# Algorithme de segmentation avancé: Grabcut

Laboratoire 4

# Algorithme GrabCut

- Segmentation entre:
  - Objets
  - Arrière-plan

- Annotation d’images (Utile pour l’entraînement Deep Learning)
- Utiliser dans produits commerciaux: Powerpoint

# Algorithme itératif

But: Minimiser fonction de coût

$$\min_{S} \lambda |\delta S| + \sum_{p \in S} f(p)$$

## Ratio de probabilités

$$f(p) = -log\frac{Pr(I_p|S)}{Pr(I_p|\Omega - S)}$$

$Pr(I_p|S)$ Probabilité qu'un pixel d'intensité Ip appartienne à l'avant-plan (S)

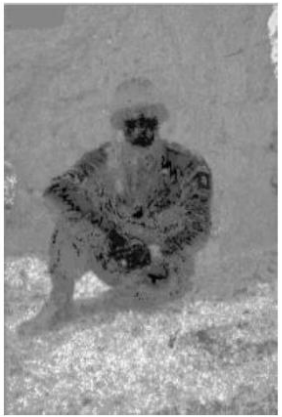
$Pr(I_p|\Omega - S)$ Probabilité qu'un pixel d'intensité Ip appartienne à l'arrière-plan ($\Omega$ - S)

# Minimisation de l'énergie

$$\min_S \sum_{p \in S} (-log Pr(I_p|S)) + \sum_{p \in \Omega - S} (-log Pr(I_p|\Omega - S))$$

Probabilité d'appartenance à l'avant-plan

Probabilité d'appartenance à l'arrière-plan

# Ajout du terme de la frontière

$$\min_S \lambda|\delta S| + \sum_{p \in S} -log(Pr(I_p|S)) + \sum_{p \in \Omega - S} -log(Pr(I_p|\Omega - S))$$

Mesure la longueur de la frontière recherchée

Favorise des frontières lisses et élimine les petites régions isolées

En ajoutant le terme de frontière

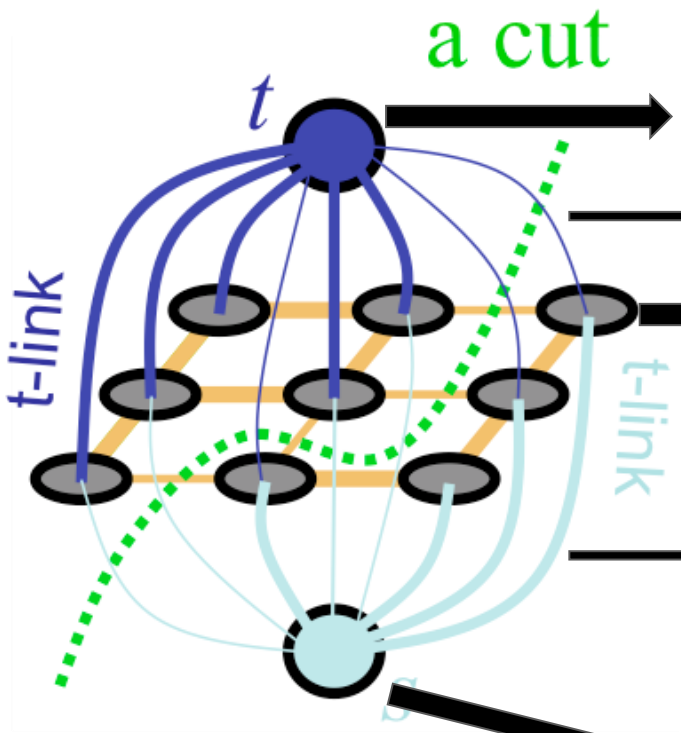
# Principe de l’algorithme itératif

À chaque itération:

Coupure de graphes selon le minimum de la fonction d’optimisation

Coupure de graphes == Segmentation

# Construction des graphes




Arrière-plan

Poid (probabilité) reliant le pixel à l'**arrière-plan**

Pixel

Poid (probabilité) reliant le pixel à l'**avant-plan**

Avant-plan

# Poids des graphes

Poids (probabilité d'appartenance) reliants chaque pixels à l'**avant-plan**:

Matrice de la même taille que l'image (M x N):

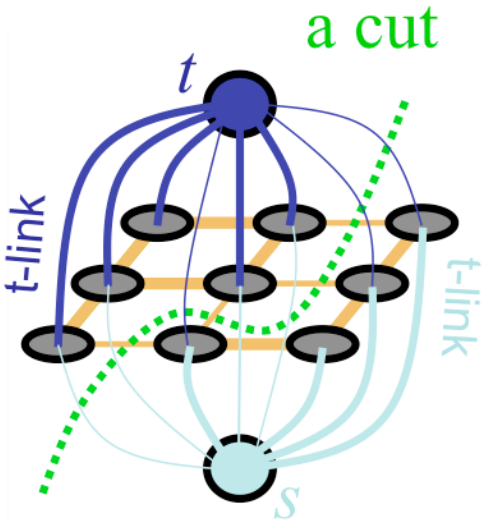
***objProbabilitees***

Poids (probabilité d'appartenance) reliants chaque pixels à l'**arrière-plan**:

Matrice de la même taille que l'image (M x N):

***bkgProbabilitees***

# Coupure des graphes (Segmentation)




En fonction de la chaîne d'arêtes dont la **somme des poids est minimale (Énergie):**

$$\min_S \boxed{\sum_{p \in S} (-logPr(I_p|S))} + \boxed{\sum_{p \in \Omega - S} (-logPr(I_p|\Omega - S))}$$

***objProbabilitees***

Probabilité d'appartenance à l'avant-plan

***bkgProbabilitees***

Probabilité d'appartenance à l'arrière-plan

# Coupure de graphes

Fonction fournie:

*[masque_seg, ~]= optimizeWithBK(prob_bkg, prob_obj)*

Permet d’obtenir la nouvelle segmentation entre l’objet et l’arrière-plan

Jusqu’à la convergence vers la solution optimale

# À faire

1. Lire, exécuter et comprendre l’implémentation actuelle:
    - Une seule itération de l’algorithme (GrabCut.m)
    - Image en ton de gris seulement (calculerProbabilitésParPixel)

# À faire: Prise en compte d'images couleur (RGB)

Modifier ***calculerProbabiliteParPixel(image, masque_seg)***:

- Probabilités approximées à partir d'histogrammes
- Implémentation actuelle: Seulement en **ton de gris**

Taille: Nbins

- Définir et compiler l'histogramme pour **chaque canal de couleur**

Taille: Nbins x Nbins x Nbins

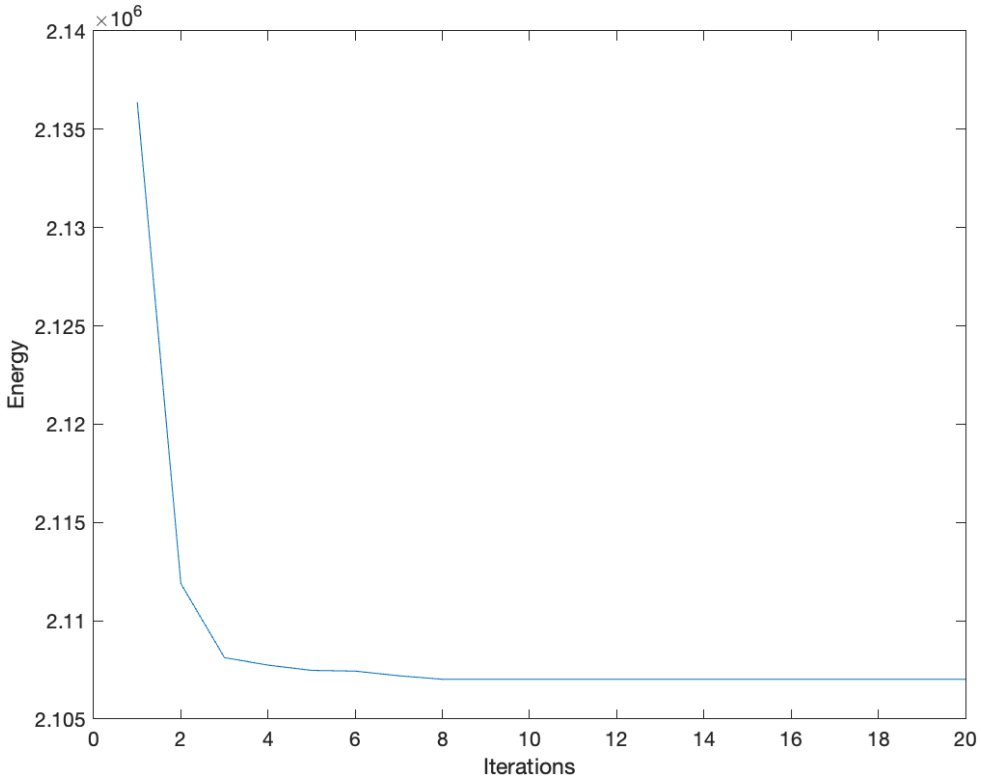
Canal: R G B

# À faire - Itération jusqu'à convergence

Modifier ***GrabCut.m***:

À chaque itération jusqu'à **convergence**:

- Segmentation - Coupure de graphe (***optimizeWithBK***)
- Comptabilisation de l'énergie (***computeEnergy***)
- Recalculer les matrices de probabilités à l'aide du nouveau masque de segmentation (***calculerProbabilitesParPixel***)



# À faire - Ajout de contraintes manuelles

Modifier ***GrabCut.m***:

- Deux matrices:
  - ***contraintePousserVersAV***: taille M x N
  - ***contraintePousserVersAR***: taille M x N
- Si on veut appliquer des contrainte, saisi des zones de contraintes:

[x, y, largeur, hauteur] = *getrect()*

- Assignation d'une grande valeur de pénalité: ex. $10^9$
- Addition des contraintes aux matrices de probabilités

***objProbabilitees = objProbabilitees + contraintePousserVersAR***

***bkgProbabilitees = bkgProbabilitees + contraintePousserVersAV***

# À faire - Itération jusqu'à convergence avec contraintes

Nouvelle boucle d'itération jusqu'à **convergence**

Tenir compte des contraintes manuelles saisies:

***objProbabilitees = objProbabilitees + contraintePousserVersAR***

***bkgProbabilitees = bkgProbabilitees + contraintePousserVersAV***