Computer Vision Winter Workshop 2009, Adrian Ion and Walter G. Kropatsch (eds.)
Eibiswald, Austria, February 4–6
Publisher: PRIP, Vienna University of Technology, Austria

Depth Map Fusion with Camera Position Refinement

Radim Tyleček and Radim Šára

Center for Machine Perception
Faculty of Electrical Engineering,
Czech Technical University, Czech Republic
tylecr1@cmp.felk.cvut.cz, sara@cmp.felk.cvut.cz,

Abstract *We present a novel algorithm for image-based surface reconstruction from a set of calibrated images. The problem is formulated in Bayesian framework, where estimates of depth and visibility in a set of selected cameras are iteratively improved. The core of the algorithm is the minimisation of overall geometric L_2 error between measured 3D points and the depth estimates.*
In the visibility estimation task, the algorithm aims at outlier detection and noise suppression, as both types of errors are often present in the stereo output. The geometrical formulation allows for simultaneous refinement of the external camera parameters, which is an essential step for obtaining accurate results even when the calibration is not precisely known. We show that the results obtained with our method are comparable to other state-of-the-art techniques.

1 Introduction

Reconstruction of a 3D model from multiple views has been an active research field in the past years. A wide range of methods with excellent results exist, but there is still a lot of space where the results can be improved, specifically in terms of accuracy and completeness on large and complex scenes. A suitably chosen representation, also depending on the application, is in the core of every reconstruction algorithm. Following the taxonomy of [14], we can divide the representations in four major groups.

Voxel-based or volumetric representations work with a 3D grid, where scene objects are indicated by an occupancy function defined on every grid cell. A successful recent representative is [20], however this representation is in practice limited to smaller closed objects, due to a low scalability of memory requirements, which can be improved by use of an octree only when an initial solution is known. Interesting GPU implementation of volumetric range image fusion [21] shows the speed is important for some applications.

Similarly, the *level-set* representations define a scalar function on the 3D grid, where its zero-crossing indicates the object surface. Again, such representations suit well for closed objects only [7].

The last 3D representation is a *polygonal mesh*, usually in the form of connected triangular facets. This representation is commonly used in computer graphics as it allows efficient rendering, therefore other representations are usually converted to a mesh at a particular stage of the reconstruction process. Alternatively, rectangular patches can be used instead of triangles, but in this case the connectivity is not implicitly defined. Currently best performing method in this area is [8], where initial matched features are grown in space. After the transformation of patches into triangle mesh with [12], its vertices are photometrically refined.

Finally, representation with *depth maps* is a different approach, where depth values are assigned to all pixels of a set of images, which allows seamless handling of the input data from both passive and active sensors. It leads to integration of the data in the image space, and such process can be described as *depth map fusion*. The drawback of depth map representation lies in the fact it is not intrinsic, as it works only with a projection of the 3D surface. Also the final step of creating a 3D mesh requires more effort when compared to a direct 3D representation. This representation was first used by [18] for global estimation of depth maps, later Strecha used it in his two algorithms: In [16] he first uses the probabilistic EM [5] algorithm to jointly estimate depths and visibility, which is supplied with hidden Markov Random Field in [17] to model inlier and outlier processes generating the images. However, the methods do not scale well and the accuracy of the second algorithm is lower due to the discretisation of depths.

We have chosen this representation, because it is simple and can easily exploit information available in images during the process of depth map fusion. It also suits well in large outdoor scenes, where 3D methods are difficult to apply. Goesele [10] and Bradley [4] recently proved that this concept works, however the results of [10] are not complete. In [4] they use the visual hull and ordering constraints, which limits the application to indoor objects. In contrast, our interest is also in outdoor scenes, where the background segmentation is not available. A recent method focusing on large-scale scenes was presented by [13] and uses a different representation. It builds an adaptive tetrahedral decomposition of matched keypoints using 3D Delaunay triangulation. Occupancy function on this graph is computed as optimal labelling by solving a minimum cut problem.

The current challenge of obtaining more complete and accurate results has reached the level where the accurate camera calibration is essential for further improvement. Furukawa [9] has recently taken this into account, when he iteratively switches between the camera calibration with a stan-

dard bundle adjustment and recovering of the scene with refined camera parameters. Our incorporation of this problem is different: geometric constraints allow us to solve jointly for both depths and camera centres.

2 Algorithm overview

The input to the proposed algorithm is a set of images $\mathcal{I} = \left\{\mathbf{I}_p^i \mid i = 1,\dots,c; p = 1,\dots,n^i\right\}$, where c is the number of cameras and n^i is the number of pixels in image i. The possibly inaccurate camera calibration $\mathcal{P} = \left\{\mathbf{P}^i \mid i = 1,\dots,c\right\}$ is obtained from a reconstruction pipeline [6]. Disparities are then computed on rectified image pairs with a publicly available dense matching stereo algorithm GCS [3]. With triangulation we obtain a point cloud $\mathcal{X}$ in the form of pair-wise disparity maps back-projected to space.

The goal is to get Bayesian estimate of depth maps $\Lambda = \left\{\lambda_p^i \mid i = 1,\dots,c; p = 1,\dots,n^i\right\}$, where $\lambda_p^i \in \mathbb{R}$ is a reconstructed depth in pixel p of image i, and visibility $V = \left\{v_p^i \mid i = 1,\dots,c; p = 1,\dots,n^i\right\}$, where $v_p^i \in \{0,1,2\}$ is the visibility of pixel p of image i in all cameras $i = 1,\dots,c$ such that $v_p^i = 0$ marks invisible and $v_p^i > 0$ visible pixels; exact meaning will be described later. The task leads to the maximisation of the posterior probability, which can be formally written as

$$(\mathcal{X}^*, \Lambda^*, V^*, \mathcal{C}^*) = \arg\max_{\mathcal{X},\Lambda,V,\mathcal{C}} P(\mathcal{X}, \Lambda, V, \mathcal{C} \mid \mathcal{I}). \quad (1)$$

The intended output is (Λ^*, V^*) while the estimation of $(\mathcal{C}^*, \mathcal{X}^*)$, where $\mathcal{C}$ is a set of camera centres, should be interpreted as an effort to repair the input data. Because of the presence of joint probabilities it is necessary to decompose the problem. The solution algorithm alternates between two sub-problems conditioned on each other: estimation of $(\Lambda, \mathcal{C}, \mathcal{X})$ and V. The output of the first subproblem is used as the input to the second, and vice versa. Internally, the subproblem of estimation of $(\Lambda, \mathcal{C}, \mathcal{X})$ is also divided, so that the result of the optimisation task of $(\Lambda, \mathcal{C})$ is used to repair the corresponding points $\mathcal{X}$. This proposal is a modification of EM algorithm [5], inspired by [16], where visibility V corresponds to the hidden variables.

With k as the number of iteration, the overall iterative procedure can be described as

1. Input corresponding points $\mathcal{X}$ and cameras $\mathcal{P}$ are given. A subset of cameras, where the depth and visibility maps will be estimated, is manually chosen.

2. In all cameras $i = 1, 2, \dots, c$, initialise visibility maps $V(0)$ and depths maps $\Lambda(0)$ from input data $\mathcal{X}$.

3. Solve the visibility estimation task (18) and get estimate of visibility V^*. Update the current value of visibility maps $V(k)$ to $V(k+1) := V^*$.

4. Solve the depth estimation task (4) and get estimate of depths $(\Lambda^*, \mathcal{C}^*)$. Update the value of depths $\Lambda(k)$ to $\Lambda(k+1) := \Lambda^*$. Update the value of camera centres $\mathcal{C}(k)$ to the new position $\mathcal{C}(k+1) := \mathcal{C}^*$.

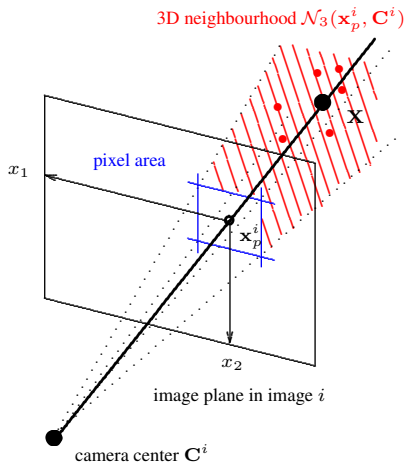

Figure 1: 3D neighbourhood. Red dots represent matching correspondences $\mathbf{X}_{pq}^{ij}$.

5. Correct the positions of $\mathcal{X}$ based on updated cameras $\mathcal{C}$ and depth estimates Λ.

6. Repeat Steps 3–5 for a given number of iterations.

7. Project depth maps to 3D space to obtain cloud of points with normals estimated from points neighbouring in the image.

8. Merge the points into continuous surface with PSR [12] and filter the result to remove introduced big triangles based on average edge size.

The individual tasks of depth and visibility estimation will be now discussed in more detail. A full description of the algorithm is given in [19].

3 Depth estimation

In this section, the highly redundant input point cloud will be integrated in the image space. The idea allowing it is represented by a *3D neighbourhood* $\mathcal{N}_3$, see Figure 1. Given image i and pixel p, we define $\mathcal{N}_3(\mathbf{x}_p^i, \mathbf{C}^i)$ as a pyramidal region in the vicinity of an image ray defined by pixel area with centre $\mathbf{x}_p^i$ and camera centre $\mathbf{C}^i$. The set of points lying in this region will be denoted as

$$\chi_p^i = \left\{\mathbf{X}_{pq}^{ij} \mid \mathbf{X}_{pq}^{ij} \in \mathcal{N}_3(\mathbf{x}_p^i, \mathbf{C}^i)\right\}, \quad (2)$$

where $j \in \{1,\dots,c\}; q \in \{1,\dots,n^j\}$ and $\mathbf{X}_{pq}^{ij} \in \mathbb{R}^3$ is a point in space computed from correspondence between pixel p in camera i and pixel q in camera j. The set of all corresponding points $\mathcal{X}$ (measurement) coming from disparity maps can be then parametrised as

$$\mathcal{X} = \left\{\chi_p^i \mid i = 1,\dots,c; p = 1,\dots,n^i\right\}, \quad (3)$$

where χ_p^i is a set of correspondences of pixel p in image i from all disparity map pairs ij. Note that by this choice our discretisation of the image space is given: we work with natural pixel resolution.

Now we can formulate the depth estimation task: given measurements $\mathcal{X}$ and visibility V, we search for the estimate

$$(\Lambda^*, \mathcal{C}^*) = \arg\max_{\Lambda,\mathcal{C}} P(\mathcal{X} \mid \Lambda, \mathcal{C}, V)\, P(\Lambda, \mathcal{C}, V). \quad (4)$$

The solution of the problem does not depend on $P(\mathcal{X}, V)$.

3.1 Projection constraints

Probability $P(\mathcal{X} \mid \Lambda, \mathcal{C}, V)$ from (4) can be expressed as

$$P(\mathcal{X} \mid \Lambda, \mathcal{C}, V) = \prod_{i=1}^{c} \prod_{p=1}^{n} p(\chi_p^i \mid \lambda_p^i, \mathcal{C}, V), \quad (5)$$

where χ_p^i is the set of correspondences projecting to pixel p in image i and λ_p^i is estimated depth at this point. We choose

$$p(\chi_p^i \mid \lambda_p^i, \mathcal{C}, V) = \begin{cases} \frac{1}{T_\lambda}\, e^{-\frac{\left(\bar{\lambda}_p^i - \lambda_p^i\right)^2}{2\,\sigma_\lambda^2}} & \text{if } v_p^i = 2, \\ 1 & \text{otherwise,} \end{cases} \quad (6)$$

where $T_\lambda = \sigma_\lambda \sqrt{2\pi}$ and $\bar{\lambda}_p^i = \bar{\lambda}(\chi_p^i, \mathcal{C})$ is a depth estimating function from the set of all correspondences χ_p^i. It is computed as a result of the least squares minimisation subtask

$$\bar{\lambda}(X_p^i, \mathcal{C}) = \arg\min_{\bar{\lambda}_p^i} \sum_{(j,q)\in\chi_p^i;\; v_q^j \geq 1} \|\bar{\mathbf{X}}_p^i - \mathbf{X}_{pq}^{ij}\|^2, \quad (7)$$

where j, q are all correspondences visible also in the corresponding cameras j and $\bar{\mathbf{X}}_p^i = \phi(\mathbf{x}_p^i, \bar{\lambda}_p^i, \mathbf{C}^i)$ is a back-projection, assigning a point in space $\bar{\mathbf{X}}_p^i$ to the depth $\bar{\lambda}_p^i$ and image point $\mathbf{x}_p^i$ in the projective camera with centre $\mathbf{C}^i$:

$$\bar{\mathbf{X}}_p^i = \mathbf{C}^i + \bar{\lambda}_p^i\, (\mathbf{R}^i)^\top (\mathbf{K}^i)^{-1} \mathbf{x}_p^i, \quad (8)$$

where $\mathbf{R}^i$ is rotation of the camera and $\mathbf{K}^i$ are internal camera parameters. Similarly $\mathbf{X}_{pq}^{ij} = \phi(\mathbf{x}_q^j, \lambda_q^j, \mathbf{C}^j)$.

Let us build a system of projective equations for all such correspondences in $(\chi_p^i \mid V)$. The part $\|\cdot\|^2$ in (7) minimises the L^2 distance between back-projected points and the estimate.

The equation for one correspondence pair (j, q) according to Figure 2 comes from the necessary condition for the minimum of (7), which is the equality of points $\bar{\mathbf{X}}_p^i = \mathbf{X}_{pq}^{ij}$. With decomposition of the camera matrices[1], the resulting constraint becomes

$$\mathbf{R}^{j\,(3)}\mathbf{C}^i - \mathbf{R}^{j\,(3)}\mathbf{C}^j + \bar{\lambda}_p^i\, \mathbf{R}^{j\,(3)}\, (\mathbf{R}^i)^\top (\mathbf{K}^i)^{-1} \mathbf{x}_p^i = \lambda_q^j \quad (9)$$

where $\cdot^{(3)}$ denotes a third row of a matrix, and $\bar{\lambda}_p^i, \mathbf{C}^i, \mathbf{C}^j$ are considered unknowns. Note that the use of geometric constraints allows us to include re-estimation of camera centres $\mathcal{C}$ in the task. This gives us one of equation from the set of linear equations of geometric constraints to $\bar{\lambda}_p^i$, forming an over-defined system.[2]

[1] See [19] for details.

[2] Only if there are more correspondences for the given pixel.

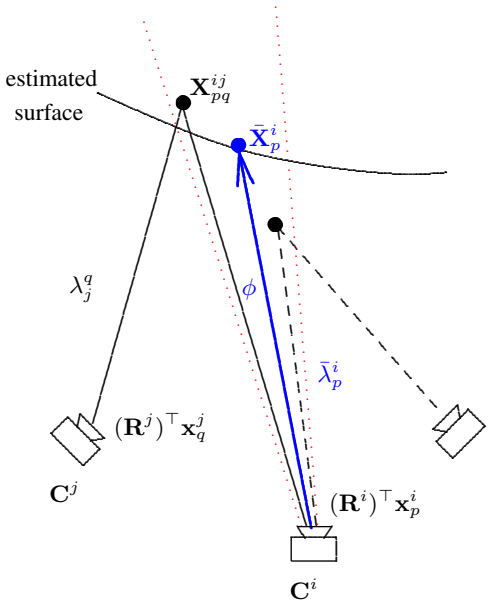

Figure 2: 2D view of a situation with two correspondences. Dotted lines are borders of 3D neighbourhood of pixel p in image i (red). $\bar{\mathbf{X}}_p^i$ and $\bar{\lambda}_p^i$ are two representations of the same point (blue).

3.2 Surface model

The surface model performs regularisation to smooth out the noisy data. Probability $P(\Lambda, V, \mathcal{C})$ from (4) can be written under the assumption of the Markov property as

$$P(\Lambda, V, \mathcal{C}) = \prod_{i=1}^{c} \prod_{(p,\bar{p})\in\mathcal{N}_2(i)} p(\lambda_p^i, \lambda_{\bar{p}}^i \mid v_p^i, v_{\bar{p}}^i)\, p(v_p^i, v_{\bar{p}}^i), \quad (10)$$

where $\mathcal{N}_2(i)$ is the set of all neighbouring pixel pairs $(p, \bar{p})$ in the image i (2D neighbourhood). The pairs are defined for all edges of the image grid, as can be seen in Figure 3.

The solution of task (4) does not depend on $p(v_p^i, v_{\bar{p}}^i)$, as it is fixed.

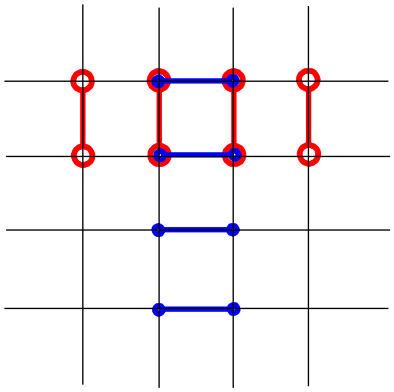

Figure 3: 2D neighbourhood $\mathcal{N}_2 \mid V$ on the image grid, pixels are at the intersection of grid lines, $v_p^i \geq 1$ where a dot is present. *Blue:* horizontal pairs in one column. *Red:* vertical pairs in one row.

The choice of $p(\lambda_p^i, \lambda_{\bar{p}}^i \mid v_p^i, v_{\bar{p}}^i)$ depends on the used model of surface, and the probability distribution of a general surface model can be described as

$$p(\lambda_p^i, \lambda_{\bar{p}}^i \mid v_p^i, v_{\bar{p}}^i) = \begin{cases} \frac{1}{T_\lambda}\, e^{-\frac{(\varepsilon_p^i)^2}{2\,(\sigma_{c,\bar{p}}^i)^2}} & \text{if } v_p^i \geq 1, v_{\bar{p}}^i \geq 1, \\ 1 & \text{otherwise,} \end{cases} \tag{11}$$

where $T_\lambda = \sigma_{c,\bar{p}}^i \sqrt{2\pi}$ and the value of $\sigma_{c,\bar{p}}^i$ is proportional to size of gradient of image function $\| \nabla \mathbf{I} \|^2$ at pixel p in image i, based on color difference of neighbouring pixels of I_p^i. This comes from the assumption that image regions with higher variance contain more information and the belief in the results of the stereo algorithm is higher, therefore we can penalise differences in λ less allowing the solution to follow data more closely. The difference ε^2 will be defined as

$$(\varepsilon_p^i)^2 = (\Phi_p^i - \Phi_{\bar{p}}^i)^2 \tag{12}$$

where Φ is a surface property, such as depth, normal or curvature, and Φ_p^i is the value of the property at given point p in image i.

For $\Phi \equiv \lambda$, the equation (11) describes continuity of order 0, this means all neighbouring depths are ideally equal, favouring locally a fronto-parallel plane. The first order continuity $\Phi \equiv \mathbf{N}$ forces the normals of surface in neighbouring points to be equal, which locally leads to a plane with arbitrary normal. The computation of the normal is complex, so we reduce problem to the constancy of the first derivative of the depth function λ along the image axes. This choice reduces the number and complexity of equations for the surface model at the cost of the loss of the intrinsicity of the normal constancy. After discretising partial derivatives we obtain the simplest approximation of the first order:

$$\varepsilon^2 = (\lambda_{-1,0} - 2\lambda_{0,0} + \lambda_{+1,0})^2 + (\lambda_{0,-1} - 2\lambda_{0,0} + \lambda_{0,+1})^2, \tag{13}$$

where we have used simplified notation $\lambda_{a,b} = \lambda(x+a, y+b)$. As a result, constancy of the depth is replaced by constancy of the gradient.

The second order means the constancy of the mean curvature along the surface, $\Phi \equiv \mathcal{H}$, and the difference (12) becomes $\varepsilon^2 = (\mathcal{H}_p^i - \mathcal{H}_{\bar{p}}^i)^2$. Following the same simplification scheme as for the first order, we obtain the following approximation for the second order:

$$\begin{aligned} \varepsilon^2 = &\left[\lambda_{-1,0} - 3\lambda_{0,0} + 3\lambda_{+1,0} - \lambda_{+2,0}\right]^2 + \\ &+ \left[\lambda_{0,-1} - 3\lambda_{0,0} + 3\lambda_{0,+1} - \lambda_{0,+2}\right]^2. \end{aligned} \tag{14}$$

In our implementation, we use the model of the second order on the majority of the visible surface, it improves the results as it better preserves surface features such as edges. However, the approximation requires certain visible area around a given pixel, so in practice we use models of zero and first orders near borders.

3.3 Energy minimisation

The problem of (4) can be factorised to individual camera contributions. After application of negative logarithm we get energy minimisation

$$(\Lambda^*, \mathcal{C}^*) = \arg\min_{\Lambda, \mathcal{C}} \sum_{i=1}^{c} E(\Lambda^i, \mathcal{C} \mid \mathcal{X}, V). \tag{15}$$

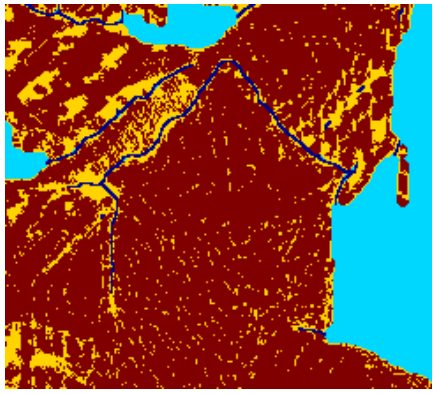

Figure 4: A visibility map. Red pixels $v_p^i = 2$ indicate presence of data points. Yellow pixels $v_p^i = 1$ indicate depth interpolation from neighbouring data. Blue pixels $v_p^i = 0$ mark invisible regions, which include dark blue pixels where discontinuity was detected.

Energy $E_\lambda^i = (\Lambda^i, \mathcal{C} \mid \mathcal{X}, V)$ of depths in camera i, from (15) with use of the first-order surface model is then

$$E_\lambda^i = \frac{1}{2\sigma_\lambda^2} \sum_{p=1}^{n} (\bar{\lambda}_p - \lambda_p)^2 + \sum_{(p,\bar{p}) \in \mathcal{N}_2(i|V)} \frac{1}{2\sigma_{c,p}^2} (\lambda_p - \lambda_{\bar{p}})^2, \tag{16}$$

where the first part expresses the data energy, the second part expresses the surface model energy and coefficients σ define their mutual weights.

The necessary condition for an extremum gives

$$\frac{1}{\sigma_\lambda^2} (\bar{\lambda}_p - \lambda_p) + \sum_{(p,\bar{p}) \in \mathcal{N}_2(i|V)} \frac{1}{\sigma_{c,p}^2} (\lambda_p - \lambda_{\bar{p}}) = 0. \tag{17}$$

The system of linear equations builds up from projective equations (9) and energy minimisation equations (17) can be represented as $Ax = b$, where A is a large sparse matrix, b is a right side vector, which we solve for unknown depths and camera centres $x = \left[\Lambda\ \mathbf{C}^1\ \mathbf{C}^2\ \ldots\ \mathbf{C}^c\right]$. We employ quasi-minimal residual method to solve our over-determined system with the initial or previous estimate as the starting point close to the optimum. The number of iterations of the solver is limited to a number proportional to the size of the problem.

4 Visibility estimation

The visibility estimation task in the step 3 of the algorithm overview in Section 2 will be now presented. It pursues several principal goals: primarily, outliers in the point cloud are detected, next the discontinuities are indicated to prevent smoothing over them and finally the compactness of the visible regions is enforced. According to Figure 4, visibility defines if a given pixel and its estimated depth will be used for reconstruction of the surface ($v_p^i \geq 1$) or not ($v_p^i = 0$), and also if there is data support from projected correspondences at this point ($v_p^i = 2$). Then given depths Λ, correspondences $\mathcal{X}$ and image values $\mathcal{I}$, it is searched for

$$V^* = \arg\max_V P(\mathcal{I} \mid V, \Lambda, \mathcal{X})\, P(V, \Lambda, \mathcal{X}). \tag{18}$$

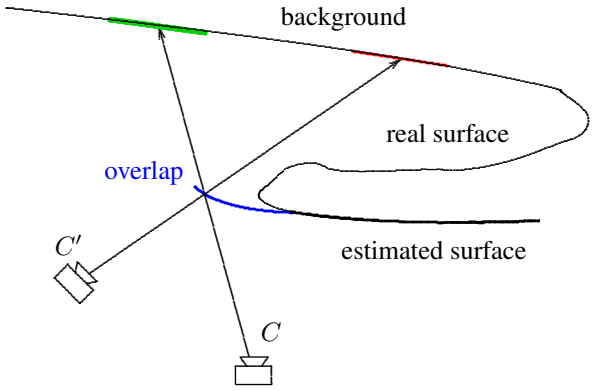

Figure 5: Overlapping borders. False overlapping surface is marked blue. Green(bold) and red illustrate different colours of the background projected to the different cameras.

The conditional probability $P(\mathcal{I} \mid V, \Lambda, \mathcal{X})$ can be expressed as

$$P(\mathcal{I} \mid V, \Lambda, \mathcal{X}) = \prod_{i=1}^{c} \prod_{p=1}^{n} \prod_{(q,j);\, v_q^j \geq 1} p(\mathbf{I}_q^j, \mathbf{I}_p^i \mid v_q^j, v_p^i), \tag{19}$$

where (q, j) is a list of visible corresponding pixels j in camera q from χ_p^i. We choose

$$p(I_q^j, I_p^i \mid v_q^j, v_p^i) = \begin{cases} \frac{1}{T_I}\, e^{-\frac{(\mathbf{I}_p^i - \mathbf{I}_q^j)^2}{2\sigma_I^2}} & \text{if } v_q^j = v_p^i = 2, \\ h(\mathbf{I}_p^i) & \text{otherwise,} \end{cases} \tag{20}$$

where $T_I = \sigma_I \sqrt{2\pi}$ and where $h(\mathbf{I}_p^i)$ is probability of observing an "invisible" colour (like colour of the sky), based on regions invisible according to previous estimate of visibility $\mathbf{V}$. The image difference and invisible colour matching mostly indicate errors on borders of surfaces, where the results of stereo tend to overlap the real surface and continue a few pixels "into the air", as in Figure 5. These overlaps then have the colour of the background (surface in higher depth) and typically occur where the background is light and the object is dark, because of the behaviour of the correlation function in stereoscopic matching.

Probability $P(V, \Lambda, \mathcal{X})$ can be rewritten as

$$P(V, \Lambda, \mathcal{X}) = P(V, \mathcal{X}) \cdot P(\Lambda \mid V, \mathcal{X}). \tag{21}$$

It is assumed that the surface is locally flat, and big local changes in depth are either discontinuities due to occlusion or errors in stereo. The discontinuities should be represented as an line of invisible pixels and erroneous data should be hidden. This assumption can be expressed as

$$P(\Lambda \mid V, \mathcal{X}) = \prod_{i=1}^{c} \prod_{(p,\bar{p}) \in \mathcal{N}_2(i \mid V)} \frac{1}{T_\lambda}\, e^{-\frac{(\lambda_p^i - \lambda_{\bar{p}}^i)^2}{2(\sigma_{\lambda,p}^i)^2}}, \tag{22}$$

where $\sigma_{\lambda,p}^i$ is estimated from residual of depth λ_p^i in the depth optimisation task (4). The residual is higher at the points where the surface model could not be fitted on the data, indicating possible outliers. Expression $(p, \bar{p}) \in \mathcal{N}_2(i \mid V)$ are visible neighbouring pixels, as defined in (10) where additionally $v_p^i \geq 1, v_{\bar{p}}^i \geq 1$.

Compactness of visible and invisible regions is assumed:

$$P(V, \mathcal{X}) = \prod_{i=1}^{c} \prod_{(p,\bar{p}) \in \mathcal{N}_2(i)} \frac{1}{T_v}\, e^{-\frac{(v_p^i - v_{\bar{p}}^i)^2}{2\sigma_v^2}}. \tag{23}$$

The expression $(v_p^i - v_{\bar{p}}^i)^2$ means that pixels neighbouring a visible pixel with data support ($v = 2$) are more likely to be visible ($v = 1$) rather than invisible ($v = 0$).

The value $v_p^i = 1$ implies interpolation of depth λ_p^i and in the terms of (23) should be done only in regions near visible data ($v_p^i = 2$). This corresponds to filling holes in the projected data, but also should not cause interpolation of depths far from data, as the probability of guessing such depth correctly is low.

After applying negative logarithm on (18) and some manipulations we get

$$V^* = \arg\min_V \sum_{i=1}^{c} E(V^i), \tag{24}$$

$$E(V^i) = \sum_{p=1}^{n} E(v_p^i) + \frac{1}{2\sigma_v^2} \sum_{(p,\bar{p}) \in \mathcal{N}_2(i)} (v_p^i - v_{\bar{p}}^i)^2, \tag{25}$$

where

$$E(v_p^i) = \sum_{(q,j) \in \chi_p^i;\, v_q^j \geq 1} E(v_p^i, v_q^j) + \sum_{(p,\bar{p}) \in \mathcal{N}_2(i \mid V)} \frac{(\lambda_p^i - \lambda_{\bar{p}}^i)^2}{2(\sigma_{\lambda,p}^i)^2} \tag{26}$$

$$E(v_p^i, v_q^j) = \begin{cases} \frac{(\mathbf{I}_p^i - \mathbf{I}_q^j)^2}{2\sigma_I^2} & \text{if } v_p^i = v_q^j = 2 \\ -\log h(\mathbf{I}_p^i) & \text{otherwise.} \end{cases} \tag{27}$$

In this task the data visibility $v_p^i = 2$ cannot be assigned without an existing support of at least one correspondence in χ_p^i. We avoid here the occlusion problem and do not create a new correspondence. As a result, only some data can be hidden by this task (visibility set to $v = 0$), and otherwise the data visibility is fixed.

Following this analysis, we can transform our problem (26) of three labels $v_p^i \in \{0, 1, 2\}$ into a binary segmentation problem, which allows us to use the minimum cut solution [11]. Specifically, penalty function is $E(v_p^i)$ plus a data visibility term, and the second term in (25) maps to the interaction potential. We use the implementation from [2]. The output of the min-cut problem is a binary labelling assigning visibility to all pixels in the image. Its interpretation for data visibility $v = 2$ is that it keeps $v_p^i = 2$ where the label 1 was assigned.

We observed during the development that the model described above cannot itself handle discontinuities completely, because it has to take continuity into account. While undetected discontinuities are critical in the depth model assuming continuity, we decided to focus on them in a subtask.

Let us introduce a new variable, discontinuity map $D^i = \{d_p^i \mid p = 1, \ldots, n^i\}$, where $d_p^i \in \{0, 1\}$ is presence of discontinuity in pixel p in camera i. Set of all discontinuity maps will be $\mathcal{D} = \{D^i \mid i = 1, \ldots, c\}$. The subtask of discontinuity detection can be formally expressed as a search

for estimate $\mathcal{D}^*$:

$$\mathcal{D}^* = \arg\max_{\mathcal{D}} P(\Lambda \mid \mathcal{D}, V) P(\mathcal{D}, V). \tag{28}$$

In this task the solution does not depend on $P(V, \Lambda)$. Probability (28) is difficult to calculate explicitly. The proposed solution solves the task indirectly in every camera i. First, gradient of depth $\delta_p^i = \|\nabla \lambda_p'^i\|^2$ is calculated on the visible data. If the depth is unknown at the moment, it is interpolated as the median of the closest known depths. Note this is an edge-preserving operation. Afterwards the gradient size is normalised to $\delta_p'^i = \frac{\delta_p^i}{\lambda_p^i}$, what is equivalent to gradient size at depth $\lambda = 1$. A threshold on normalised gradient is chosen and the initial discontinuity map D^{i0} is obtained. Finally, D^{i0} is processed with binary morphological operations to reduce regions to lines to obtain D^{i*} and estimated discontinuities are propagated to the visibility maps, $v_p^i = 0$ is set for each pixel where $d_p^{i*} = 1$.

5 Update of correspondences

One of the possible solutions to the energy-minimisation equations for visibility (25) and depths (15) is that nothing is visible in all cameras and the energy reaches its minimum $E(V) = E(\Lambda) = 0$.

This could happen if the input does not match the assumptions and parameters of the used model, i.e. the surface is not locally simple or the exposure of input images is very different. Up to a certain limit, this case is avoided by the robust estimation of parameters σ_λ and σ_v.

The visibility according to the previous section builds on the data visibility, so the data can be only hidden by the visibility task ($v < 2$) and propagated to other cameras, the iteration could gradually hide all data. This is avoided by the repair of hidden correspondences, so that the data visibility of previously hidden points $v = 2$ is restored.

The correspondences $\mathcal{X}$ are repaired in the step 5 of our algorithm in the overview in Section 2, after depths are properly estimated in each iteration. A correspondence, $\mathbf{X}_{pq}^{ij}$, is to be repaired when it is visible in at least one camera without data support, that is either $v_p^i = 1$ or $v_q^j = 1$. If either only $v_p^i = 1$ or $v_q^j = 1$, then correspondence is a back-projection of the currently estimated depth, for $v_p^i = 1$ it is

$$\mathbf{X}_{pq}^{ij} = \mathbf{C}^i + \lambda_p^i \, (\mathbf{R}^i)^\top (\mathbf{K}^i)^{-1} \mathbf{x}_p^i, \tag{29}$$

similarly for $v_q^j = 1$. If both $v_p^i = 1$ and $v_q^j = 1$, then correspondence $\mathbf{X}_{pq}^{ij}$ is updated to the value back-projected in camera i as in previous case, and a new correspondence $\mathbf{X}_{p'q'}^{ij}$ is created according to the value back-projected in camera j. The visibility of the repaired correspondence is enforced by setting $v_p^i = v_q^j = 2$, where the indices p, q are obtained after the new projection of $\mathbf{X}_{pq}^{ij}$.

Because the correspondence is restored at the point determined by the model, it minimises the energy $E(V)$ and $E(\Lambda)$. Since the visibility task keeps data that match the model, repaired correspondences will not be later hidden again with high probability.

A different approach in the place of updating data points, which we would like to consider in the future is to run stereo matching again with refined camera positions.

Dataset	#img	#main	complet.	accuracy
fountain-P10	10	6	80.19 %	35.45 %
Herz-Jesu-P8	8	8	78.82 %	39.38 %
entry-P10	10	10	79.43 %	31.84 %
castle-P19	19	7	79.93 %	40.82 %

Table 1: Results on datasets with accurate calibration. The accuracy and completeness correspond to the percentage of pixels with error $< 2\sigma$ and 10σ respectively, where σ is the ground truth error from [15]

6 Experimental results

We have tested the proposed algorithm on two groups of datasets, first with available accurate calibration and second with less accurate camera parameters. In all cases, experiments were run in the Matlab environment on 2.4GHz PCs with 16GB memory.

The first group consists of datasets available for on-line evaluation from [15]. We have chosen it because all datasets present there are real outdoor-scenes captured in a high resolution. We have run our algorithm with same parameters[3] on four datasets, which are included in Table 1, and the results are available on-line[4]. The number of main cameras refers to the subset of cameras, in which the depths are estimated, and it was adjusted according to the number of input images. The number of iterations of the proposed algorithm was fixed to 5 and all images were sampled down to half-size resolution.

The automatic evaluation in [15] is performed against ground truth acquired with time-of-flight laser measurement. Evaluated scene is projected to the input cameras, and obtained depths are compared with the ground truth depths in the scale of its accuracy σ in Figure 6. The update of camera centres was not performed in this group to keep the idea of calibrated evaluation, however it was still variable in (9).

The accuracy of the results on the first two datasets can be observed in Figure 6, where a comparison with a number of other methods can be performed. In both cases the results are almost identical to the best performing methods publicly presented at the time of submission. From rendered results it shows that many methods, including our, fail to reconstruct the ground plane in the scenes. The reason for this behaviour was found in the almost pure horizontal motion in the datasets. The specific behaviour of our method can be demonstrated on the error distribution in the image. We believe the camera calibration in this dataset is still not fully accurate, which results in less errors (light) on the right side and more errors (dark) on the left side in the all other algorithms' results, like in Figure 7 e). In contrast, our overall geometric L_2 minimisation allowed a small shift of the camera resulting in the error being distributed equally in the image, see Figure 7 d).

In the second group, the input point cloud in correspondences, camera pairs and camera parameters were obtained from the reconstruction pipeline of [6]. We have selected a scene that shows a part of paper model of the Prague cas-

[3] Their values and choice is given in [19, Chapter 3].

[4] http://cvlab.epfl.ch/ strecha/multiview/denseMVS.html

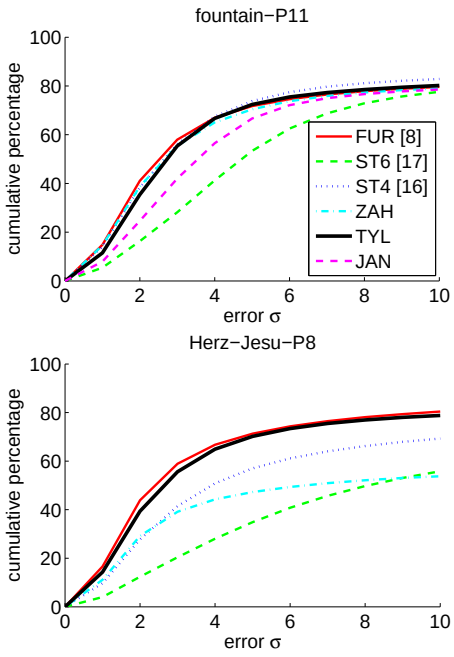

Figure 6: Cumulative histograms from [15] represent percentage of pixels with an error $< n\sigma$. The accuracy manifests at the value of 2σ, completeness at 10σ. Proposed method is labelled "TYL".

tle with Daliborka tower [1]. In this case the input data are noisy and the pair-wise reconstructions do not align with each other well, because of inaccurate camera parameters. Some surfaces have no texture, therefore they were not matched by the stereo algorithm. Figure 8 shows that the estimation of the camera positions converge to a stable position after a significant update in the second iteration. However the number of iterations necessary to refine inaccurate camera positions is higher than in the accurate case, in this case the algorithm was run for 11 iterations.

Finally, Figure 10 shows the successful suppression of the stereo overlap error, resulting in better accuracy near edges.

7 Conclusion and future work

We have presented a novel multi-view stereo algorithm that is suitable both for scenes with accurate and inaccurate camera calibration. The suitably chosen techniques, such as invisible colour matching, edge and discontinuity preservation help to produce very faithful results. The chosen depth map representation has linear complexity in the number of images and allows application to large outdoor scenes. The experiments show that the fused pair-wise stereo matching can achieve same results as simultaneous multi-view matching.

Having an accurate global solution, we now see improvement potential in the local mesh refinement.

Acknowledgement

The author was supported by Ministry of Education as a part of the specific research at the CTU in Prague and by the Czech Academy of Sciences under project No. 1ET101210406.

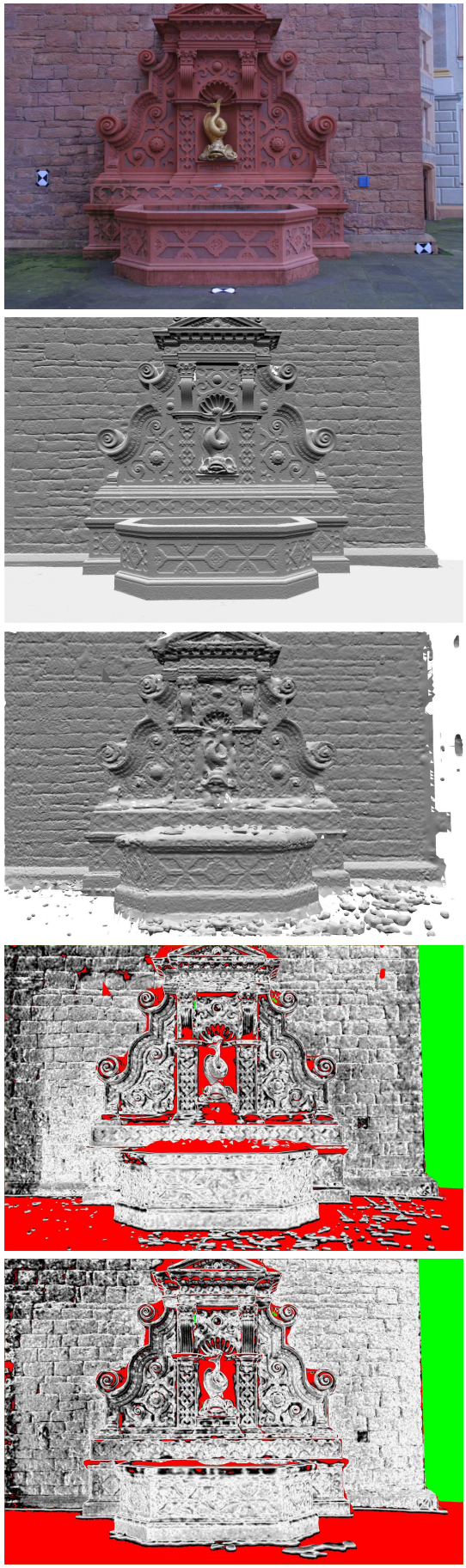

Figure 7: Fountain-P11 dataset [15]. From top: a) one of the 11 input images, b) ground truth rendering, c) our method result, d) our method error, accurate regions are white, missing reconstruction is red, green area was not evaluated, e) error of [8]

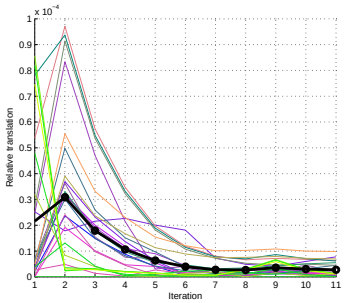

Figure 8: Relative translation in iterations, Daliborka dataset. Scale is given so that the average distance between cameras is 1. Each one of 23 cameras is represented by a different color, bold black line is the average of all cameras.

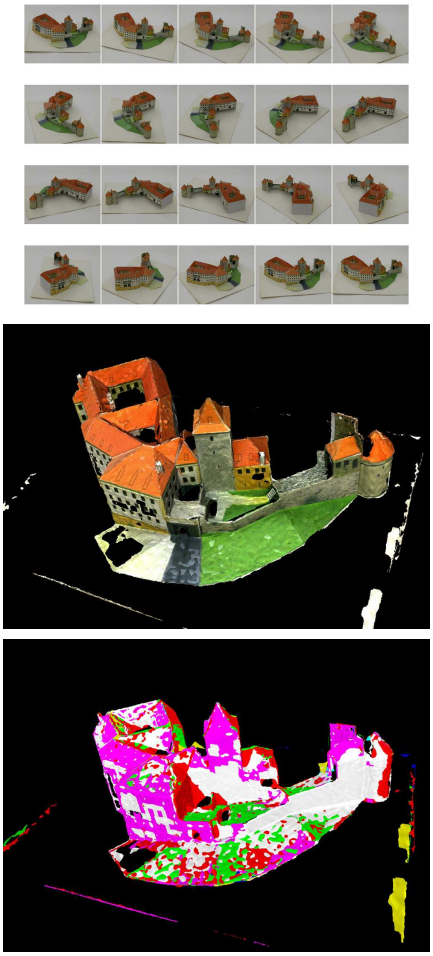

Figure 9: Daliborka dataset. From top: a) some of 27 input images, b) textured result of our method, c) composition of surface from different views, one color per camera.

Figure 10: Detail from Daliborka scene. Overlapping borders (left) are successfully removed after one iteration of the proposed algorithm (right).

References

[1] Betexa ZS, s.r.o. The Prague castle, paper model of the biggest castle complex in the world, 2006. Scale 1:450.

[2] Yuri Boykov and Vladimir Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Computer Vision. *IEEE Trans. on PAMI*, 26(9):1124–1137, September 2004.

[3] Jan Čech and Radim Šára. Efficient sampling of disparity space for fast and accurate matching. In *BenCOS 2007: CVPR Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*, 2007.

[4] W. Heidrich. D. Bradley, T. Boubekeur. Accurate Multi-View Reconstruction Using Robust Binocular Stereo and Surface Meshing. In *Proc. CVPR*, pages 1–8, 2008.

[5] Dempster, A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Statist. Soc.*, (39):1–38, 1977.

[6] Kamberov, George et al. 3D geometry from uncalibrated images. In *Proc. ISVC*, number 4292 in LNCS, pages 802–813, November 2006.

[7] O. Faugeras and R. Keriven. Complete dense stereovision using level set methods. *LNCS*, 1406:379–393, 1998.

[8] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *Proc. CVPR*, pages 1–8, 2007.

[9] Y. Furukawa, J. Ponce, and W. Team. Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment. In *Proc. CVPR*, pages 1–8, 2008.

[10] M. Goesele, B. Curless, and S. M. Seitz. Multi-View Stereo Revisited. In *Proc. CVPR*, volume 2, pages 2402–2409, 2006.

[11] Hartley, R. and Zisserman, A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[12] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proc. Eurographics symposium on Geometry processing*, pages 61–70, 2006.

[13] P. Labatut, J.P. Pons, and R. Keriven. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In *Proc. ICCV*, pages 1–8, 2007.

[14] Steve Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Rick Szeliski. A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms. In *Proc. CVPR*, pages 1–8, 2006.

[15] C. Strecha, E. CVLab, W. von Hansen, L. Van Gool, E. CVLab, P. Fua, and U. Thoennessen. On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *Proc. CVPR*, pages 1–8, 2008.

[16] Christoph Strecha, Rik Fransens, and Luc Van Gool. Wide-Baseline Stereo from Multiple Views: A Probabilistic Account. In *Proc. CVPR*, pages 552–559, 2004.

[17] C. Strecha, R. Fransens, and L. Van Gool. Combined Depth and Outlier Estimation in Multi-View Stereo. In *Proc. CVPR*, pages 2394–2401, 2006.

[18] R. Szeliski. A multi-view approach to motion and stereo. In *Proc. CVPR*, pages 1–8, 1999.

[19] Radim Tyleček. Representation of Geometric Objects for 3D Photography. Master's thesis, Czech Technical University, Prague, January 2008.

[20] G. Vogiatzis, P. H. S. Torr, and R. Cipolla. Multi-View Stereo via Volumetric Graph-Cuts. In *Proc. CVPR*, pages 391–398, 2005.

[21] C. Zach. Fast and high quality fusion of depth maps. In *Proc. 3DPVT*, pages 1–8, 2008.