

Medium

 Search

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



語意切割過渡演算法



WZX

10 min read · Just now



Share



More

Algorithmn 流程

輸入：

- 起始語義圖 A (semantic_map_a_np)
- 結束語義圖 B (semantic_map_b_np)
- 總插值幀數 (num_total_frames)
- 類別總數 (num_classes)
- 形態學操作配置 (針對出現、消失、形變)
- 插值參數 (base_threshold, presence_factor)

輸出：

- 插值後的語義圖序列 (interpolated_map_sequence)

步驟：

```
對於`每一幀` (從 0 到 num_total_frames - 1): # 外層迴圈
  a. 算當前幀的插值因子 `alpha = k / (num_total_frames - 1)` (alpha從 0.0 變化到 1)
  b. 對於每一個類別 `class_id` (從 0 到 num_classes - 1) # 內層迴圈
    i. 提取類別 `class_id` 在圖 A 中的二值遮罩 `mask_A_c`
    ii. 提取類別 `class_id` 在圖 B 中的二值遮罩 `mask_B_c`
    iii. 判斷該類別的過渡狀態：
```

出現 (Appear): 如果 `mask_A_c` 為空, 但 `mask_B_c` 存在
 消失 (Disappear): 如果 `mask_A_c` 存在, 但 `mask_B_c` 為空
 形變/位移 (Morph): 如果 `mask_A_c` 和 `mask_B_c` 都存在。
 都不存在 (Absent): 如果 `mask_A_c` 和 `mask_B_c` 都為空 (跳過此類別)

iv. 核心插值步驟 (距離變換 DT):

出現:
 計算 `mask_B_c` 的帶符號距離變換 `dt_b`
 計算當前閾值 `threshold_appear = base_threshold + (1 - alpha) * presence`
 從底部中心擴展的 `reveal_limiter_mask` (大小隨 `alpha` 變化)
 插值遮罩 `interpolated_mask_for_class = grown_mask & reveal_limiter_mask`
 消失:
 計算 `mask_A_c` 的帶符號距離變換 `dt_a`
 計算當前閾值 `threshold_disappear = base_threshold + alpha * presence`
 插值遮罩 `interpolated_mask_for_class = (dt_a >= threshold_disappear)`
 形變/位移:
 計算 `mask_A_c`, `mask_B_c`
 線性插值距離場: `interpolated_dt = (1 - alpha) * dt_a + alpha * dt_b`
 都不存在:
 `interpolated_mask_for_class` 為空

v. 數學形態學操作:

1. 根據過渡狀態 (出現/消失/形變) 選擇對應的形態學操作配置
2. 配置存在且 `interpolated_mask_for_class` 非空:
 - 根據 `alpha` 和配置中的 "early"/"late" 標記調整操作強度 (如腐蝕/擴張)
 - 將形態學操作應用於 `interpolated_mask_for_class`

vi. 將處理後的 `interpolated_mask_for_class` 中為 True 的像素, 在 `current_interpolated_map` 添加到 `interpolated_map_sequence` 列表

c. 將 `current_interpolated_map` 添加到 `interpolated_map_sequence` 列表

返回 `interpolated_map_sequence`

Distance Transform(距離變換)

實現出現、消失、位移的核心機制

DT (距離變換值)

對於二值圖像中的每個像素, 計算它到最近的背景像素 (或前景像素) 的距離。

`scipy.ndimage.distance_transform_edt` 基於歐幾里得距離的精確距離變換 (Euclidean Distance Transform, EDT)。

```
dt = distance_transform_edt(mask) —
(distance_transform_edt(np.logical_not(mask)) — 1)
```

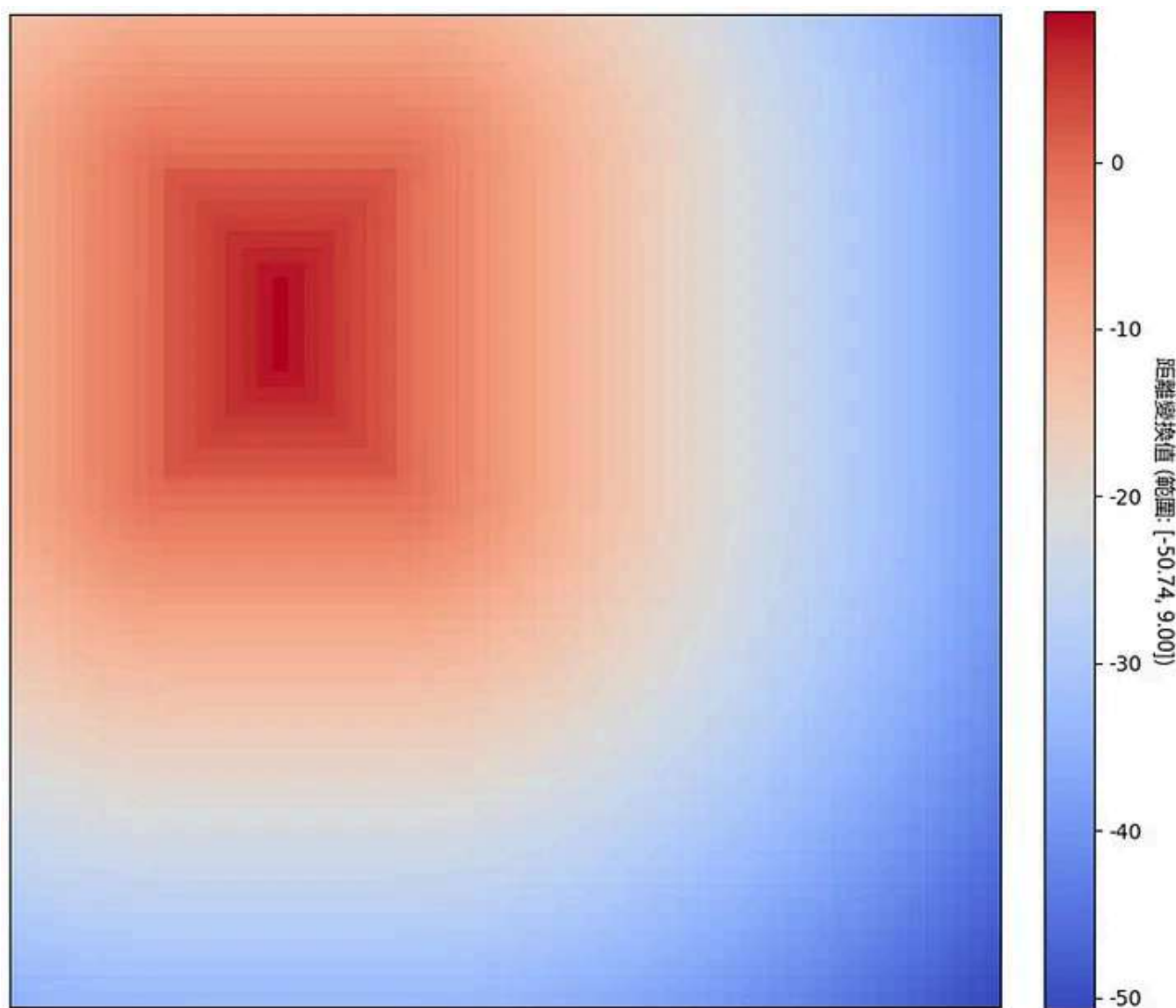
`distance_transform_edt(mask)`: `mask` 中為 True (前景) 的像素到最近的 False (背景) 像素的距離。對於 `mask` 內部深處的點, 這個值會比較大。

`distance_transform_edt(np.logical_not(mask))`：計算原始 `mask` 的外部點（新前景）到最近的原始 `mask` 邊緣（新背景）的距離。對於 `mask` 外部越遠的點，這個值會比較大。

減去 1 為了讓邊界上的點距離為 0 (而不是 1)

dt:

- **正值**：像素位於原始遮罩的**內部**。越遠離原始邊緣，正數就越大
- **零值**：像素恰好位於原始遮罩的**邊緣**
- **負值**：像素位於原始遮罩的**外部**。越遠離原始邊緣，數值本身越小



DT Threshold(距離變換閾值)

對每一個像素計算距離變換，如果該像素的 DT 值 \geq DT Threshold，該像素就被認為是當前插值幀中的**前景**，該像素的 DT 值 $<$ DT Threshold 就是**背景**

出現

$$\text{threshold} = \text{base_threshold} + (1 - \alpha) * \text{presence_factor}$$

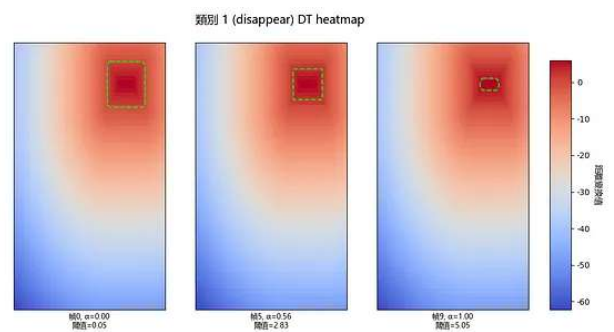
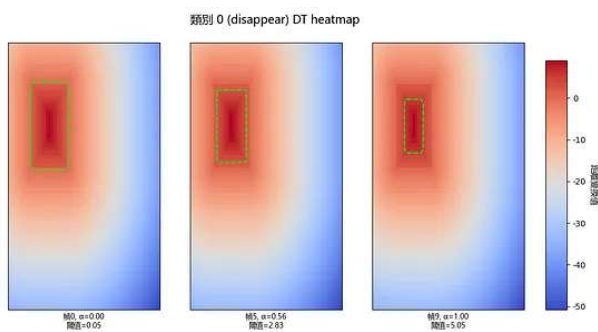
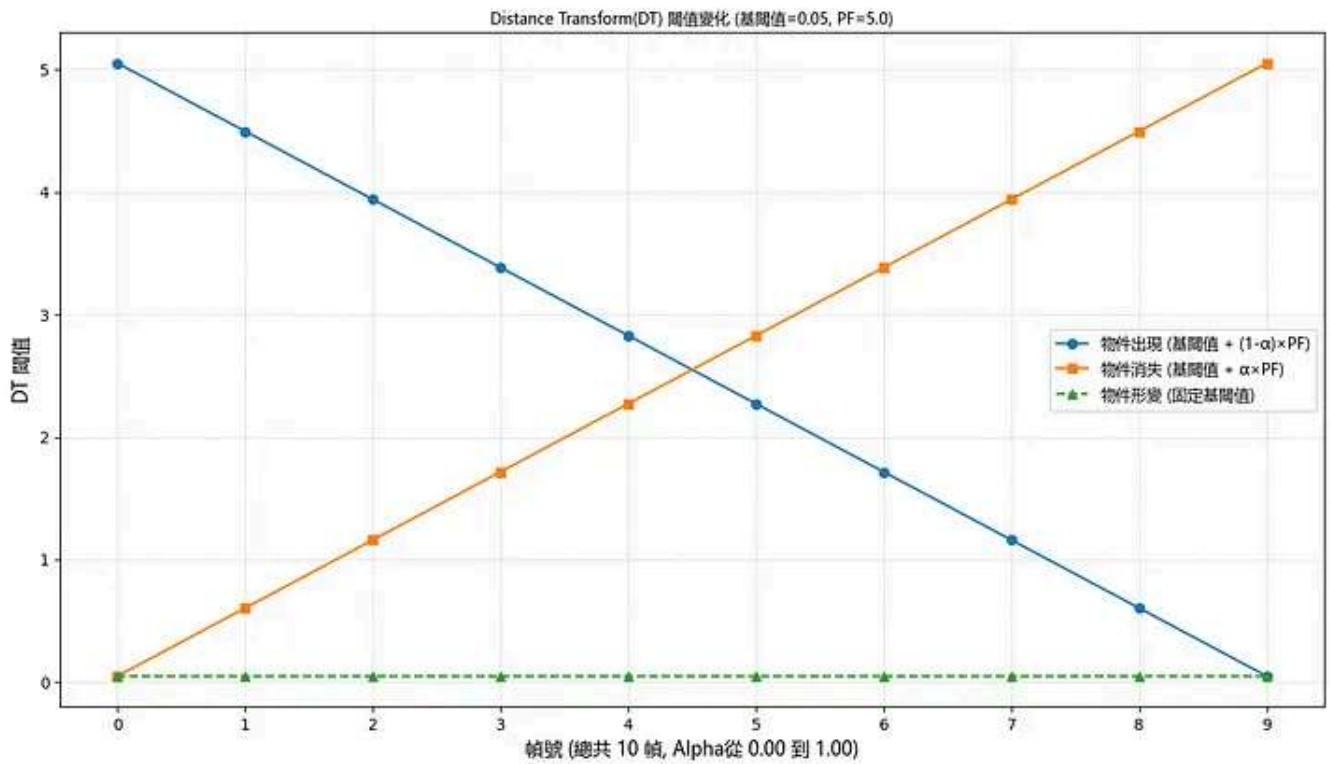
消失

$$\text{threshold} = \text{base_threshold} + \alpha * \text{presence_factor}$$

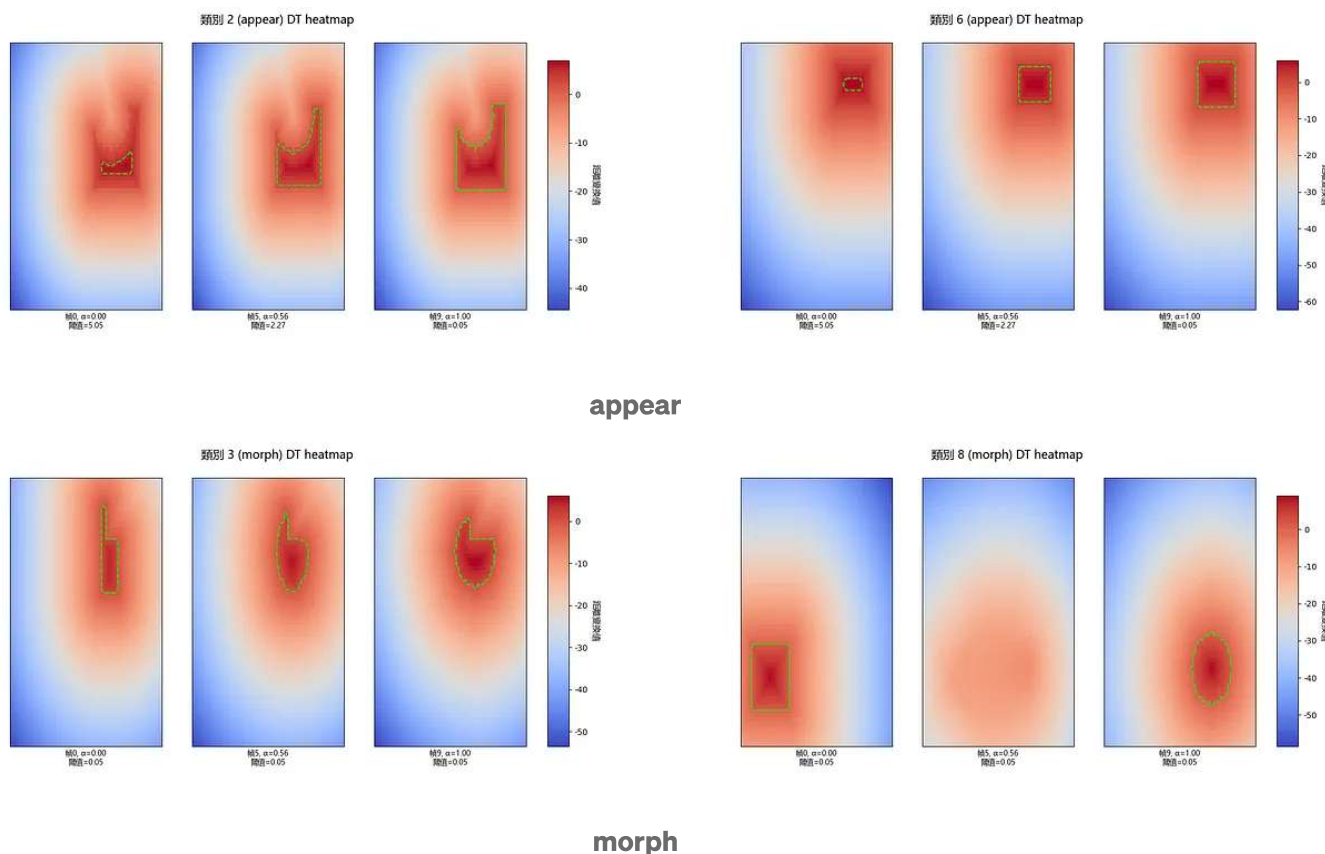
形變

$$\text{threshold} = \text{base_threshold}$$

$$\text{interpolated_dt} = (1 - \alpha) * \text{dt_a} + \alpha * \text{dt_b}$$



Disappear



Alpha

插值進度，從 0 變化到 1，初始圖為0，終止圖為1

Presence Factor(存在因子)

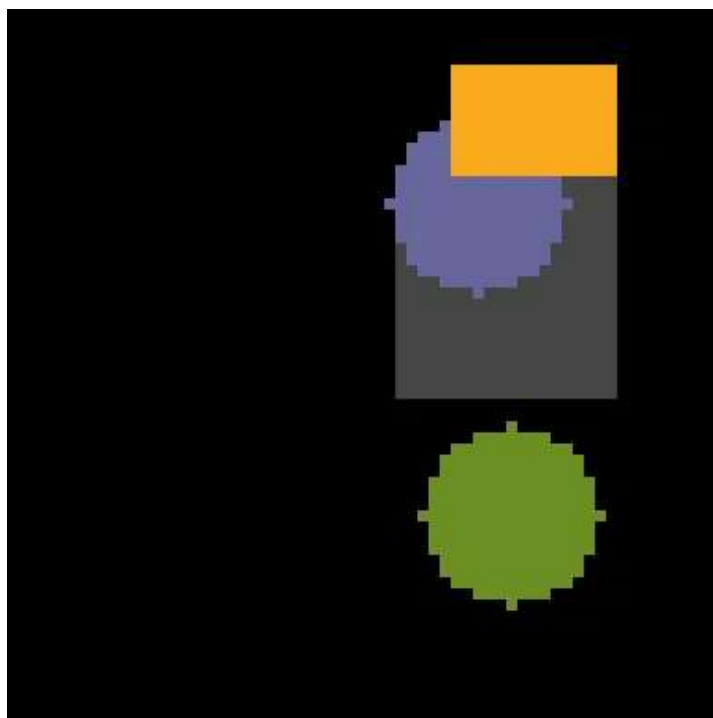
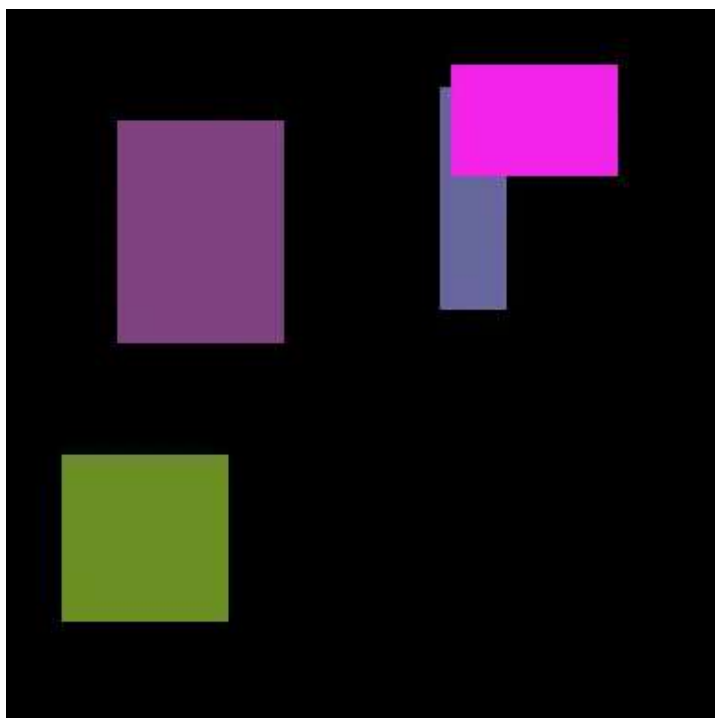
主要控制在物件**出現**和**消失**這兩種過渡狀態下，需要手動調整(根據公式計算出的dt設定)

出現：高 PF -> 從小點開始，低 PF -> 初始就較大

消失：高 PF -> 徹底消失，低 PF -> 消失不徹底

Base Threshold(基閾值)

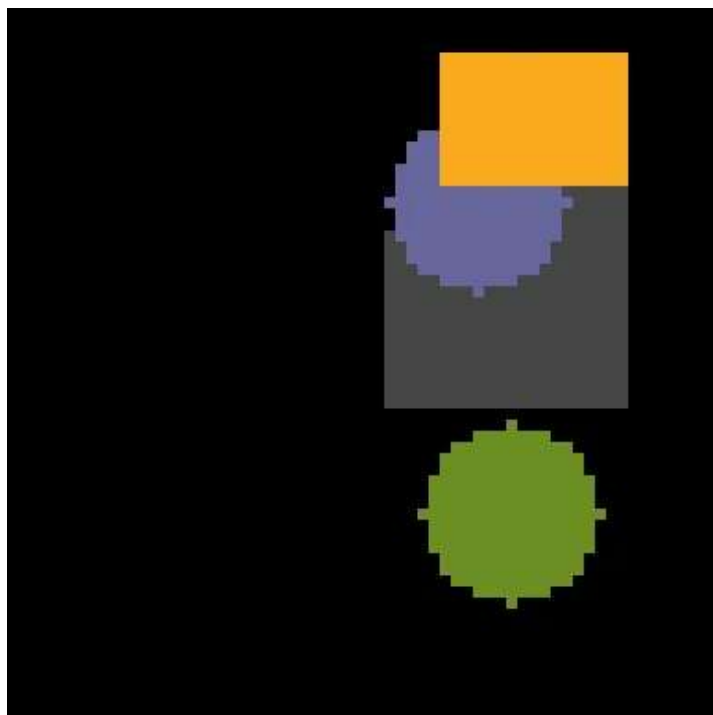
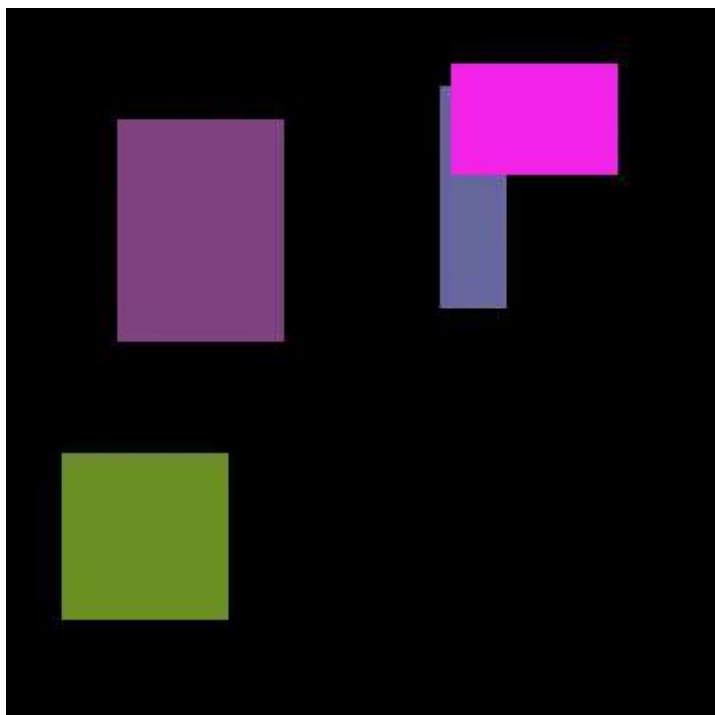
前面公式在原始遮罩的**緊鄰邊緣的那一圈像素**上，計算出的 DT 值也可能為0，從而被 ≥ 0 的閾值條件包含進來，造成了輕微的**膨脹**效果



初始圖 中止圖

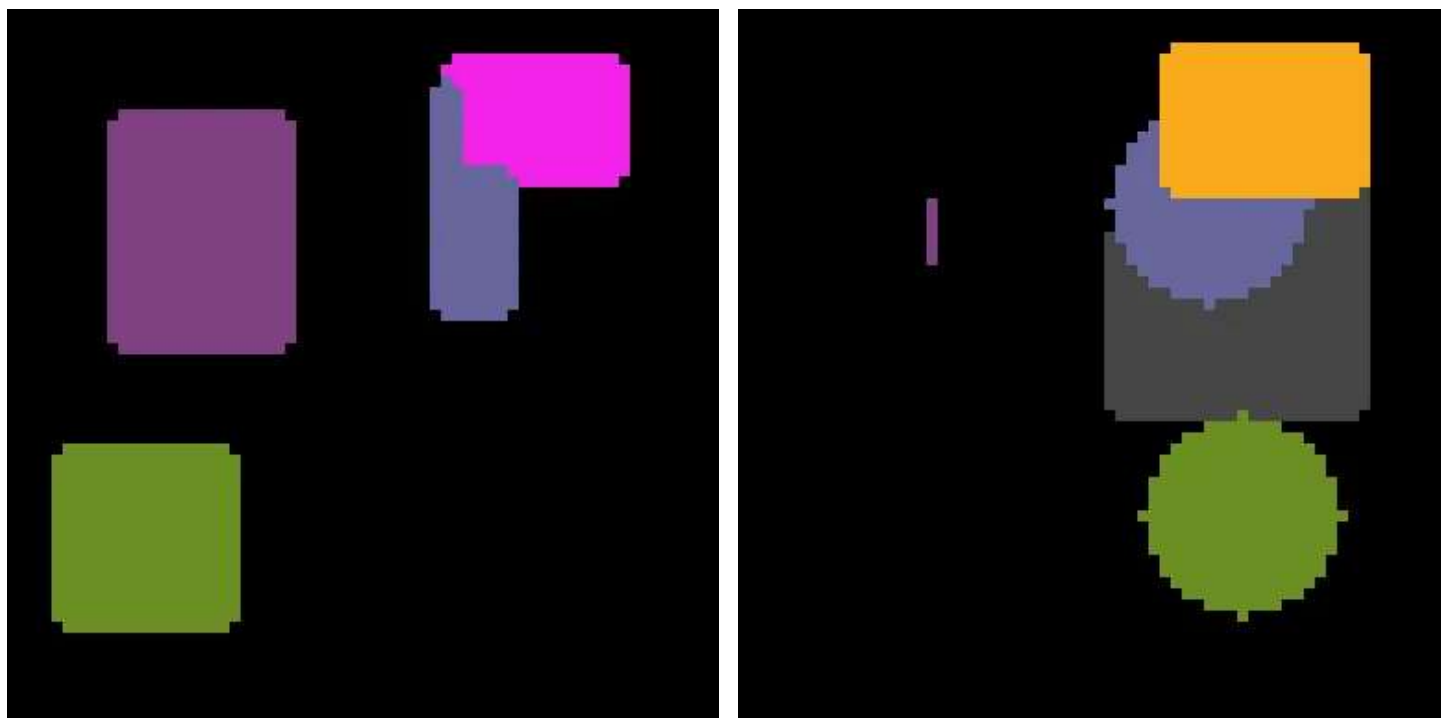
設置非常小的正數

Base Threshold = 0.05



最早和最晚的幀

Base Threshold = 0.00



最早和最晚的幀

Mathematical Morphology(數學形態學)

形狀之間的平滑過渡改善質量、增強視覺效果和處理細節

出現(Appear)

`erode_early` (早期侵蝕) + `dilate_late` (後期擴張)

`erode` 會去除物件邊緣的像素，特別是細小的突出部分或獨立的噪點像素。讓物件的初始形態更乾淨，避免出現一些零散的、不自然的像素點。

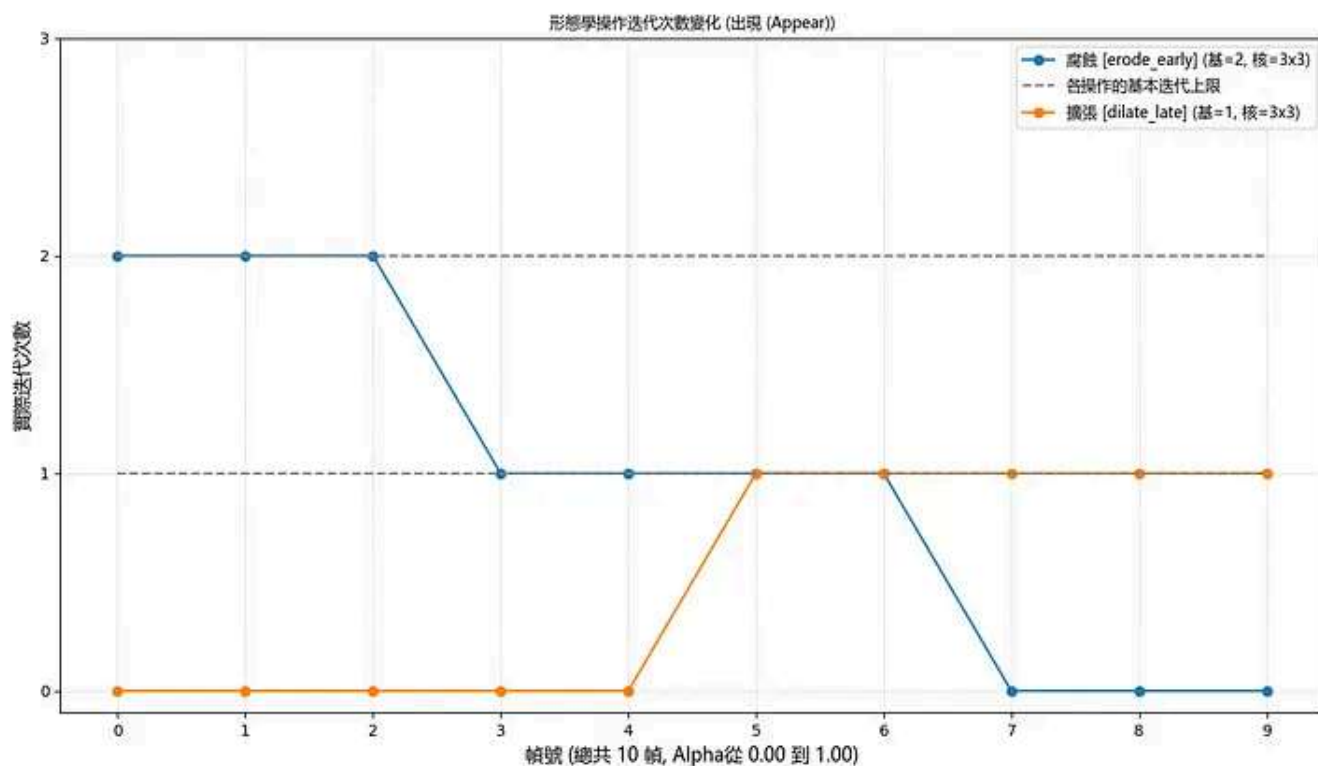
在早期 `erode` 掉一些噪點和細小部分後，後期的 `dilate` 可以幫助物件恢復其應有的飽滿度，使得之前被 `erode` 的部分能夠長回來，並且使物件的整體形狀更平滑。隨著 `alpha` 增加，`erode_early` 的影響**減弱**，而 `dilate_late` 的影響**增強**。

類似形態學中的開運算 (Opening)

去除小的物件（噪點）和平滑物件的輪廓，同時盡量保持大物件的整體形狀和大小。

視覺動態：

erode_early，讓物件在最初出現時顯得**更小、更纖細**，即使 DT 插值本身已經產生了一個較小的初始形態，erode_early 可以進一步強化這種**從小到大的**視覺效果。



消失(Disappear)

erode_late (後期侵蝕)

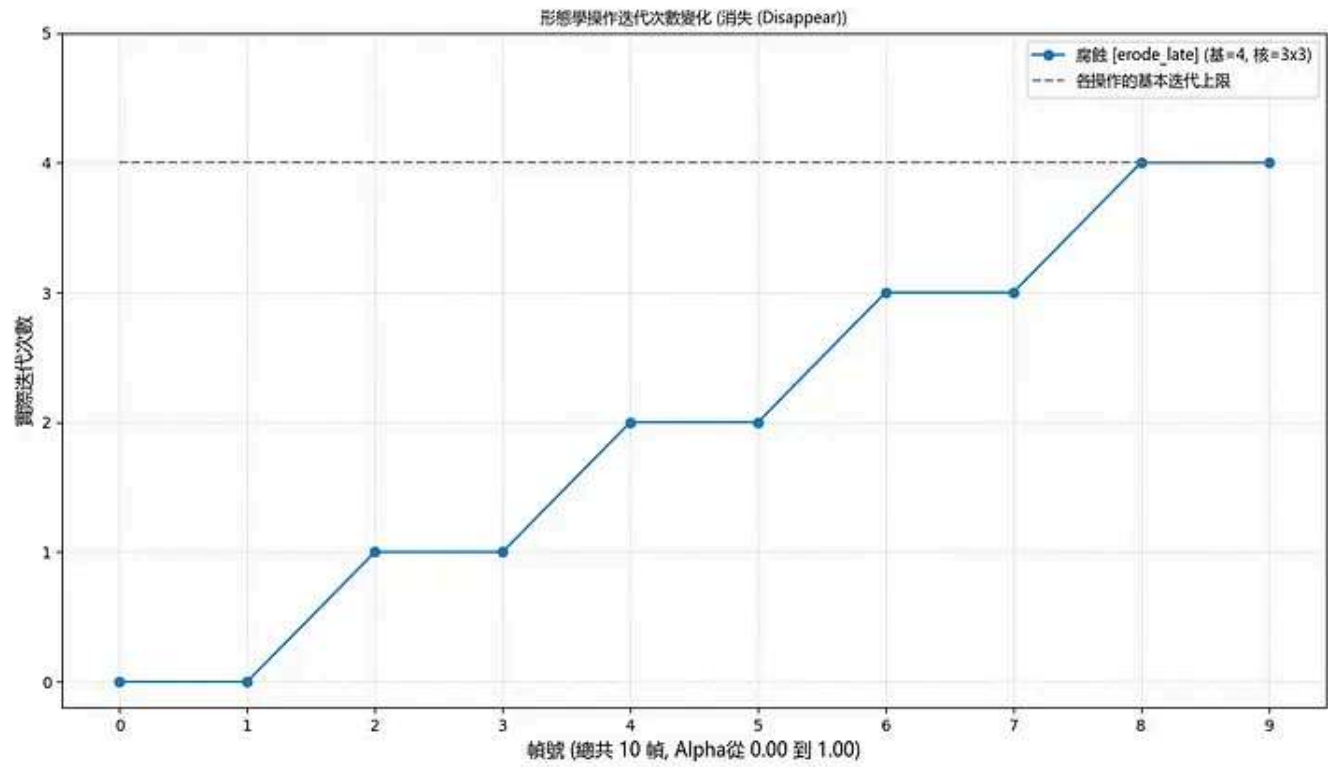
在物件消失過程的初期（alpha 較小，迭代次數接近0），腐蝕操作的影響很小或幾乎沒有。此物件的形態主要由距離變換 (DT) 插值的閾值升高而導致的自然縮小決定。

隨著 alpha 逐漸增大並趨近於 1（物件即將完全消失），erode_late 的強度（迭代次數）顯著增加，達到其配置的最大值。

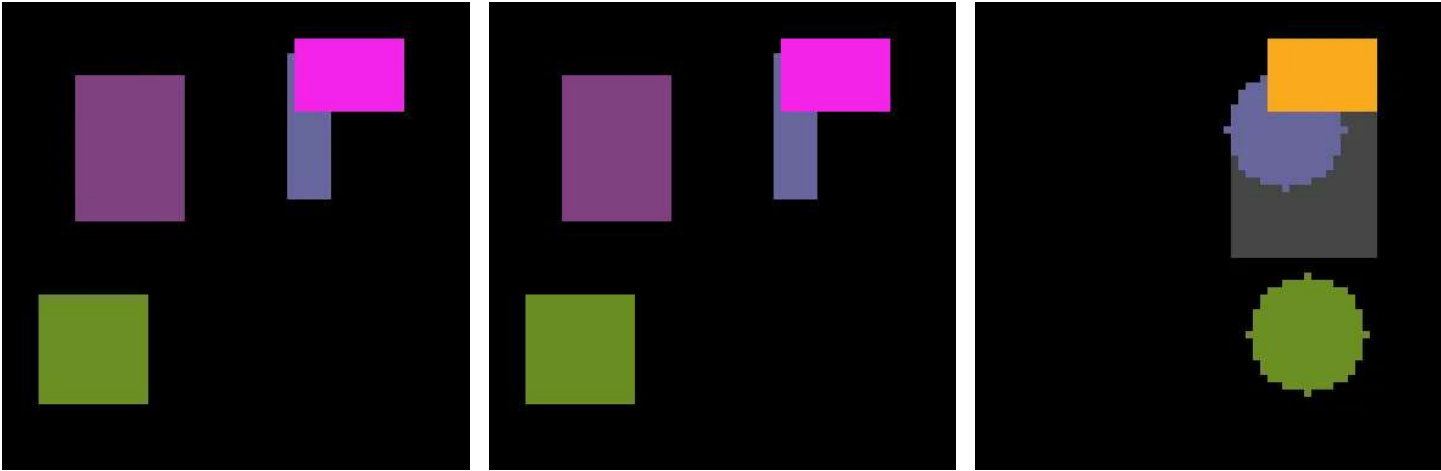
後期增強的強烈侵蝕，用於**確保物件能夠徹底消失**。侵蝕包括由於DT插值閾值不夠高而未能完全移除的物件核心區域（DT值較高的部分）。避免在動畫的最後幾幀留下微小的餘像素點。

視覺動態：

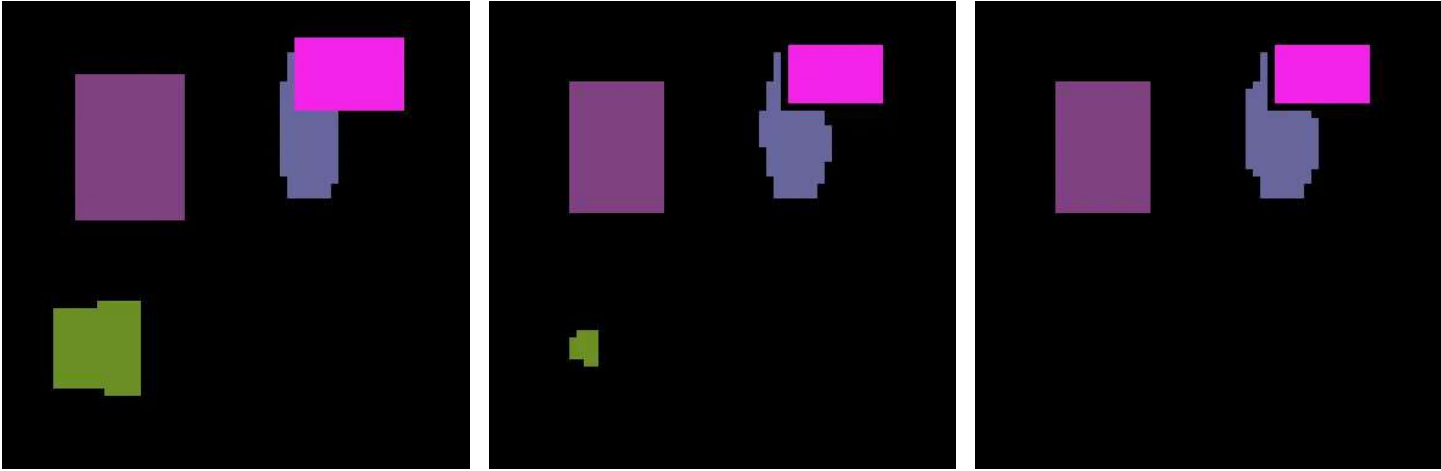
物件在消失的初期，形態變化較為平緩，主要表現為 DT 閾值造成的縮小。當 alpha 變大時，erode_late 效果**加劇**，物件的收縮速度明顯加快，直至完全從畫面上消失。

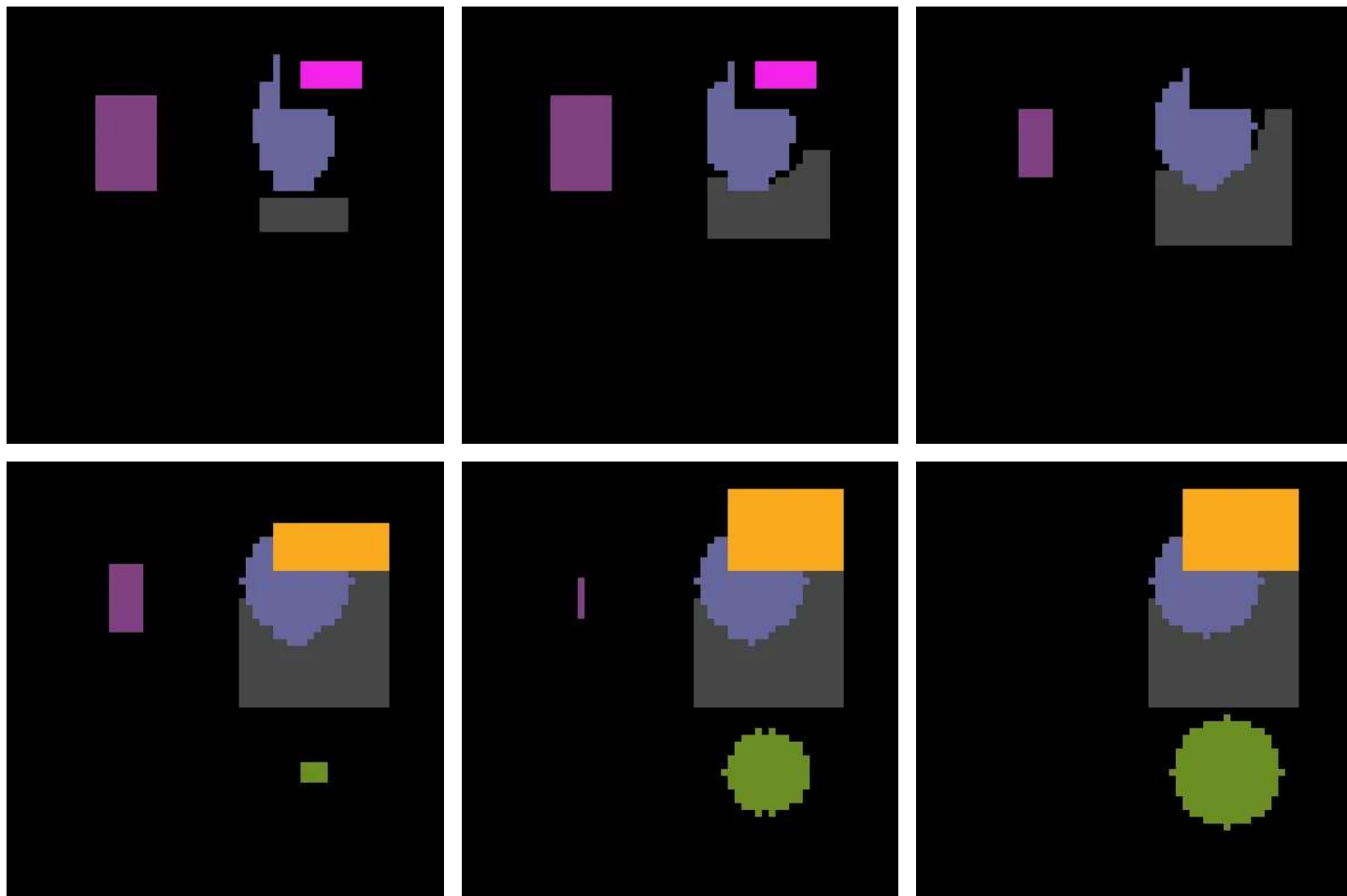


簡單範例



左:初始圖 中:動畫 右:終止圖

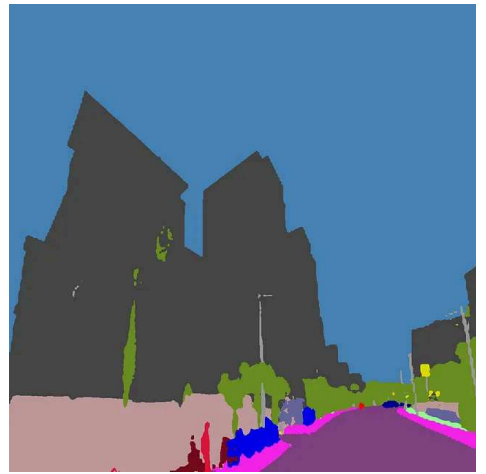
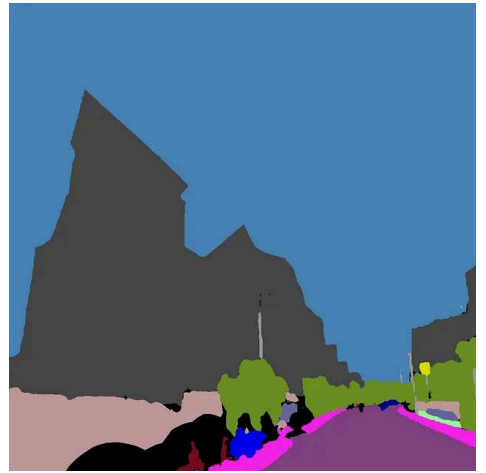
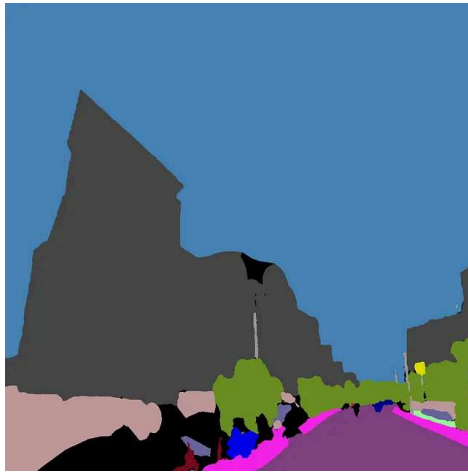




實作



左:初始圖 中:動畫 右:終止圖



Distance Transform

Mathematical Morphology

Semantic Segmentation

Algorithms



Edit profile