

# DIP report

b04902054 王曄庭、b04703001 蔡明宏  
組別: 4

## Paper title

**Global Contrast Based Salient Region Detection[1]**

Ming-Ming Cheng, Niloy J. Mitra, Xiaolei Huang, Philip H. S. Torr, and Shi-Min Hu

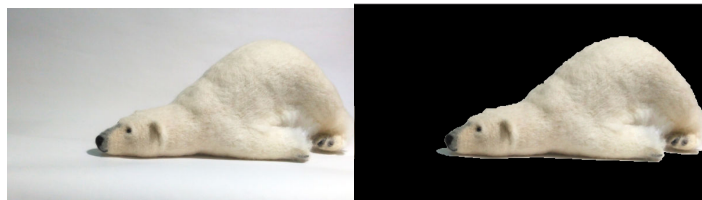
## Motivation

人的眼睛擅長於捕捉環境中鮮明的物體(salient objects)，而相對地，電腦要捕捉、辨識出 salient objects，並不是那麼容易的事情。如何快速、準確的從容易混淆的背景中找到 salient objects，便是其中的一大挑戰。因此我們希望能夠找出一個有效的方法，盡可能辨識出一張圖像中主角的位置，以利於之後針對該圖中的主角進行一些其他的處理。

此外，若是我們順利得到理想的結果，便會考慮將此方法融合 Shape matching and object recognition [2]，先利用我們的方法去偵測出主角的區域，再將該區域的形狀 map 到另一張圖的主角，並進行比對。利用 2 張圖中主角的形狀的相似程度，應用於圖片檢索。

## Problem definition

給定一張圖形，設法找出該圖中的主角，標記出其所在的範圍。



# Algorithm

## 演算法概念

主角物體通常會是整張圖片最「突兀的」，也就是對比程度較大的區域。同樣的物體，顏色的分佈相近，因此直觀上，一系列相近顏色可以大約代表一個物體。我們先定義每一種顏色在整張圖片的突兀程度，找出能量最大的區域，再透過算法的剪裁，輸出演算法所找到的輪廓二分圖。

### Step1

計算每一個像素的突兀程度，製作突兀能量分佈圖

分成 (1) 全局計算 (2) 區域計算。其中全局版本的效能較好，但成品較為粗糙。區域版本的考慮了地域的資訊，成品較細緻，但效能較差。

#### (1)全局版本

定義突兀程度

$$S(I_k) = \sum_{I_i \in I} D(I_k - I_i)$$

其中 $D(I_k - I_i)$ 為顏色 $I_k$ 和 $I_i$ 在 $l \times a \times b$ 空間的距離

#### (2)區域版本

第一步：先利用 graph-based segmentation 的演算法圈出各個區域

第二步：定義突兀程度

$$S(r_k) = w_s(r_k) \sum_{r_i \neq r_k} e^{\frac{D_s(r_k - r_i)}{-\sigma_s^2}} w(r_i) D_r(r_k, r_i)$$

其中 $w_s(r_k) = \exp(-9d_k^2)$ ， $d_k$ 為區域與圖片中心的平均距離  
 $d_s(r_k, r_i)$ 是兩個區域質心的距離， $\sigma_s$ 控制距離的影響程度  
 $w(r_i)$ 是區域所包含的像素數量

$$D_r(r_k, r_i) = \sum_{i=1}^{n1} \sum_{j=1}^{n2} f(c_{1,i}) f(c_{2,j}) D(c_{1,i} - c_{2,j})$$

$f(c_{k,i})$ 是在區域 $k$ 中全部 $n_k$ 種顏色的第 $i$ 個顏色 $c_{k,i}$ 在整張圖片出現的機率

以直觀來解釋每一個參數， $w_s$ 讓靠近圖片中心的權重較大，更容易選為主角物體。 $d_s$ 讓鄰近區域的比較，有較大的權重。 $\sigma_s$ 控制距離的影響程度。 $w(r_i)$ 讓面積較大的區域有較高的權重。綜合考量區域的資訊後，雖然犧牲了些許效能，卻能讓最後的突兀分佈圖有較好的成果。

## Step 2

### 針對突兀圖中能量最強的區域做剪裁

主要的概念是將圖片切割成三種狀態，分別是背景，不確定，和物體。將三元圖餵給 GrabCut 演算法，輸出顯著物體的區域。接著，利用剛才的輸出再製作一個三元圖，重新餵給 GrabCut。利用循環來逼近物體的邊緣，增加精準度。

#### (1)第一步 初始化

設立門檻，將圖片二分成背景，和不確定。在初始化的時候，並沒有確定的區域，而是希望不確定的區域經過循環的 GrabCut 可以漸漸描繪出確定的部分。

#### (2)第二步 循環的分割

將三分圖餵給 GrabCut 演算法，輸出主角物體的區域。根據輸出的區域，重新描繪三分圖，描繪的方式如下。將區域做 dilation，放寬標準，如果仍然不在 dilation 區域的像素標記成背景。將區域做 Erosion，限縮標準，仍然在 Erosion 區域內的標記為物體。其他的區域則標記為不確定。最後將三分圖重新餵給 GrabCut，循環直到主角物體的區域沒有變化為止。

## Expected result



## Reference

[1]<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6871397>

[2]<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=993558>