Information Processing and Computer Vision (TIVA) TP5 - Stereo Matching with Loopy Belief Propagation (LBF).

Yonatan DELORO

For May 2017, the 24^{th}

1 Computing messages & beliefs at each iteration of LBF

(i.) The vector of messages that node p will send to node q at iteration t, if a general pair-wise cost $\mathcal{V}$ (defined in the paper) is used, is given by the formula :

$$\forall l_q \in \{0, ..., d_{max}\}, m^t_{p,q}(l_q) = \underset{l_p \in \{0,\dots,d_{max}\}}{min} \{\mathcal{D}_p(l_p) + \mathcal{V}(l_p - l_q) + \sum_{r \in \mathcal{N}(p),\ r \neq q} m^{t-1}_{r,p}(l_p)\}$$

We consider in this paper $\mathcal{N}(p)$ as the four neighbors of pixel p.

Thus, for instance, the messages that are transmitted from nodes to their right neigbors are computed as follows :

$$\begin{aligned} \forall l \in \{0, ..., d_{max}\}, \quad m^t_{(x,y),(x+1,y)}(l) = \underset{l' \in \{0,\dots,d_{max}\}}{min} \quad & \{\mathcal{D}_{(x,y)}(l') + \mathcal{V}(l' - l) + m^{t-1}_{(x-1,y),(x,y)}(l') \\ & + m^{t-1}_{(x,y+1),(x,y)}(l') + m^{t-1}_{(x,y-1),(x,y)}(l')\} \end{aligned}$$

(ii.) What is the complexity of computing this vector ? Let us give $l_q \in \{0, ..., d_{max}\}$. For each $l_p \in \{0, ..., d_{max}\}$, we need to compute $\mathcal{D}_p(l_p) + \mathcal{V}(l_p - l_q) + \sum_{r \in \mathcal{N}(p),\ r \neq q} m^{t-1}_{r,p}(l_p)$ in order to consider the minimum of these values. For each l_p, the cost is constant ($|\mathcal{N}(p)| \leq 4$).

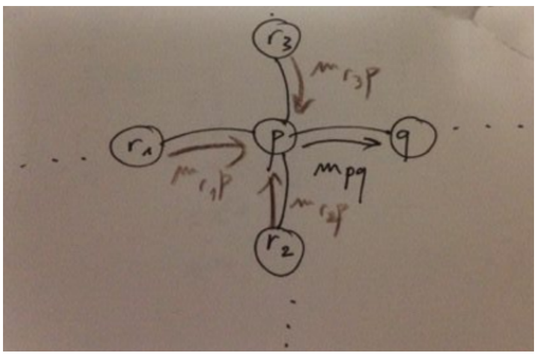

FIGURE 1 – Transmission of message from p to q

.

Thus the complexity of computing the vector $m^t_{p,q}$ is in $\mathcal{O}(d_{max} * d_{max}) = \mathcal{O}(d^2_{max})$. It is quadratic in the number of disparity values considered, or in the "thinness" of disparity discretization one could say.

(iii.) The beliefs b^t_q of node q at iteration t are given by the sum of the data cost term in q and the messages sent to q at this iteration :

$$\forall l_q \in \{0, ..., d_{max}\}, b^t_q(l_q) = \mathcal{D}_q(l_q) + \sum_{p \in \mathcal{N}(q)} m^{t-1}_{p,q}(l_q)$$

We get the MAP label of node q at iteration t by taking the "disparity" l_q that minimizes the "energy" given by the belief b_q, that is to say :

$$l^{t\star}_q = \underset{l_q \in \{0,...,d_{max}\}}{argmin} \; b^t_q(l_q)$$

2 The specific case of Potts model

For pair-cost, we use the Potts model $\mathcal{V}(x) = \lambda\delta_{x\neq0}$.

Thus the pairwise term is just a function of the labels l_p and l_q (of their difference to be precise), thus it is independent from the positions of the pixels p and q.

We can re-write the formula of 1.(i) as follows : $\forall l_q \in \{0, ..., d_{max}\}$,

$$\begin{aligned} m^t_{p,q}(l_q) &= \min\{\mathcal{D}_p(l_q) + \textstyle\sum_{r \in \mathcal{N}(p),\ r\neq q} m^{t-1}_{r,p}(l_q), \min_{l_p \neq l_q}\{\mathcal{D}_p(l_p) + \textstyle\sum_{r\in\mathcal{N}(p),\ r\neq q} m^{t-1}_{r,p}(l_p)\} + \lambda\} \\ &= \min\{\phi_{p,q,t}(l_q), \min_{l_p\neq l_q} \phi_{p,q,t}(l_p) + \lambda\} \end{aligned}$$

where we have defined $\forall l, \phi_{p,q,t}(l) := \mathcal{D}_p(l) + \sum_{r\in\mathcal{N}(p),\ r\neq q} m^{t-1}_{r,p}(l)$

Thus, we just need to compute once the minimum Q of $\phi_{p,q,t}$ on $\{0, ..., d_{max}\}$. As $\lambda \geq 0$, for each $l_q \in \{0, ..., d_{max}\}$, the message $m^t_{p,q}(l_q)$ is the minimum of $Q + \lambda$ and $\phi(l_p)$. Thus the complexity cost to compute the message is the complexity cost of computing $\phi_{p,q,t}(l)$ for each l.

Consequently, using the Potts model as $\mathcal{V}$, the complexity of computing the vector becomes linear in $\mathcal{O}(d_{max})$, while it was quadratic in the general case.

3 Implementation of the min-sum LBP algorithm

We fill the functions that compute at a given iteration the new messages, beliefs and labels, the cost of the data term and the whole energy.
Please see the code.

4 Purpose and effect of normalizing messages

The function named **normalize_messages.m** centers at a given iteration the messages that are transmitted from node to node :

$$\forall l_q, \quad \widetilde{m}^t_{p,q}(l_q) = m^t_{p,q}(l_q) - \overline{m^t_{p,q}(l_q)}^{p,q}$$

Thus,

$$\overline{\widetilde{m}^t_{p,q}(l_q)}^{p,q} = 0$$

The purpose is to avoid overflow, as we do not want the messages to increase too significatively (what could cause difficulty to find the labels l_q that minimize the beliefs $b^t_q(l_q) = \mathcal{D}_q(l_q) + \sum_{p\in\mathcal{N}(q)} m^{t-1}_{p,q}(l_q)$).

Theoretically, it does not affect the labelling that is obtained in each iteration. Indeed, as we remove the same quantity from each message at a given iteration, both the labbeling at the same iteration and the computing of the messages at the next iteration are not affected ($argmin(a + c, b + c) = argmin(a, b)$, and see formulas 1.(i) and 1.(iii))

5 Results of LBF : disparity map and energy evolution

We run the Loopy Belief Propagation Algorithm for different values of regularization penalty $\lambda = 1, 10$ and 1000. We display in each case the obtained disparity map and the energy curb over the LBP Algorithm iterations.

About convergence For both values of lambda, the Loopy Belief Propagation converges toward a local minimum of energy, as we can see it with the curbs of evolution of the energy over the iterations. Thus, if our model is good enough to compute "real" depth, we can expect from the results of the algorithm maps that are not so far from the ground truth.

We also see that the greater λ is, the faster the convergence is.

About the disparity maps We observe that the greater λ is, the smoother the disparity map is. It is logical as λ penalizes the affectation of different disparities for neighbouring pixels.

The extreme values $\lambda = 1$ and $\lambda = 1000$ appear as bad choices.

— If $\lambda = 1$, depth varies deeply from pixel to pixel. However, we do not live in a world where there are such discontinuities. ! The penalizatin of assigning different labels for neigbouring pixels is not large enough.

— If $\lambda = 1000$, the smooth effect is too powerful. It even destroys the frontiers between the objects that are classically regions where depth changes significatively. Indeed, the data term becomes on this case very small with respect to the enormous term of regularization in the whole energy.

On the contrary, $\lambda = 10$ appears as a good choice of regularization, even if the research of the parameter could be probably refined. Even if they are still some defects, we manage with this parameter to rebuild the different levels of depth we see on the ground truth.

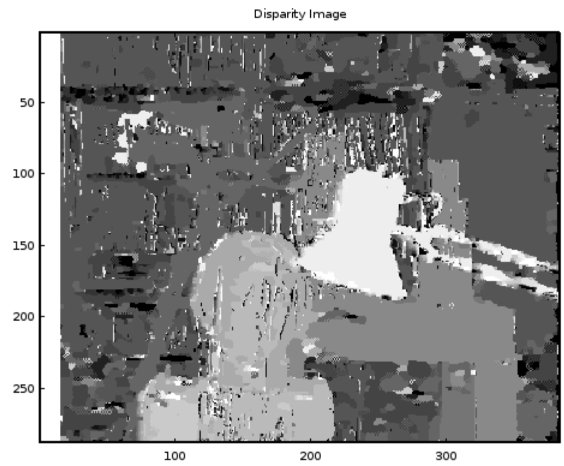

FIGURE 2 – Disparity map for $\lambda = 1$

.

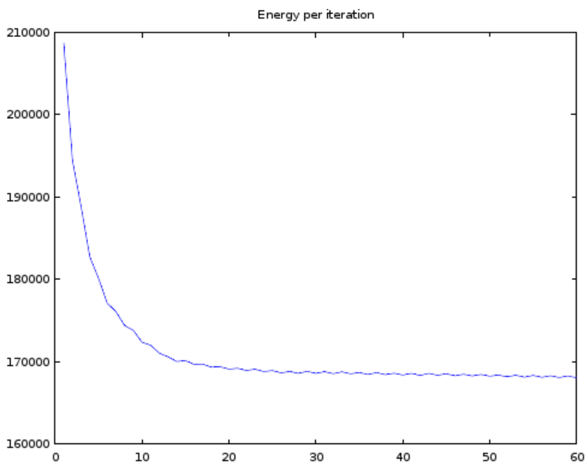

FIGURE 3 – Evolution of energy over the LBP iterations for $\lambda = 1$

.

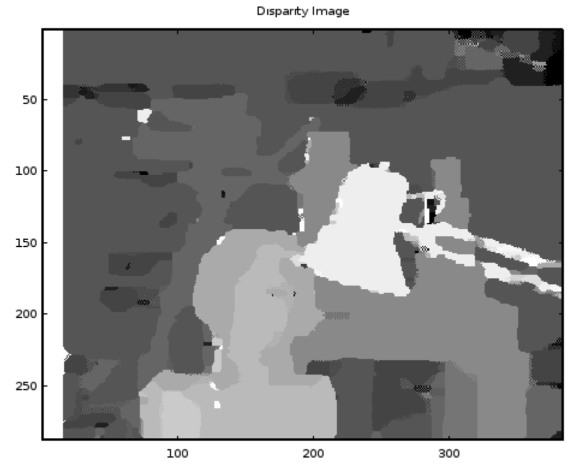

FIGURE 4 – Disparity map for $\lambda = 10$

.

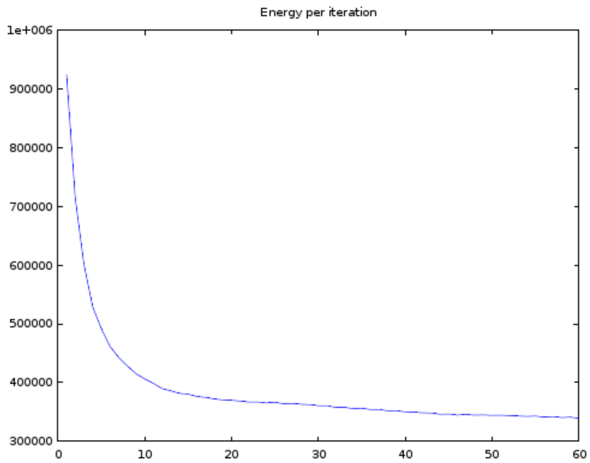

FIGURE 5 – Evolution of energy over the LBP iterations for $\lambda = 10$

.

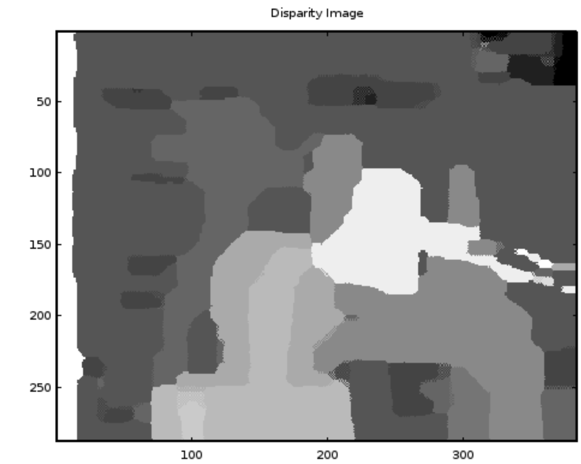

FIGURE 6 – Disparity map for $\lambda = 1000$

.

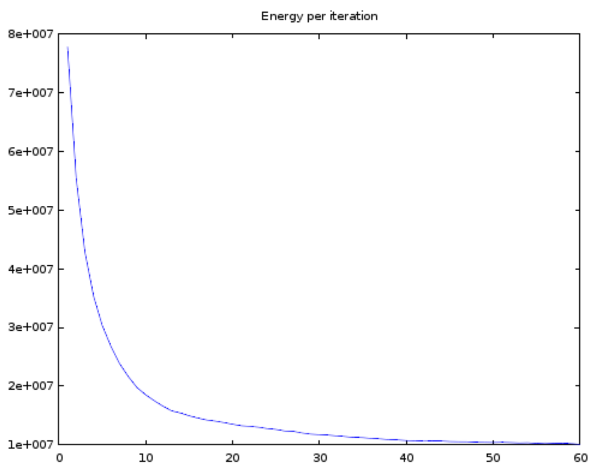

FIGURE 7 – Evolution of energy over the LBP iterations for $\lambda = 1000$

.

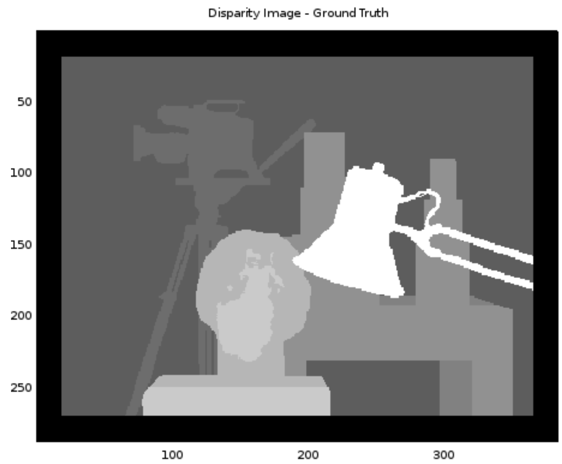

FIGURE 8 – Ground truth

.