Spatio-Temporal Point Path Analysis and Optimization of a Galvanoscopic Scanning Laser Projector

Simon Willi* Anselm Grundhöfer†

Disney Research

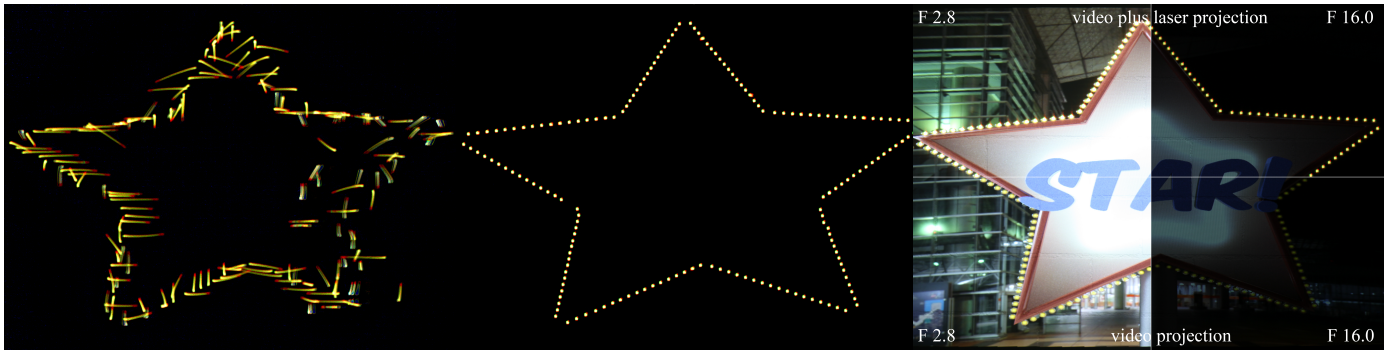


Fig. 1: A star-shaped point pattern projected by a galvanoscopic scanning laser projector. Left, the pattern is displayed in random order without any additional (invisible) control points. The desired spatial locations are not accurately hit, and points are rendered as strokes. Adding a specific number of control points to each vertex location neutralizes these problems at the expense of a reduced overall scanning speed (cf. middle). This slowdown can lead to undesirable perceived flickering. Finding the optimum drawing order and the required minimal number of control points to still achieve spatial accuracy is the goal of our proposed optimization. Registering the device with regular video projectors enables local high dynamic range projections (cf. right).

Abstract—Galvanoscopic scanning laser projectors are powerful vector graphic devices offering a tremendous local brightness advantage compared to standard video projection systems. However, such devices have inherent problems, such as temporal flicker and spatially inaccurate rendering. We propose a method to generate an accurate point-based projection with such devices. To overcome the mentioned problems, we present a camera-based method to automatically analyze the laser projector's motion behavior. With this information, a model database is generated that is used to optimize the scanning path of projected point sequences. The optimization considers the overall path length, its angular shape, acceleration behavior, and the spatio-temporal point neighborhood. The method minimizes perceived visual flickering while guaranteeing an accurate spatial point projection at the same time. Comparisons and timing measurements prove the effectiveness of our method. An informal user evaluation shows substantial visual quality improvement as well.

Keywords: Projector-camera systems, Calibration and registration of sensing systems, Display hardware, including 3D, stereoscopic and multi-user Entertainment, broadcast

1 Introduction

Spatial Augmented Reality, also called shader lamps, video mapping or projection mapping gained a lot of attention within the last decade. It is a method to augment objects, for example buildings, with registered graphics using video projectors. To achieve this goal, either manual or automated geometric calibration methods are used to register a virtual representation to the real object to augment (The interested reader is referred to [1] for an in-depth introduction). Although video projectors have a lot of advantages such as a high spatial resolution and fast frame rates, their restricted peak brightness is still a limiting factor. Scanning laser projectors enable a significant local brightness advantage compared to standard video projection systems. This makes them well suited, for example, for special effects applications in the entertainment industry, in combination with video projectors for local HDR effects or due to its enormous brightness also for daylight augmentations. In contrast to video projectors using lenses for image formation, galvanoscopic mirrors are used in such devices to steer the laser beam. The beam first strikes a mirror controlling the vertical or horizontal axis, then strikes the second mirror controlling the other axis. Given this orientation the laser beam is directed onto a projection surface. In order to perceive the content as static graphics by human observers, the mirrors have to be moved at a high speed. However, the speed of this galvanoscopic mirror movement is constrained by the underlying mechanical structure, the resulting heat generation, and resonant vibrations. These physical limitations can lead to undesired problems like blurring, inertia issues, and flickering. To overcome blurring and inertia, the speed has to be further reduced by a certain number of additional invisible control points per vertex. On one side this ensures a precise spatial rendering (cf. Figure 1 for an example) but on the other side these speed reductions also encourage flickering. In order to avoid flickering effects while still guaranteeing a precise spatial rendering the complexity of the displayed content has to be limited. The perceived strength of the above mentioned problems is influenced by a variety of factors, such as overall path length, sequence of path

*e-mail:simon@disneyresearch.com
†e-mail:anselm@disneyresearch.com

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 2 November 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

traversal, number of vertices to display, and temporal path consistency. Having knowledge about the detailed behavior of the laser system, a model can be set up that can be used to optimize the scanning such that blurring, inertia, and flickering are reduced.

To minimize these effects while simultaneously maximizing the number of displayable points, we present a model-based optimization method using a one-time, camera-based calibration and measurement step that automatically estimates the laser's behavior under varying scanning paths. The captured images are analyzed and compared to the input data. Based on this information, a model is generated that accurately describes the behavior of the laser with respect to the relevant properties. This model is then used to calculate an approximated optimal scanning path that generates a spatially accurate projection of the input points within a minimal amount of time while simultaneously minimizing perceived flickering artifacts, considering the spatio-temporal path distribution. Furthermore, we present a simple registration method to optionally register such devices with standard video projectors.

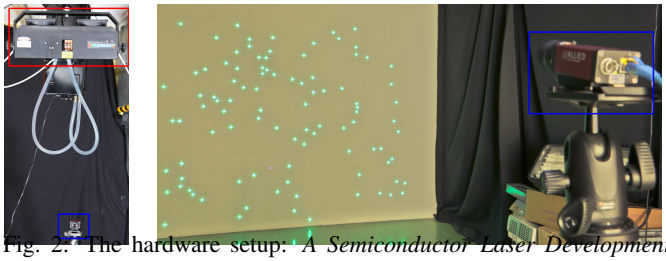


Fig. 2: The hardware setup: *A Semiconductor Laser Development Compact 2.7W RGB* projector (red square left) is used in combination with an *Allied Vision Manta MG504C* machine vision camera (blue square) for calibration, model acquisition, and evaluation.

1.1 Related Work

To our knowledge, little research has been conducted on calibrating and optimizing galvanoscopic scanning laser projection systems. Manakov et al. [12] developed an accurate model of the ray parameters of such systems that can be used to generate a calibrated output by driving the mirror controls directly. Most professional laser projectors, however, offer higher level interfaces, such as defining vectors on a virtual two-dimensional image plane, and therefore can only be accessed and calibrated on a higher abstraction level. We are using a Semiconductor Laser Development Compact 2.7W RGB projector controlled using Pangolin's LD2000 control software [15]. The same company also offers a commercial software tool (LCMax) that is said to automatically optimize the laser path [14]. This is accomplished by using an unknown black-box optimization routine. Applying the algorithm, the results do not seem to generate spatial or temporal optimality. In [17], and [6] Halabi et al. present methods to model the properties of such a higher level controlled system to optimize line drawings by adjusting angles between consecutive line segments and by improving the scanning speed. The parameter modeling was carried out based on human observations, which makes it difficult to generalize. Temporal consistency and point path optimization are not considered in their research.

We focus on the adaptation and extension of the latter methods. In the first step, we carry out a fully automated system calibration and model parameter estimation. In the second step, we use the generated model to optimize the displaying of point data with respect to minimal temporal flickering effects, spatial accuracy, and scanning speed. Finding optimal spatio-temporal paths has been applied in fields such as video data for event detection [20],[21]. In our research, we consider those relationships to further decrease the perceived flickering of the galvanoscopic laser.

2 Model Generation

The speed and spatial precision of the displayed laser path are mainly influenced by the physical movement of the galvanoscopic mirrors. Inertia, potential resonant vibrations, and the required deceleration and acceleration to display individual points exactly at the desired locations are aspects that require a reduction of the overall scanning speed. Because communicating with the laser using abstract interfaces, such as the LD2000 software, allows only high-level control, the galvanoscopic mirrors cannot be influenced directly in their orientation but can only be controlled by several global parameters, such as overall scanning speed, and the number of additional control points the system automatically adds to each point. By fixing these values and analyzing the system's output response using a camera, we discovered that the spatial point drawing accuracy mainly depends on the following factors:

- angle between consecutive points
- distance between consecutive points
- number of additional control points

Based on these properties, we derive a model that can be used to calculate the minimal number of control points for an arbitrary point path, while still guaranteeing a spatially accurate rendering. Therefore, various combinations of these input parameters (angles, distances and control points) were systematically displayed with the galvanoscopic laser projector and captured by a camera. The acquired image data was processed and analyzed to fill a model database. An overview of this process is shown in Figure 3. Please note that the model acquisition needs to be done only once per projector as long as the global parameters have not been changed. This model is then used to define a cost function for a path optimization algorithm, which will be described in Section 3. In the following, each step of this process is explained in more detail (cf. Figure 3).

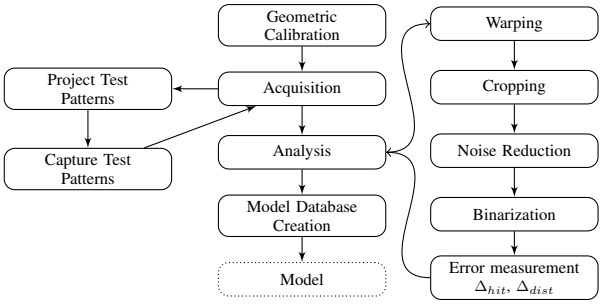


Fig. 3: Overview of the model generation steps. After the initial geometric calibration, the data acquisition is carried out. Next, the data is analyzed to generate the model database.

2.1 Geometric Calibration

To achieve the goal of fast but spatially accurate projection of point sequences, we propose to carry out a camera-based calibration procedure to gather details about the laser's scanning behavior and to ensure that the displayed content is projected onto the desired spatial locations.

Therefore, we use a machine vision camera[1] (cf. Figure 2) to capture specific patterns that are processed and analyzed by comparing them to rasterized versions of the input vector data. To accomplish this, it must be ensured that the camera and projector are either sharing the same image plane — which might be infeasible to achieve — or that a dense mapping exists to warp camera images onto the projector's virtual image plane. To achieve the latter, a $2D$ calibration is carried out by projecting point patterns with binary encoded x/y coordinates onto a close-to-planar surface without discontinuities. An accurate spatial mapping of the points is guaranteed by adding a sufficient number (10) of control points to each displayed point. To speed up acquisition time, only a subset of all possible laser point positions are temporally encoded and projected. After decoding the point coordinates using a blob detection algorithm [11], a mapping is generated between the sub-pixel locations on the camera's image plane

[1] Allied Vision Manta MG504C

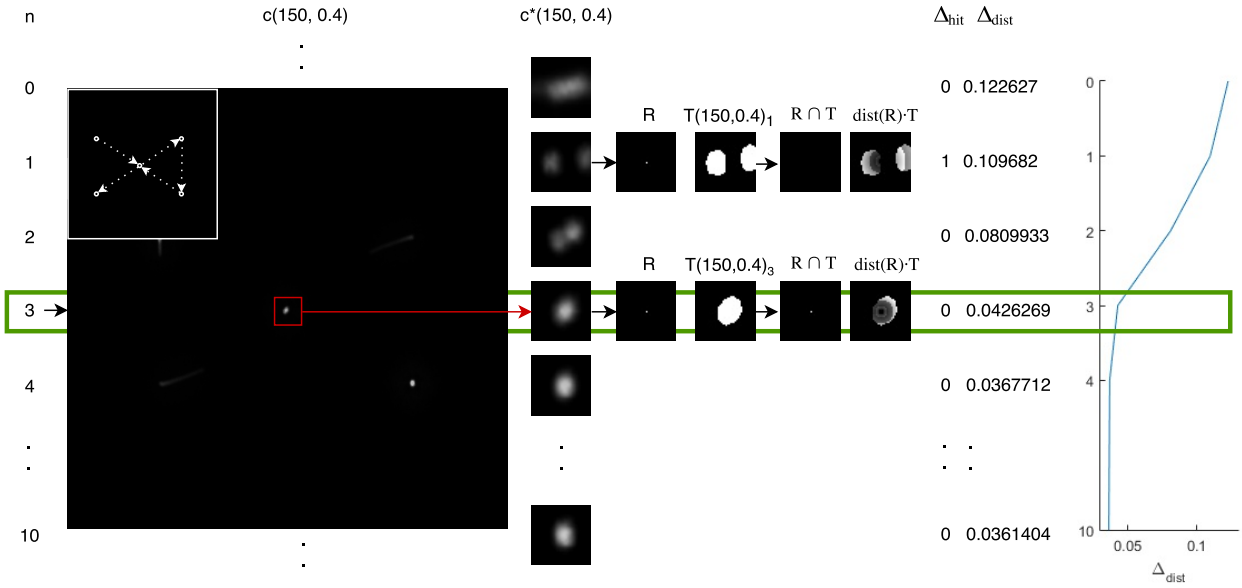


Fig. 4: Sample model generation workflow for the projected pattern with angle $\alpha = 150$ and distance $\delta = 0.4$. On the left, the image that was captured during the measurement using three added control points ($n = 3$) is shown. The input point trajectory is shown in the inset. For analysis, this image is first warped onto the projector's virtual image plane and cropped to a region around the central point (red square). Next, adaptive thresholding is applied and the hit Δ_{hit} and distance Δ_{dist} errors are calculated by comparing the cropped image with the reference point coordinate (R). Section 2.3 describes how the optimal number of $n = 3$ control points (highlighted green) is calculated.

and the corresponding projector coordinates. Missing points are interpolated using thin-plate-spline-based interpolation [4] to generate a dense look-up table (LUT), which enables the precise warping of camera images onto the projector's virtual image plane. If a structured light-based registration is also carried out for an overlapping video projector, both devices can be registered with each other using the LUTs to display superimposed image content using both devices, as shown in Fig 1.

2.2 Acquisition

The last section explained how to geometrically register the laser and the camera. This spatial mapping enables the accurate displaying of points at desired locations in the specific camera view, which serves as the basis to estimate the relevant model parameters.

To measure the laser's behavior with respect to the factors mentioned above, a variety of patterns are projected and captured. For the model data acquisition, a set of point patterns $P(\alpha, \delta)$ with varying angles $(0.0^\circ \leq \alpha \leq 180.0^\circ)$ and normalized distances $(0.2 * d \leq \delta \leq 0.8 * d)$ is projected several times with an increasing number n of control points up to a defined maximum (N), which we will denote as $\{p(\alpha, \delta)_0, ..., p(\alpha, \delta)_N\} \in P(\alpha, \delta)$. d corresponds to half of the height of the virtual image plane and equals $\frac{\sqrt{0.5}}{2}$ of the diagonal. The patterns all follow the same shape: a three-point sequence, starting from the top, down to the center, and up again with an adjusted angle α in between. The same is also rendered vertically mirrored (please refer to the left side of Figure 4 for an example image).

All patterns are captured using a short shutter time, thus avoiding the occurrence of saturated intensities. To ensure that the laser path is fully captured, multiple images of the same pattern are taken with random delays of a few ms in between and averaged, which also significantly reduces the amount of noise in the images. In our measurements, capturing and averaging eight camera images for each projected pattern resulted in the desired image quality.

2.3 Analysis and Modeling

Having acquired the series of input images $\{c(\alpha, \delta)_0, ..., c(\alpha, \delta)_N\} \in C(\alpha, \delta)$, each individual configuration of α and δ is analyzed to find the minimum n that generates the desired result. This is carried out by comparing the images to rasterized representations $R(\alpha, \delta)$ of the input data. Therefore, all members of $C(\alpha, \delta)$ are first warped onto the virtual image plane of the projector using the LUT generated as described in Section 2.1, resulting in $C^*(\alpha, \delta)$.

As we are only interested in the analysis of the central point of the pattern, the images are cropped to a small region around its location. The corresponding cropped image of $R(\alpha, \delta)$ results in a single white pixel at the image center, and as it is the same for all α, δ it will be denoted by R in the following. Next, noise reduction and binarization via intensity thresholding is applied to $C^*(\alpha, \delta)$. As the overall surface illumination, and as a consequence the intensity of the captured images, varies depending on the projected patterns — primarily on the number of added control points — we used the adaptive thresholding method proposed in [2] in combination with a fixed low-intensity threshold to robustly segment all patterns automatically. The resulting binary images for all N added control points $\{t(\alpha, \delta)_0, ..., t(\alpha, \delta)_N\} \in T(\alpha, \delta)$ are compared to R using two error metrics:

- *Hit error*, describing whether the desired point location was illuminated at all:

$$\Delta_{hit} = \begin{cases} 0, & \text{if } R \cap T(\alpha, \delta) \neq \emptyset \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

- *Distance error*, describing how far off the drawing is from the intended point location.

$$\Delta_{dist} = \sum_{\forall pixel} dist(R) \cdot T(\alpha, \delta), \quad (2)$$

where $dist$ is the L^2 distance transform [5].

To estimate the optimal model parameters, only those Δ_{dist} provide valuable information where $\Delta_{hit} = 0$ because, if that is not the case, the desired spatial accuracy cannot be achieved with the current number of control points for this configuration of α and δ. To take this into account, the overall error term for each specific combination of n, α,

and δ is defined as follows:

$$e(\alpha,\delta)_n = \begin{cases} \Delta_{dist}, & \text{if } \Delta_{hit} = 0 \\ \infty, & \text{otherwise} \end{cases} \quad (3)$$

Obviously, Δ_{dist} should be minimized to draw the desired point as accurately as possible.

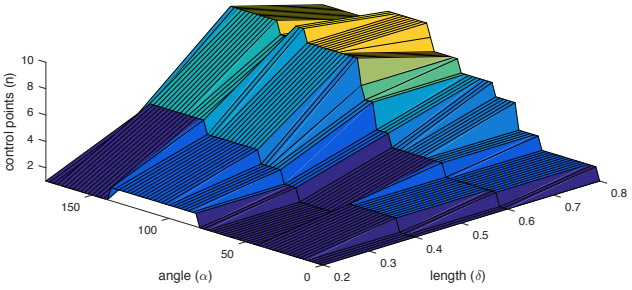


Fig. 5: Plot of the derived model parameters M. The required minimum number of control points is shown with respect to line length and angle.

Having analyzed the patterns with various control points n for all α and δ, a model database describing the laser's displaying behavior is defined, as described in the following. The optimal number of control points for α and δ is declared as $M(\alpha,\delta)$ and the corresponding error is declared as $E(\alpha,\delta)$. The model database entry of $M(\alpha,\delta)$ is initialized with zero control points $M(\alpha,\delta) = 0$, and accordingly their initial error values $E(\alpha,\delta) = e(\alpha,\delta)_0$. The number of control points $n \in [0, N]$ gets iteratively increased by one and their error values $e(\alpha,\delta)_n$ are evaluated and compared to $E(\alpha,\delta)$. This procedure is visualized in Figure 4. The optimal number of control points $M(\alpha,\delta)$ gets increased by one if there is an n for which $e(\alpha,\delta)_n < E(\alpha,\delta) * u, u \in (0,1]$. The corresponding error value $E(\alpha,\delta)$ gets accordingly updated to $E(\alpha,\delta) = e(\alpha,\delta)_n$. In our work, we set $u = 0.75$. The analysis then continues with $n = M(\alpha,\delta) + 1$.

Because both errors are sensitive to noise in the input data, two post-processing steps are applied:

1. For consecutive angles α_j, α_{j+1} at a fixed distance δ_k, the model is restricted to have fewer control points $M(\alpha_{j+1},\delta_k) < M(\alpha_j,\delta_k)$ only if the corresponding error also decreases $E(\alpha_{j+1},\delta_k) < E(\alpha_j,\delta_k)$. Otherwise, $M(\alpha_{j+1},\delta_k)$ is set to $M(\alpha_j,\delta_k)$.

2. An outlier removal procedure is applied to remove single outliers between consecutive angles of the same distance. If a whole sequence of consecutive angles has non-monotonic changes, all corresponding $M(\alpha,\delta)$ get assigned to the maximum number of control points within this sequence.

Figure 5 visualizes the result of the measured database entries of M using our laser hardware [2].

3 Optimization

Our main goal is the generation of a point sequence path such that the overall scanning time is minimized while still guaranteeing that the content is accurately displayed at the desired locations. This is achieved by a combination of:

- minimizing the overall Euclidean path length of the point sequence and
- finding a path that requires the least amount of control points.

[2]The relevant settings in LD2000 were set to: Scan rate: 35,000 samples per second. We operated in vector mode, and settings in the 'vector display' tab were all set to the minimum. Point spacing was set to 1,500 for blanked lines and to 250 for visible lines.

Furthermore, temporal sequences should be projected such that the spatio-temporal point distribution is homogeneously distributed as much as possible to evenly distribute and thus to further suppress perceived flickering. A general overview of the optimization procedure is provided in Figure 6.

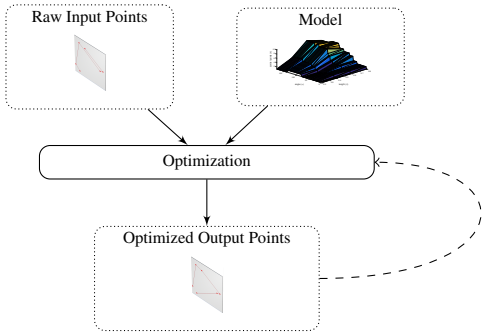


Fig. 6: Using the generated model (cf. Section 2), raw input points are optimized so that they appear at accurate positions within a minimal scanning time. If the input consists of a temporal vector graphics sequence, the optimized output points of the previous frame are used as additional input for the upcoming frame.

Finding the globally optimal scanning path for a random input point sequence is an NP-complete problem [16] and therefore cannot be solved easily within the desired reasonably short amount of time. Therefore, we decided to try to find an approximate solution that is close enough to the global optimum to achieve acceptable performance improvements. Because a first implementation using simulated annealing (SA) [8] did not lead to a satisfying result within an acceptable time frame, we chose to implement k-opt-based local search algorithms (2-opt and 3-opt) [9], which proves to compare well with more sophisticated SA methods (cf. [7]). Further improvement could be achieved using more sophisticated algorithms, such as the LK-heuristic [10]. However, as we obtained satisfactory and presumably close to optimal results in a fast computation time of several seconds per frame, we chose to keep the 2-opt implementation, especially as further path optimizations did not lead to a significant reduction in projection speed. Please refer to Section 4 for timing comparisons between the results of a 2-opt and combined 2-opt and 3-opt optimizations.

3.1 Still Frame Optimization

If the scanning path of only a single still frame containing l points $\{p_0, ..., p_{l-1}\} \in P$ should be optimized, the calculation is carried out considering: minimization of the total path length, which is equivalent to solving the traveling salesman problem [19], and optimal angularity, which means it is traversed such that the scanning path requires the least amount of control points while still rendering a precise output. To achieve this, the following error costs are minimized.

1. Distance cost, which is defined by:

$$\epsilon_\delta = \sum_{i=0}^{l-1} \Gamma(p_i) \quad (4)$$

with

$$\Gamma(p_i) = |p_{i-1}(x,y) - p_i(x,y)| \quad (5)$$

To make the distance cost independent of the resolution of the virtual image plane, the maximal possible distance over the diagonal is normalized to 1.0.

2. Angle cost, which is defined by:

$$\epsilon_\alpha = \sum_{i=0}^{l-1} \Theta(\Upsilon(p_i), \Gamma(p_i), M), \quad (6)$$

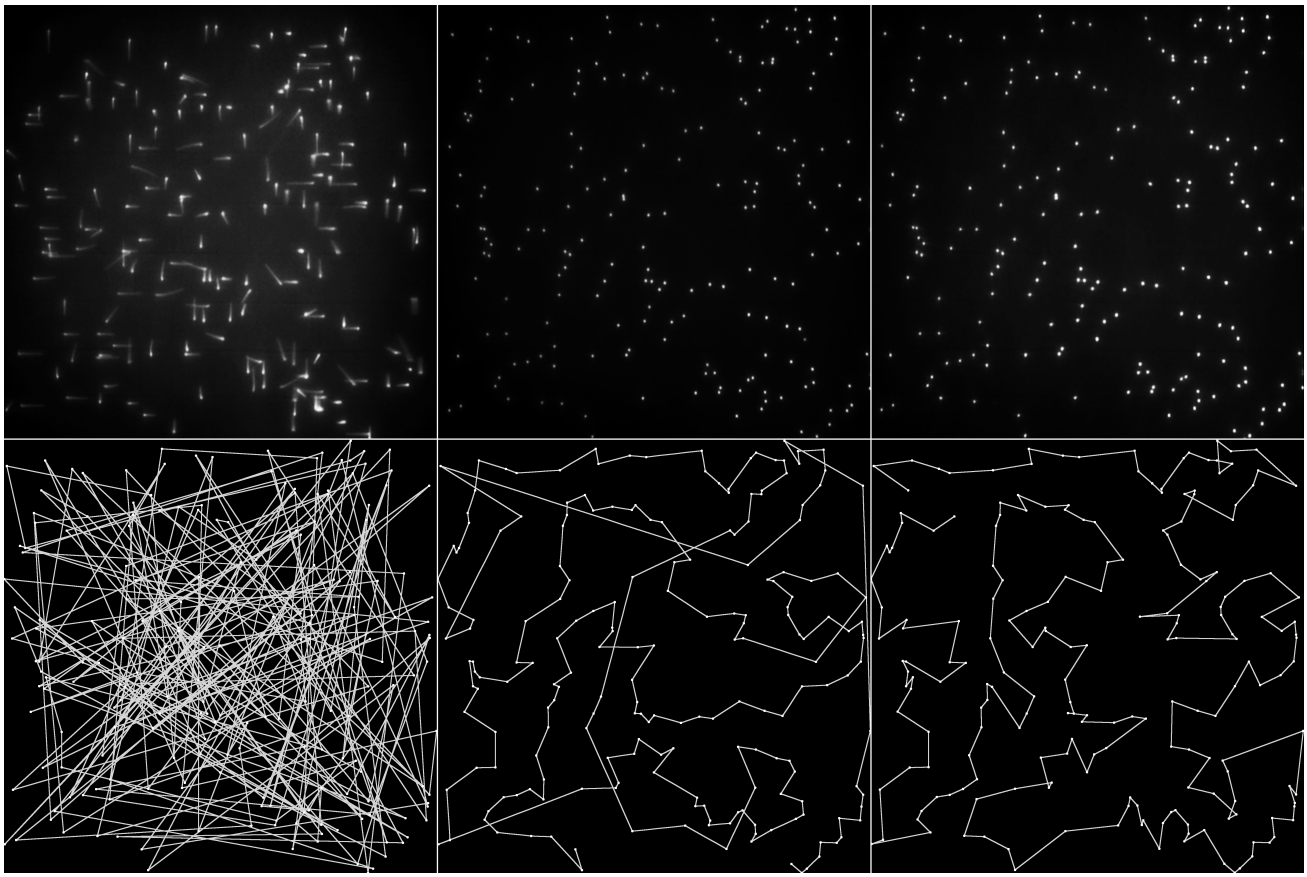


Fig. 7: Sample frame of one of the evaluation sequences. Upper row: captured projections of (left to right): random path, optimization results using LCMax, result of our proposed method. Bottom row: the actual point paths. Both optimized versions are able to draw points correctly at the desired locations. The overall intensity of the proposed method is higher compared to the LCMax solution because it requires fewer invisible control points and thus is able to spend more time to display colored points.

where $\Upsilon(p_i)$ is the angle at point p_i formed by the path $p_{i-1} \rightarrow p_i \rightarrow p_{i+1}$ and $\Theta(\Upsilon(i), \Gamma(p_i), M)$ is the look up to the nearest entry in the model database storing the required number of control points.

The total cost is given by the weighted sum of the two terms:

$$\epsilon = \omega_\delta \cdot \epsilon_\delta + \omega_\alpha \cdot \epsilon_\alpha \tag{7}$$

The optimal values for ω_δ and ω_α were estimated as described in Section 4.1.

This error is minimized using the k-opt method. We apply 2-opt on the randomized input path, as long as there are path improvements. To further improve the solution, a 3-opt is applied afterwards. The results of a sample frame are shown in Figure 7.

3.2 Temporal Sequence Optimization

If an animated point sequence of S frames should be projected instead of a single vector image, the optimization also takes care to equalize the spatio-temporal distribution of points in consecutive frames. This ensures that points that are spatially located in proximity are all drawn within approximately equal temporal offsets to ensure an equal light distribution over the whole frame, which reduces the perception of locally varying flickering. The overall temporal optimization concept is illustrated schematically in Figure 8.

Therefore, the aforementioned still frame optimization is extended to consider spatio-temporal consistency. Solution Ω_s of frame f_s at position $s \in [0, S-1]$ is initialized with a nearest neighbor (NN) approach to the previous solution Ω_{s-1}. We implemented the NN approach using a kd-tree and applying the FLANN method [13]. For every point in f_s we find the order position of the NN in Ω_{s-1} and normalize that position by the number of points l_{s-1} in Ω_{s-1} to $[0, 1]$. Please note that the number of points l in each frame can be arbitrary. Having found this relative order position (Π), the point is mapped to the corresponding absolute order position for f_s. Once we found the NN, we sorted them so that we start our paths with those points that are closest to the beginning of the optimal path in the previous solution. With this method, the spatio-temporal error cost is defined by:

$$\epsilon_\theta = \sum_{i=0}^{l_s-1} \left| \Pi_s(p_i) - \Pi_{s-1}(NN_{s-1}(p_i)) \right|. \tag{8}$$

The overall error term for the temporal optimization is then defined as an extension of Equation 7:

$$\epsilon = \omega_\delta \cdot \epsilon_\delta + \omega_\alpha \cdot \epsilon_\alpha + \omega_\theta \cdot \epsilon_\theta. \tag{9}$$

The temporal weighting factor ω_θ was experimentally set to 1, while ω_δ and ω_α were optimally set to 10 and 0.1 using a calculation described in Section 4.1. Please note that for Equations 4, 6, and 8 for the case when index i goes out of bounds, i is mapped back to the indices space in a circular manner.

4 Evaluation

As initially stated, two main goals were targeted by our optimization method. The first was the generation of an approximated spatially optimal and accurate point path traversal minimizing the overall drawing time, and the second was a reduction of perceived flickering. The first goal can be evaluated by objective measurements using a camera to measure accuracy and speed, while to evaluate the second, a user evaluation had to be carried out to assess any perceived improvement in visual image quality. Several patterns and temporal sequences were generated and used to evaluate the proposed method under varying conditions:

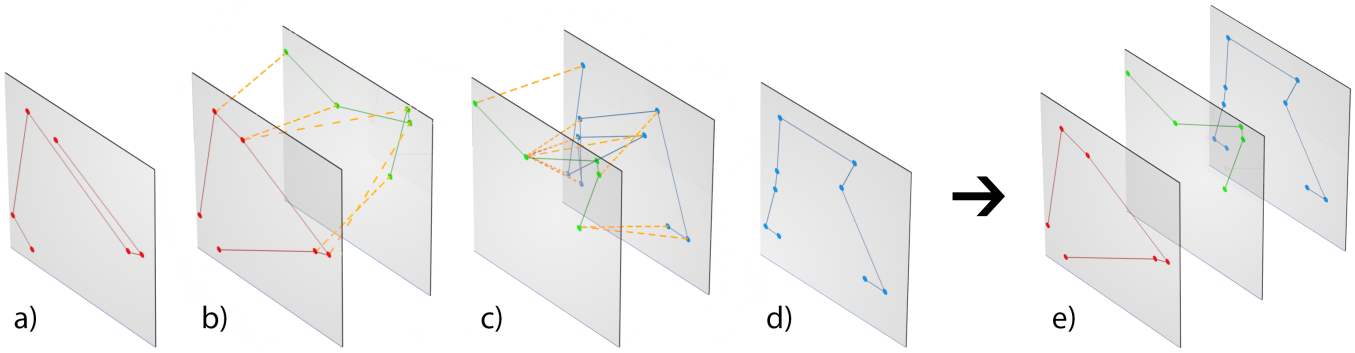


Fig. 8: Visualization of the optimization steps: (a) The random path of the first frame of the sequence (red) is optimized by distance and angularity using M (b). For the next frame (green), its points are mapped to the closest points of the optimized last frame, and an initial path is generated as described in Sec.3.2. This serves as the starting point to optimize that frame using the additional temporal error cost (c). The procedure is repeated for the consecutive (blue) frames (d) to compute the spatio-temporal paths minimizing scanning time and flickering (e).

1. **Random points (RP):** Single frames consisting of 100, 200, and 400 randomly distributed points.

2. **Star:** A temporal sequence of a rotating star consisting of 72 frames. The star contains 151 points.

3. **Random points sequence:** A temporal sequence of 100 points per frame located partially at new locations and partially at their last locations. The sequence consists of 400 frames.

4. **Varying random points:** A temporal sequence with a varying number of points located partially at new locations and partially at their last locations. The potential point counts per frame range from 0 to 300 points. As before, the sequence contains 400 frames.

As mentioned previously, we used a 2-opt optimization to calculate an approximated optimal scanning path. As we are aware that much more sophisticated algorithms exist that are able to find more optimized solutions or even the global optimum, the question remained whether it would be worth spending resources to implement better optimization methods or if the additional theoretical performance improvement might not lead to a practical speeding up of the projection. To determine whether further path optimizations significantly improve the performance of the projection system, we compared the scanning speeds of point paths that were optimized once with the 2-opt and once with a combined 2-opt and 3-opt method. The results are shown in Table 1. As can be seen, while the overall theoretical error is further reduced by $\sim$6.9%, this is not directly reflected in the actual projection speed, which is only marginally accelerated by $\sim$2.4%. Therefore, we decided to keep the 2-opt approximation because the negligible performance improvement of the projection system would not justify the additional implementation effort of more sophisticated algorithms.

In the following, we will explain how the optimal initial weights for the cost function optimization were calculated. Later, the camera-based objective evaluation focusing on an accurate measurement of scanning time per frame and exact drawing locations will be given,

Method	Avg. error	Comp. time [s]	Proj. speed [s]
unopt. input	-	0.0	3.342
2-opt	0.0618	110.8	2.436
2-opt and 3-opt	0.0578	4630.1	2.378

Table 1: This table lists the average error of a temporal subset (172 frames) of sequence #4, the times needed to compute these point sequences using two different optimization methods, and the actual projection speed. While the computation time definitely can be reduced further using more sophisticated optimization algorithms, the computed error reduction of the more sophisticated 2-opt and 3-opt method ($\sim$ 6.9%) is not directly reflected in the actual projection speedup ($\sim$ 2.4%) when compared to the simpler 2-opt method.

and finally the results of the user evaluation investigating whether perceived temporal flicker can be subjectively reduced will be presented.

4.1 Weights estimation

To estimate the optimal weighting factors for the error terms ϵ_α, ϵ_δ, and ϵ_θ a grid-based search was applied. Pattern (#1) with 200 points was optimized with variations of ω_α and ω_δ ranging from 0.001 to 100. The results were displayed by the laser projector, and captured by the camera, and the overall drawing speed was measured as described in the following section. The fastest scan rate was achieved for $\omega_\delta = 10$ and $\omega_\alpha = 0.1$, and these settings were used for all further processing. As the temporal weighting ω_θ has only minimal impact on the overall scanning speed, it was set to $\omega_\theta = 1$ in which the perceived flickering for the optimized sequence (#2) appeared to be subjectively minimized well based on several human observations.

Method	100 RP	200 RP	400 RP
Raw	23.3	50.9	98.2
LCMax	45.5	85.9	163.5
Our	16.76	30.7	59.5

Table 2: Scanning speeds in $[ms]$ for 100, 200, and 400 random points.

4.2 Still Frames

The quality of the laser projection is influenced by how spatially accurate the frame is drawn and how much flickering is encountered. Given these two criteria, we compared the scanning time and location correctness of un-optimized and optimized frames using the LCMax optimization software[3].

In order to evaluate the proper locations of the points of pattern (#1), the projected frames were captured, warped onto the projector's virtual image plane, and binarized as described in Sec. 2.3. These binarized images T were then compared to the rasterized ground truth data R. Extending Δ_{hit} described in Equation 1 over all points $r \in R$ gives the overall location error:

$$\Delta_{loc} = \frac{\sum_r \Delta_{hit}(r)}{l}, \tag{10}$$

where r represents all rasterized points and l represents the total number of $r \in R$. Both optimization methods, LCMax and our proposed one, are able to accurately draw all points of all tested samples at the correct locations, while the raw frames missed the desired locations of more than 90% of the sample points (cf. Figure 7 for a visual comparison of the projected points). Since our as well as LCMax optimization is able to draw all points at correct locations, the captured images

[3]The LCMax optimization was tuned by an expert to achieve optimum results.

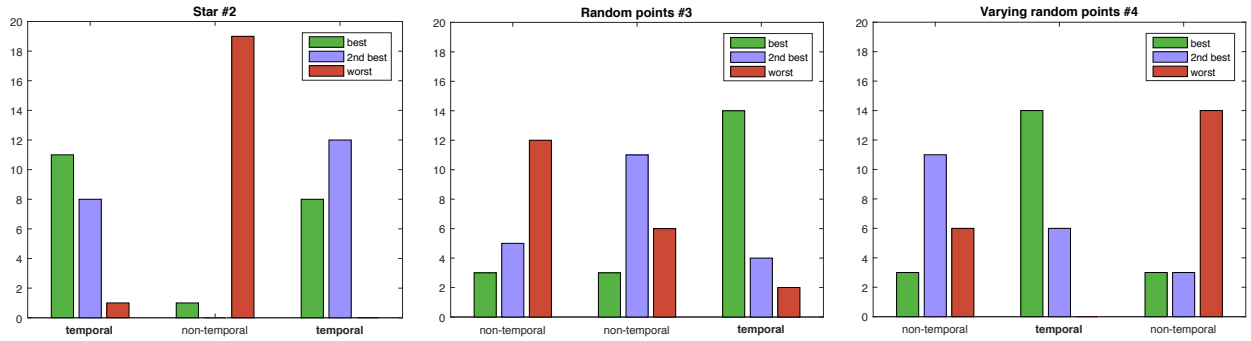


Fig. 9: Results of the user evaluation comparing temporal vs non-temporal optimized sequences. In all sequences the vast majority of test subjects voted the non-temporal optimized projections as worst (red bars) while the visual quality of the temporal optimized sequences was preferred by almost all test subjects (green bars).

Method	Star	Const # RP	Varying # RP
Raw	2.848	9.768	7.56
LCMax	4.052	14.792	11.3
Our (nt)	1.588	6.608	5.296
Our (t)	1.592	6.716	5.364

Table 3: Drawing times [s] for all tested sequences: Our proposed method outperforms the raw input and the optimized LCMax data. The temporally (t) optimized data is slightly slower than the non-temporal (nt) optimized data due to its additional optimization constraints. A visual comparison of the star sequence is given in the supplementary video.

shown in the upper center and right look very similar. However, they have different intensities, since both methods use different strategies to achieve the desired spatial accuracy: While the LCMax version we used adds a constant number of invisible control points for each visible point, our method adaptively inserts the minimal number of required control points. This leads to a reduced overall number of necessary control points. Thus, with our method more time is spent projecting visible points instead of invisible control points even if both methods trace the same path.

To measure the drawing time of a specific frame, the projection is continuously captured by the machine vision camera while interactively adjusting the exposure time until no flickering is visible in the captured video stream. We noticed two different kinds of flickering: 1) some points disappear irregularly and 2) some points appear brighter than others. In the first case, the exposure time was too short to capture a whole scanning pass. In the second case, some of the points were projected twice. By interactively adjusting the exposure time in between these two situations until no flickering occurs, the current exposure time corresponds to the scanning time of that frame.

The results are shown in Table 2. Our proposed method is able to display the content faster than the un-optimized, spatially inaccurate, raw data due to its optimized scanning path. Compared to the accurately drawn LCMax frames, our algorithm is able to reduce the drawing time by almost up to a factor of 3.

4.3 Temporal Sequences

To evaluate temporal sequences, animations were created in which the frames of patterns #2—4 were all displayed exactly once, that is as fast as could be supported by the device, and were recorded by a regular video camera. All animations were played 10 times in a loop, and the measured time was divided by 10 to reduce measurement inaccuracies due to the limited video frame rate of $25Hz$. The overall duration was measured by manually determining the start and end frames of the sequence in the video. As shown in Table 3, our method outperforms not only the slower but spatially accurate LCMax data but also the un-optimized raw input data that contains no additional control points at all. Note that the scanning time is slightly reduced for the non-temporal sequences. However, the perceived visual quality degrades compared to the temporal optimization: Since the displayed points do not have any afterglow, i.e. each point is only displayed for a few microseconds, they are only perceived as static if they are projected at a frequency far above the critical flicker frequency [18]. To minimize that disturbing flickering, we chose to equalize the temporal offset of the spatially distributed points by considering spatio-temporal relationships. This leads to a flickering - although unavoidable - having an almost uniform frequency all over the projection area. If this is not carried out, a spatially varying flickering is observed whose intensity distribution changes over time. As shown in the user evaluation, the temporal component improved the visual quality further by minimizing the flickering impression although the overall scanning speed was slightly reduced.

4.4 User Evaluation

We employed an informal user evaluation by asking 20 people to compare the perceived visual quality of the projected sequences #2—4. The participants were shown three projections of each sequence and were asked to rank them relatively. Within the three projections, only two different sequences were shown, temporally (t) and non-temporally (nt) optimized sequences. For sequence #2, the order was [t, nt, t], and 19/20 participants rated the temporally optimized sequence as the best one (95%). For sequences #3 [nt, nt, t] and #4 [nt, t, nt], 14/20 participants rated the temporally optimized sequence as the best sequence (70%). However, it should be mentioned that for sequence #3, four of the six participants who did not rank the t as best ranked it as second best, and for sequence #4 all of these six people ranked the temporal sequence as second best, which shows a clear preference for temporally optimized sequences. A visualization can be found in Figure 9.

Although this initial informal evaluation lacks a thorough statistical analysis, we think it nicely shows the preference for our temporally optimized projection.

5 Conclusions

We presented a method to enable an accurate point-based projection by spatio-temporally optimizing the tracing path of a galvanoscopic scanning laser projector to increase the scanning rate and to reduce perceived flickering. Using a camera to estimate the required model parameters enables a fully automatic, robust calibration process that allows adaptation to the actual projector settings and ensures accuracy and repeatability. The evaluation showed that our method significantly improves scanning speed for completely unoptimized input and also increases speed compared to commercially available optimization methods. The result of the user evaluation clearly shows the effectiveness of our proposed spatio-temporal optimization to further reduce the perception of disturbing flickering compared to a pure spatial/angular optimization.

The physical limitations of such a projector, in particular the maximum rotation speed of the mirrors, however, still remain the limiting factor that cannot be changed without developing new projection hardware systems. This constrains the application of such devices to local

high dynamic range augmentations. Other emerging projection technologies, such as those recently presented in [3], might help push the boundaries of high dynamic range projections further.

While a pure geometrically optimized point path can be achieved with our method, other aspects, such as generating a calibrated color output of the projector, extending the method to consider line segments, and a comprehensive user evaluation, are part of our future research goals.

REFERENCES

[1] O. Bimber and R. Raskar. *Spatial augmented reality - merging real and virtual worlds.* A K Peters, 2005.

[2] D. Bradley and G. Roth. Adaptive thresholding using the integral image. *Journal of Graphics, GPU, and Game Tools*, 12(2):13–21, 2007.

[3] G. Damberg, A. Ballestad, E. Kozak, J. Minor, R. Kumaran, and J. Gregson. High brightness hdr projection using dynamic phase modulation. *ACM SIGGRAPH 2015 Emerging Technologies*, page 13:1, 2015.

[4] G. Donato and S. Belongie. Approximate thin plate spline mappings. *Proc. of the 7th Europ. Conference on Computer Vision*, 2002.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8(19):415–428, 2012.

[6] O. Halabi and N. Chiba. Efficient vector-oriented graphic drawing method for laser-scanned display. *Displays*, 30(3):97 – 106, 2009.

[7] D. S. Johnson and L. A. McGeoch. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, 1:215–310, 1997.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[9] S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44(10):2245–2269, 1965.

[10] S. Lin and B. W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 21(2):pp. 498–516, 1973.

[11] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318.

[12] A. Manakov, H.-P. Seidel, and I. Ihrke. A mathematical model and calibration procedure for galvanometric laser scanning systems. *16th International Workshop on Vision, Modeling and Visualization*, pages 207–214, 2011.

[13] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. *International Conference on Computer Vision Theory and Application*, pages 331–340, 2009.

[14] Pangolin. Pangolin laser systems lasershow converter max. `http://www.pangolin.com/LD2000/lc-max_overview.htm`.

[15] Pangolin. Pangolin laser systems ld 2000. `http://www.pangolin.com/LD2000/LD2000_sdk.htm`.

[16] C. H. Papadimitriou. The euclidean travelling salesman problem is np-complete. *Theoretical Computer Science*, 4(3):237 – 244, 1977.

[17] A. Purkhet, O. Halabi, T. Fujimoto, and N. Chiba. Accurate and efficient drawing method for laser projection. *The Journal of the Society for Art and Science*, 7(4):155–169, 2008.

[18] H. Shankar and K. Pesudovs. Critical flicker fusion test of potential vision. *Journal of Cataract and Refractive Surgery*, 33(2):232 – 239, 2007.

[19] D. B. Shmoys, A. H. G. Rinnooy Kan, J. K. Lenstra, and E. L. Lawler. *The Traveling salesman problem: a guided tour of combinatorial optimization*. Wiley interscience series in discrete mathematics and optimization, 1985. J. Wiley, 1987.

[20] D. Tran and J. Yuan. Optimal spatio-temporal path discovery for video event detection. In *CVPR*, pages 3321–3328. IEEE Computer Society, 2011.

[21] D. Tran, J. Yuan, and D. Forsyth. Video event detection: From subvolume localization to spatiotemporal path search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):404–416, Feb 2014.