Image Processing and Pattern Recognition

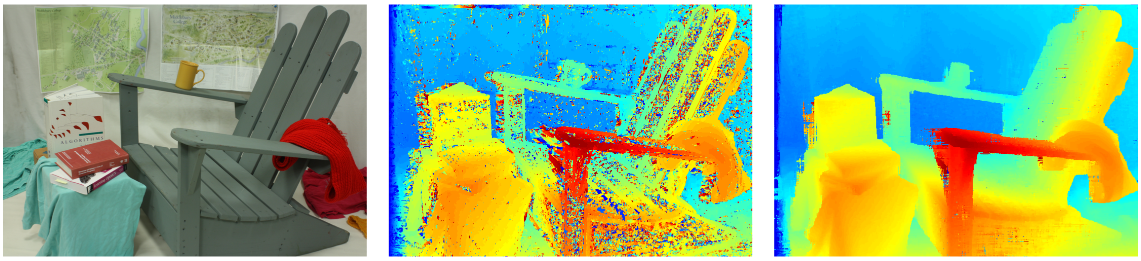
Assignment 3 - Stereo

January 13, 2020

Deadline: January 27, 2020 at 23:55h.
Submission: Upload your report as a .pdf file and your implementation as a .py file to the TeachCenter. Please do not zip your files and use the provided framework-file for your implementation.

1 Goal

The goal of the second assignment is to implement the two most important parts of a stereo method. This is (1) the computation of the cost-volume and (2) the regularization of the often erroneous initial result. In the first part we compute the cost-volume using different local matching costs and compare their respective performance. In the second part we regularize the result with the famous Semi-Global Matching (SGM) algorithm. Fig. 1 shows a sample image plus the winner-takes-all (WTA) result and the SGM result.



(a) Reference Image (b) WTA Result (c) SGM Result

Figure 1: Algorithm visualization. (a) shows the reference image, (b) shows the winner-takes-all result and (c) shows the regularized result. Note how we got rid of a lot of noise.

2 Local Matching

In the first part of the assignment, we compare three different metrics for local matching. For now, we consider two patches p and q which we want to compare. The most often used similarity measure is the the *sum of squared differences* similarity measure or short SSD, which is defined as

$$SSD(p,q) = \sum_{x=1}^{W}\sum_{y=1}^{H}(p(x,y) - q(x,y))^2. \tag{1}$$

The second measure is called the *sum of absolute differences* (SAD). The name already indicates that the square is now replaced by the absolute function, *i.e.*

$$SAD(p,q) = \sum_{x=1}^{W}\sum_{y=1}^{H}|p(x,y) - q(x,y)|. \tag{2}$$

Last, we use a slightly more sophisticated similarity measure called *normalized cross correlation* (NCC). It is defined as

$$NCC(p,q) = \frac{\sum_{x,y}(p(x,y)-\bar{p})(q(x,y)-\bar{q})}{\sqrt{\sum_{x,y}(p(x,y)-\bar{p})^2 \sum_{x,y}(q(x,y)-\bar{q})^2}}. \tag{3}$$

The NCC additionally normalizes the patches by subtracting their means, $\bar{p} = \frac{1}{H \cdot W}\sum_{x,y} p(x,y)$ and $\bar{q} = \frac{1}{H \cdot W}\sum_{x,y} q(x,y)$.

Cost Volume We use the similarity measures to compute a cost-volume for a pre-defined range of disparities D. The cost-volume can be computed with

$$cv(x,y,d) = s(I_0(x,y), I_1(x-d,y)), \quad d \in \mathcal{D}, \tag{4}$$

where x and y are the coordinates of the pixels and $\mathcal{D} = \{0,\ldots,D-1\}$ are all valid disparities. Note that the cost-volume is a volume. It contains the similarity for all pixels for all pre-defined disparities in the 3^{rd} dimension.

Winner-takes-all Using the cost-volume, we can directly compute the WTA solution as visualized in Fig. 1(b). It can be computed by the pixel-wise arg-min over the disparity dimension, *i.e.*

$$\bar{d}(x,y) \in \arg\min_d cv(x,y,d). \tag{5}$$

3 Semi-Global Matching

In the second part of the assignment, we introduce *pairwise* costs. They penalize deviation between neighboring pixels. We aim to minimize the following energy with SGM:

$$\min_z \underbrace{\sum_{i\in\mathcal{V}} g_i(z_i)}_{\text{data term}} + \underbrace{\sum_{ij\in\mathcal{E}} f_{i,j}(z_i,z_j)}_{\text{pairwise term}}. \tag{6}$$

Note that we used the following notation here: i and j are neighboring nodes which correspond to a specific position x and y. The variable z is a complete labeling (=disparity map in our case) of the scene. $\mathcal{V}$ are all vertices of the image, i.e. the image pixels, and $\mathcal{E}$ are edges on a 4-connected grid, $g_i : \mathcal{D} \to \mathbb{R},\ \ g_i \in \mathbb{R}^D$, *i.e.* a function which measures the cost for all disparities. In our case the g_i are represented by the cost-volume Eq. (4). The functions $f_{i,j} : \mathcal{D}\times\mathcal{D} \to \mathbb{R},\ \ f_{i,j} \in \mathbb{R}^{D\times D}$ define penalties for jumps between neighboring pixels. In this assignment, we define the pairwise costs as

$$f_{i,j}(z_i,z_j) = \begin{cases} 0 & \text{if } z_i = z_j \\ L_1 & \text{if } |z_i - z_j| = 1 \\ L_2 & \text{else,} \end{cases} \tag{7}$$

where L_1 and L_2 are hyper-parameters of our model.

Algorithm The task is now to compute the result of the minimization problem Eq. (6) using SGM The algorithm works with a two-stage procedure. First, we compute the messages for all disparity directions, i.e. left-right, right-left, up-down and down-up. This can be done with the following procedure:

$$m^a_{i+1}(t) = \min_{s\in\mathcal{D}}(m^a_i(s) + f_{i,i+1}(s,t) + g_i(s)), \tag{8}$$

where all messages are initialized with zeros. Second, we need to compute the beliefs with

$$b_i(s) = g_i(s) + \sum_{a\in\{L,R,U,D\}} m^a_i(s) \tag{9}$$

The beliefs are defined as the sum of all messages in node i and the unary costs in node i (9). The last step is the to compute the final result based on the beliefs with

$$\hat{d}(x,y) \in \arg\min_d b(x,y,d). \tag{10}$$

Fig. 2 shows a visualization of all terms involved for computing the result of one row.

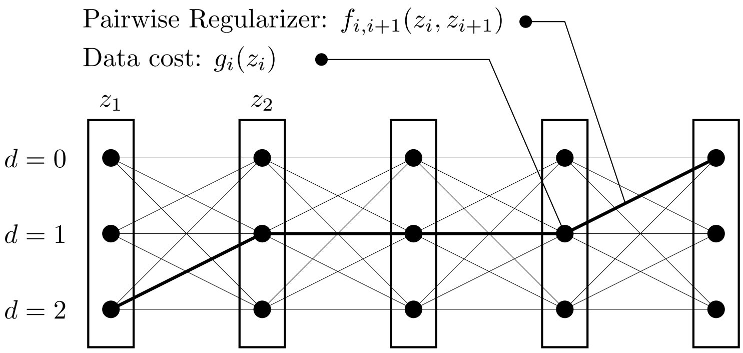


Figure 2: Visualization of Eq. (8) for one row. The black nodes denote the data cost, the edges between the nodes denote the pairwise costs.

4 Evaluation

In the final part of the assignment, we want to evaluate the performance of the described stereo method. Therefore, we compare the output of the WTA result and the regularized result with a given ground-truth disparity map. One standard measure is the *accX* measure. It measures the number of pixels with an error less than or equal to X disparities.

$$accX(z, z^*) = \frac{1}{Z} \sum_{x,y} m(x, y) \cdot \begin{cases} 1 & \text{if } |z(x, y) - z^*(x, y)| \leq X \\ 0 & \text{else,} \end{cases} \tag{11}$$

where z is the prediction ($\bar{d}$, $\hat{d}$ in our case), z^* is the ground-truth disparity map, m is the mask containing zeros for invalid pixels and ones for valid pixels and Z is the number of valid pixels.

5 Task

- Local Matching
  - Implement the local similarity measures Eqs. (1) to (3).
  - Compute the cost-volume as defined in Eq. (4).
  - Compute the WTA solution as defined in Eq. (5).
- SGM
  - Implement the algorithm as defined in Eqs. (8) and (9)
  - Hint: Try to implement the algorithm with the blackboard example I have shown in the exercise. This allows easy debugging and you can easily confirm the correctness of your implementation. Then add the other directions and apply the algorithm to the stereo task.
- Evaluation
  - Compute the accuracy accX with $X = \{1, 2, 3\}$ (11) of the WTA solution (5) for the local similarity measures (SSD, SAD, NCC). For each similarity measure use the radii $\{1, 2, 5\}$. What is the difference between SSD, SAD, NCC in terms of performance (compare the numbers in a table)? How does the result change when the radius of the matching window is changed? Plot and describe the results of all variants in your report.
  - Same as before, but now with the regularized result, ie $\hat{b}$.
  - Use the best performing model and vary the hyper-parameters L_1 and L_2. They should be always positive and $L_2 > L_1$. Plot and describe the results and describe the effect of the hyper-parameters in your report.
  - Use the best parameter setting and compute the optimal solutions using your SGM implementation.

- Bonus Task: Add an edge-dependent weighting term to the regularizer. Therefore replace $f_{i,j}$ with

  $$\tilde{f}_{i,j} = w_{i,j} f_{i,j}, \tag{12}$$

  where $w_{i,j}$ are some weights computed from the edges in the reference image.

We provide the three input stereo pairs. You should use all of them in your evaluation.

- `cones_left.png` + `cones_right.png`
- `Adirondack_left.png` + `Adirondack_right.png`
- `Adirondack_left.png` + `AdirondackE_right.png`

The algorithm as described above works with gray-scale images. Thus, you must convert the color images to gray-scale images in your implementation.