Monocular Road Mosaicing for Urban Environments

Andreas Geiger
Institut für Mess- und Regelungstechnik
Universität Karlsruhe (TH), Germany
geiger@mrt.uka.de

***Abstract*—Marking-based lane recognition requires an unobstructed view onto the road. In practice however, heavy traffic often constrains the visual field, especially in urban scenarios such as urban crossroads.**

In this paper we present a novel approach to road mosaicing for dynamic environments. Our method is based on a multistage registration procedure and uses blending techniques. We show that under modest assumptions accurate registration is possible from monocular image sequences. We further demonstrate that fusing visual information from previous frames into the current view can greatly extend the camera's field of view.

I. Introduction

Intelligent vehicles require a good view onto the road when visual positioning according to lane markers is required [1], [2]. Often however, the field of view is strongly limited. This is because traffic signs, other traffic participants or buildings frequently occlude large parts of the road, as illustrated in figure 1 and 2. Treating occlusions as noise can lead to unsatisfying results in such situations.

We tackle this problem by making use of the parallax effect – the apparent displacement of an object viewed along two different lines of sight. Knowing the ego-position and -orientation of the vehicle and identifying a light ray as coming from a particular point on the road allows for incrementally generating a *virtual map* of the road.[1] This map can be used to reconstruct parts of the road which become occluded in future image frames. Figure 1 shows this idea. While at frame $t+1$ the view of the ego-vehicle is partly barred by the other car, the occluded part of the road is visible at frame t due to the relative displacement of object and camera. In order to remove artificial edge artifacts from the mosaic we use blending techniques which are a well-studied tool in the computer graphics community.

In recent years ego-positioning and mapping experienced a lot of attention [3]–[7]. Recent advances have shown that even long distances can be robustly mapped by mobile robots [8]. Many of the techniques, however, use stereo-information, landmark points, additional sensor types and/or assume static scenes. Our method also differs from traditional stitching techniques [9], [10] since we have to deal with low resolution in far field due to the tilted and low camera position with respect to the road.

The method described in this paper solely uses information from a monocular grayscale camera, thus supporting a cheap and easy application in automotive systems. Since we work with monocular images only, ego-positioning is challenging and we have to make assumptions about the environment which are summarized as follows:

- An estimate of the camera intrinsics (focal length, principal point, distortion) and extrinsics (camera height, inclination) is given.
- The road surface to be considered can be approximated by a plane and offers enough texture (i.e. road markings) for the registration process.

Note that the first assumption is valid for all calibrated camera setups and the second assumption holds for a large number of interesting inner-city scenarios.

The remainder of this paper is structured as follows: First we show how representative frames can be found in the image sequence. Then our four-stage registration process is described followed by a description of the image-blending techniques we use. Finally the paper is concluded with an outlook on future work.

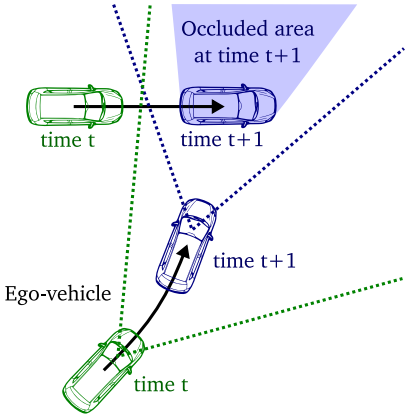


Fig. 1. **The parallax effect.** Combining information from previous frames with the current view increases the visual field with respect to the road.

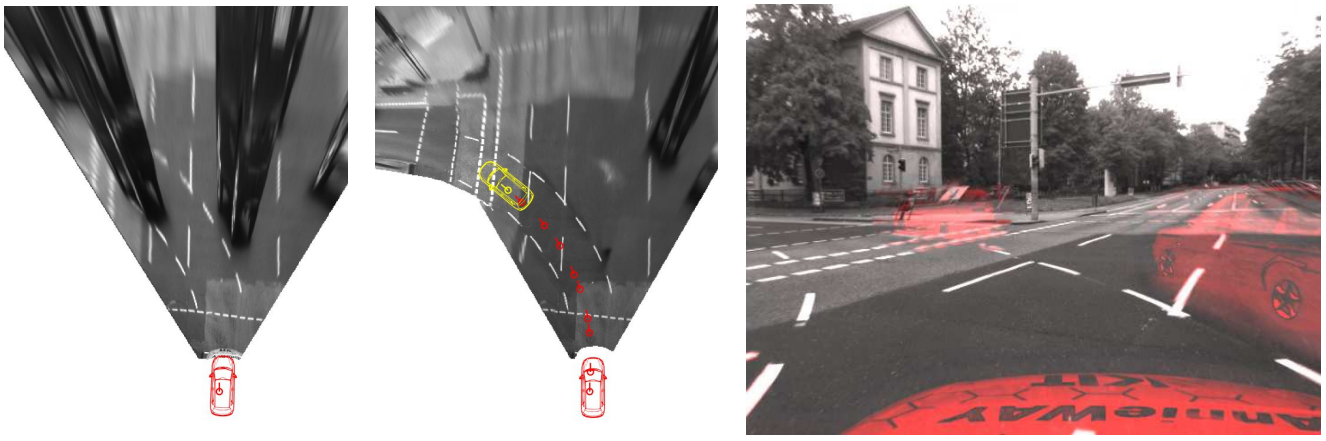
II. Parameter initialization

A. Geometrical scene description

Approximating the road surface by a plane enables us to describe the perspective mapping from one image plane to another image plane and from image planes to the road via homographies [11]

$$\mathbf{p}_j \simeq \mathbf{H}\mathbf{p}_i$$

[1] We call this map *virtual*, since we are only interested in the correct pose parameters. The target surface to project on can be either a bird's eye view (figure 2(b)) or a single camera view (figure 2(c)).



(a) First frame (bird's eye view) (b) Mosaicing result (c) Overlay (best viewed in color)

Fig. 2. **Comparison of single image road-views with road mosaics.** Figure 2(a) shows a bird's eye perspective of the road when projecting the current camera image only. Figure 2(b) shows the mosaicing result of our algorithm with the visually estimated ego-position of the camera for 10 keyframes. The images are combined in a way that removes spurious obstacles from the road and leads to a better overlook. Figure 2(c) shows an artificial overlay (red) of a single frame with the mosaicing result computed only from previous frames (gray values).

where $\mathbf{p}_i$ is a homogeneous point lying on plane i and $\mathbf{H}$ is a 3×3-matrix describing the projection from plane i to plane j. Here $\simeq$ denotes that the equation is defined up to a scale factor.

In theory the 8 homography parameters could be estimated from frame to frame using standard techniques like DLT[2]. However we do not know which of the correspondences between two subsequent frames actually lie on the road. Furthermore small registration errors would accumulate quickly [3], [7], leading to bad results when projecting points along the *" homography chain"*.

Thus we decided to use a direct parameterization scheme which further features interpretable parameters in contrast to elementwise homography parameterization. This also allows for putting priors on the parameters in order to perform temporal integration and improve the optimization result. Figure 3 shows the transformation matrices $\mathbf{T}_{rr}$ which project from road to road, $\mathbf{T}_{rc}$ which project from road to camera, $\mathbf{K}$ which project from camera to the image plane and $\mathbf{T}_{ii}$ which project from image to image. Due to the planarity assumption all projections can be expressed as 3×3 homography matrices. Transforming a point from one coordinate system into the other is easily expressed via concatenations and inversions of these matrices. The individual mappings are parameterized as

$$\mathbf{K}(\mathbf{f},\mathbf{c}) = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T}_{rc}(h,\theta,\phi) = \begin{bmatrix} \cos\phi & -\sin\phi\cos\theta & -\sin\phi\sin\theta h \\ \sin\phi & \cos\phi\cos\theta & \cos\phi\sin\theta h \\ 0 & \sin\theta & -\cos\theta h \end{bmatrix}$$

[2]Direct Linear Transform [11].

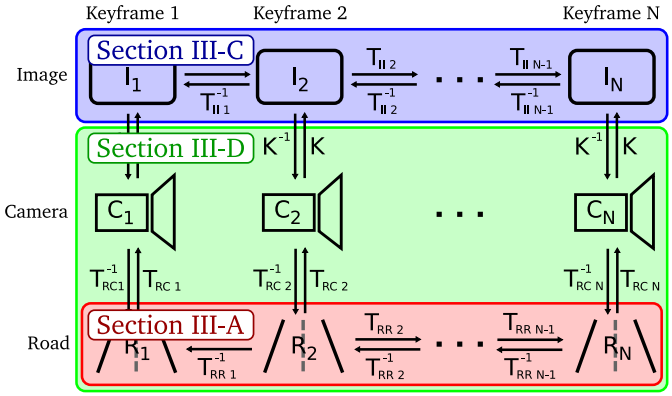


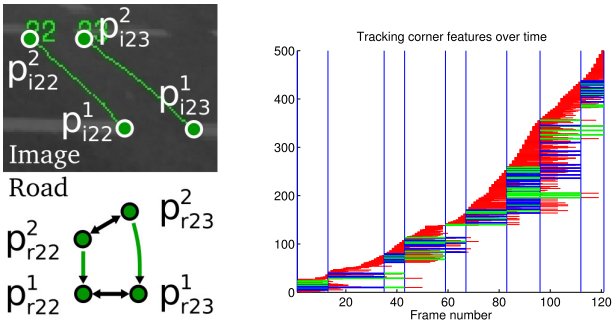
Fig. 3. **Geometrical description of the scene.** This figure shows the transformations we use for projecting between different coordinate systems. The colored boxes denote associate sections in this paper.

$$\mathbf{T}_{rr}(\alpha,\mathbf{t}) = \begin{bmatrix} \cos\alpha & -\sin\alpha & t_1 \\ \sin\alpha & \cos\alpha & t_2 \\ 0 & 0 & 1 \end{bmatrix}$$

where **f**,**c** are the camera intrinsics, h is the camera height, **t** represents the camera translation over ground and ϕ,θ,α stands for roll, pitch and yaw. While the calibration matrix $\mathbf{K}$ is assumed constant over time, $\mathbf{T}_{rc}$ and $\mathbf{T}_{rr}$ vary. The goal in registration is now to estimate the parameter set $\mathbf{\Theta} = (\mathbf{f},\mathbf{c},[h_1,\theta_1,\phi_1]...[h_N,\theta_N,\phi_N],[\alpha_1,\mathbf{t}_1],...,[\alpha_{N-1},\mathbf{t}_{N-1}])$ which relates N keyframes to each other.

B. Keyframes and keyfeatures

Since we can not make use of landmark points, accumulation of errors is a severe problem averting the use of all frames for registration. By picking only a subset as *keyframes* we also save computational power because optimizing for the parameters is done in a lower-dimensional space. Thus we



(a) Harris corner tracking (b) Snakes over time

Fig. 4. **Tracking Harris corners.** Figure 4(a) depicts two Harris corner tracks between two consecutive keyframes in the image *(top)* and road *(bottom)* coordinate system. Sustaining tracks are shown over time in figure 4(b). The green tracks are selected as keyfeatures for parameter initialization as described in section III-A. Keyframes are indicated by blue vertical bars.

search for the smallest set of keyframes which still allows for accurate registration.

To do so we first track Harris corners [12], [13] over the sequence using template matching via cross-correlation. New *"tracks"* are initialized at points distant from existing active tracks. Tracks end when the cross-correlation value falls below a threshold τ or simple smoothness or geometrical constraints are violated. Figure 4 depicts this process.

As mentioned earlier, we are interested in a small set of representative keyframes with a large number of tracks which connect two consecutive keyframes. Furthermore we want the keyframes to be uniformly distributed. In other words we wish to minimize

$$E_{key} = aN - b\sum_{i=1}^{N-1} t_i^{i+1}\psi(C_i^{i+1}) + c\sum_{i=1}^{N-1}|t_i^{i+1} - \frac{t_1^N}{N}| \quad (1)$$

with respect to the number of keyframes N. Here t_i^j denotes the time between keyframe i and keyframe j, C_i^{i+1} is the number of tracks connecting keyframe i and keyframe $i+1$ and ψ stands for the logistic function. In our experiments we set $a = b = c = 1$ and choose the logistic parameters such that 4-5 connections between consecutive keyframes are considered sufficient. Because minimization of equation 1 with respect to N leads to exponential complexity we approximate the solution by a simple greedy algorithm: We initialize the set of keyframes to the set of all frames and remove the frame with the highest gain in E_{key} until no further improvement is possible. A typical solution of this algorithm is shown in figure 4(b).

Reliable Ego-pose initialization further requires selecting a subset of tracks connecting pairs of keyframes which lie on or close to the road. To achieve this we project feature candidates onto the road[3] via $\mathbf{p}_r^j = \mathbf{T}_{rc}^{-1}\mathbf{K}^{-1}\mathbf{p}_i^j$ for each pair of consecutive keyframes $j,j+1$. Since distances on the road are invariant under global rotation and translation we select those features as keyfeatures which exhibit the smallest change in distance $|\|\mathbf{p}_{r1}^j - \mathbf{p}_{r2}^j\|_2 - \|\mathbf{p}_{r1}^{j+1} - \mathbf{p}_{r2}^{j+1}\|_2|$ from each other. Figure 4(a) illustrates this process for a single pair.

[3]Here the a-prior knowledge about h,θ and ϕ is used.

III. Registration

Accurate image alignment is of utmost importance since it has a direct impact on the mosaicing result. As it turns out, even small registration errors in the image coordinate system can cause large errors in the road coordinate system. This is because the camera is highly tilted with respect to the relevant road plane leading to decreasing image resolution and registration accuracy in the far field. Note that this is not the case for traditional stitching tasks.

Because of this we employ a four-stage registration algorithm which first estimates the ego-pose, secondly segments keyframes into foreground and background, afterwards finds additional road correspondences via a *RANSAC*[4]-based search and finally performs bundle adjustment in a probabilistic setup with Gaussian Process priors.

A. Ego-pose initialization

We first initialize the calibration parameters $\mathbf{f}, \mathbf{c}$ and the road-to-camera transformation parameters $(h_1, \theta_1, \phi_1) = ... = (h_N, \theta_N, \phi_N)$ according to our prior knowledge about the camera setup. The only missing parameters to complete the vector $\mathbf{\Theta}_0$ are the road-to-road transformation parameters.

As shown in [14] the least squares problem to fitting a 3D point set to another can be solved in closed form by computing the singular value decomposition of a 3×3 matrix. We make use of this algorithm for our 2D point fitting problem in order to minimize

$$E_{point}^j = \sum_i \|\mathbf{p}_{ri}^j - \mathbf{p}_{ri}^{j+1}\|^2 \quad (2)$$

where $\mathbf{p}_{ri}^j$ denotes road point i at keyframe j. Solving equation 2 results in the initial estimate for the missing translation and rotation parameters $(\alpha_j, \mathbf{t}_j)$ for $j = \{1, ..., N-1\}$.

B. Road segmentation

Given our initial estimate $\mathbf{\Theta}_0$ we intend to refine the parameters in a global optimization scheme. This requires accurate road-to-road correspondences between keyframes which are sought in the next section. A first step consists of segmenting foreground objects from the background (road). Since we are using a monocular setup, stereo-information is not accessible for this task. We tackle this problem by observing that points on the road usually transform in a way distinct from points lying on moving or static objects (e.g. cars).

Applying the sparse iterative version of Lucas-Kanade optical flow estimation (as described in [15]) on Harris corners below the horizon line[5], results in 2-dimensional optical flow vectors $\mathbf{v}_{real}$ for salient pixels $\mathbf{p}_i$, where the index i denotes that it lies in the image plane. The virtual optical flow $\mathbf{v}_{virt}$ to compare against is calculated by projecting the *"road flow"* into the image, using the estimed parameters from section

[4]RANdom SAmple Consensus.

[5]The horizon line can be easily determined by h,θ and ϕ.

III-A. The final error value v_{err} is computed by weighting the norm of the optical flow differences with the angular agreement of both vectors, thereby penalizing contradicting vector directions.

$$
\begin{aligned}
v_{err} &= \|\mathbf{v}_{real} - \mathbf{v}_{virt}\|_2 \left(2 - \frac{\mathbf{v}_{real}^T \mathbf{v}_{virt}}{\|\mathbf{v}_{real}\|_2 \|\mathbf{v}_{virt}\|_2}\right) \\
\mathbf{v}_{real} &= (\frac{\partial \mathbf{p}_i}{\partial u}, \frac{\partial \mathbf{p}_i}{\partial v})^T \\
\mathbf{v}_{virt} &= \mathbf{K}\mathbf{T}_{rc}(\mathbf{p}_r + \mathbf{p}_v) - \mathbf{p}_i \\
\mathbf{p}_v &= \frac{\mathbf{T}_{rr}^j \mathbf{p}_r - \mathbf{T}_{rr}^{j-1\,-1} \mathbf{p}_r}{(t_{j+1} - t_{j-1})} \\
\mathbf{p}_r &= \mathbf{T}_{rc}^{-1} \mathbf{K}^{-1} \mathbf{p}_i
\end{aligned}
$$

Here $\mathbf{p}_r$ denotes road coordinates and $\mathbf{p}_v$ is the road flow vector for this point, obtained by an approximation to the trajectory tangent. After calculating v_{err} for all points which exhibit enough texture [13] we apply nearest-neighbor clustering and remove clusters with less than 10 points. We extend the convex hull of all remaining point sets to approximately cover the whole object. Points inside the polygons, above the horizon line or on the engine hood are masked while the remaining regions are assumed to stem from the road. An example of the segmentation result is depicted in figure 5 for a single frame.

C. RANSAC-based correspondence refinement

Having segmented the keyframes into road and non-road regions enables us to find correspondences more precisely. We employ an iterative correspondence search making use of the planarity assumption. To do so we warp the image and mask from keyframe $j+1$ to the image coordinate system of keyframe j using bilinear interpolation and the homography

$$\mathbf{T}_{ii}^j = \mathbf{K}\ \mathbf{T}_{rc}^{j+1}\ \mathbf{T}_{rr}\ \mathbf{T}_{rc}^{j\ -1}\ \mathbf{K}^{-1}$$

as depicted in figure 6. We then calculate Harris corners for all non-masked pixels in the warped image $j+1$ and search for the best match in image j. Template matching and *RANSAC* are employed for finding inlier correspondences and estimating the homography $\mathbf{T}_{ii}^j$ at the same time, making use of the *DLT* algorithm for calculating each random sample. This procedure is repeated for a smaller search area, a bigger template size and the new $\mathbf{T}_{ii}^j$ until convergence (typically after 2-3 iterations). The remaining inlier correspondences are then used in the bundle adjustment stage described in the next section.

D. Global optimization with priors

This section describes how we find the optimal parameters using all information available to us. We perform temporal integration via bundle adjustment with priors on the parameters. The regularization term helps when dealing with ambiguities or small registration errors as in our case.

We seek to maximize the posterior probability of the parameters given the observations with respect to the parameters

$$P(\mathbf{\Theta}|\mathbf{P}_1, ..., \mathbf{P}_{N-1}) \propto P(\mathbf{P}_1, ..., \mathbf{P}_{N-1}|\mathbf{\Theta})P(\mathbf{\Theta}) \qquad (3)$$

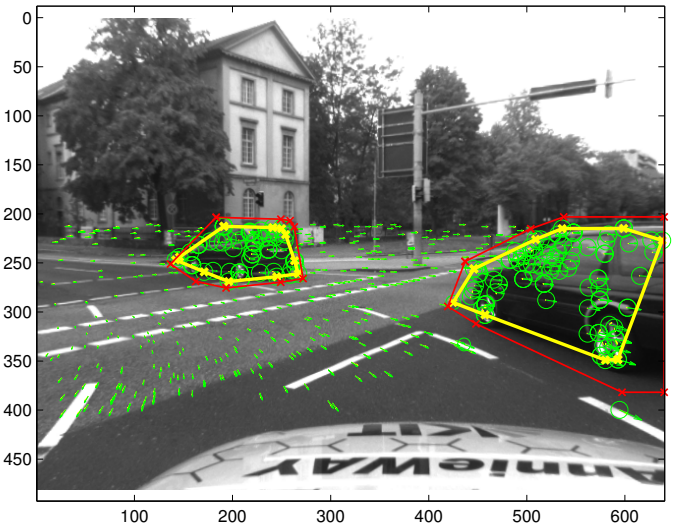
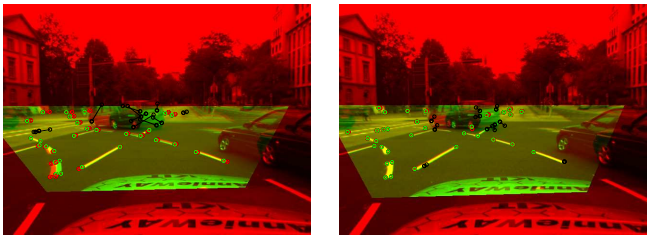


Fig. 5. **Road segmentation using virtual optical flow.** Here we depict the optical flow (yellow) and the virtual optical flow (green) for keyframe 7 of the test sequence from figure 4(b). To give robust results, we do not use dense optical flow, but instead calculate the optical flow only at locations with high saliency. Green circles indicate deviations in optical flow and the red polygon shows the refined convex hull which is used for masking objects.



(a) Iteration 1 (b) Iteration 2

Fig. 6. **RANSAC correspondence refinement.** This figure illustrates iteration 1 and 2 of the *RANSAC* correspondence search. While the red channel shows keyframe j, the green channel displays keyframe $j+1$ warped to the image coordinate system of keyframe j. Inlier correspondences are depicted in red and green while outliers are marked black. For the sake of clarity this figure does not show the object masks which were used.

where $\mathbf{\Theta}$ are the parameters and $\mathbf{P}_j$ is the match probability which depends on the correspondences found between keyframe j and $j+1$ and the geometric parameters.

Since we do not consider correspondences between non-consecutive keyframes the pairwise likelihood is conditionally independent

$$P(\mathbf{P}_1, ..., \mathbf{P}_{N-1}|\mathbf{\Theta}) = \prod_{j=1}^{N-1} P(\mathbf{P}_j|\mathbf{\Theta}) \qquad (4)$$

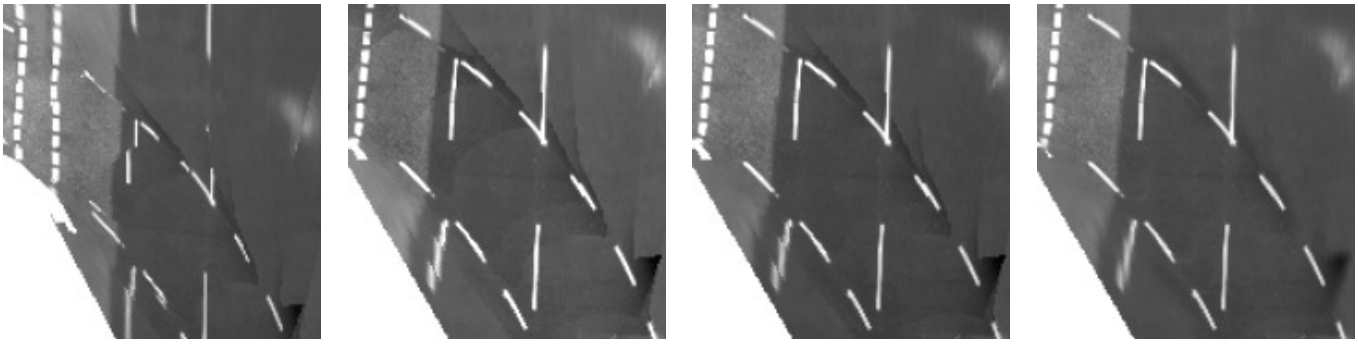
and we assume white noise on the correspondences

$$P(\mathbf{P}_j|\mathbf{\Theta}) \propto \exp(-\frac{1}{2}\mathbf{d}_j^T \mathbf{\Sigma}^{-1} \mathbf{d}_j)$$

with $\mathbf{\Sigma} = \text{diag}(\sigma_d^2, ..., \sigma_d^2)^T$ and the reprojection error distance

$$d_j^i = \|\mathbf{T}_{ii}^j \mathbf{p}_j^{i-} - \mathbf{p}_j^{i+}\|_2.$$

Here $\mathbf{p}_j^{i-}$ denotes the i'th correspondence of $\mathbf{P}_j$ in the image coordinate system of keyframe j, $\mathbf{T}_{ii}^j = \mathbf{T}_{ii}^j(\mathbf{\Theta})$ represents the



(a) Initialization (b) Global optimization result (c) Gain compensation result (d) Gain compensation & blending

Fig. 7. **Road mosaicing.** While figure 7(a) shows the initialization configuration, figure 7(b) depicts the combination of base images after optimizing the parameters with equation 3. Artificial edge artifacts emerge due to different gains, small registration errors and object shadows. We reduce this effect by compensating for the gain (7(c)) and blending the base images at multiple bands as depicted in figure 7(d). However, as can be seen from the marker at the bottom left of 7(d), small registration errors persist due to the high sensitivity of the extrinsic parameters and the low resolution of the camera.

homography for transforming $\mathbf{p}_j^{i-}$ into the image coordinate system of keyframe $j+1$ and $\mathbf{p}_j^{i+}$ is the corresponding point in keyframe $j+1$.

To circumvent ambiguities and compensate for accumulating registration errors we do not optimize the log-likelihood $\log P(\mathbf{P}_1, ..., \mathbf{P}_{N-1}|\mathbf{\Theta})$ directly, but rather add priors on the parameters and optimize the log-posterior $\log P(\mathbf{\Theta}|\mathbf{P}_1, ..., \mathbf{P}_{N-1})$.

Assuming independence of the parameters we have

$$P(\mathbf{\Theta}) = P(\mathbf{f}, \mathbf{c})P(h)P(\theta)P(\phi)P(\mathbf{v})P(\omega) \tag{5}$$

where $\mathbf{v}$ and ω are the velocity and the angular rate of the camera over ground. Since we assume the camera intrinsics to be fixed we model the probability

$$(\mathbf{f}, \mathbf{c}) \sim \mathcal{N}(\mu_c|\mathbf{\Sigma}_c) \tag{6}$$

according to our knowledge from the calibration process.

The remaining parameters are assumed to vary over the sequence, i.e. are functions over time. We encourage smoothness by putting Gaussian Process priors [16] on the function space and model the mean function of the individual processes via our prior knowledge about the camera setup (e.g. height of the camera). Thus we have

$$f(t) \sim \mathcal{GP}(\mu_f(t), \sigma_f(t, t')) \tag{7}$$

with $f \in \{h, \theta, \phi, v_x, v_y, \omega\}$. For all f we set the mean function identical to the initial estimate of the parameter $\mu_f(t) \equiv f_0$ and we model the covariance function $\sigma_f(t, t')$ using the squared exponential kernel

$$\sigma_f(t, t') = \sigma_{f_l}^2 \exp(-\frac{(t-t')^2}{2\sigma_{f_w}^2}) + \delta_{t=t'}\sigma_{f_\eta}^2$$

with kernel height $\sigma_{f_l}^2$, width $\sigma_{f_w}^2$ and noise σ_{f_η} for each function f. The hyperparameters $\sigma_{f_l}^2$, $\sigma_{f_w}^2$ are set according to our belief about the variance and the smoothness of f. A small noise term is added to increase stability when calculating the inverse of the covariance matrix for $P(f)$.

Integrating the priors (5) and the likelihood (4) into (3) and taking the logarithm yields the log-posterior $\log P(\mathbf{\Theta}|\mathbf{P}_1, ..., \mathbf{P}_{N-1})$ which can be maximized by standard gradient descent techniques like Scaled Conjugate Gradients [17].

IV. Mosaicing

Simple merging of road images leads to unsatisfactory results due to differences in gain, vignetting, object shadows and registration errors as depicted in figure 7(b). Thus, in order to create the final road mosaics we perform two additional steps. After generating *"base images"*, we first compensate for the camera gain which might have changed during the ride. Then we combine the road images using multi-band blending [9], [10] to generate a visually pleasing result and remove artificial edges.

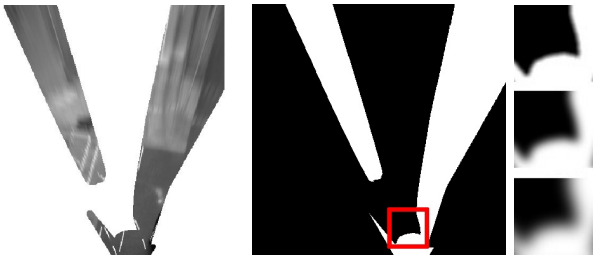
A. Creating base images

Having estimated the parameters $\mathbf{\Theta}$, image points can be transformed from each keyframe to any other keyframe or a global road coordinate system. Thus occluded pixels in one image can be replaced by fusing intensity information from other images. After selecting a target coordinate system (e.g. the road coordinate system for generating top-down views) we generate one *base image* per keyframe which contains all visible pixels warped into the target coordinate system using bilinear interpolation (this is illustrated in figure 8(a)). Combining the base images to the final mosaic is done by taking pixels from the *closest* base image. Here *closest* refers to the Euclidian distance from the camera center of the base image to the pixel's global road position.

B. Gain compensation

In this section, we show how we can solve for the overall gain (a photometric parameter) in closed-form. The gain is denoted as the vector $\mathbf{g} = (g_1, ..., g_N)^T$ where g_i represents the gain of base image I_i. Our goal is to minimize the error function

$$E_{gain} = \sum_{i \neq j} (g_i \mu_{ij} - g_j \mu_{ji})^2 \tag{8}$$



(a) Base image (b) Blending weights

Fig. 8. **Base image and blending weight.** Figure 8(a) shows the base image $\mathbf{I}_i^0$ for a keyframe of the test sequence. The corresponding blending weight $\mathbf{W}_i^{k\sigma}$ is depicted in figure 8(b) for $\sigma = 1.5$ and $k = 0$. The small images on the right side show the red patch for bands $k = \{1, 3, 5\}$.

where μ_{ij} stands for the mean of pixels in image I_i which overlap with image I_j. We exclude the trivial solution $\mathbf{g} \equiv \mathbf{0}$ via the constraint $\mathbf{g}^T\mathbf{g} = N$ using Lagrange multipliers. This avoids the choice of parameters as in [10]. Equation 8 is quadratic in $\mathbf{g}$, thus the problem can be rewritten as minimizing $E_{gain} = \mathbf{g}^T\mathbf{A}\mathbf{g}$ subject to $\mathbf{g}^T\mathbf{g} = N$. This leads to the eigenvalue problem

$$\mathbf{A}\tilde{\mathbf{g}} = \lambda\tilde{\mathbf{g}}$$

where

$$A_{ij} = \begin{cases} \sum_{k \neq i} \mu_{ik} & \text{for } i = j \\ -\mu_{ij}\mu_{ji} & \text{otherwise} \end{cases}$$

The solution is given by $\mathbf{g} = \sqrt{N}\tilde{\mathbf{g}}$ with $\tilde{\mathbf{g}}$ being the eigenvector corresponding to the smallest eigenvalue of $\mathbf{A}$. The result of compensating the gain is depicted in figure 7(c).

C. Multi-band blending

Following [9], [10] we use multi-band blending to confine the effect of artificial edges due to the reasons discussed above. The idea is to blend low frequencies over a large spatial range, and high frequencies over a short range. Therefore the blending weights $\mathbf{W}_i$ and base images $\mathbf{I}_i$ need to be smoothed for the different bands and the bandpass images $\mathbf{B}_i^{k\sigma}$ are calculated by differencing the smoothed base images. For $k \geq 1$ we have

$$\begin{aligned} \mathbf{W}_i^{k\sigma} &= \mathbf{W}_i^{(k-1)\sigma} \otimes g_{\tilde{\sigma}(k)} \\ \mathbf{I}_i^{k\sigma} &= \mathbf{I}_i^{(k-1)\sigma} \otimes g_{\tilde{\sigma}(k)} \\ \mathbf{B}_i^{k\sigma} &= \mathbf{I}_i^{(k-1)\sigma} - \mathbf{I}_i^{k\sigma}. \end{aligned}$$

where the standard deviation of the Gaussian blurring kernel is set to $\tilde{\sigma}(k) = \sqrt{2k+1}\sigma$ such that the range of wavelengths does not change for subsequent bands. Here $\mathbf{W}_i^0$ and $\mathbf{I}_i^0$ denote the base weights and base images respectively and $\otimes$ is the convolution operator. Blending weights for a single keyframe are depicted in figure 8(b).

Combining overlapping images for each band linearly

$$\begin{aligned} I_\Sigma^{k\sigma}(u,v) &= \frac{\sum_i W_i^{k\sigma}(u,v) I_i^{k\sigma}(u,v)}{\sum_i W_i^{k\sigma}(u,v)} \\ B_\Sigma^{k\sigma}(u,v) &= \frac{\sum_i W_i^{k\sigma}(u,v) B_i^{k\sigma}(u,v)}{\sum_i W_i^{k\sigma}(u,v)} \end{aligned}$$

results in the final road mosaic

$$I_{mosaic} = I_\Sigma^{K\sigma} + \sum_k B_\Sigma^{k\sigma}$$

with K the total number of bands. The blending result is depicted in figure 7(d).

V. Conclusion and Future Work

In this paper we have shown that accurate road mosaicing is possible from monocular image sequences only, even in heavy traffic situations. This leads to an important gain in visual information for subsequent processing steps. Further work will focus on an iterative version of the algorithm which is able to add keyframes to an existing map in real-time. Since road segmentation (described in section III-B) is key to finding accurate correspondences we also intend to consider additional features like appearance or stereo information for this step.

Acknowledgment

The author would like to thank the *Karlsruhe School of Optics and Photonics* and the *Deutsche Forschungsgemeinschaft* (Collaborative Research Center *Cognitive Automobiles*) for supporting this work.

References

[1] C. Duchow, "A novel, signal model based approach to lane detection for use in intersection assistance," *Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE*, pp. 1162–1167, 2006.

[2] M. Aly, "Real time detection of lane markers in urban streets," *Intelligent Vehicles Symposium, 2008 IEEE*, pp. 7–12, June 2008.

[3] A. Milella and R. Siegwart, "Stereo-based ego-motion estimation using pixel tracking and iterative closest point," *Computer Vision Systems, 2006 ICVS '06. IEEE International Conference on*, pp. 21–21, Jan. 2006.

[4] J. Horn, A. Bachmann, and T. Dang, "Stereo vision based ego-motion estimation with sensor supported subset validation," *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 741–748, June 2007.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 6, pp. 1052–1067, 2007.

[6] T. Y. Tian, C. Tomasi, and D. J. Heeger, "Comparison of approaches to egomotion computation," *CVPR*, p. 315, 1996.

[7] R. Goecke, A. Asthana, N. Pettersson, and L. Petersson, "Visual vehicle egomotion estimation using the fourier-mellin transform," *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 450–455, June 2007.

[8] M. Pollefeys, "Detailed real-time urban 3d reconstruction from video," *IJCV*, vol. 78, no. 2-3, pp. 143–167, July 2008.

[9] P. J. Burt and E. H. Adelson, "A multiresolution spline with application to image mosaics," *ACM Transactions on Graphics*, vol. 2, pp. 217–236, 1983.

[10] Brown, Matthew, Lowe, and David, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, August 2007.

[11] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[12] C. Harris and M. Stephens, "A combined corner and edge detection," in *Proceedings of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[13] J. Shi and C. Tomasi, "Good features to track," *CVPR*, pp. 593–600, Jun 1994.

[14] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *PAMI*, vol. 9, no. 5, pp. 698–700, 1987.

[15] J. Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm," 2002. [Online]. Available: http://robots.stanford.edu/cs223b04/algo_tracking.pdf

[16] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[17] M. F. Moeller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, vol. 6, pp. 525–533, 1993.