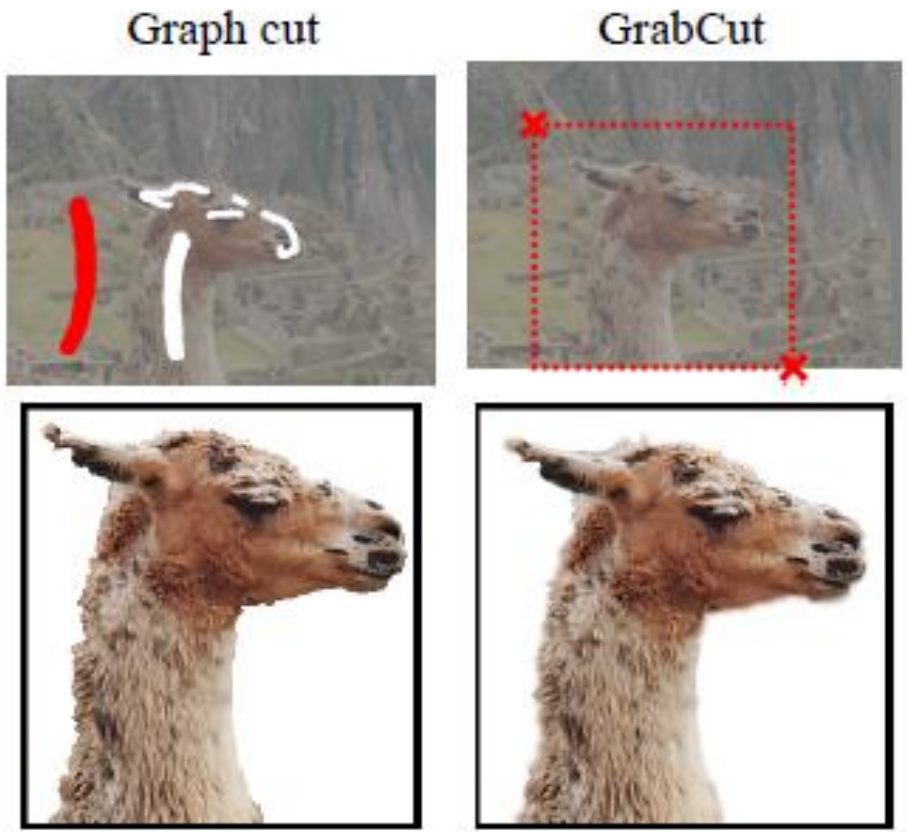
«GrabCut» — Interactive Foreground Extraction using Iterated Graph Cuts.

«Graph cut» vs «GrabCut»



The main task of segmentation

Original image: $z = (z_1, z_2, \dots, z_N)$.

The segmentation of the image is expressed as an array of "opacity" values $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_N)$ at each pixel:

– usually $0 \leq \alpha_i \leq 1$;

– hard segmentation $\alpha_i \in \{0,1\}$.

«Graph cut»(1)

- Two histograms are calculated: one for the foreground, the second for the background:

$$\underline{\theta} = \{h(z;\alpha), \alpha = 0,1\}$$

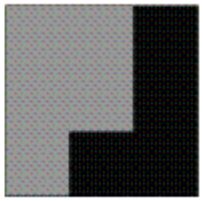
- Energy function : $\mathbf{E}(\underline{\alpha},\underline{\theta},\mathbf{z}) = U(\underline{\alpha},\underline{\theta},\mathbf{z}) + V(\underline{\alpha},\mathbf{z})$.

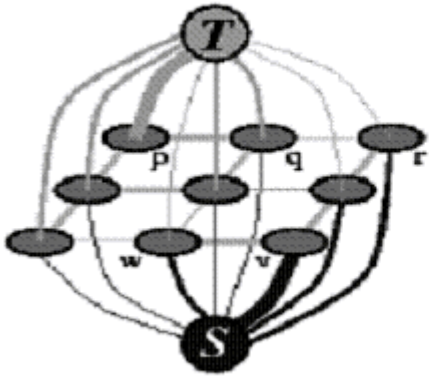
$$U(\underline{\alpha},\underline{\theta},\mathbf{z}) = \sum_n -\log h(z_n;\alpha_n)\,. \qquad V(\underline{\alpha},\mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} dis(m,n)^{-1}\,[\alpha_n \neq \alpha_m] \exp -\beta(z_m - z_n)^2$$

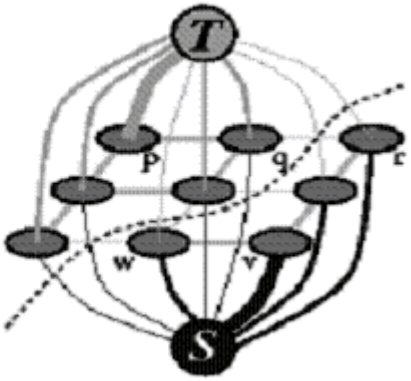
- The global minimum is estimated

$$\underline{\hat{\alpha}} = \arg\min_{\underline{\alpha}} \mathbf{E}(\underline{\alpha},\underline{\theta})$$

«Graph cut»(2)







GrabCut(1)

- Use GMMs: $K = 5$
- Energy function : $\mathbf{E}(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) + V(\underline{\alpha}, \mathbf{z}),$

$$U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z}) = \sum_n D(\alpha_n, k_n, \underline{\theta}, z_n).$$

$$V(\underline{\alpha}, \mathbf{z}) = \gamma \sum_{(m,n)\in\mathbf{C}} [\alpha_n \neq \alpha_m] \exp -\beta \|z_m - z_n\|^2.$$

GrabCut(2)

Initialisation

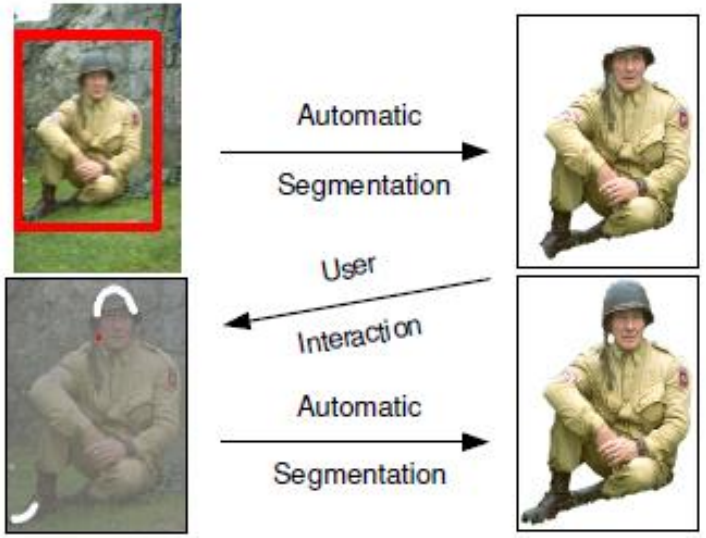
- User initialises trimap T by supplying only T_B. The foreground is set to $T_F = \emptyset$; $T_U = \overline{T}_B$, complement of the background.
- Initialise $\alpha_n = 0$ for $n \in T_B$ and $\alpha_n = 1$ for $n \in T_U$.
- Background and foreground GMMs initialised from sets $\alpha_n = 0$ and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

Iterative minimisation

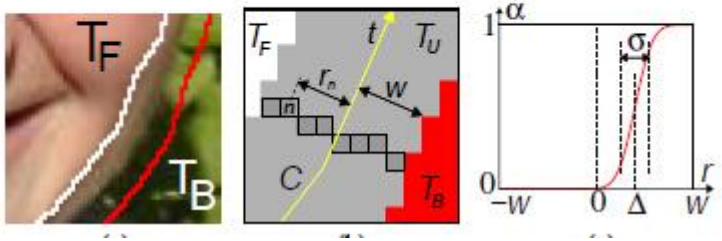
1. *Assign GMM components to pixels:* for each n in T_U,
$$k_n := \arg\min_{k_n} D_n(\alpha_n, k_n, \theta, z_n).$$
2. *Learn GMM parameters* from data z:
$$\underline{\theta} := \arg\min_{\underline{\theta}} U(\underline{\alpha}, \mathbf{k}, \underline{\theta}, \mathbf{z})$$
3. *Estimate segmentation:* 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).

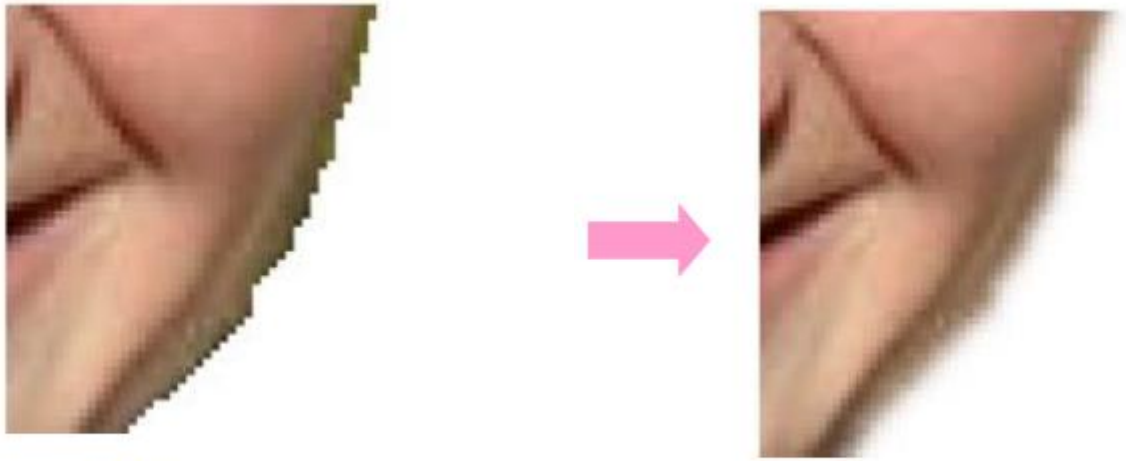
User editing

- *Edit:* fix some pixels either to $\alpha_n = 0$ (background brush) or $\alpha_n = 1$ (foreground brush); update trimap T accordingly. Perform step 3 above, just once.
- *Refine operation:* [optional] perform entire iterative minimisation algorithm.



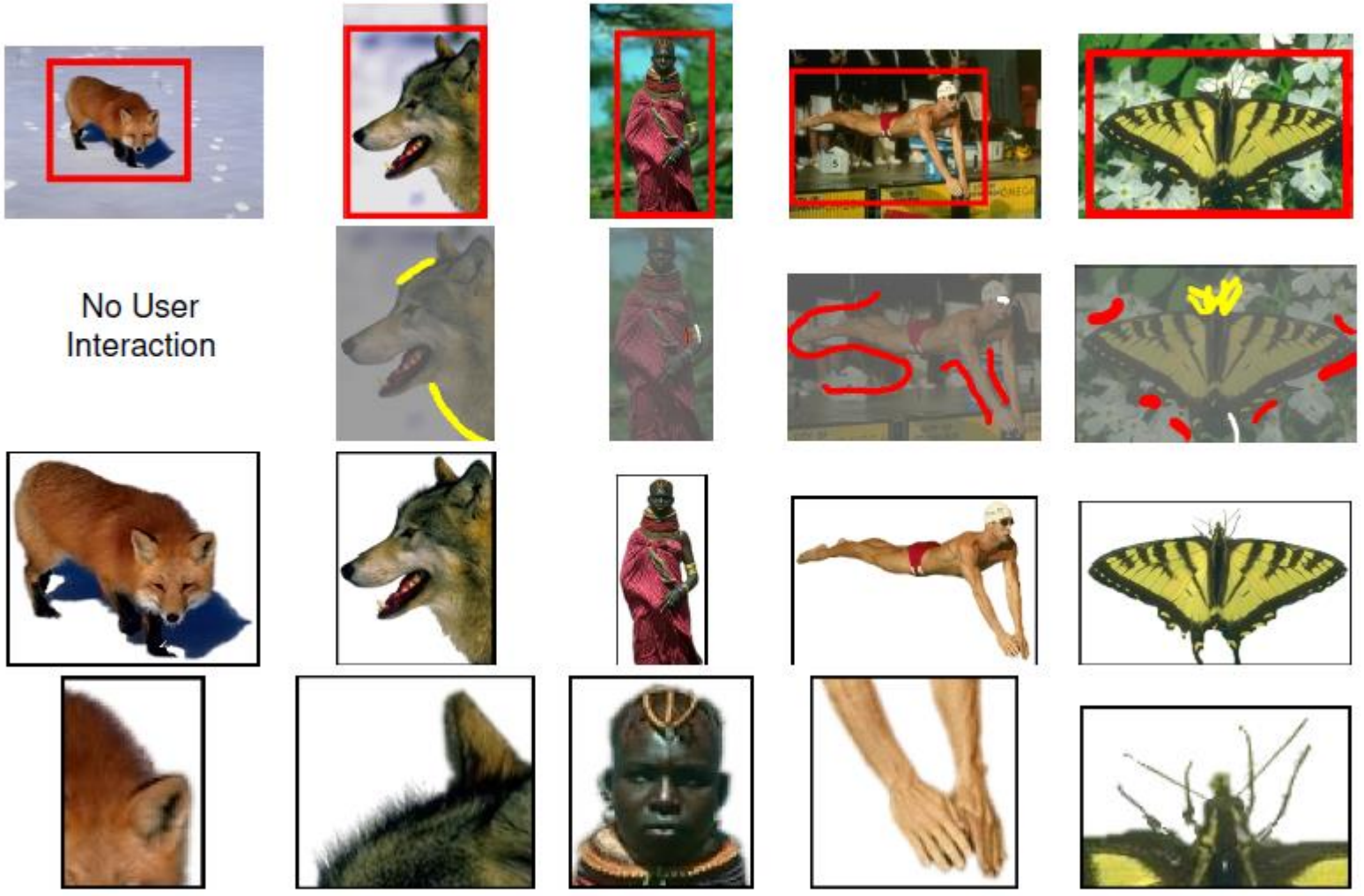
GrabCut(3)





Жесткая сегментация

Мягкая сегментация


No User
Interaction