Chapter 2

Theoretical Background

This chapter describes the theoretical background of this thesis. It starts with defining the Legendre-Fenchel transformation that is a prerequisite for the dense reconstruction. Then the possibility to be invariant to illumination change by using photometric invariants is introduced. After these prerequisites, the sparse reconstruction will be defined as its results are needed by the dense reconstruction. In the end, the theory of the dense reconstruction is shown.

2.1 Legendre-Fenchel Transformation

It is common to use the dual problem to solve an optimization problem. To get the dual form of a problem, it is transformed into another space in order to make solving the problem easier. The problem is then called primal in the original space and dual in the other space. The Legendre-Fenchel transformation can be used to create a dual problem out of a continuous problem that does not have to be differentiable. The dual form will then be convex and differentiable. This section refers to [6].

The Legendre-Fenchel transformation $f^*(p)$ of a function f is defined as

$$f^*(p) = \sup_{x \in \mathbb{R}} \{px - f(x)\} . \tag{2.1}$$

As the supremum of $px - f(x)$ is searched, the derivate of this function has to be 0. This leads to

$$p = f'(x) . \tag{2.2}$$

A tangent at point $(x, f(x))$ will then be defined as

$$f(x) = px - c . \tag{2.3}$$

This can be transformed to

$$c = px - f(x) , \tag{2.4}$$

with c being the negative intersection of the tangent with the y-axis. Comparing (2.1) and (2.4) shows that the Legendre-Fenchel transformation $f^*(p)$ is the negative intersection of a tangent with the slope p and the y-axis. The Lengendre-Fenchel transformation can also be written in a vector notation as

$$f^*(\mathbf{p}) = \sup_{\mathbf{x} \in \mathbb{R}^n} \{\mathbf{x}^{\mathrm{T}}\mathbf{p} - f(\mathbf{x})\} . \tag{2.5}$$

© Springer Fachmedien Wiesbaden 2016
S. Painer, *Variation Based Dense 3D Reconstruction,* BestMasters,
DOI 10.1007/978-3-658-12698-8_2

In this thesis, the Legendre-Fenchel transformation of the Huber norm is needed. The Huber norm is defined as

$$\|\mathbf{x}\|_\varepsilon = \begin{cases} \frac{\|\mathbf{x}\|_2^2}{2\varepsilon} & \text{if } \|\mathbf{x}\|_2 \leq \varepsilon \\ \|\mathbf{x}\|_1 - \frac{\varepsilon}{2} & \text{otherwise} \end{cases} . \tag{2.6}$$

To compute the Legendre-Fenchel transformation of the Huber norm, both cases have to be calculated separately. The first case is called f_1 and the second case is called f_2.

First, f_1 has to be inserted into (2.5) leading to

$$f_1^*(\mathbf{x}) = \sup_{\mathbf{x}\in\mathbb{R}^n} \{\mathbf{x}^\mathrm{T}\mathbf{p} - \frac{\|\mathbf{x}\|_2^2}{2\varepsilon}\} . \tag{2.7}$$

The derivative has to be calculated and then the equation has to be rearranged to solve for $\mathbf{x}$ which is equivalent to $f_1'^{-1}(\mathbf{p})$.

$$f_1'^{-1}(\mathbf{p}) = \mathbf{x} = \varepsilon\mathbf{p} \tag{2.8}$$

The tangent at point $(\mathbf{x}, f_1(\mathbf{x}))$ is then defined as

$$f_1(\mathbf{x}) = \mathbf{x}^\mathrm{T}\mathbf{p} - c . \tag{2.9}$$

By replacing $\mathbf{x}$ with $f_1'^{-1}(\mathbf{p})$, the equation becomes

$$f_1(f_1'^{-1}(\mathbf{p})) = f_1'^{-1}(\mathbf{p})^\mathrm{T}\mathbf{p} - c . \tag{2.10}$$

(2.8) is inserted into (2.10). Rearranging the equation to solve for c leads to the Legendre-Fenchel transformation of f_1

$$f_1^*(\mathbf{p}) = c = \frac{1}{2}\varepsilon\|\mathbf{p}\|_2^2 . \tag{2.11}$$

The condition for the first case in (2.6) has to be changed to be dependent on $\mathbf{p}$. Therefore, (2.8) is inserted into the condition leading to $\|\mathbf{p}\|_2 \leq 1$.

The Legendre-Fenchel transformation of the L^1 norm, that is used in f_2, is given in [6] as

$$f_{\mathrm{L}^1}^*(\mathbf{p}) = \begin{cases} 0 & \text{if } \|\mathbf{p}\| \leq 1 \\ \infty & \text{otherwise} \end{cases} . \tag{2.12}$$

Taking into account the condition space for f_2, only the second case of (2.12) has to be used.

The complete Legendre-Fenchel transformation of the Huber norm is then given as

$$f^*(\mathbf{p}) = \begin{cases} \frac{1}{2}\varepsilon\|\mathbf{p}\|_2^2 & \text{if } \|\mathbf{p}\|_2 \leq 1 \\ \infty & \text{otherwise} \end{cases} . \tag{2.13}$$

2.2 Photometric Invariants

When examining the abdomen with a mini-laparoscope, the only light source is mounted at the tip of the mini-laparoscope. This fact leads to a varying illumination as the light source moves with the mini-laparoscope. When doing 3D reconstruction, features in one image have

to be matched to features in another image. In order to be able to match these features in mini-laparoscopic sequences, it is necessary to deal with the changing illumination.

One possibility is to model the light source and surface reflectance. This results in models with a high amount of parameters. Debevec et al. used a special setup and more than 2000 frames to estimate the reflectance model of a face [3]. This is even more difficult inside the abdomen.

Another possibility is to transform the colors from the illumination-variant RGB color space into an illumination-invariant color space. When doing this, it is not necessary to estimate any parameters. This section refers to [17].

To understand the meaning of photometric invariants, first the dichromatic reflection model has to be introduced. A color at point $\mathbf{x} = (x,y)^{\mathrm{T}}$ is defined as

$$\mathbf{c}(\mathbf{x}) = (R(\mathbf{x}), G(\mathbf{x}), B(\mathbf{x}))^{\mathrm{T}} . \tag{2.14}$$

It can also be defined as a combination of the interface reflection component $\mathbf{c}_i$ and body reflection component $\mathbf{c}_b$

$$\mathbf{c}(\mathbf{x}) = \mathbf{c}_i(\mathbf{x}) + \mathbf{c}_b(\mathbf{x}) . \tag{2.15}$$

The interface reflection component describes specularities or highlights. The body reflection component describes the reflectance of the matte body.

If spectrally uniform illumination is assumed, (2.15) can be further decomposed into

$$\mathbf{c}(\mathbf{x}) = e(m_i(\mathbf{x})\widehat{\mathbf{c}}_i(\mathbf{x}) + m_b(\mathbf{x})\widehat{\mathbf{c}}_b(\mathbf{x})) , \tag{2.16}$$

with e being the overall intensity, $m(\mathbf{x})$ being the geometrical reflection factor and $\widehat{\mathbf{c}}(\mathbf{x})$ being the reflectance color. As spectrally uniform illumination is assumed, all three channels of $\widehat{\mathbf{c}}_i(\mathbf{x})$ have to be equal. $\widehat{\mathbf{c}}_i(\mathbf{x})$ will then be called $\mathbf{w}_i(\mathbf{x})$.

If also neutral interface reflection is assumed, $\mathbf{w}_i(\mathbf{x})$ becomes independent of $\mathbf{x}$. Then (2.16) becomes

$$\mathbf{c}(\mathbf{x}) = e(m_i(\mathbf{x})w_i\mathbf{1} + m_b(\mathbf{x})\widehat{\mathbf{c}}_b(\mathbf{x})) , \tag{2.17}$$

with $\mathbf{1}$ being the vector $(1,1,1)^{\mathrm{T}}$. Using (2.17), photometric invariance can be defined as a color $\mathbf{c}(\mathbf{x})$ being invariant to at least one parameter e, m_b or m_i.

There are three different classes of photometric invariance characterizing the independence on these three parameters. The first one is only independent on e and therefore handles global multiplicative illumination changes. The second class is independent on e and m_b. This is true at least for matte surfaces ($m_i = 0$) and handles shadow and shading. The last class is independent on all three parameters and handles highlights and specular reflections.

There are different strategies to get photometric invariants. Basically, these are normalization techniques, log-derivatives and transformations to other color spaces. In this thesis, the spherical transformation into the $r\phi\theta$ color space is used. It is defined as

$$(R,G,B)^{\mathrm{T}} \mapsto \begin{cases} r = \sqrt{R^2+G^2+B^2} \\ \theta = \arctan\left(\dfrac{G}{R}\right) \\ \phi = \arcsin\left(\dfrac{\sqrt{R^2+G^2}}{\sqrt{R^2+G^2+B^2}}\right) \end{cases} . \tag{2.18}$$

In (2.18), θ and ϕ are invariant to shadow and shading and r is no photometric invariant.

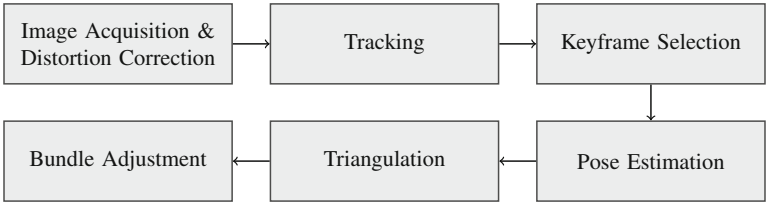

Figure 2.1: Overview of the sparse reconstruction. The necessary steps for the Structure from Motion approach are shown.

2.3 Sparse Reconstruction

When doing 3D reconstruction from a sequence of images, the scene and the camera positions have to be estimated. It is also possible to estimate the intrinsic parameters of the camera, but in this thesis calibrated cameras are used. This has the advantage of becoming more robust because less parameters have to be estimated. A Structure from Motion approach, based on the one proposed by Pollefeys et al. [21], is used to do the sparse reconstruction. It is called sparse reconstruction as it does not give 3D information for each pixel. It works with feature points that are good to track and only gives 3D information for these points.

Figure 2.1 shows the different steps of the Structure from Motion approach that is used. During the first step, the images are captured. Afterwards, a distortion correction is done. Within the second step the points, which are good for tracking, are located. Now these points are matched in different frames. After matching, a keyframe selection has to be done to decide which frames should be used for pose estimation and triangulation. During pose estimation, the positions of the cameras of the different frames will be estimated. The feature points are backprojected into the 3D space during the triangulation step and finally bundle adjustment is done to optimize the positions of the cameras and the 3D points.

2.3.1 Tracking

For reconstructing the camera positions and 3D points, it is necessary to estimate the motion of the image points between two frames first. Therefore, distinctive points have to be found in the frames. These points are called keypoints. From these keypoints, vectors can be extracted that describe the keypoints. These vectors are called descriptors. The descriptors can be matched together to receive the motion of the pixels. This motion can be used to estimate the motion of the cameras later. The Scale Invariant Feature Transform (SIFT) is used to get the keypoints and the descriptors and was introduced by Lowe [12]. SIFT features are invariant to scale and rotation. In addition they are robust to changes in illumination and changes in the viewpoint.

To get robust results of the following steps, long trajectories of matching keypoints over the sequence are important. It is possible that keypoints are outside the viewpoint for a few frames or that single uninformative frames during the sequence appear. These circumstances can become a problem if the matching is only be done between consecutive frames. The trajectories

will then end at these frames. Therefore, matching is not only done with the directly preceding frame, but also with some more preceding frames. This could lead to the problem that more than one point has to be inserted into the same trajectory. Schlüter et al. described four strategies to deal with that problem [22]. The strategy *New Trajectory for All* is used in this thesis as Schlüter et al. have also shown that this is the best strategy. It gives the highest absolute number of trajectories and also the highest relative number of correct trajectories. In case of a conflict, none of the conflicting keypoints are inserted into the trajectory and new trajectories are created for all of them.

2.3.2 Keyframe Selection and Pose Estimation

The correspondences resulting from the tracking are noisy. If the motion of the keypoints is very small between two frames, the assumed motion due to noise can be bigger than the real motion of the keypoints. Therefore, the motion of the keypoints between two frames has to be as big as possible to increase the signal-to-noise ratio and to enable a robust estimation of the camera positions.

In addition to the distance of the keypoints, it is also important to have enough correspondences. Normally, five to eight correspondences are sufficient, depending on the algorithm that is chosen for pose estimation, but besides noise rigidity is another problem. These algorithms perform pose estimation for rigid transformations. The liver in case of cirrhosis is rigid, but the deformation of the abdomen during respiration is non-rigid. To handle noise and rigidity, a Random Sample Consensus (RANSAC) approach, as first described by Fischler et al. [4], is used. During RANSAC, a set of points is chosen to be inliers and a corresponding model is estimated. Then all points are checked whether they can be considered being inliers to this model. All inliers are then called the consensus set. The model is taken as a candidate if the number of points in the consensus set is big enough. These steps are repeated a fixed number of times. Models with a greater number of points in the consensus set are taken as the new candidate. In the end, the model with the biggest number of points in the consensus set is taken as the true model and outliers of this model can be discarded. In order to use this RANSAC approach, more correspondences than the minimum number for the algorithm are required. Therefore, besides the condition of the minimum distance of the keypoints, there is also a condition of a minimum number of correspondences.

In this thesis, a simple algorithm for keyframe selection is chosen. Each new frame is compared to the last keyframe. If the motion of the keypoints between these two frames and the number of correspondences are both higher than a threshold, the frame is considered as being a keyframe. Estimation of the camera poses as well as triangulation is only done using the chosen keyframes.

The estimation of the motion between the keyframes is done by using the five-point-algorithm as described by Nistér [19]. This algorithm estimates an essential matrix E between the two frames that is decomposed into a rotation matrix R and a translation vector $\mathbf{t}$ as also shown in Nistér's paper. First, the matrix

$$\mathrm{D} = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.19}$$

is defined. Then the essential matrix E is decomposed via singular value decomposition into $\mathrm{U}\mathrm{diag}(1,1,0)\mathrm{V}^{\mathrm{T}}$. U and V are chosen in a way that their determinant is greater than zero. This decomposition is defined up to scale. The translation vector is

$$\mathbf{t} \sim [u_{13}\ u_{23}\ u_{33}]^{\mathrm{T}} . \tag{2.20}$$

For the rotation matrix there are the two possible solutions, namely

$$\mathrm{R}_a = \mathrm{U}\mathrm{D}\mathrm{V}^{\mathrm{T}} \tag{2.21}$$

and

$$\mathrm{R}_b = \mathrm{U}\mathrm{D}^{\mathrm{T}}\mathrm{V}^{\mathrm{T}} . \tag{2.22}$$

The four possible transformation matrices are $\mathrm{P}_A = [\mathrm{R}_a|\mathbf{t}]$, $\mathrm{P}_B = [\mathrm{R}_a|-\mathbf{t}]$, $\mathrm{P}_C = [\mathrm{R}_b|\mathbf{t}]$ and $\mathrm{P}_D = [\mathrm{R}_b|-\mathbf{t}]$, but only one is physically valid. In this valid configuration, the points are in front of both cameras. In the other three solutions, the points are behind one or both of the cameras.

There can be problems with motions that are nearly sidewards as von Öhsen et al. have shown [24]. They have also proposed an algorithm to handle these problems. The first step is to check whether the motion is sidewards. This is done by evaluating the inequality

$$\|\mathrm{I}-\mathrm{R}\|_F \leq 0.05 , \tag{2.23}$$

with $\|\cdot\|_F$ being the Frobenius norm and I being the identity matrix. If this inequality is fulfilled, the correct direction $\tilde{\mathbf{t}}$ of the translation vector is determined as

$$\tilde{\mathbf{t}} = \begin{cases} \mathbf{t} & \text{if median}(\mathbf{x}-\mathbf{x}') > 0 \\ -\mathbf{t} & \text{if median}(\mathbf{x}-\mathbf{x}') \leq 0 \end{cases} . \tag{2.24}$$

2.3.3 Triangulation

The next step after estimating the camera positions is to reconstruct the 3D points. When the camera parameters are known, it is possible to exactly calculate the point in the image plane $\mathbf{x}$ where a 3D point $\mathbf{X}$ is being projected to by calculating

$$\mathbf{x} = \mathrm{K}\mathrm{P}\mathbf{X} . \tag{2.25}$$

$\mathrm{K} \in \mathbb{R}^{3\times3}$ contains the intrinsic camera parameters and is defined as

$$\mathrm{K} = \begin{pmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{pmatrix} , \tag{2.26}$$

with $\alpha_x = fm_x$ and $\alpha_y = fm_y$ being the focal lengths in pixel dimensions in x- and y-direction, $(x_0, y_0)^{\mathrm{T}}$ being the principal point in pixel dimensions and s being the skew factor. $\mathrm{P} \in \mathbb{R}^{3\times4}$ contains the extrinsic camera parameters and is defined as

$$\mathrm{P} = [\mathrm{R}|\mathbf{t}] , \tag{2.27}$$

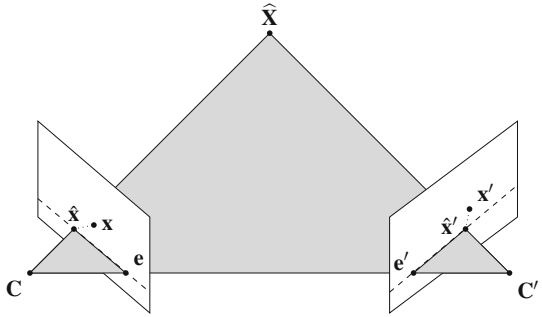

Figure 2.2: Overview of the epipolar geometry. The camera centers are located at $\mathbf{C}$ and $\mathbf{C}'$ and the epipoles are denoted by $\mathbf{e}$ and $\mathbf{e}'$. The 3D point $\widehat{\mathbf{X}}$ is projected into both images onto the points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$, respectively. These could be shifted to $\mathbf{x}$ and $\mathbf{x}'$ due to noise. In both image planes the epipolar lines are given by a dashed line.

with $\mathrm{R} \in \mathbb{R}^{3\times3}$ being the rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ being the translation vector. Details about the camera parameters can be found in [7].

It can also be seen by the matrix dimensions that the opposite problem, to find a 3D point corresponding to an image point, can not be solved unambiguously. It is only possible to determine a line on which the point has to be. Therefore, a corresponding point in another image has to be known. Figure 2.2 illustrates this situation. If the two cameras and the image points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are known, it is possible to find the 3D point $\widehat{\mathbf{X}}$ by constructing the lines through the camera centers and the image points, and determining the intersection of these two lines. This is called triangulation.

The problem is that the correspondences are noisy and therefore the lines do not intersect. In order to have an intersection, the epipolar constraint

$$\hat{\mathbf{x}}'^{\mathrm{T}}\mathrm{F}\hat{\mathbf{x}} = 0 \tag{2.28}$$

has to be fulfilled. F is the fundamental matrix and is related to the essential matrix E by

$$\mathrm{E} = \mathrm{K}'^{\mathrm{T}}\mathrm{F}\mathrm{K}\,. \tag{2.29}$$

The case of noisy correspondences is illustrated by the points $\mathbf{x}$ and $\mathbf{x}'$ in Figure 2.2. These points do not fulfill (2.28). Hartley and Zisserman proposed a method to correct these correspondences in order to fulfill the epipolar constraint and to enable a simple triangulation method [7]. They minimize the function

$$C(\mathbf{x},\mathbf{x}') = d(\mathbf{x},\hat{\mathbf{x}})^2 + d(\mathbf{x}',\hat{\mathbf{x}}')^2 \tag{2.30}$$

with respect to the epipolar constraint. The function $d(\cdot,\cdot)$ denotes the euclidean distance between the two points. A pair of points, that fulfills the epipolar constraint, has to lie on two corresponding epipolar lines $\mathbf{l}$ and $\mathbf{l}'$. In Figure 2.2, the epipolar lines are given by dashed lines.

All points on these lines will fulfill the epipolar constraint, but only the ones being nearest to $\mathbf{x}$ and $\mathbf{x}'$ will minimize the function $C(\mathbf{x},\mathbf{x}')$. Therefore, (2.30) can be written as

$$C(\mathbf{x},\mathbf{x}') = d(\mathbf{x},\mathbf{l})^2 + d(\mathbf{x}',\mathbf{l}')^2 \,, \tag{2.31}$$

where $d(\mathbf{x},\mathbf{l})$ now denotes the perpendicular distance of the point $\mathbf{x}$ to the line $\mathbf{l}$. The lines $\mathbf{l}$ and $\mathbf{l}'$ can be all possible pairs of epipolar lines. The points $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}'$ are then chosen to be the closest points to $\mathbf{x}$ and $\mathbf{x}'$ on the lines. If the pencil of epipolar lines is parametrized by t, (2.31) becomes

$$\min_t \; C(\mathbf{x},\mathbf{x}') = d(\mathbf{x},\mathbf{l}(t))^2 + d(\mathbf{x}',\mathbf{l}'(t))^2 \,. \tag{2.32}$$

Details on the minimization can be found in [7].

2.3.4 Bundle Adjustment

After estimating the camera positions and 3D points, an optimization step has to be done. For each 3D point, there are many corresponding 2D points in different frames because of the trajectories found during tracking. Normally, for each 2D point

$$\mathbf{x}_j^i = \mathrm{P}^i \mathbf{X}_j \tag{2.33}$$

has to be fulfilled, but due to noise this is not the case. Therefore, projection matrices $\hat{\mathrm{P}}^i$ and 3D points $\widehat{\mathbf{X}}_j$ that fulfill (2.33) and minimize the geometric error between the reprojected image point and the measured image point have to be found. This leads to

$$\min_{\hat{\mathrm{P}}^i,\widehat{\mathbf{X}}_j} \sum_i \sum_j d(\hat{\mathrm{P}}^i \widehat{\mathbf{X}}_j, \mathbf{x}_j^i)^2 \,, \tag{2.34}$$

where $d(\cdot,\cdot)$ denotes the geometric image distance between the two points. This minimization is called bundle adjustment and can be found in [7].

2.4 Dense Reconstruction

The sparse reconstruction gives 3D points for features that are suitable to track. If there are no features in a region of interest, there will also be no 3D information about this region. The dense reconstruction, in contrast to the sparse reconstruction, gives depth information for each pixel of a reference frame and therefore 3D information for each pixel. The dense reconstruction is done by using a variational approach and is handled in detail in this section. This section refers to [18].

2.4.1 Variational Approach

In a variational approach, the solution of a problem is considered as an energy functional and this energy functional has to be minimized. Horn and Schunck are considered to be the first to

use a variational approach in computer vision [8]. They applied it to compute the optical flow between two consecutive images in a sequence. Wedel and Cremers refined this approach in [25]. Newcombe et al. used a variational approach to do a dense reconstruction [18].

Newcombe et al. proposed a method to compute an inverse depth map ξ. It is called inverse because the values of this map are the inverse of the depth. They formulated the energy E_ξ, which has to be minimized to compute the inverse depth map, as

$$E_\xi = \int_\Omega \{g(\mathbf{u})\|\nabla\xi(\mathbf{u})\|_\varepsilon + \lambda C(\mathbf{u},\xi(\mathbf{u}))\}\mathrm{d}\mathbf{u}\,. \tag{2.35}$$

This equation consists of a regularizer and a cost volume. The regularizer is given by $\|\nabla\xi(\mathbf{u})\|_\varepsilon$ and the cost volume is given by $C(\mathbf{u},\xi(\mathbf{u}))$. The cost volume contains the data terms and will be defined in detail in Section 2.4.2. It is weighted by the factor λ. In the regularizer, $\|\cdot\|_\varepsilon$ denotes the Huber norm, which is defined in (2.6). The regularizer is weighted for each pixel by the function $g(\mathbf{u})$, which is defined as

$$g(\mathbf{u}) = e^{-\alpha\|\nabla \mathrm{I}_r(\mathbf{u})\|_2^\beta}\,, \tag{2.36}$$

where $\mathrm{I}_r(\mathbf{u})$ is the intensity of the pixel at location $\mathbf{u}$. This weighting function decreases the influence of the regularizer at edges so that they can remain in the inverse depth map.

2.4.2 Cost Volume

The cost volume $C(\mathbf{u},\xi(\mathbf{u}))$ encapsulates the data terms and therefore gives information about the image content. It is a three-dimensional space, that maps each depth sample d for each pixel $\mathbf{u}$ of a reference frame r to costs for this tuple. The computation is done as

$$C_r(\mathbf{u},d) = \frac{1}{N_{r,\mathrm{inside}}(\mathbf{u},d)} \sum_{m\in\mathcal{I}(r)} \|\rho_r(\mathrm{I}_m,\mathbf{u},d)\|_1\,. \tag{2.37}$$

For each pixel $\mathbf{u}$ and each depth sample d, the backprojected 3D points are projected into all other frames nearby, $\mathcal{I}(r)$, and the L^1 norms of the photometric errors $\rho_r(\mathrm{I}_m,\mathbf{u},d)$ are summed up. The number of projections that are inside the other frames is denoted as $N_{r,\mathrm{inside}}(\mathbf{u},d)$ and I denotes the image. Figure 2.3 illustrates the calculation of the cost volume.

For the computation of the cost volume, the photometric error is used that is defined as

$$\rho_r(\mathrm{I}_m,\mathbf{u},d) = \mathrm{I}_r(\mathbf{u}) - \mathrm{I}_m(\pi(\mathrm{K}\mathrm{T}_{rm}\pi^{-1}(\mathbf{u},d)))\,, \tag{2.38}$$

where T_{rm} denotes the transformation of a point in the coordinate system of camera r into the coordinate system of camera m. The dehomogenisation of a point $\mathbf{x} = (x,y,z)^\mathrm{T}$ is given as $\pi(\mathbf{x}) = (x/z, y/z)^\mathrm{T}$ and $\pi^{-1}(\mathbf{u},d) = \frac{1}{d}\mathrm{K}^{-1}\mathbf{u}$ is the backprojection of an image point into the 3D space. In (2.38), the data terms of the images at a pixel are compared. Newcombe et al. used gray values as data terms but they are not invariant to illumination changes. Marcinczak et al. used spherical data terms in order to cope with illumination changes [14]. Therefore, the $r\phi\theta$ color space is used. The transformation into this color space can be found in (2.18). When applying the spherical transformation, the photmetric error becomes

$$\rho_r(\mathrm{I}_m,\mathbf{u},d) = \begin{bmatrix}\theta_r(\mathbf{u}) - \theta_m(\pi(\mathrm{K}\mathrm{T}_{rm}\pi^{-1}(\mathbf{u},d)))\\ \phi_r(\mathbf{u}) - \phi_m(\pi(\mathrm{K}\mathrm{T}_{rm}\pi^{-1}(\mathbf{u},d)))\end{bmatrix}\,. \tag{2.39}$$

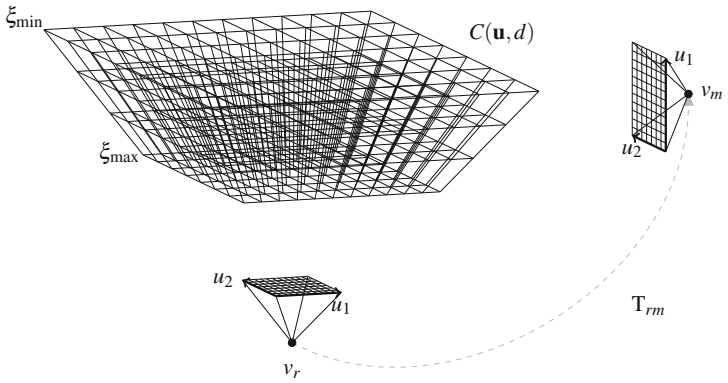

Figure 2.3: Illustration of the procedure to compute the cost volume. All pixels of the reference frame v_r are backprojected into the 3D space for all depth samples d between $\xi_{\min}$ and $\xi_{\max}$. In this example, there are three depth samples. This voxel volume is then projected into another frame v_m. The values of the pixel in the reference frame and the pixel in the other frame are then taken to compute the cost volume. This figure is taken from [14].

As the component r of the $r\phi\theta$ color space is no photometric invariant, it is not used in (2.39).

An inverse depth map can be constructed from the cost volume by using the inverse depth value with the minimum costs for each pixel. Figure 2.4 (a) shows such a resulting inverse depth map. It can be seen that this inverse depth map is very noisy. Figure 2.4 (b) shows the photometric error for the pixel $\mathbf{u}$ that is marked in Figure 2.4 (a). Ideally, there should be only one distinct minimum, but in regions with low information this is not the case and leads to the noise in the resulting inverse depth map. Therefore, some regularization has to be done on the inverse depth map as described in Section 2.4.4.

2.4.3 Coupling to Sparse Reconstruction

The dense reconstruction needs information from the sparse reconstruction. On the one hand, the camera positions have to be known. For the construction of the cost volume, the transformation from the reference frame to the frames nearby has to be known. This knowledge is inevitable. When there is no information about the scene so far, this information comes from the sparse reconstruction. Later, it can also come from tracking in the dense domain.

On the other hand, the 3D points from the sparse reconstruction can be used to guide the dense reconstruction. Tests have shown that coupling the dense reconstruction to the result of the sparse reconstruction gives considerably better results. However, there have been no quantitative evaluations. There are two possibilities to couple the dense reconstruction to the

Springer

http://www.springer.com/978-3-658-12697-1

Variation Based Dense 3D Reconstruction
Application on Monocular Mini-Laparoscopic Sequences

Painer, S.

2016, XV, 78 p. 34 illus., Softcover
ISBN: 978-3-658-12697-1