Gain Adaptive Real-Time Stereo Streaming

S. J. Kim[+], D. Gallup[+], J.-M. Frahm[+], A. Akbarzadeh[*], Q. Yang[*], R. Yang[*], D. Nistér[*] and M. Pollefeys[+]

[+]Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, USA

[*]Department of Computer Science
University of Kentucky
Lexington, USA

Abstract. This paper introduces a multi-view stereo matcher that generates depth in real-time from a monocular video stream of a static scene. A key feature of our processing pipeline is that it estimates global camera gain changes in the feature tracking stage and efficiently compensates for these in the stereo stage without impacting the real-time performance. This is very important for outdoor applications where the brightness range often far exceeds the dynamic range of the camera. Real-time performance is achieved by leveraging the processing power of the graphics processing unit (GPU) in addition to the CPU. We demonstrate the effectiveness of our approach on videos of urban scenes recorded by a vehicle-mounted camera with auto-gain enabled.

1 Introduction

In the last few years visualizations of the world from aerial imagery became popular in applications like GoogleEarth and Microsoft Virtual Earth. These data have a poor quality for ground-based visualization. To achieve high-quality ground-level visualization one needs to capture data from the ground. A system that automatically generates texture-mapped, ground-level 3D models should be capable of capturing large amounts of data while driving through the streets and of processing these data efficiently. We introduce a capture system and a processing approach that fulfills these requirements. It reconstructs 3D urban scenes from video data automatically.

The processing has the goal to automatically process very large amounts of video data acquired in an unconstrained manner. This forces us to take shape from video out of the laboratory and to achieve a fieldable system. Additionally the system design is strongly driven by the performance goal of being able to process the large video datasets in a time comparable to the acquisition time. A first overview of our system was presented in [1]. In this paper we introduce the improvements of the system to deal with high dynamic ranges over time of the captured scene and the achievement of real-time processing for the system.

One of the challenges that the system has to overcome is the high dynamic range of the captured scene. In order to capture the full brightness range of natural scenes where parts are in the shadow and others are in bright sunlight the camera has to use automatic gain control. Accordingly the appearance of

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

the same scene point in the video images varies. Most current techniques do not account for those changes automatically. We introduce a real-time tracking technique that estimates the global gain change between consecutive video frames while tracking salient features. The novel proposed stereo algorithm efficiently adapts for those gain changes and runs in real-time on the graphics card.

The paper is organized in the following way. First we discuss the work related to the introduced system and its components. Section 3 gives an overview of the system. Then we introduce our new improved real-time feature tracking which additionally delivers the gain of the images. Afterwards the improved real-time stereo-estimation that employs the estimated gain in section 5 is introduced. In section 6 the evaluation of the proposed algorithms is presented.

2 Related work

Several researchers in photogrammetry and computer vision address the problem of 3D reconstruction from images. There are many systems which are highly flexibile and are economic in size, weight and cost. The challenges for those systems are mostly the well-documented inaccuracies in 3D reconstruction from 2D measurements. We approach the problem using passive sensors only, employing the work on structure from motion and shape reconstruction within the computer vision community in the last two decades [2–5]. Since this literature is too large to survey here, the interested reader is referred to [6–9]. Our emphasis here is to develop a fully automatic reconstruction system which is able to operate in continuous mode without the luxury of capturing data from selected viewpoints since capturing is performed from a moving vehicle constrained to the vantage points of urban streets.

An essential step in image based reconstruction is the identification of salient features and the tracking of those throughout the sequence. Commonly the KLT tracker [10] is used to perform this task. It inherently assumes a constant appearance of the feature through out the video. In the implementation by Birchfield [11], a simple method is used to account for the gain change between images. For each feature patch in the first image, an individual gain is computed using the current estimate of the location of the patch in the second image. The gain ratio is computed by the ratio of mean intensity values of the two patches. The estimated ratio is used to normalize the intensity of neighborhoods of the point in the second image to proceed with the tracking process. Notice that even though the gain ratio is a global parameter for the image it is computed for each individual feature independently. In [12, 13] illumination invariance is achieved by solving for a gain and bias factor in each individually tracked patch. Those approaches again treat the gain as individual parameter of each patch. With a linear camera response function or a known response function, we can solve for the gain as a global parameter instead of for each tracked patch which is both computationally more efficient as well as more stable with respect to noise.

To achieve a 3D reconstruction of the scene a dense stereo estimation is required. We will use planesweeping stereo which was first introduced by Collins [14].

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

It provides a simple way to correlate multiple images without the need for rectification between image pairs. Planesweeping has since proven to be an ideal candidate for efficient implementation. It is especially well suited for an implementation on the graphics processing unit (GPU) since it is heavily optimized to perform the key operation of planesweeping stereo, namely, rendering images onto a planar surface. In [15] Yang and Pollefeys presented a real-time stereo using the GPU. The proposed algorithm performs correlation-based stereo using images from two or more distinct cameras. One important way in which our algorithm differs from theirs, is that we assume a system with a single camera in motion.

Our system is targeted towards the reconstruction of urban environments. A system that reconstructs simple geometric models of urban environments from ground based video of a moving stereo camera was introduced in [16, 17]. Alternatively there are also approaches of the reconstruction of urban environments from LIDAR [18]. The next section will give a brief overview over the components of our system.

3 System overview

The proposed processing pipeline performs the following steps to achieve a fast 3D reconstruction. First the system identifies salient features in each video stream. These salient features are then tracked through the video. Lost salient features are replaced by new features in order to maintain a sufficient number of features for the following processing steps. For this step, our novel proposed tracking technique also estimates the global gain change between consecutive frames to accommodate the varying brightness of the different scene parts.

The tracked features are used to estimate the relative camera positions needed for stereo depth estimation over time with a single camera. Afterwards the estimated camera poses and the gains are employed by our novel stereo with gain adaptation to estimate the depth of all pixels in each video frame. In the next sections we will explain the different components of the system.

4 Feature tracking with gain estimation

The process of finding and tracking good features is a vital step for an automatic 3D reconstruction system. One of the most widely used methods for tracking features is the KLT tracker ([19, 20]). The KLT tracking algorithm computes the displacements of features (dx, dy) between consecutive video frames when the image brightness constancy constraint (Eq. (1)) is satisfied.

$$I(x + dx, y + dy, t + dt) = I(x, y, t) \tag{1}$$

The brightness constancy constraint (Eq. (1)) means that the appearance of an object in consecutive images doesn't change. Usually this is achieved by keeping exposure time and gain of the camera constant. For our application of driving in urban environments this limitation is not acceptable. We need to vary the gain of the camera to capture the full brightness range of natural scenes.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

If the gain of the camera changes to adjust to the brightness of the scene, the brightness constancy constraint (Eq. (1)) is no longer satisfied. Accordingly the performance of the KLT tracker degrades significantly. We propose a variant of the KLT tracker that estimates the gain change of the camera which is a global parameter for the image if the camera response function is linear.

Assuming that the radiometric response function of the camera is linear, we incorporate the gain ratio $\beta = 1 + d\beta$ between the image at time t and the image at time $t + dt$ to Eq. (1) to obtain the following equation

$$I(x + dx, y + dy, t + dt) = (1 + d\beta)I(x, y, t). \tag{2}$$

Assuming equal displacement for all pixels of a patch around each feature (P_i), the displacement for each feature (dx_i, dy_i) and the gain ratio β are estimated by minimizing the following error function:

$$E(dx_i, dy_i, d\beta) = \sum_{x,y \in P_i} (I(x + dx_i, y + dy_i, t + dt) - (1 + d\beta)I(x, y, t))^2, \tag{3}$$

where n is the number of features. Applying the Taylor expansion, Eq. (3) can be approximated by

$$E(dx_i, dy_i, d\beta) = \sum_{x,y \in P_i} (I_x dx_i + I_y dy_i + I_t - d\beta I)^2, \tag{4}$$

where $I = I(x, y, t)$, $I_x = \frac{\partial I}{\partial x}$, $I_y = \frac{\partial I}{\partial y}$, $I_t = \frac{\partial I}{\partial t}$. This expression is minimized when all partial derivatives towards the unknowns are zero. Accordingly for each feature the following equation needs to be solved

$$\underbrace{\begin{bmatrix} \mathbf{U_i} & \mathbf{w_i} \\ \mathbf{w_i}^T & \lambda_i \end{bmatrix}}_{\mathbf{A_i}} \mathbf{x} = \begin{bmatrix} \mathbf{b_i} \\ c_i \end{bmatrix} \text{ with } \mathbf{U_i} = \begin{bmatrix} \sum_{P_i} {I_x}^2 & \sum_{P_i} I_x I_y \\ \sum_{P_i} I_x I_y & \sum_{P_i} {I_y}^2 \end{bmatrix}, \mathbf{w_i} = \begin{bmatrix} -\sum_{P_i} I_x I \\ -\sum_{P_i} I_y I \end{bmatrix},$$

$$\lambda_i = \sum_{P_i} I^2, \mathbf{b_i} = \begin{bmatrix} -\sum_{P_i} I_x I_t & -\sum_{P_i} I_y I_t \end{bmatrix}^T, c_i = \sum_{P_i} I I_t, \mathbf{x} = [dx_i, dy_i, d\beta]^T \tag{5}$$

The unknown displacements (dx_i, dy_i) and the global gain ratio β can be estimated by minimizing the error $E(dx_1, dy_1, ..., dx_n, dy_n, \beta)$ for all features simultaneously. Accordingly Equation (4) can be written as

$$\mathbf{A}\mathbf{x} = \begin{bmatrix} \mathbf{U} & \mathbf{w} \\ \mathbf{w}^T & \lambda \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ c \end{bmatrix} \tag{6}$$

with

$$\mathbf{U} = \begin{bmatrix} \mathbf{U_1} & 0 & \dots & 0 \\ 0 & \mathbf{U_2} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & \dots & \mathbf{U_n} \end{bmatrix}, \quad \mathbf{w} = [\mathbf{w_1}, \dots \mathbf{w_n}]^T, \quad \lambda = \sum_{i=1}^{n} \lambda_i \text{ and } c = \sum_{i=1}^{n} c_i$$

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

$$\mathbf{b} = \left[b_1 \ldots b_n \right]^T, \quad \mathbf{x} = \left[dx_1 \; dy_1 \; dx_2 \; dy_2 \ldots dx_n \; dy_n \; d\beta \right]^T.$$

Since the matrix $\mathbf{A}$ is a sparse matrix, we can take advantage of the structure to find a computationally efficient solution. Both sides of the Eq. (6) are multiplied on the left by $\begin{bmatrix} \mathbf{I} & 0 \\ -\mathbf{w}^T\mathbf{U}^{-1} & 1 \end{bmatrix}$ resulting in

$$\begin{bmatrix} \mathbf{U} & \mathbf{w} \\ 0 & -\mathbf{w}^T\mathbf{U}^{-1}\mathbf{w} + \lambda \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{w}^T\mathbf{U}^{-1}\mathbf{b} + c \end{bmatrix} \tag{7}$$

From Eq. (7), the global gain ratio $\beta = 1 + d\beta$ can be solved by

$$(-\mathbf{w}^T\mathbf{U}^{-1}\mathbf{w} + \lambda) d\beta = -\mathbf{w}^T\mathbf{U}^{-1}\mathbf{b} + c \tag{8}$$

Notice that the inverse of $\mathbf{U}$ can be computed by inverting each 2×2 diagonal block in $\mathbf{U}$ separately (which corresponds to the amount of work needed for the traditional KLT). Once $d\beta$ is found, solving for displacements becomes trivial. For each patch i, dx_i and dy_i are solved by back substituting $d\beta$. Hence the proposed estimation adds one additional equation (8) to solve to the original KLT tracking equations.

5 Gain adaptive multi-view real-time stereo

The multi-view stereo module takes as input the camera poses, gains and images from a single video stream and estimates a depth map that contains the depth for each pixel of the frame assuming the scene is static. It uses the plane-sweep algorithm of Collins[14], which is an efficient multi-image matching technique that is based on projective texture mapping. The multi-view stereo algorithm is summarized in Algorithm 1.

In stereo, it is critical to account for changes in exposure in the images while computing the cost C_L, C_R in step 2 in Algorithm 1. We show how to account for changes using the gain ratio β obtained from KLT tracking as described in section 4. To compare intensity values recorded in different images, they must first be normalized for the gain change. According to equation 2, this is done for consecutive frames simply by multiplying by the image's gain ratio β_i which is the ratio of the gain of image $i+1$ and the gain of image i. For non-consecutive frames, the gain ratio $\beta(i,j)$ between frame i and frame j may be obtained by multiplying the gain ratios of the frames in between

$$\beta_{i,j} = \prod_{l=i}^{j-1} \beta_l \text{ and } \beta_{j,i} = {\beta_{i,j}}^{-1}$$

Thus in stereo, we define the dissimilarity measure C_L (resp. C_R) between the reference view I_{ref} and the warped view I'_i as follows

$$C_L(y,x) = \sum_{i<ref} |\beta_{i,ref} I'_i(y,x) - I_{ref}(y,x)|, \tag{9}$$

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

where $\beta_{i,ref}$ is the gain ratio between image I_{ref} and image I_i. Note that by normalizing with respect to the reference image we avoid the necessity to handle the full dynamic range.

To achieve real-time performance, the steps 2, 3, 4 in Algorithm 1 are performed on GPU. In step 2, for each view, the four vertices are projected onto

Algorithm 1 Multi-view Stereo

```
1: A plane is swept through space in steps along a predefined direction, typically
   parallel to the optical axis of the reference view.
2: At each position of the plane, all images ($I_i$) are projected on the plane and rendered
   in the reference view ($I_{ref}$), and two slices of matching cost are calculated as:
       $C_L(y,x) = \sum_{i<ref} |\beta_{i,ref} I'_i(y,x) - I_{ref}(y,x)|$
       $C_R(y,x) = \sum_{i>ref} |\beta_{ref,i} I'_i(y,x) - I_{ref}(y,x)|$
   where $I'_i$ are the projected views.
3: Under the local smoothness assumption, a boxcar filtering is performed throughout
   the two sets of the slices of matching cost. For each pixel, the one with smaller match
   cost is selected.
4: Depth values are computed by selecting the minimal aggregated correlation volume
   at each pixel.
```

the reference view by the plane that is swept through space in steps, and the indices of the remaining pixels are computed by the interpolation of the four vertices representing the image corners (using vertex shader, a programmable unit in the graphics hardware). In addition, since the graphics hardware is most efficient at processing 4-channel (RGB + alpha) color images, this allows us to compute four depth hypotheses at once. Once the projective mapping is completed, we use the pixel shader to calculate the gain corrected absolute difference (10) of the projected views and the reference view, which is written to an output texture. Since the time performance of our stereo is bound by the projective texture mapping on the GPU, gain correction does not add a measurable overhead. One may observe that by delaying gain correction until after image warping we must perform the per-pixel multiplication for every plane hypothesis. The image could be normalized before warping, thus saving computation time. However, this would require higher precision for image storage to account for the possible dynamic range of natural scenes. Moreover a higher precision would add a significant requirement for memory bandwith. To avoid this overhead the image values are stored as 8-bit integers and only converted to higher precision floating point numbers during the computation of the dissimilarity. In our implementation, we have verified that there is no observable time penalty for applying gain correction.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

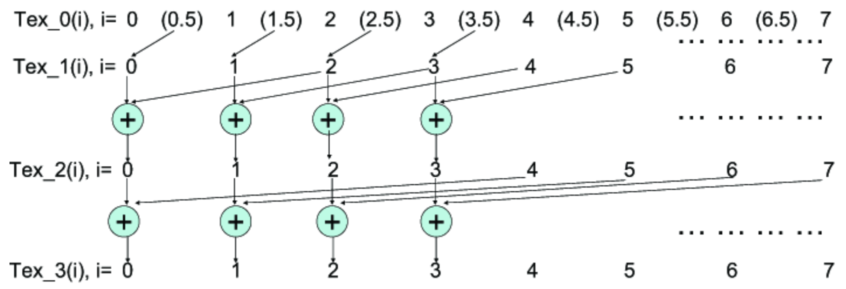
Tex_0(i), i= 0 (0.5) 1 (1.5) 2 (2.5) 3 (3.5) 4 (4.5) 5 (5.5) 6 (6.5) 7
Tex_1(i), i= 0 1 2 3 4 5 6 7
Tex_2(i), i= 0 1 2 3 4 5 6 7
Tex_3(i), i= 0 1 2 3 4 5 6 7

Fig. 1. On the right are the indices of the textures, on the left textures that are used to save the aggregation results.

Cost aggregation is performed in step 3. We implement a GPU boxcar filter with multiple passes. In the first pass, we take advantage of the graphics card's bilinear texture interpolation to compute a 2×2 boxcar filtering result. For a given pixel (x, y) we access the cost image C at address $(x + 0.5, y + 0.5)$. The graphics card's built-in bilinear interpolation returns the average of four pixel costs, $(C(x, y) + C(x + 1, y) + C(x, y + 1) + C(x + 1, y + 1))/4$, which we then multiply by 4 before storing the result in an 8-bit texture. This avoids losing precision to discretization for low-cost regions while the potential saturation in high-cost regions has no impact on the best cost result. The result is written to an output texture C_1 which stores the 2×2 boxcar result. In each subseqent pass, the results from previous boxcars are combined to produce a boxcar result of twice the size in each dimension. Thus in pass i for pixel (x, y), we compute the $2^i \times 2^i$ boxcar result from the previous result C_{i-1} as $C_{i-1}(x, y) + C_{i-1}(x + 2^{i-1}, y) + C_{i-1}(x, y + 2^{i-1}) + C_{i-1}(x + 2^{i-1}, y + 2^{i-1})$. Figure 1 summarizes the algorithm. The advantage of this approach is that the memory access is continuous. Although more texture memory accesses are required, this approach is faster than alternative approaches.

6 Experimental results

The experimental evaluation of the of the proposed techniques is discussed in the following. First we evaluate the feature tracking with gain estimation from section 4. The improvement of the stereo estimation by involving the estimated gain will be discussed afterwards.

First the gain value of a natural image was changed by multiplying all gray values with a constant β to test the gain estimation in the tracking. This image was used to compare the tracking performance of the original KLT tracker and our new KLT tracker with gain estimation. Fig. 2 shows result for a gain ratio of $\beta = 0.8$. While the normal KLT shows poor performance, our new algorithm performs comparably to the case with no gain change. Our technique estimated a gain ratio of $\beta = 0.7997$ for this test.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

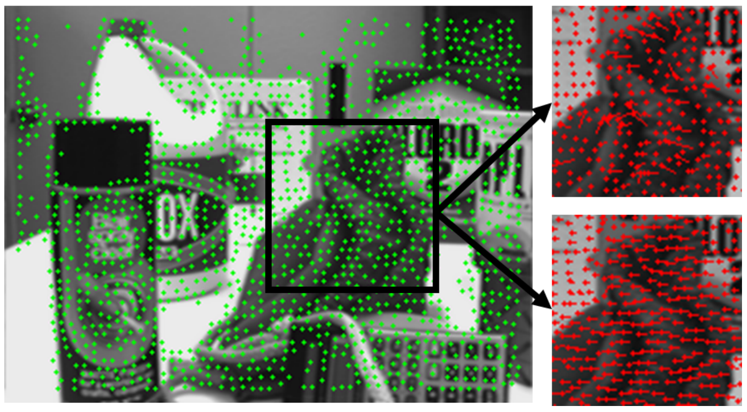

Fig. 2. (left) First image and the extracted features (top right) second image (gain = 0.8) and the features tracked by the normal KLT tracker (bottom right) second image (gain = 0.8) and the features tracked by our new KLT tracker

Additionally the system was evaluated on video captured by a 1024x768 resolution Point Grey Flea with a 40 degree field of view and a linear response function. The camera operated at 30 fps and all camera settings were held fixed except for automatic gain control to account for the large variation in illumination of the scene. The camera was mounted on a vehicle moving at roughly 10 km/h during video capture. This particular camera also featured the ability to embed the automatically selected gain factor in the image data. We used this information to verify the gain estimation of our KLT tracker.

The object of interest was the exterior wall of a building approximately 6 meters from the camera. This building was surveyed to obtain a ground truth model of the building with centimeter accuracy[1]. Additionally, a GPS/INS system was mounted on the vehicle to assist in pose estimation and to provide a geo-located coordinate frame for comparison with the ground truth model.

The change of the gain for the sequence of 400 images was computed with the new proposed tracker and is shown in Figure 3. To evaluate our new stereo algorithm, we selected a part of the video in which the gain changes significantly. This change is caused by capturing first in bright sunlight and afterwards in shadow. For the experiments we used a camera that is able to report the current gain value to have an approximate ground truth measurement. The recovered cumulative gain ratio is 1.4668 over 11 frames.

We measure the results of our stereo implementation by comparing the recovered depth to the ground truth model. Each point in the depth map is converted into a 3D point in the geo-located coordinate frame. The error is then computed as the minimum distance to the ground truth surface.

In Figure 4 we see that our system is able to correctly handle the large change in gain and produce an accurate depth map. The gain-corrected depth map has

[1] data are courtesy of the UrbanScape project

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

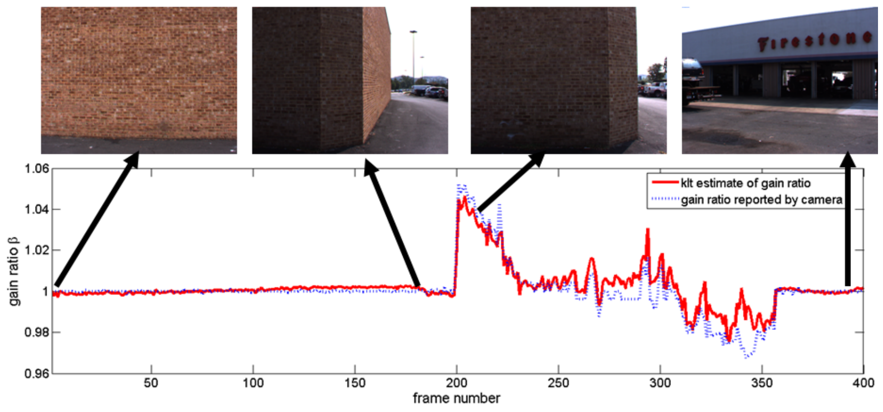

Fig. 3. Estimated gain for the video sequence and example images.

1.9 cm average error. Without accounting for gain, the depth map produced has severe errors: 37.4 cm on average.

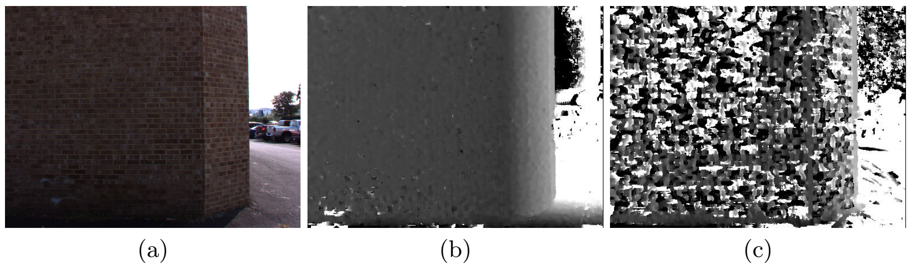

Fig. 4. (a) video frame, (b) novel gain corrected stereo, (c) standard stereo.

7 Conclusions

We presented a novel system for real-time streaming stereo computation. To achieve a robust approach we proposed a new 2D-tracking approach that estimates the gain change in a video sequence while tracking the 2D features. This improved the 2D tracking performance and delivered the gain value. Additionally we proposed a mechanism to use the computed gain value to improve stereo depth estimation. Finally we showed that the novel system has a significantly improved performance that allows the streaming of video with auto-gain.

Acknowledgements This work was partially funded by the NSF Career Award IIS 0237533 and the Packard foundation. This work is also partially supported by DARPA under the UrbanScape project, which is lead by the Geo-Spatial Technologies Information Division of SAIC. This document has been approved for public release, distribution unlimited, by DARPA.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

References

1. Akbarzadeh, A., Frahm, J.M., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Merrell, P., Phelps, M., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H., Nister, D., Pollefeys, M.: Towards urban 3d reconstruction from video. In: 3DPVT. (2006)
2. Pollefeys, M., Koch, R., Van Gool, L.: Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. Int. J. of Computer Vision **32**(1) (1999) 7–25
3. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. on Pattern Analysis and Machine Intelligence **26**(6) (2004) 756–777
4. Nistér, D.: Automatic passive recovery of 3d from images and video. In: 3DPVT. (2004) 438–445
5. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. Int. J. of Computer Vision **59**(3) (2004) 207–232
6. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. of Computer Vision **47**(1-3) (2002) 7–42
7. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithm. In: Int. Conf. on Computer Vision and Pattern Recognition. (2006) I: 519–528
8. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000)
9. Faugeras, O.: Three-Dimensional Computer Vision: A Geometric Viewpoint. MIT Press (1993)
10. Lucas, B., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Int. Joint Conf. on Artificial Intelligence. (1981)
11. Birchfield, S.: Derivation of Kanade-Lucas-Tomasi Tracking Equation (1997)
12. Jin, H., Favaro, P., Soatto, S.: Real-time feature tracking and outlier rejection with changes in illumination. (2001) I: 684–689
13. Baker, S., Gross, R., Matthews, I., Ishikawa, T.: Lucas-kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (February 2003)
14. Collins, R.: A space-sweep approach to true multi-image matching. In: Int. Conf. on Computer Vision and Pattern Recognition. (1996) 358–363
15. Yang, R., Pollefeys, M.: Multi-Resolution Real-Time Stereo on Commodity Graphics Hardware. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (2003) 211–218
16. Cornelis, N., Cornelis, K., Gool, L.J.V.: Fast Compact City Modeling for Navigation Pre-Visualization. In: In Proceedings CVPR. (2006)
17. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.V.: 3D City Modeling using Cognitive Loops. In: International Symposium on 3D Data, Processing, Visualization and Transmission. (2006)
18. Früh, C., Zakhor, A.: An automated method for large-scale, ground-based city model acquisition. Int. J. of Computer Vision **60**(1) (2004) 5–24
19. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. In: Conference on Artificial Intelligence. (1981) 674–679
20. Shi, J., Tomasi, C.: Good Features to Track. In: IEEE Conference on Computer Vision and Pattern Recognition. (1994) 593–600

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de