Team 10 Final

徐近庭 李欣怡 洪郡辰

Paper Title & Motivation

選題：

《“GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts》，作者為Carsten Rother、Vladimir Kolmogorov及Andrew Blake。

動機：

在作業三實作的k-means演算法讓我們聯想到圖片編輯時很常會使用到的一項技術—去背。因為我們常常只需要前景的物件，而不希望保留圖片的背景，所以在人工選取位置進行去背的做法中，如果能有個聰明的演算法，讓每次更新過後的圖片能更接近使用者想要的，這樣去背的過程將會變得更有效率。

Problem Definition

問題描述：

給定一張圖片後，能透過簡單標註大致分出前後景，再依據使用者需求調整直至使用者滿意區分結果。

處理方式：

為了達成此目標，我們找到grab-cut，它延伸自去背演算法 graph-cut ，優化graph-cut的處理步驟及結果。

Algorithm - 特點

GrabCut 的核心為 iterative estimation 和 incomplete labelling。

每次迭代時都會進行 graph cut optimization，利用顏色、位置、GMM參數來區分每個像素是對應到前景還是後景，同時使用smoothness term，提高相鄰近的像素被分在同一類別的機率。

在 完 成 iterative estimation 後 ， 使用者只要在每次迭代結果上簡單label出background 中想設為前景的部分、或foreground中想設為後景的地方，就可以依此進行下一輪的update，直到使用者滿意最後生成的結果。

Algorithm – 步驟

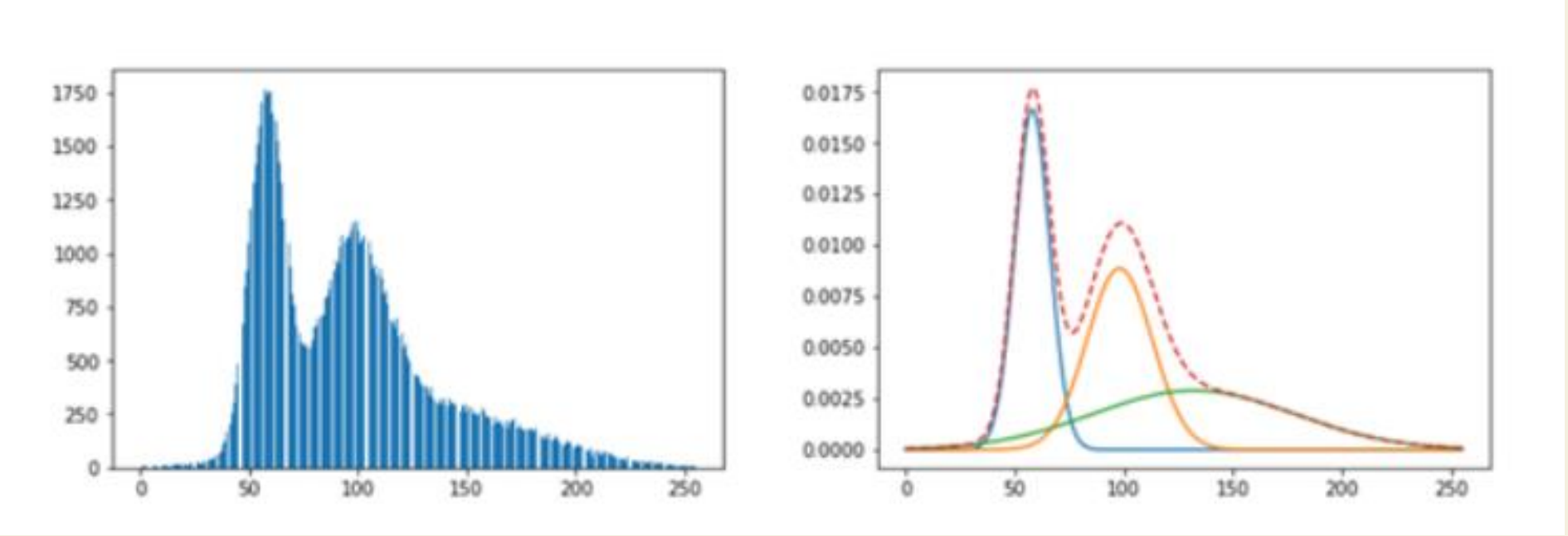
1. 初始化：使用者透過簡單的手繪標記圖像的前景和背景區域，比如用一個矩形涵蓋前景區域，或在圖像中手動標記前景和背景。
2. 前景背景模型估計：先假設前景和背景的像素符合高斯分佈，而後基於用戶的初始標記，使用高斯混合模型（GMM）粗估前景和背景。
3. 估計圖割：通過最小化能量函數（energy minimization)，將圖像分 割為前景和背景。
4. 迭代更新：不斷更新模型和圖割，直到energy收斂。
5. 手動優化:使用者能根據需求進行labeling優化目前結果。

initial

Initialisation

- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and $\alpha_n = 1$ respectively.

Gaussian Mixture Model(GMM)



Expectation-Maximization

E-step:根據data及GMM參數計算每個data屬於每個高斯分布的機率。

M-step:重新計算GMM參數，使得data分配到某個高斯分布的機率能夠最大化。

iterative estimation

Iterative minimisation

1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$
2. *Learn GMM parameters* from data $\mathbf{z}$:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* use min cut to solve:
$$\min_{\{\alpha_n:\ n \in T_U\}} \min_{\mathbf{k}} \mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}).$$
4. Repeat from step 1, until convergence.
5. Apply border matting (section 4).

Graph-cut

The following table gives weights of edges in $\mathcal{E}$

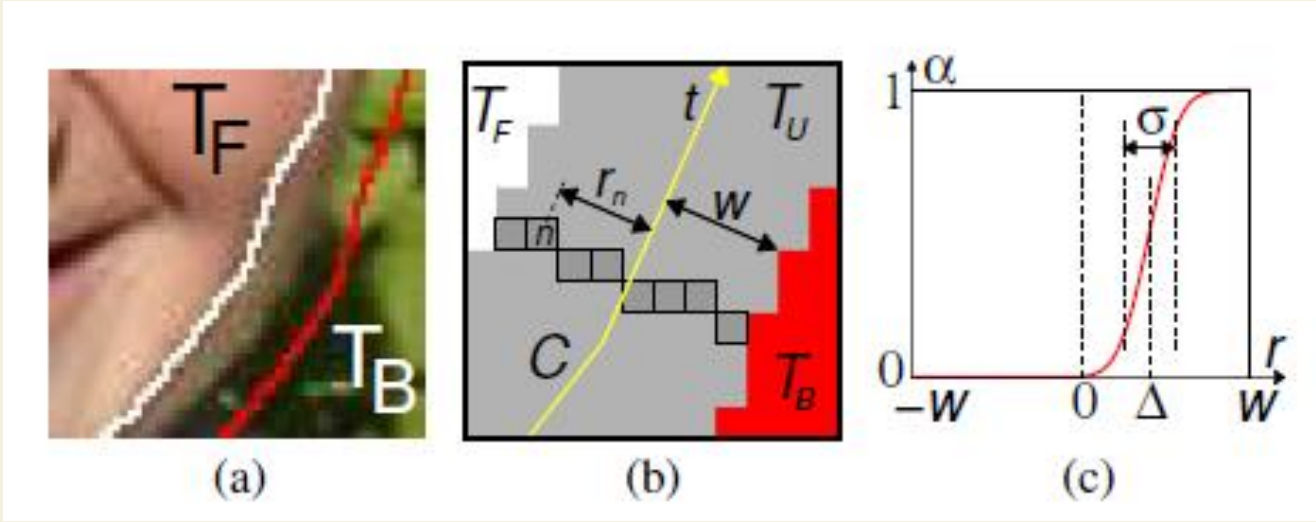
edge	weight (cost)	for
$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in \mathcal{N}$
$\{p,S\}$	$\lambda \cdot R_p(\text{“bkg”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	K	$p \in \mathcal{O}$
	0	$p \in \mathcal{B}$
$\{p,T\}$	$\lambda \cdot R_p(\text{“obj”})$	$p \in \mathcal{P},\ p \notin \mathcal{O} \cup \mathcal{B}$
	0	$p \in \mathcal{O}$
	K	$p \in \mathcal{B}$

where

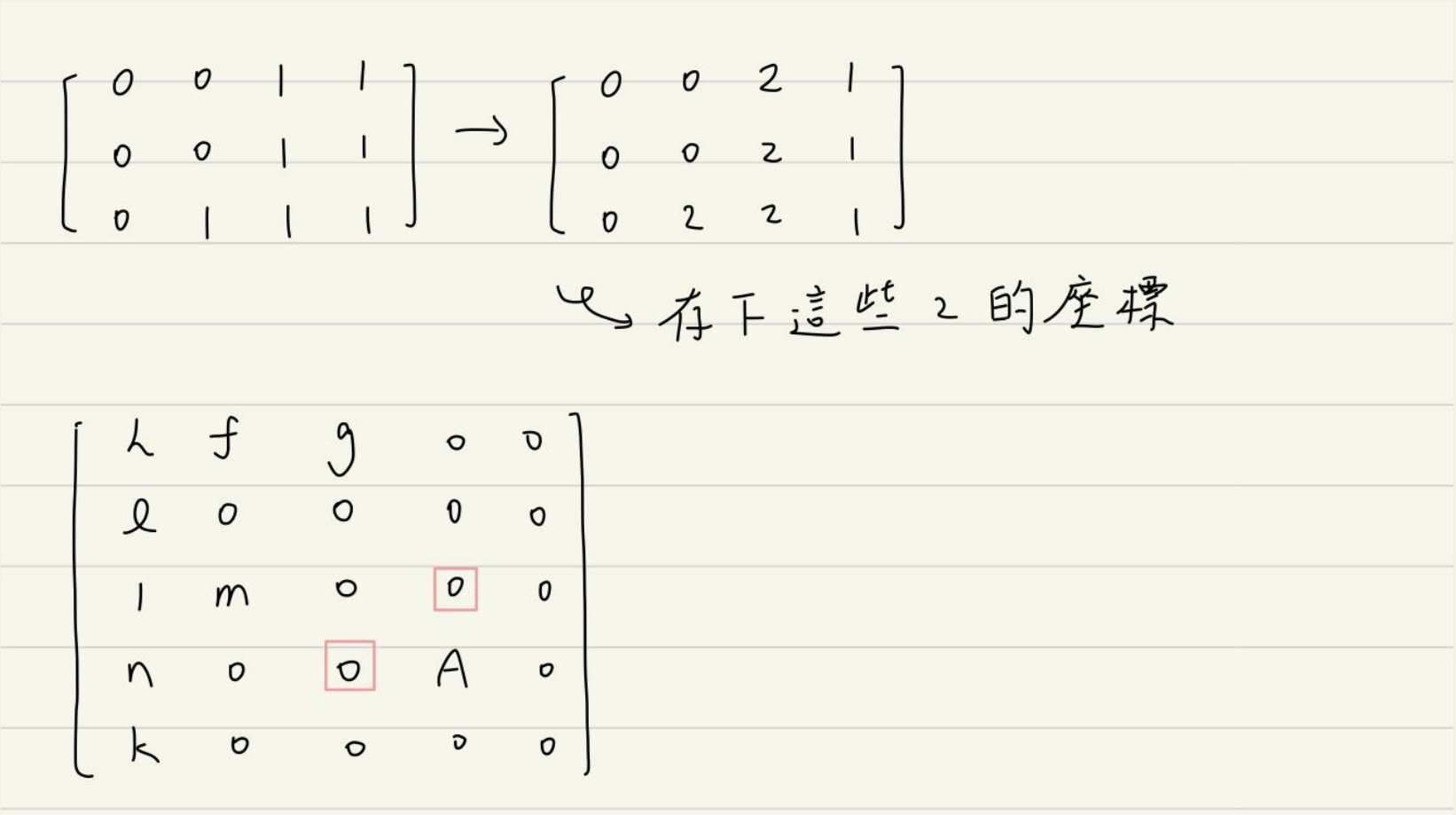
$$K = 1 + \max_{p \in \mathcal{P}} \sum_{q:\ \{p,q\} \in \mathcal{N}} B_{\{p,q\}}.$$

t-link	initial cost	add	new cost
$\{p,S\}$	$\lambda R_p(\text{“bkg”})$	$K + \lambda R_p(\text{“obj”})$	$K + c_p$
$\{p,T\}$	$\lambda R_p(\text{“obj”})$	$\lambda R_p(\text{“bkg”})$	c_p

Border Matting



Implementation



Energy Function

$$E = \sum_{n \in T_U} \tilde{D}_n(\alpha_n) + \sum_{t=1}^{T} \tilde{V}(\Delta_t, \sigma_t, \Delta_{t+1}, \sigma_{t+1}) \qquad (12)$$

Smoothing Regularizer

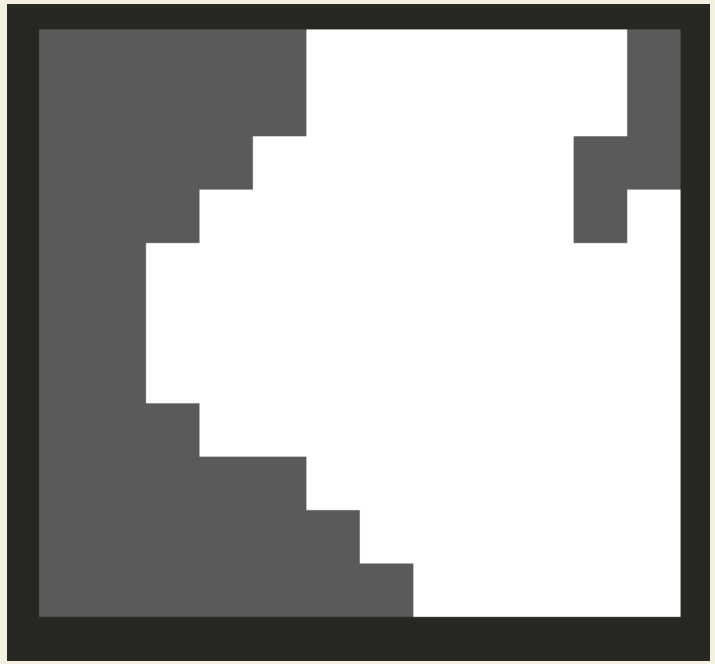
$$\tilde{V}(\Delta, \sigma, \Delta', \sigma') = \lambda_1(\Delta - \Delta')^2 + \lambda_2(\sigma - \sigma')^2, \qquad (13)$$

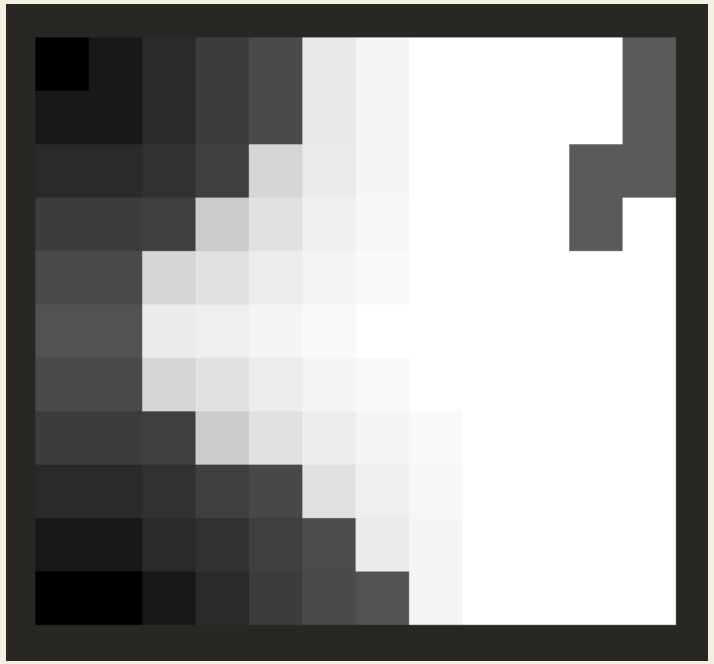
Data term

$$\tilde{D}_n(\alpha_n) = -\log \mathbf{N}\left(z_n; \mu_{t(n)}(\alpha_n), \Sigma_{t(n)}(\alpha_n)\right) \tag{14}$$

$$\begin{aligned} \mu_t(\alpha) &= (1-\alpha)\mu_t(0) + \alpha\mu_t(1) \\ \Sigma_t(\alpha) &= (1-\alpha)^2\Sigma_t(0) + \alpha^2\Sigma_t(1). \end{aligned} \tag{15}$$

Results





Results

操作方式

指令：

python grabcut.py

參數：

--img_file

需要去背的檔案路徑，預設為 sample1.png

--output_file

輸出去背圖片時使用的檔案名，預設為 output.jpg

互動方式：

使用鍵盤滑鼠互動

流程

初始化：

畫長方形

標註前後景

調整：

標註前後景

執行border matting

輸出：

結束去背並輸出圖片

流程控制

v: 畫長方形

使用滑鼠拉出一個長方形，包含所有前景物件

b: 標註前後景

使用不同按鍵切換筆刷

用滑鼠在圖片上標記

n: 確認筆刷

模型會根據標記結果進行去背

esc: 結束並輸出圖片

筆刷切換

1: 背景(background)，黑色

2: 前景(foreground)，白色

3: 可能是背景(possiable background)，藍色

4: 可能是前景(possiable foreground)，紅色

<: 縮小筆刷

>: 放大筆刷

其他按鍵

s: 儲存當前繪圖結果

z: 回到上次儲存的結果

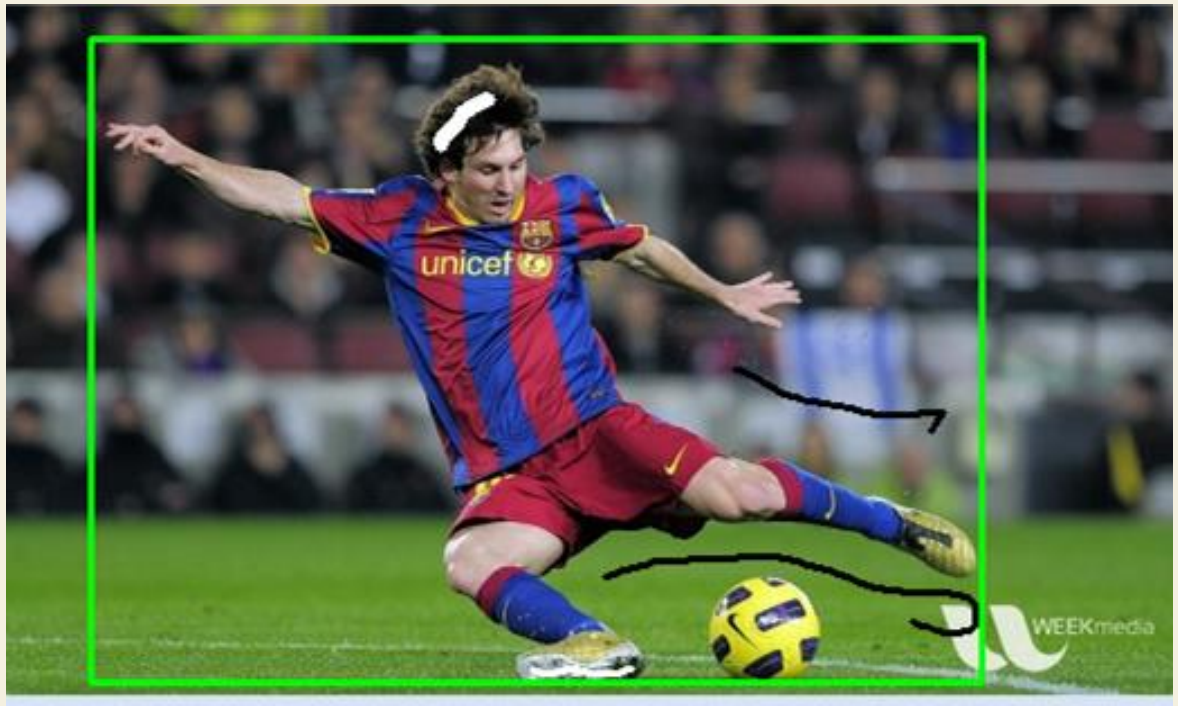
m: 捨棄繪圖結果並進行border matting

r: 重新開始

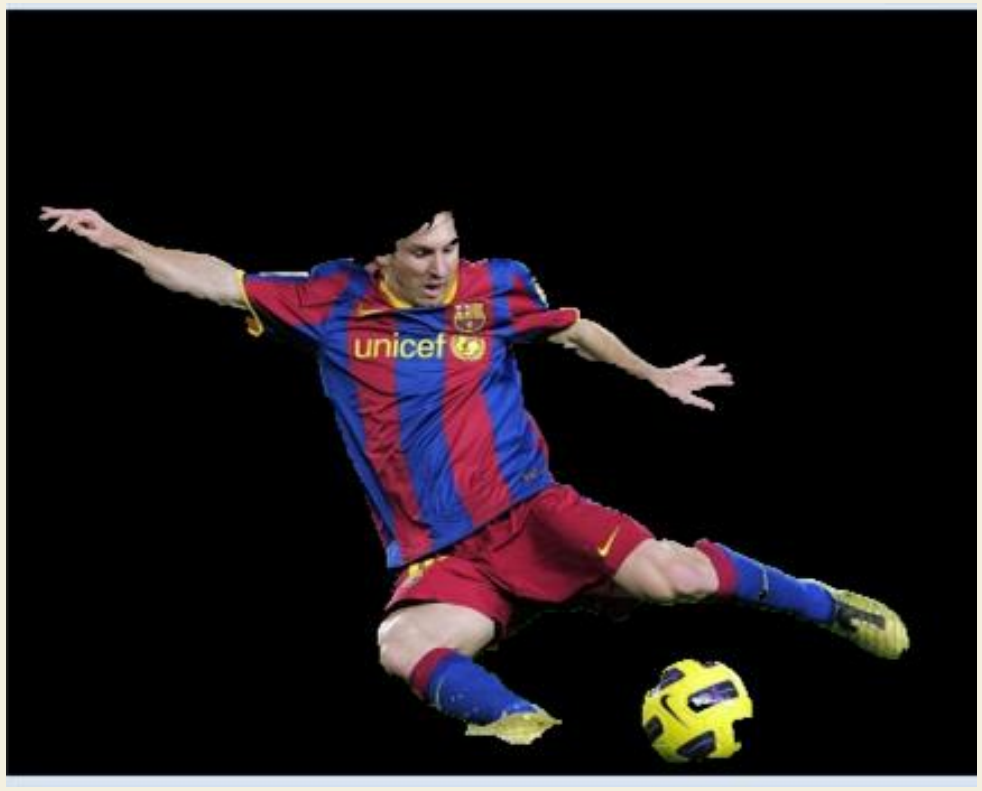
a: 以動畫方式展現目前去背過程

h: 顯示所有指令

Result-Sample



Result-no brush



Result-brush

Result-border matting

Result-other example

Result-border matting

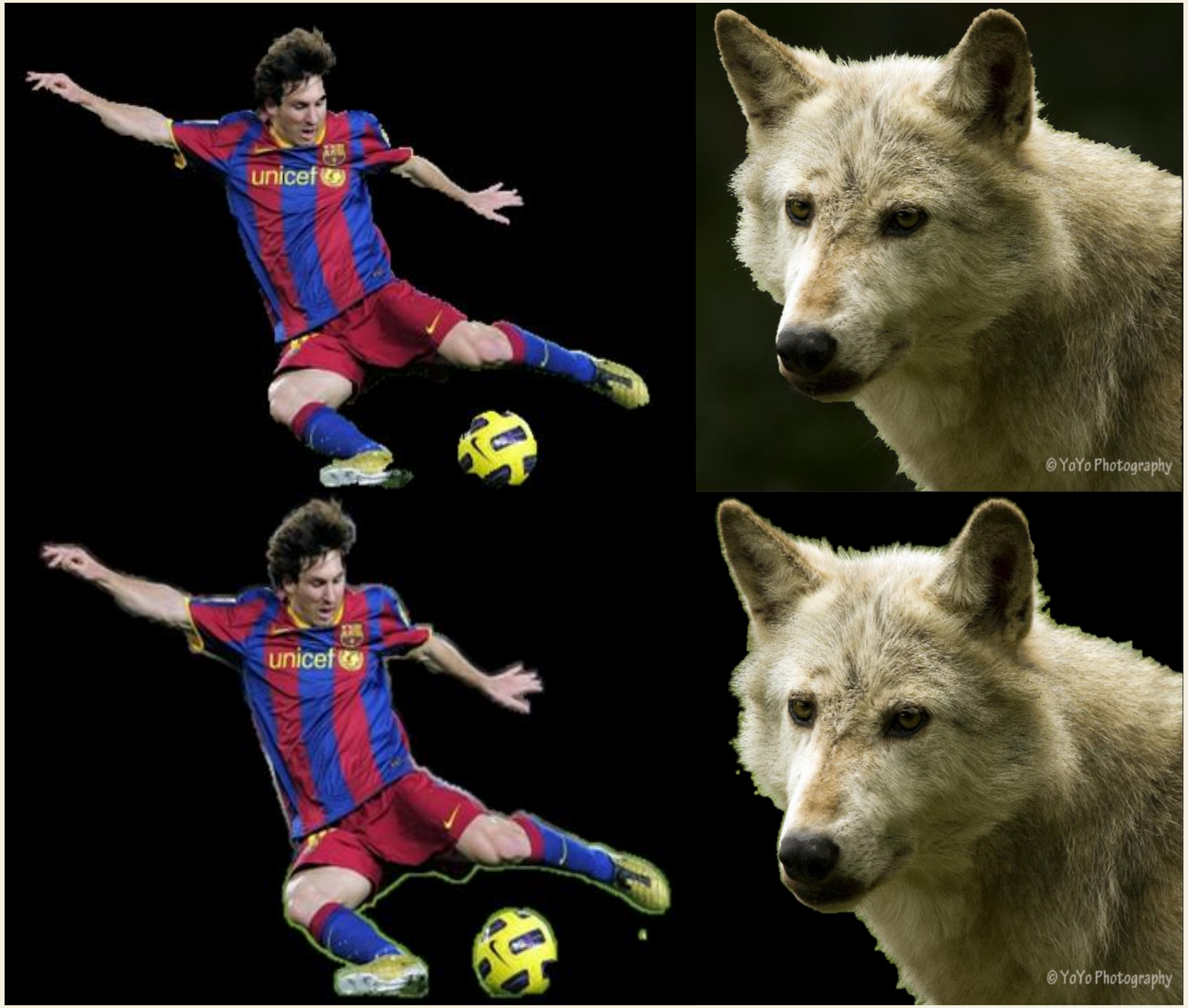
References

"GrabCut" | ACM SIGGRAPH 2004 Papers

- 篇名：“GrabCut” — Interactive Foreground Extraction using Iterated
  Graph Cuts
- 作者：Carsten Rother, Vladimir Kolmogorov, Andrew Blake
- 發表期刊：SIGGRAPH '04: ACM SIGGRAPH 2004 Papers
- 發表時間：2004年8月

Original

Border Matting



Thanks for listening