# libBICOS – An Open Source GPU-Accelerated Library implementing BInary COrrespondence Search for 3D Reconstruction

Christoph Liebender, Michael Bleier, Andreas Nüchter

Computer Science XVII - Robotics, Julius-Maximilians-Universität Würzburg, Germany
{christoph.liebender,michael.bleier,andreas.nuechter}@uni-wuerzburg.de

**Technical Commission II**

**Keywords:** Multi Shot Stereo Vision, Structured Light, 3D Reconstruction, Pixelwise Correspondence Search, Binary Descriptors, Free Implementation.

**Abstract**

In this paper, we present an implementation and publish an open source library for binary correspondence search (BICOS), an efficient method for accurate 3D reconstruction from structured light stereo imagery. Starting with two stacks of stereo-rectified images of a scene illuminated by a statistical light pattern the proposed method solves the problem of a pixelwise correspondence search. Our GPU-accelerated implementation reduces the latency of disparity computation using 7MP images on modern hardware down to 20 milliseconds. Based on the algorithm introduced by Dietrich et al. (2019), we extend their approach by increasing the descriptor size while augmenting postprocessing to increase its applicability on other types of pattern projections. Lastly, we provide benchmarks and example reconstructions using a stereo camera setup combined with an off-the-shelf projector to validate the algorithm's performance. While many state-of-the-art single-shot stereo implementations are included in common computer vision libraries, high performance multi-shot methods are not broadly available. By publishing this method as a freely available library, in both a CUDA and CPU implementation, we hope for others to quickly gain traction in this field. The source code with build instructions and command-line tooling is available at `https://github.com/JMUWRobotics/libBICOS` under the GNU LGPLv3.

## 1. Introduction

STEREO vision using cameras is a cost-effective and versatile way of 3D reconstruction. A common solution here are single-shot stereo vision methods, where only a single, rectified and synchronized pair of input images is used to compute the depth of a scene. However, the accuracy of those single-shot approaches is limited. Especially badly lit, unstructured or otherwise uncooperative scenes cause errors in these measurements in the best case and render 3D reconstruction impossible in the worst case. A solution to this problem are active stereo vision systems, where a statistical light pattern is projected into the scene. This makes it possible to work in environments with little to no light, while adding structure to structureless areas, lifting the requirement on the scene's cooperativeness.

Projecting changing light patterns opens up the field of multi-shot stereo vision, since the change of a object point's projected pixel intensity can be recorded over time to be matched in the second view of the scene. Unfortunately, computing the disparity map from two stacks of high resolution images proves to be a computationally expensive process, limiting the applicability of this approach. With increasing camera resolution and framerate, typical methods based on computationally expensive correlations of temporal intensity vectors for disparity computation quickly become unapplicable in real-time scenarios.

Still, correlation-based methods provide accurate correspondence results. Ideally, the amount of pixels to be processed by correlation is reduced to a number that is smaller than the initial $\mathcal{O}(HW^2)$, with image height $H$ and width $W$. This stems from the fact that for pixelwise matching, every pixel in the reference image $(H \times W)$ requires searching on its epipolar line in the match image $(W)$. BInary COrrespondence Search (BICOS) implements such a preprocessing, by reducing the correlation problem of intensity values to one of comparing binary vectors.

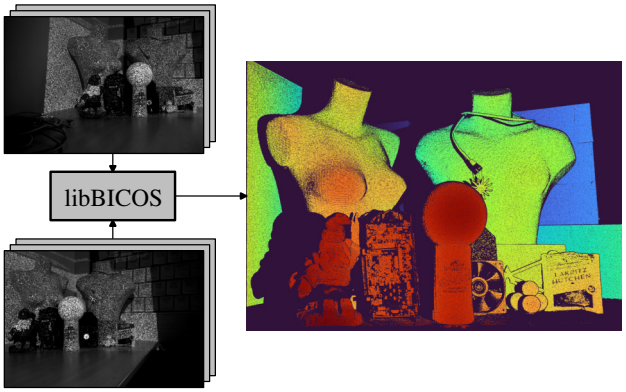


Figure 1. Overview of the inputs and outputs of libBICOS. Left: two stacks of rectified multi-shot structured light stereo images as input. Right: computed disparity map as output, colorized.

This paper's main contributions are:

- A thorough description of the implementation details of the BICOS algorithm.
- Extensions and modifications to the original approach of Dietrich et al. (2019) to improve applicability for different measurement setups.
- An accuracy evaluation of an example stereo vision setup using reference shapes and additional free-form scans.
- The proposed implementation's publication as a freely-available library.

## 2. Related Work

Narrow baseline stereo vision can generally be considered a well-understood problem of computer- and machine vision. Regarding single shot approaches, Semi-global Matching (SGM) of Hirschmüller (2008) is a widely used baseline. Via initial pixelwise cost aggregation of matching costs over vertical, oblique and horizontal paths and subsequent disparity computation by cost-minimization, it is able to create dense disparity maps from single shot stereo images. For cost computation, different possibilities exist: Hirschmüller (2008) proposes the usage of mutual information (Kim et al., 2003). However, publicly available implementations use other metrics: The implementation *libSGM* (fixstars Development Team, 2024) uses the census transform (Zabih and Woodfill, 1994), while the variant in *OpenCV* (OpenCV Development Team, 2024) employs a sample-insensitive metric (Birchfield and Tomasi, 1998).

Latest works achieving state-of-the-art single-shot stereo performance on challenges like ETH3D (Schops et al., 2017) and KITTI (Geiger et al., 2012) use learned methods utilizing complex neural network architectures. Li et al. (2022) use feature extraction networks to compute the correlation of features along an input image pyramid, while Li et al. (2024) do both monocular and binocular disparity estimation, to refine an initial disparity estimate using local structure information based on normal maps, followed by upsampling of the disparity result to the original input resolution. Work of Min and Jeon (2024) using a U-net transformer (Ronneberger et al., 2015) shows that learned methods are able to output a confidence map of the resulting estimated disparity map, for helping decision based applications like autonomous driving or robotics. However, learned approaches on depth estimation from stereo imagery rely on large amounts of training data, which is mainly sourced from synthetically generated datasets. Thus, it is not clear whether or not those methods are reliable enough for high-accuracy 3D reconstruction.

Instead of optimizing the matching process of passive stereo imagery, higher accuracy is achieved by combining light pattern projection and matching algorithms adapted to the respective projection. One approach is to project laser lines into the scene to either match points on these lines in both cameras (Liu et al., 2024), or to only have one camera with a calibrated projector (Bleier and Nüchter, 2017). Others project light patterns: either speckles or point patterns (Gu et al., 2020; Schaffer et al., 2010, 2011; Zhong et al., 2019), or vertical fringes (Heist et al., 2016; Scharstein and Szeliski, 2003; Zhang et al., 2018) that are easily set up by placing a goes-before-optics (GOBO) wheel in-between a high power light source and a lens system. These varying light patterns lead to multi-shot approaches where a time-series of left and right views are used for correspondence search (Dietrich et al., 2019; Heist et al., 2015). Lastly, if measuring latency is not critical, entirely coded patterns help matching correspondences by assigning a unique code to each pixel (Batlle et al., 1998; Salvi et al., 2004; Young et al., 2007).

## 3. Algorithm

In this section, we show the algorithm's methodology behind each step shown in Figure 2. Starting with two stacks of images $\mathbf{I}_i$, the descriptor images are computed as $\tilde{I}_i$. These are then exhaustively matched as the raw disparity $\tilde{D}$. Then, $\tilde{D}$ and $\mathbf{I}_i$ are combined to produce the resulting filtered disparity $D$.

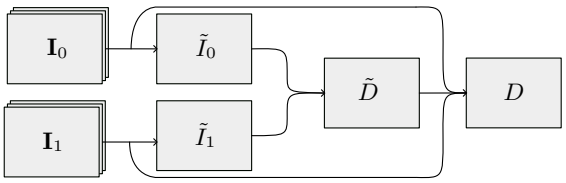


Figure 2. Flowchart of the matching process. Stacks of images $\mathbf{I}_i$. Descriptor images $\tilde{I}_i$. Raw disparity $\tilde{D}$. Disparity $D$.

### 3.1 Descriptor Transform

Initially, we start with two stacks $\mathbf{I}_i$ of $n$ images each, from both reference and match camera. To be able to preprocess these stacks via binary correspondence search, we first need to compute descriptors $\tilde{I}_i$ from these stacks.

Given a stack of images, a pixelwise termporal block is defined as the vector of intensities $I_t$ over time of a pixel $p$ at top-left centered sensor coordinates $(x, y)$. Then, the bits of the binary descriptor at $p$ get set according to Dietrich et al. (2019) by comparing the intensities $I_t$ at different time points, visualized in Figure 3. This yields:

- $n-1$ comparisons of neighboring pairs $I_t, I_{t+1}$,
- $n$ comparisons against the mean intensity $\bar{I}$,
- $n-2$ comparisons of non-neighboring pairs $I_t, I_{t+2}$,
- $n^2 - 5n + 6$ comparisons of non-overlapping pairsums.

The last bits are generated from comparing each pairsum to every other pairsum, such that pairsums don't overlap in their summands. The number of bits generated from this last part is quadratic in $n$, thus, they have a large influence on the resulting descriptor. Because of this, we also implement a 'limited' variant for our evaluation, where only neighboring, non-overlapping pairsums are compared, as shown in Figure 3. This yields $n-4$ bits.

While Dietrich et al. (2019) limit descriptor size to a maximum of 64 bits, we increase this limit to 128 bits since modern compilers support it via `__uint128_t`. In practice, we dynamically truncate the integer width to either 32 or 64 bits, if the input image stack size allows for it.

### 3.2 Exhaustive Epipolar Correspondence Search

Given two descriptor images $\tilde{I}_i$, we now exhaustively search for correspondences on epipolar lines. Assuming ideal stereo rectification, we compare a descriptor in $\tilde{I}_0$ against every descriptor in the corresponding row in $\tilde{I}_1$.

As a matching cost, we follow Dietrich et al. (2019) to compute the bitwise hamming distance, defined as the number of bits that differ between both descriptors:

$$C_{\mathrm{ham}}(p, d) = \sum_j \tilde{I}_0(p, j) \oplus \tilde{I}_1(q, j) \;, \tag{1}$$

over $j$ bits in both descriptors, where $q = (x - d, y)$. Modern processor architectures are able to reduce this expression to just two intstructions: `xor` and `popcount`, given that our descriptors

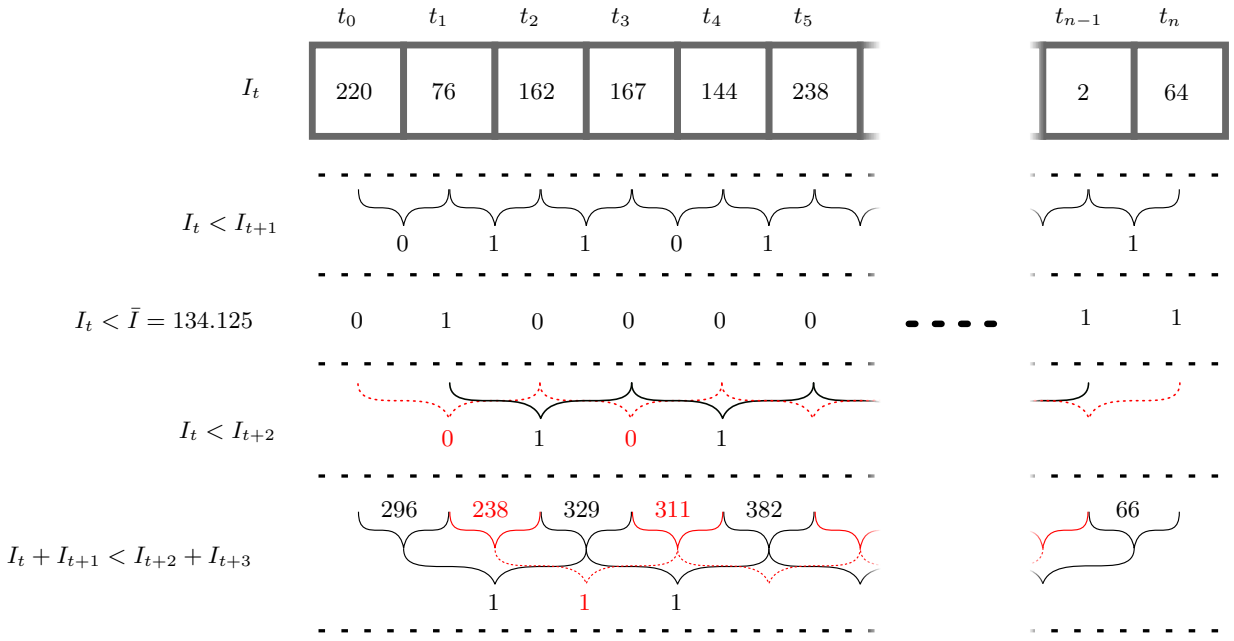


Figure 3. Principle of the binary descriptor generation. Given a sequence $t_{0\dots n}$ of intensities $I_t$ of a single pixel, the binary descriptor is composed of different comparisons of these intensities, pairsums, or the mean intensity $\bar{I}$.

can be easily represented as unsigned integers. The disparity at $p$ is then determined as:

$$d(p) = \arg\min_d C_{\text{ham}}(p, d) \ . \tag{2}$$

A high number of pixels per image row may cause multiple, local minima to exist. To address this, only disparities where a global mimimum is found on the epipolar line are accepted.

This contrasts Dietrich et al. (2019), where they choose to do a reverse-search along the epipolar line in the reference image. We empirically found our way to be more efficient, since for each search of a descriptor at $p$ in $\tilde{I}_0$, only the epipolar line of $\tilde{I}_1$ needs to be cached in memory.

### 3.3 Crosscorrelation- and Variancefilter

The resulting raw disparity map $\tilde{D}$ still contains many outliers; to obtain a usable result, additional postprocessing is necessary to finally obtain $D$. For this step, we use the normalized cross correlation:

$$\text{NXC}(p, d) = \frac{\text{Cov}\left[I_0(p), I_1(q)\right]}{\sqrt{\text{Var}\left[I_0(p)\right] \cdot \text{Var}\left[I_1(q)\right]}} \ , \tag{3}$$

where we skip normalizing by $n$ for brevity:

$$\text{Cov}\left[I_0(p), I_1(q)\right] = \sum_{t \in n} \left[I_0(p,t) - \bar{I}_0(p)\right] \left[I_1(q,t) - \bar{I}_1(q)\right] \ , \tag{4}$$

$$\text{Var}\left[I(p)\right] = \sum_{t \in n} \left[I(p,t) - \bar{I}(p)\right] \ . \tag{5}$$

Given a raw correspondence $\tilde{d}$ at $p$, we now test the actual pixel intensities in the stacks $\mathbf{I}$ for correlation. For this, we introduce a configurable threshold $\theta_{\text{NXC}} \in (0, 1)$, and invalidate each raw disparity where the computed NXC $< \theta_{\text{NXC}}$. Note that we do not compute this correlation for each intensity vector on the epipolar line, only for existing raw correspondences of $\tilde{D}$ where some disparities with non-global minima of $C_{\text{ham}}$ have been removed. Thus, this step is in worst case $\mathcal{O}(HW)$.

Adding onto the correlation-filter which was proposed by Dietrich et al. (2019), we introduce an additional variance-based filter. This is motivated by the fact that depending on the stereo setup and its projected light pattern, not every pixel in $\mathbf{I}$ may vary sufficiently for correlation-based methods to yield good results. Especially in cases where it is not possible to cover the whole field of view of the camera system with a varying light pattern, pixels either don't change their intensity entirely or only based on noise, which produces many, well correlated outliers. (See input stereo pairs of Figures 1 and 12 for an example where not every pixel is covered by structured light.) By introducing a second configurable parameter $\theta_{\text{Var}}$, we allow to filter for the majority of coarse outliers caused by non-ideal setups by discarding disparities where $\text{Var}\left[I\right] < \theta_{\text{Var}}$, for either $I_{0,1}$. Since computing variances is required for cross-correlation, we do not need to add another step to our filtering process. In contrast to Dietrich et al. (2019), we do not apply additional median-filtering on the resulting disparity map $D$ since with a higher descriptor size and proposed modifications very few outliers remain. Different filtering strategies can still be applied as a postprocessing step to improve the 3D reconstruction results.

### 3.4 Subpixel-Interpolation

Lastly, to refine our disparity map, we implement subpixel-interpolation as proposed by Dietrich et al. (2019) as follows. Given a raw disparity $\tilde{d}$, a second-degree polynomial $\tilde{I}_1(z, t) = a(t) \cdot z^2 + b(t) \cdot z + c(t)$ is fit through the spatial neighborhood $q \pm 1$ on the epipolar line of the match pixel, with:

$$\begin{bmatrix} a(t) \\ b(t) \\ c(t) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \\ -1 & 0 & 1 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} I_1(q-1, t) \\ I_1(q, t) \\ I_1(q+1, t) \end{bmatrix} \ , \tag{6}$$

such that a maximum NXC on that curve with $z \in [-1, 1]$ can be compared against the filters of previous section. In practice, we use a discretized, configurable pixel stepsize $\sigma_p \in (0, 0.5]$.

Figure 4. Experimental stereo camera system with mounted projector.

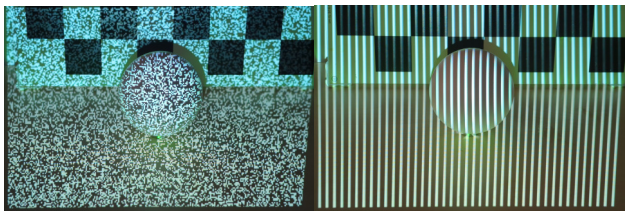

Figure 5. Projected light patterns. Left: varying binary noise similar to static noise from analog television. Right: fringes where inter-fringe distance and fringe width vary over time.

## 4. Experiments

For our experiments, we employ the stereo camera system shown in Figure 4, comprised of two synchronized 7MP ORX-10G-71S7C-C cameras from Teledyne mounted with a baseline of approximately 38 cm. Instead of a GOBO-projector as used by the original authors, we use an unsynchronized aftermarket projector that projects either a binary noise pattern, similar to the static noise in analog television, or a fringe pattern; both are shown in Figure 5. The stereo setup is calibrated with 80 image-pairs of a ChArUco-pattern using the method of Zhang (2000).

### 4.1 Scans of Known Artefacts

To give a reference on the achievable accuracy with our implementation, the scans are evaluated on a reference plane, sphere and dumbbell (Figure 6), using either binary noise or fringe projection, based on the error metrics defined in the industry standards (VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2012). Note that this evaluation does not include typical postprocessing present in commercial 3D sensors. That is: scans of the artifacts are acquired by computing the pointcloud from a single depth map of our BICOS implementation, without multi-scan fusion or median filtering. We only remove outliers based on a $6\sigma$ rule, where points exceeding 6 standard deviations are considered as outliers, and the respective fit is recomputed.

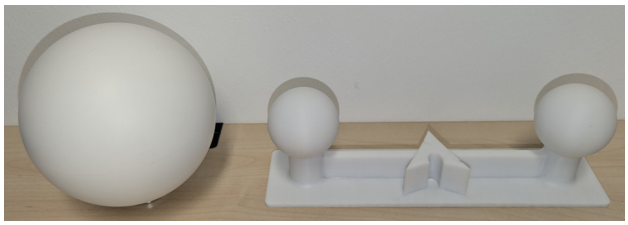

Figure 6. Spherical scanning artefacts. Left: reference sphere with diameter of 145 mm. Right: SLA-printed dumbbell with 60 mm diameter spheres, spaced 220 mm apart.

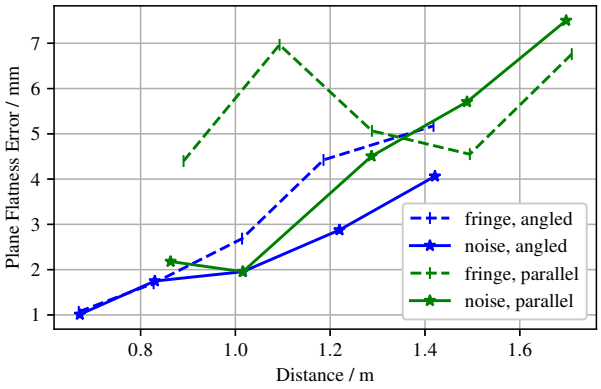


Figure 7. Plane flatness error of fringe (dashed) and binary noise (solid) pattern projections, with both a parallel (green) and angled (blue) reference plane. Using the $6\sigma$ rule, we exclude a maximum of 6.47‰ points for the case of parallel plane and fringe projection.

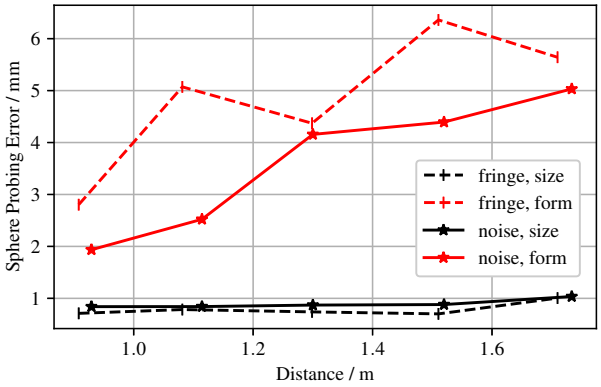


Figure 8. Sphere probing errors of fringe (dashed) and binary noise (solid) pattern projections, with size (black) and form (red) errors. The $6\sigma$ rule excludes 7.68‰ points on form measurements with fringe projection. No points are excluded for size measurements.

**4.1.1 Plane Flatness Error** The flatness error is shown in Figure 7, over a measurement volume of 0.5-1.7 m. It is computed from the thickness of a best-fit plane, which in turn is the result of a least-squares fit. Two plane-configurations are measured: one where the plane is perpendicular to the scanner's main optical axis (fronto-parallel), another where it is angled at approximately 45°. At small distances, flatness-errors as low as 1-2 mm are observed.

**4.1.2 Sphere Probing Error** Consisting of two metrics, we compute the least-squares fit of a variable-radius sphere, its thickness (form) and size-deviation from a known radius over the same measurement range as in our evaluation of the plane flatness error. The results in Figure 8 show that size-errors are in the sub-millimeter range, and form-errors are in the low millimeters from 2 mm to 5 mm in the case of binary noise projection. Both plane flatness and sphere probing errors show that for this measurement setup more accurate results are achieved by using binary noise projection.

**4.1.3 Sphere Spacing Error** Lastly, we measure a dumbbell of two spheres with known distance. By fitting two, fixed-radii spheres, we compute the length error shown in Figure 9. Over a

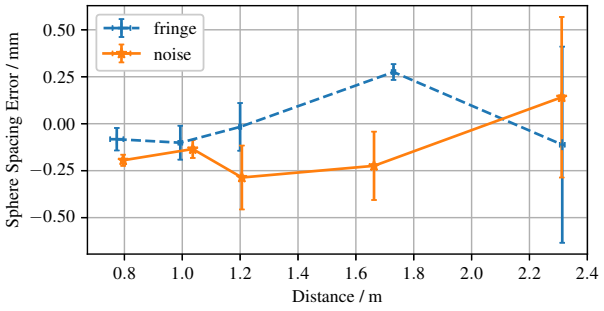


Figure 9. Sphere spacing error of fringe (dashed, blue) and binary noise (solid, orange) projections. For fringe projections, 6.54‰ points are outside of $6\sigma$. Errorbars indicate 1 standard error of the mean, over different poses of the dumbbell, at the respective measurement distances.

larger measurement range than in previous measurements, length errors do not exceed 0.5 mm. Only the standard error of the mean increases at a larger distance of 2.3 m.

### 4.2 Free-Form Scans

Secondly, we record images using binary noise projection of a free-form scene 'Adam & Eve' and a garden gnome, at a distance of 1 m and 30 cm respectively.

**4.2.1 Adam & Eve** A 3D reconstruction of the scene is shown in Figure 16. To show the difference between both descriptor variants, top of Figure 16 displays the meaningful difference between 12 and 33 images as black points, such that black points in this scan are points present in the scan with 12 images but missing when using 33 images. Thus, these black points are effectively outliers that are removed by increasing the sequence size. The rest of both scans is practically identical.

**4.2.2 Garden Gnome** Similarly, Figure 13 shows outliers removed by changing variants and increasing the sequence size. Given a ground-truth scan of the garden gnome which was created using a FARO ScanArm, we align our scan and show that cloud-to-cloud distances don't exceed 0.75 mm.

### 4.3 Performance Benchmarks

Since we implement the algorithm in CUDA, a very low reconstruction latency is achieved. We benchmark the latency of disparity computation over three configurable parameters: temporal block size $n$, subpixel stepsize $\sigma_p$ and single/double computational precision. The dataset for our benchmarks consists of the same 7MP images taken to compute the reconstruction of Figure 16. Our implementation does have other configurable parameters; however, they do not influence reconstruction latency as much as the following.

**4.3.1 Temporal Blocksize** It is clear that processing more images increases latency. However, we can see in Figure 10 that truncating the descriptor – or in other words: the size of the unsigned integer – significantly speeds up the reconstruction. With full descriptors and six images per camera, we achieve a latency as low as 12.3 ms. Note that our variant with limited descriptors can use 32-bit descriptors with up to 9 images, 64-bit with 17 and 128-bit with 33 images. The limited 128-bit descriptors with 33 images are only marginally slower than full descriptors with 12 images. The remaining parameters of benchmarks in Figure 10 are: $\theta_{\mathrm{NXC}} = 0.95$, $\theta_{\mathrm{Var}} = 1.0$, with no subpixel interpolation.

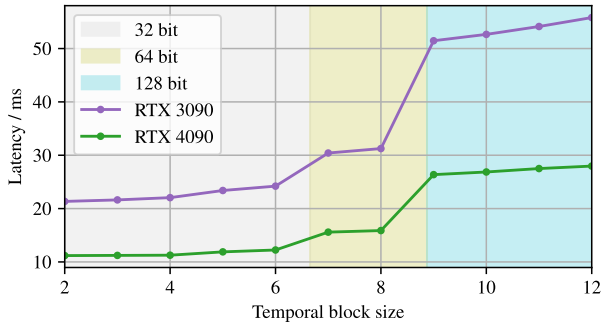


Figure 10. Latency of libBICOS using full descriptors over increasing temporal block sizes, on both Nvidia RTX 3090 and 4090. Jumps in latency are caused by increases in descriptorsize, where up to 6 images per stack fit into 32-bit descriptors, 8 into 64-bit and 12 into 128-bit.

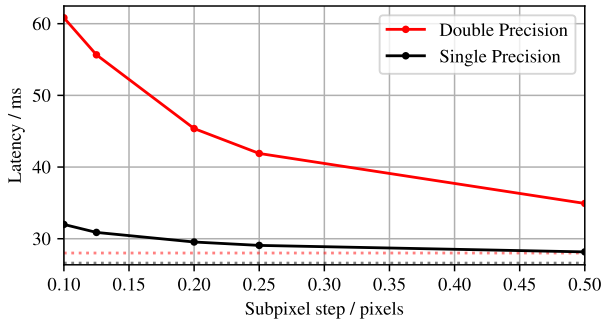


Figure 11. Latency of libBICOS using full descriptors over subpixel stepsize $\sigma_p$, with either double or single computational precision. Dotted lines indicate latency without subpixel interpolation.

**4.3.2 Subpixel Step** The second significant influence on latency is subpixel interpolation, since the amount of computations increases with a lower $\sigma_p$. Figure 11 shows how the latency rises with lower values of $\sigma_p$, when run on a RTX 4090. Since we are working with floating-point thresholds, we may not necessarily require the precision of full 64-bit floating point. Therefore, we enable reducing the computational precision to 32-bit floating point. This effectively halves the latency, with little influence on reconstruction quality. This way, subpixel interpolation with a stepsize of $\sigma_p = 0.25$ only adds 5 ms to the matching process. In this case, we perform benchmarks with full descriptors and $n = 12$ as well as $\theta_{\mathrm{NXC}} = 0.95$, $\theta_{\mathrm{Var}} = 1.0$.

### 4.4 Comparison against libSGM

Comparing libBICOS against libSGM (fixstars Development Team, 2024) as a performant implementation of a single-shot 3D-reconstruction algorithm, we note the following: running on a single pair of 7MP input images, libSGM uses 14GB of VRAM while taking 37 ms on a RTX 4090 to compute a disparity map. However, this particular implementation limits the disparity search range to a maximum of 256 pixels, which will result in any true correspondences with a disparity higher than that to be represented as a wrong match somewhere else in the disparity map. Therefore, it is not well suited for real-time stereo reconstruction on high-resolution images. In contrast, libBICOS required a maximum of 3.2GB of VRAM during our benchmarks, while not having any constraints on disparity search

Figure 12. The garden gnome reference object. Left: overview of the gnome placed on a chair. Right: input stereo image pair (enhanced contrast for display).

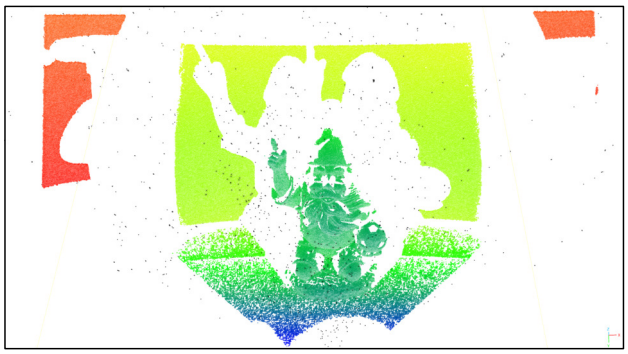

Figure 13. Computed point cloud of the gnome using the same settings as those of Figure 16, colored by depth. Outliers not present in the scan with 33 images are colored in black.

range. Still, libBICOS requires multi-shot structured light, while libSGM is applicable to single-shot stereo reconstruction.

## 5. Conclusion

This work presented an implementation for binary correspondence search (BICOS) for stereo reconstruction. By implementing this algorithm using the CUDA framework, we were able to achieve a disparity computation latency on 7 megapixel input images of less than 1/10th of a second. We extended the already proposed algorithm of Dietrich et al. (2019) by a variable descriptor size of 32-, 64- or 128-bit. Moreover, by adding a variance-threshold on the temporal intensity block the algorithm's flexibility is improved. With its increased flexibility, our implementation is applicable to different kinds of statistical structured light projection, lifting the requirement on sophisticated, customized projector and stereo camera setups. Additionally, we enforce unique descriptor matches between left and right images by formulating a epipolar-line-wise global minimum hamming distance constraint. Experiments using a medium-baseline stereo vision setup and different types of light pattern projection showed that millimeter accuracy is achievable at close range with small baselines. By publicly making our implementation available at `https://github.com/JMUWRobotics/libBICOS` under the GNU Lesser General Public License (LGPLv3), we hope to enable other researchers as well as industry to easily pick up on structured light, multi-shot stereo vision for 3D reconstruction.

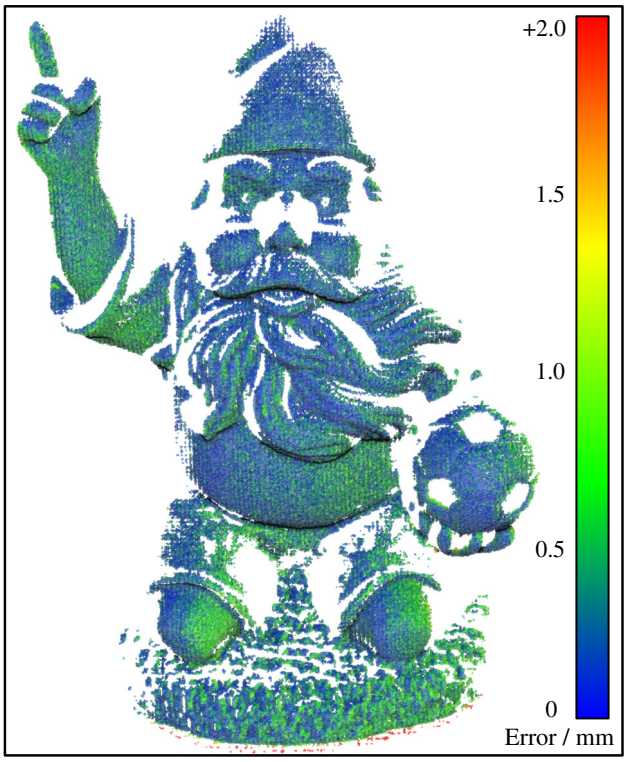


Figure 14. Segmented gnome after aligning it to a ground truth point cloud using the ICP algorithm, colored by cloud-to-cloud distance, with scalebar in millimeters.

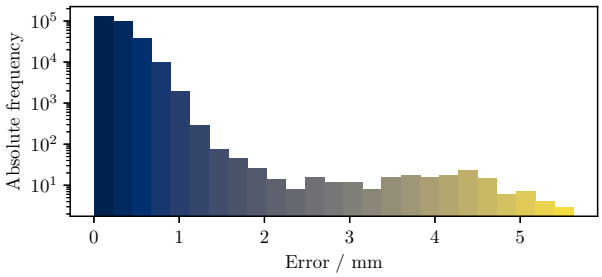


Figure 15. Absolute frequencies of binned cloud-to-cloud distances between scanned gnome and ground truth of Figure 14.


## Acknowledgement

This work was supported by the project 'Mixed Media Scanning' funded by the German Research Foundation (DFG NU230/14-1) and the project WISP funded by the German Federal Ministry for Economic Affairs and Climate Action (03EE3077D). We also acknowledge support from the Elite Network of Bavaria for the academic program Satellite Technology - Advanced Space Systems.


## References


Batlle, J., Mouaddib, E., Salvi, J., 1998. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern recognition*, 31(7), 963–982.

Birchfield, S., Tomasi, C., 1998. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4), 401–406.


Figure 16. Reconstructed point clouds of the scene 'Adam & Eve', colorized by depth. Top: Point cloud computed using 12 images with full descriptor generation. Black points are outliers present in the scan, which are missing when using limited descriptors and 33 images. Bottom left: overview of the scene. Bottom right: highlighted parts of the acquired point cloud, with both fine details of the ornament and accurate shapes of the PC fan, liquorice bag and golf balls. Other chosen parameters are: $\theta_{\mathrm{NXC}} = 0.96$, $\theta_{\mathrm{Var}} = 2.0$, $\sigma_p = 0.1$.

Bleier, M., Nüchter, A., 2017. Towards Robust Self-Calibration for Handheld 3D Line Laser Scanning. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W8, 31–36.

Dietrich, P., Heist, S., Landmann, M., Kühmstedt, P., Notni, G., 2019. BICOS – An Algorithm for Fast Real-Time Correspondence Search for Statistical Pattern Projection-Based Active Stereo Sensors. *Applied Sciences*, 9(16).

fixstars Development Team, 2024. libSGM: a CUDA implementation performing Semi-Global Matching. GitHub. `https://github.com/fixstars/libSGM` (18 October 2024).

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361.

Gu, F., Song, Z., Zhao, Z., 2020. Single-Shot Structured Light Sensor for 3D Dense and Dynamic Reconstruction. *Sensors*, 20(4).

Heist, S., Lutzke, P., Dietrich, P., Kühmstedt, P., Notni, G., 2015. Experimental comparison of laser speckle projection and array projection for high-speed 3D measurements. *Optical Measurement Systems for Industrial Inspection IX*, 9525, SPIE, 282–289.

Heist, S., Lutzke, P., Schmidt, I., Dietrich, P., Kühmstedt, P., Tünnermann, A., Notni, G., 2016. High-speed three-dimensional shape measurement using GOBO projection. *Optics and Lasers in Engineering*, 87, 90–96.

Hirschmüller, H., 2008. Stereo Processing by Semi-Global Matching and Mutual Information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 328–341.

Kim, J. et al., 2003. Visual correspondence using energy minimization and mutual information. *Proceedings Ninth IEEE International Conference on Computer Vision*, IEEE, 1033–1040.

Li, J., Wang, P., Xiong, P., Cai, T., Yan, Z., Yang, L., Liu, J., Fan, H., Liu, S., 2022. Practical Stereo Matching via Cascaded Recurrent Network With Adaptive Correlation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16263–16272.

Li, K., Wang, L., Zhang, Y., Xue, K., Zhou, S., Guo, Y., 2024. LoS: Local Structure-Guided Stereo Matching. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19746–19756.

Liu, Y., Ou, P., Xu, X., Sun, J., 2024. Multi-line structured light binocular vision stereo matching method via coarse-to-fine spatial geometric constraints. *Optics & Laser Technology*, 176, 110950–110957.

Min, J., Jeon, Y., 2024. Confidence Aware Stereo Matching for Realistic Cluttered Scenario. *2024 IEEE International Conference on Image Processing*, 3491–3497.

OpenCV Development Team, 2024. OpenCV: Open Source Computer Vision Library, Version 4.10.0. GitHub. `https://github.com/opencv/opencv` (18 October 2024).

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention*, Springer, 234–241.

Salvi, J., Pages, J., Batlle, J., 2004. Pattern codification strategies in structured light systems. *Pattern recognition*, 37(4), 827–849.

Schaffer, M., Grosse, M., Harendt, B., Kowarschik, R., 2011. High-speed three-dimensional shape measurements of objects with laser speckles and acousto-optical deflection. *Optics Letters*, 36(16), 3097–3099.

Schaffer, M., Grosse, M., Kowarschik, R., 2010. High-speed pattern projection for three-dimensional shape measurement using laser speckles. *Applied Optics*, 49(18), 3622–3629.

Scharstein, D., Szeliski, R., 2003. High-accuracy stereo depth maps using structured light. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 1, I–I.

Schops, T., Schonberger, J. L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A., 2017. A Multi-View Stereo Benchmark With High-Resolution Images and Multi-Camera Videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2012. Optische 3-D-Messysteme - Bildgebende Systeme mit flächenhafter Antastung. VDI/VDE 2634.

Young, M., Beeson, E., Davis, J., Rusinkiewicz, S., Ramamoorthi, R., 2007. Coded structured light. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1–8.

Zabih, R., Woodfill, J., 1994. Non-parametric local transforms for computing visual correspondence. *Computer Vision — ECCV '94*, Springer, 151–158.

Zhang, C., Brahm, A., Breitbarth, A., Rosenberger, M., Notni, G., 2018. Single-frame three-dimensional imaging using spectral-coded patterns and multispectral snapshot cameras. *Optical Engineering*, 57(12), 123105–123105.

Zhang, Z., 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330–1334. https://ieeexplore.ieee.org/document/888718.

Zhong, F., Kumar, R., Quan, C., 2019. A Cost-Effective Single-Shot Structured Light System for 3D Shape Measurement. *IEEE Sensors Journal*, 19(17), 7335–7346.