thresholding the distance map:

$$\text{sure\_fg} = \mathbf{1}\left(D(x,y) > \alpha \cdot \max(D)\right)$$

where $\mathbf{1}(\cdot)$ is the indicator function and α is a tunable parameter.
3) Compute *sure background* by dilating the foreground mask.
4) Define the *unknown region* as sure_bg minus sure_fg.
5) Create connected-components markers from sure_fg and set unknown pixels to 0.

D. Watershed and Instance Label Output

We apply watershed on the original image with the computed markers. The output is converted into an instance label image: background $\to 0$, and each region label $\to 1..N$. Tiny regions are filtered and remaining labels are re-indexed to be consecutive.

IV. Parameters and Their Effects

Table I lists the main parameters in our implementation. Their effects are summarized afterwards.

A. Most Sensitive Parameters

Distance threshold α: Low α produces many markers (risk of over-segmentation). High α may miss markers in weak/low-contrast cells (under-segmentation).

Morphological opening: Too little opening keeps debris (false positives). Too much opening breaks thin cell regions (false negatives) and reduces marker quality.

Thresholding mode: Otsu works best under relatively consistent illumination. Adaptive thresholding can help when illumination varies, but may create fragmented masks in texture-heavy regions.

B. Recommended Starting Values

Based on qualitative validation and standard practice for distance-transform watershed:

- $\alpha \in [0.40, 0.55]$ (default 0.45)
- Morph kernel size $k \in \{3, 5\}$, opening iterations 1–3
- CLAHE clipLimit $\in [1.5, 3.0]$, tile size 8
- Blur kernel 3–7
- Min area 100–300 pixels (dataset dependent)

The truly optimal values should be selected by grid search on a small validation subset (e.g., 10 images), maximizing pixel-level F1 and IoU.

V. Evaluation Protocol

We report pixel-level (binary) F1 and IoU by comparing predicted foreground ($\hat{Y} > 0$) with ground-truth foreground ($Y > 0$). For each test image, we compute:

$$\text{IoU} = \frac{TP}{TP + FP + FN}, \quad F1 = \frac{2TP}{2TP + FP + FN}.$$

Note: The assignment requires instance-labeled outputs; however, the provided evaluation metric is pixel-level, so we treat the task as foreground-vs-background for scoring.

VI. Results and Discussion

A. Quantitative Results

Insert a table here summarizing results for at least 10 images.

B. Qualitative Results

Include a few example figures of success and failure cases.

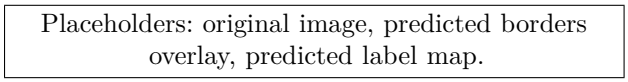

Fig. 1. Qualitative examples (add your own visuals).

C. Failure Modes

Common failure modes we observed (and what to do):

- **Severe under-segmentation:** Markers missing. Lower α, reduce opening, try a different feature channel (e.g., LAB-A or HSV-S).
- **Over-segmentation:** Too many markers due to noise. Increase blur, increase opening, increase α, reduce CLAHE strength.
- **Background leakage into cells:** Thresholding failed due to illumination. Switch to adaptive thresholding and/or apply background correction.

VII. Conclusion

Marker-based watershed with distance-transform markers provides a practical and explainable non-DL solution for microscopy cell instance segmentation, especially when touching cells are common. The method's performance is governed mainly by marker quality; thus, parameter tuning should focus on thresholding, morphology, and the distance threshold controlling internal markers.

Reproducibility

All code is provided in the file `watershed_cellseg.py`. For batch evaluation:

```
python watershed_cellseg.py --img_dir images/ --label_dir labels/ \
  --out_dir preds/ --eval
```

The automatic invert detection is enabled by default. To disable it and use manual settings, use `--no_auto_invert` along with `--invert` or `--no_invert`.