

Adaptive Colour Classification for Structured Light Systems

Philipp Fechteler and Peter Eisert

Fraunhofer Institute for Telecommunications - Heinrich-Hertz-Institute
Image Processing Department,
Einsteinufer 37, D-10587 Berlin, Germany
{philipp.fechteler / peter.eisert}@hhi.fraunhofer.de

Abstract

The authors present an adaptive colour classification method as well as specialised low-level image processing algorithms. With this approach the authors achieve high-quality 3D reconstructions with a single-shot structured light system without the need of dark laboratory environments. The main focus of the presented work lies in the enhancement of the robustness with respect to environment illumination, colour cross-talk, reflectance characteristics of the scanned face etc. For this purpose the colour classification is made adaptive to the characteristics of the captured image to compensate for such distortions. Further improvements are concerned with enhancing the quality of the resulting 3D models. Therefore the authors replace the typical general-purpose image preprocessing with specialised low-level algorithms performing on raw photo sensor data. The presented system is suitable for generating high-speed scans of moving objects because it relies only on one captured image. Furthermore, due to the adaptive nature of the used colour classifier, it generates high-quality 3D models even under perturbing light conditions.

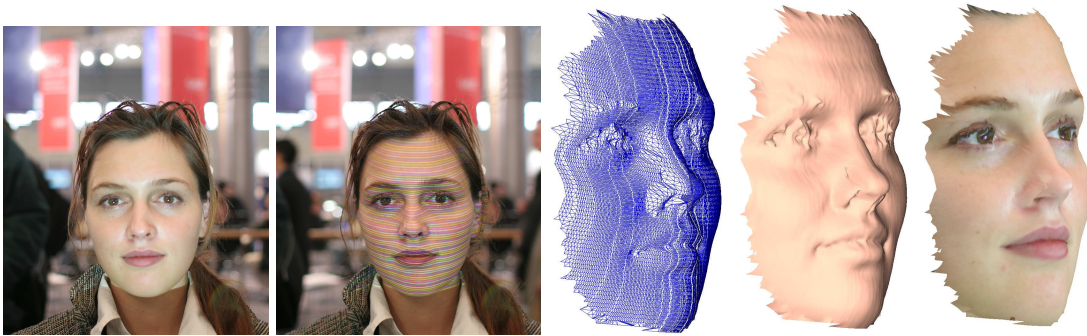


Figure 1 *Input images and resulting 3D models as wire frame model, surface and textured surface*

1 Introduction and related work

The reconstruction of 3D range data from 2D views is an longstanding goal in Computer Vision. Several Stereo Vision techniques have emerged in the last decades, see [23] for a survey. A related technique to tackle the 3D reconstruction challenge is the structured light method, also called Active Stereo Vision,

where controlled illumination is used to actively generate geometric correspondence between projector view(s) and camera view(s) as basic step for 3D reconstruction [24].

Among the earliest work in the field of one-shot 3D scanning using coloured structured light is [6]. The authors present a complete framework to measure the depth of arbitrary scenes. Matching of detected to projected stripes is performed with a "crystal growing" analogy algorithm. Starting with single matches, sequences of successive matches are computed until all consistent matches are generated followed by a heuristic selection of the most probable sequence. In [18], coloured stripes are projected onto faces. The input image is simply segmented by thresholding in HSV colour space. A graph-based matching is performed analogous to string matching followed by triangulation to reconstruct the 3D coordinates. Though pioneering work, both methods use moderate resolutions and explicitly exclude any robustness issues with respect to textured objects and ambient illumination.

Today, various structured light systems for 3D reconstruction have been developed with quite different characteristics: in [32] a real-time system is proposed which runs on specialised hardware; in [24] a method is presented for generating high-resolution depth maps of complex scenes by using multiple projectors, cameras and several snapshots per camera; in [31] the authors present an approach to capture high-resolution 3D models of faces utilising several synchronised video cameras; in [19] the formulae for scattering media (like milky water) is modelled as well as the algorithm for estimating its parameters and the way how to compensate these scattering effects are shown; in [29] a method is shown which uses just one projector and one camera running on a typical PC. This last mentioned work has motivated us to develop a structured light 3D scanner specialised for faces, which poses the foundation of the presented system [10, 9].

In recent research, significant effort has been made to enhance the performance of such systems with respect to the resulting 3D models. Improvements of structured light systems can be grouped broadly into:

- extending the method, for example adding stereo analyse techniques by using several cameras [24], adding video analysis techniques (spacetime analysis) [30], adding photometric techniques [27];
- using more refined and sophisticated algorithms to match the captured image with the projected pattern, for example GraphCut [14]; and
- enhancing the core methods, as discussed below.

In order to use structured light 3D reconstruction for non-rigid objects, like human faces, the standard approach of projecting several successively refining patterns is not possible. All pattern information has to be integrated into as few patterns as possible to minimise the influence of temporal changes. A major challenge here is the search for correspondences between the projected pattern and the captured image. The choice of patterns used to project obviously has a strong influence. Different approaches for designing good patterns have been proposed [22]. For example, in [1, 2] the authors suggest a 2D M-array pattern with a Hamming distance for fault tolerance. This means, a matrix filled up with symbols is generated so that every sub-matrix is uniquely identifiable even in case of incorrectly recovered symbols. Since the symbols used as primitives for 3D reconstruction consist of several pixels, the resolution of such an approach is limited. In the early work of [7] a method is presented to measure the scenes albedo and influence of the environment illumination in order to (a) correct the measurements accordingly and (b) generate optimised colour projection patterns. The parameters for a chromaticity model of camera and projector are measured in a prerequisite calibration. Two reference pictures are taken before the actual range imaging: one with only the environment illumination and one projected with uniform white light. Out of the two reference pictures a pixel-wise reflectance map is set up and noise thresholds are calculated. Based on the measurements a series of colour projection patterns is generated which are optimised for unique identification. With this approach the constraint of colour

neutral scenes is greatly relaxed, but at the expense of performing a calibration process and the need to capture several images. In [15] a profilometry based method is proposed for online adaptation of the projected pattern to overcome over- and under-exposure as well as aliasing effects. The authors have come up with a model for the complete signal path from projected RGB pattern to captured RGB image, considering several characteristics of camera, projector, scene albedo as well as environment illumination. With this approach high-quality 3D range data are captured; however at the expense of needing to gauge the camera and projector. Additionally, as reported, up to 16 measurements have to be taken in order to adapt the pattern and reconstruct the scene depth. So performance will degrade with non-rigid objects performing complex movements even though simple motion compensation takes place. As reported in [5] the constraint of physical rigidity during acquisition is often not easy to achieve - even for rigid objects - because of environment characteristics. Therefore, capturing a sequence of photos is often not possible or leads to disturbed results.

A different way to enhance the detection of the projected pattern is to improve directly the classification method used therefore. In [26] a simple method is presented to distinguish between projected, not-projected and uncertain pixels in a binary monochromatic structured light system. As their work relies on the separation of direct and indirect illumination as it is presented in [20], several images have to be captured in order to achieve high-quality results.

Here we present a method to enhance the search for correspondences by adapting the classification of detected pattern colours to the captured statistical characteristics. With this we achieve an improved single-shot structured light 3D scanner with respect to robustness to ambient light and reflectance characteristics of the object to be scanned. Additionally, we present low-level image processing algorithms suited for the generation of high-accuracy 3D models.

2 Framework and Architecture

A 3D model of a face is computed by first projecting a simple coloured stripe pattern onto the face. The depth information is calculated by taking into account the distortion of the stripes in the face caused by its shape. To measure the degree of distortion, correspondences between projected and detected stripes are established. The depth is evaluated for all correspondences with respect to the intrinsic and extrinsic parameters of camera and projector. The resulting cloud of 3D points is converted into a triangle mesh. This mesh constitutes the surface of the 3D model. Optionally, the mesh can be textured with an additional picture taken under regular white light.

The hardware used by our framework consists of off the shelf devices: a digital camera and a projector (see Fig. 2). Both devices are controlled by a PC running the framework. The devices are

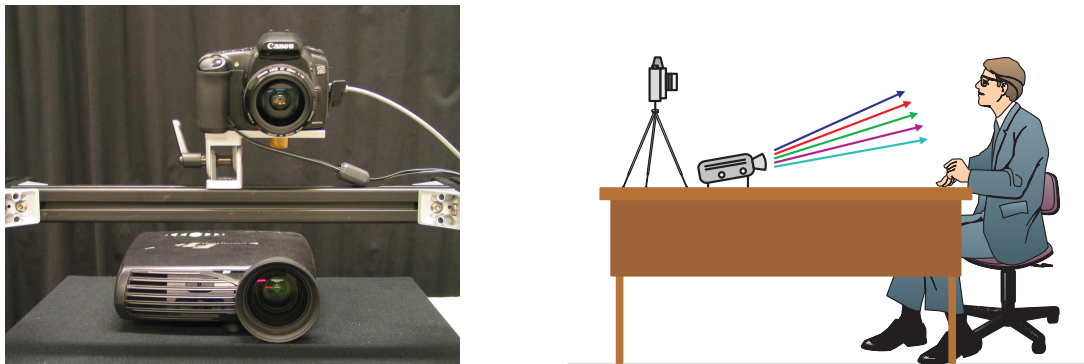


Figure 2 *Devices and setting used in this framework*

mounted so that their image centres are one upon the other.

In order to generate a 3D model of a face the following steps are performed:

1. Take image I_{input} of the face illuminated with a colour stripe pattern $I_{pattern}$ and optionally capture an image $I_{regular}$ with regular white light.
2. Extract the prospective stripes in I_{input} .
3. Estimate the colours corresponding to the prospective stripes.
4. Match the prospective stripes with the projected ones.
5. Calculate 3D coordinates of correspondences.
6. Create a triangle mesh from the 3D point cloud.
7. Optionally project $I_{regular}$ onto the surface as texture.

To create a 3D model of a face, the goal is to find the most probable correspondences between $I_{pattern}$ and I_{input} among all possibilities. This is achieved by performing a global optimisation after having extracted hypothetical stripes in I_{input} as well as their colours.

2.1 Pattern Characteristics

The pattern projected onto faces should allow an easy assignment of imaged parts to parts of the pattern. Since it is a design goal to facilitate single-shot 3D reconstruction, time-varying patterns like sinusoidal or hierarchical binary patterns are out of scope. As it was analysed in [13], for projected patterns the detectability is increased and the ambiguity is reduced when using certain high-contrast colours. Therefore a stripe pattern has been chosen with horizontal lines of fully saturated colours with empty (black) spaces in between. This reduces the search for correspondences to a 1D search along the corresponding scan columns. In order to achieve high-quality 3D reconstructions the pattern is designed in this way to allow a peak-based matching. An alternative would be to design the pattern for an edge-based 3D reconstruction where the colour transitions are matched, which generally provides a denser reconstruction result, see [29] for example. A further increase in 3D model resolution is achievable by combining both approaches to a hybrid pattern as it is presented in [21]. However, the increased resolution of edge-based as well as hybrid-based 3D reconstructions is always measurable in reduced accuracy, for example the appearance of fringe artefacts.

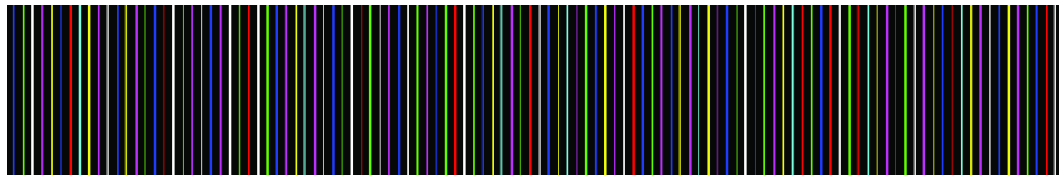


Figure 3 A cut-out of the used pattern rotated by 90°

Note in Fig. 2 that the camera and projector centres are mounted on top of each other to provide the Epipolar constraint without any image rectification. The colours in the resulting pattern image $I_{pattern}$ are (see Fig. 3): red, green, blue, white, cyan, magenta and yellow. Inspired by de Bruijn sequences [11] and to ease the unique assignment of detected stripes to projected ones we have chosen a series of stripe colours with a large period. Besides that, we introduced the constraint that two consecutive stripes have to differ in at least two colour channels. With this latter constraint we achieve an enhanced delimitation

of successive stripes, and the unique identification is simplified. The smaller periodicity because of the additional constraint in our case is no problem, as long as the period of the pattern is smaller than the largest jump in depth. Taking this into account the pattern $I_{pattern}$ can be determined with a simple depth-first search.

2.2 Image Preprocessing for Simplified Stripe Detection

After capturing an image I_{input} of a face illuminated with the pattern $I_{pattern}$ the image is reduced to the manually selected interesting region of the face. All pixels outside are set to black, so that they will be ignored by subsequent steps. In order to reduce the noise level and to enhance the visibility of the stripes the input image is preprocessed with two consecutive filters:

- with a directional 2D hourglass shaped low-pass filter [16] a smoothing as well as enhancement of horizontal stripes is achieved,
- with a 1D band-pass filter the colour of the stripe's vertical neighbourhood is removed to increase the visibility.

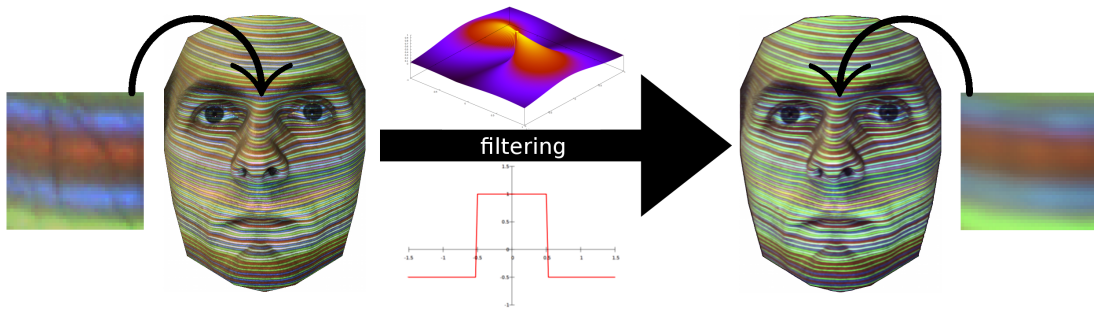


Figure 4 *Demonstration of filter*

The performance of the filter is illustrated in Fig. 4. The resulting preprocessed image $I_{preproc}$ of the face will be searched for the projected stripes.

3 Detection of Stripes

After preprocessing, the stripes corresponding to the projected pattern are located in $I_{preproc}$.

3.1 Sub-Pixel Stripe Localisation

To achieve a highly accurate 3D model the stripes are detected with sub-pixel resolution.

Therefore all the "general-purpose" image preprocessing (Bayer interpolation, gamma correction, white balancing etc.) in the camera is skipped, which is generally used to generate perceptually appealing images. Instead, the presented framework uses the plain photo sensor values with full 12-bit resolution instead of the typical 8 bits. Owing to the Bayer pattern we treat the pixel values differently depending on which sensor type they were measured on, red, green or blue. As a result, we have to deal with columns of single-channel pixels with alternating colour sensitivity: RGRGRG... and GBG-BGB...

Extracting the prospective stripes is done separately for all three colour channels as well as for every scan column. A pixel is taken as a stripe candidate if the values of the preceding and succeeding

pixels are not bigger. This results in three sets of stripe candidates, one for each sensor colour type. As we are interested in sub-pixel stripe localisations, we determine the centres of the stripe candidates by fitting parabolas through their intensities: $p(x) = ax^2 + bx + c$, with a, b, c being the parameters of the parabola $p(\cdot)$ and x the pixel location along the scan column, again for all three colour channels separately. The centre of the stripes is assumed to be at the maximum point of the parabola. Fitting the parabolas is performed via squared distance minimisation: $\min_{\mathbf{p}}(\mathbf{X} \cdot \mathbf{p} - \mathbf{y})^2$, with the parameter vector $\mathbf{p} = (a, b, c)^T$, the matrix \mathbf{X} holding the different powers of the pixels locations along the scan column, and \mathbf{y} containing the actual pixel values.

In the regular case with one pixel value bigger than its two vertical neighbours, the parabola is fitted through these three points. In cases where two adjoining pixels hold the same value bigger than the two surrounding ones, these four pixels are used for this. If there are more than two equal valued pixels, the inner most pixels are ignored. In this case the parabola is fitted with respect to the two starting and the two closing pixels of that interval. The latter case occurs when the sensor is saturated.

Most projected stripes produce responses in more than one kind of colour sensor, for example magenta light should excite the sensors for red and blue. And even if a sensor gets illuminated with light it is not sensitive to, a certain response is measurable. This is known as colour-cross talk. That is why one projected stripe often results in multiple detections which are found in the sets corresponding to the different sensor types.

In order to get one common set containing all stripes found in the preprocessed input image $I_{preproc}$ the three sets are fused to a single one. Thereby, detected stripes of different sets (emerging from different sensor types) which belong to one projected stripe are combined to one common representation. The common centre c_{common} is calculated as a weighted sum of the two original centres c_1 and c_2

$$c_{common} = \frac{c_1 \cdot p_{1,valid} + c_2 \cdot p_{2,valid}}{p_{1,valid} + p_{2,valid}}, \quad (1)$$

weighted with their likelihood of being correctly recovered (discussed in the next subsection). Here again a scan column is processed one after another, but now under consideration of the two different colour channels present. Every two parabolas in a single scan column from the two different sensor types are fused together if their centres are not too far apart, see Fig. 5.

For establishing correspondences, the colours are compared between the projected stripes and the detected ones. Therefore a colour is assigned to every prospective stripe using sub-pixel Bayer interpolation. This means, that stripe candidates get their red, green and blue values by assigning them as a weighed mean of their neighbouring sensor values. The weights used here are proportional to the inverse Euclidean distance between the stripes' centres and the pixel locations.

3.2 Probability of Stripes

In the following global optimisation, detected stripes that deviate from the pattern sequence order too much and which are not bold enough are cancelled out. For this purpose every detected prospective stripe is assigned a likelihood of being a correctly recovered projected stripe. Since most correctly recovered stripes are bright and crisp in their appearance this likelihood is derived from the luminance as well as the "sharpness" parameter a of the fitted parabola as mentioned above.

All the parabolas are opened to the lower side, so $a < 0$. Also, parabolas with a low absolute value of a close to zero are flatter than the ones with more negative values of a . To normalise the derived likelihoods $p_{i,valid}$ of a stripe candidate \mathbf{p}_i to be a valid stripe to a range of $[0, 1]$ the sharpness parameter a_i is divided by the lowest negative value of all a 's among all parabolas of the same sensor type. Together with the luminance $l_i \in [0, 1]$ of stripe candidate \mathbf{p}_i we obtain

$$p_{i,valid} = \frac{a_i}{2a_{min}} + \frac{l_i}{2} \quad (2)$$

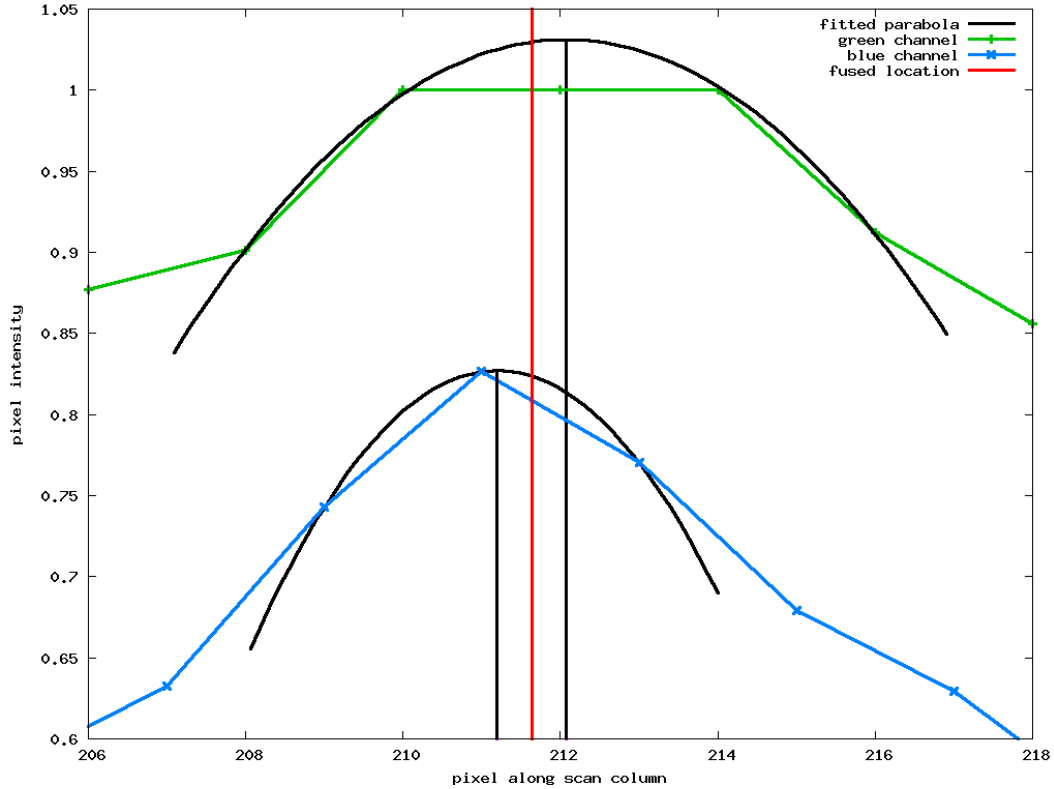


Figure 5 Parabolas fitted through maximum intensity in green respective blue channel and its resulting common location

This means, that indistinct stripes will correspond to flatter fitted parabolas which in turn results in lower probability weights. Additionally, stripe candidates with low luminance values are treated as well as less probable.

4 Colour Classification of Detected Stripes

The result of the previous steps is a set of all stripe candidates. Each stripe candidate is specified by a scan column index, a position along that scan column and a RGB colour value. For every detected stripe the likelihoods of being projected with the different colours of the pattern are derived. Hence every stripes pixel has assigned one probability value for each projected colour, seven in our case.

4.1 Classifying the Detected Colours

Experiments have shown that projected colours, reflected by skin and recorded by cameras encounter various distortions. Additionally, sensor noise as well as colour-cross talk is detected between the projector spectra and the sensor filters. In Fig. 6 two RGB space representations of pixels corresponding to prospective stripes are depicted: one captured under controlled lighting conditions and the other one in a usual office environment. Colour clusters are roughly identifiable corresponding to the projected colours, without a clear separation between them. The visible clusters are approximately shaped along straight lines which seem to be slightly displaced versions of the black→red, black→cyan etc. axes.

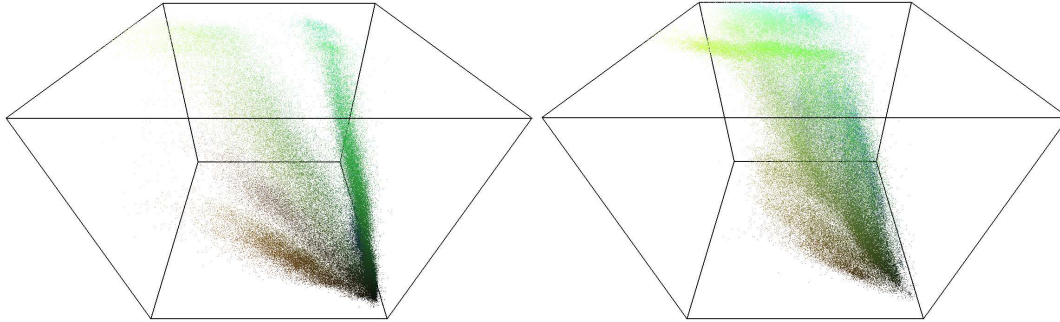


Figure 6 Pixel colours of prospected stripes in RGB space, **left:** picture captured with ideal conditions, **right:** picture captured with strong environment illumination

The plots in Fig. 6 show how crucial the light conditions of the environment are. Without any disturbing light sources in the environment, the clusters are identifiable quite clearly. But with increasing ambient light the clusters become more and more fuzzy until there is only one big blob of data points in RGB space.

4.2 Adaptation to Image Statistics

In order to determine the individual stripe colours, straight 3D lines $g_c : \mathbf{o}_c + x\mathbf{r}_c$ are fitted through the clusters in RGB space to form prototypes of these clusters; one line for each pattern colour $c \in \{r, g, b, c, y, m, w\}$. The classification of data point \mathbf{p}_i (a stripes pixel's colour) is then performed by calculating the distances $d(\mathbf{p}_i, g_c)$ of that pixel \mathbf{p}_i to all the prototype lines g_c , and assigning the colour of the prototype with the smallest distance.

This fitting of straight 3D lines through clusters is a form of orthogonal distance regression (ODR), and the classification of the projected colours is a form of model selection. The parameters for this mixture model (the straight 3D lines g_c) are determined out of the measured data. For this purpose the KMeans algorithm [17] is adapted. The classical KMeans method works broadly in the following way:

1. initialise the parameters of the classifier, the mixture of straight 3D lines
2. repeat until no significant changes in labelling are observable
 - **label** the data with current classifier parameters
 - **adapt** classifier parameters

A more detailed discussion on KMeans and clustering in general (among others) can be found in the text book [4].

There are efficient general-purpose initialisation methods for the standard KMeans method, for example [3]. These methods are not applicable for our adapted KMeansLineFit method because the cluster means are not in the same space as the data, but in parameter space of straight 3D lines. However, by knowing the originally projected colours, our initial guess of classifier parameters are straight lines, originating from black (0, 0, 0) and pointing to the fully saturated colours red (1, 0, 0), magenta (1, 0, 1) etc.

The labelling step means to assign every stripe pixel the colour label it was most probably projected with according to the current classifier. This is the colour of the prototype line g_c with the smallest distance to the stripes colour \mathbf{p}_i .

The adaptation step is slightly more complex: every prototype line g_c is moved into the centre of the data points which are currently labelled with the corresponding colour. At a first glance this can be done for every cluster independently. However, experiments have shown that the resulting lines do not lie in the cluster centres, because all the clusters are fused together at dark colours. The prototype lines do not pass from dark colours near black to lighter colours near the fully saturated ones and they do not cross the clusters in their centre. To overcome this, an artificial constraint has been introduced, that all the prototype lines must contain one common point near black, which is also adapted by the KMeansLineFit algorithm. This seems to be plausible as it is the pixels' value achieved when dimming any colour more and more.

The problem of finding all prototype lines passing through one common point poses a system of coupled equations which are not solvable in closed form. Therefore it is solved approximately by first calculating the direction vector \mathbf{r}_c for every colour c , and then determining the common offset point \mathbf{o} which fits best all the new prototypes g_c to the corresponding data. The former calculation of \mathbf{r}_c can be performed by using eigenvalue decomposition of the datas' covariance matrix for each cluster separately. The latter problem can be solved by minimising the sum of squared distances for each given data point \mathbf{p}_i and its corresponding prototype line g_c

$$\min D = \sum_{c=1}^C \sum_{i=1}^{N_c} d^2(\mathbf{p}_{ci}, g_c) = \sum_{c=1}^C \sum_{i=1}^{N_c} \|\mathbf{r}_c \times (\mathbf{p}_{ci} - \mathbf{o})\|^2 \quad (3)$$

with $C = 7$ the number of different colours in the projected pattern and N_c being the number of currently labelled pixels belonging to colour c . The offset \mathbf{o} minimising this squared distance D can be calculated by deriving D with respect to \mathbf{o} 's components and setting them to zero. The resulting three equations, resolved for \mathbf{o} 's components can be combined into an equation system of the form $\mathbf{A} \cdot \mathbf{o} = \mathbf{b}$ which can be solved with stable matrix inversion.

4.3 Robust Clustering Sensitive to Shape and Scale

Since each pattern colour is projected in nearly equal proportions the desired clusters should all contain an broadly equal number of stripes pixels. For enhanced robustness this is enforced by scaling the measured distances of stripes pixels to the prototype lines accordingly before the actual label assignment happens. Therefore an in-advance-labelling takes place as mentioned above. In a second labelling iteration, each measured distance of a stripes pixel \mathbf{p} is scaled by the proportion of desired size, which is the mean m of all clusters, and the number of stripes assigned to the current prototype line g_c in the in-advance-labelling:

$$d_{scaled}(\mathbf{p}, g_c) = \frac{m}{size(g_c)} d(\mathbf{p}, g_c) \quad (4)$$

This distance scaling, which reduces the variance of size of all the clusters, follows the label assignments according to the scaled distances.

Further robustness is gained by using a weighted covariance matrix for the eigenvalue decomposition to fit the prototype lines. The sensitivity to outliers is reduced by weighting each data point with the inverse distance for the covariance matrix calculation.

The accomplished accuracy is refined with adaptation to the shape of the clusters by means of considering correlations among the red, green and blue values of the stripes pixels. This is achieved by using the Mahalanobis distance measure instead of the Euclidean for the distance calculation of stripes pixels to prototype lines. This implies that the vectors connecting the prototype lines with the stripes pixels are considered. The means and covariance matrices needed to calculate the Mahalanobis distance are calculated out of the connecting vectors of the previous iteration.

After having iterated over the labelling-adaptation loop until no significant changes in labelling are noticed, the classifier is adapted to the statistical characteristics of the input image. For every stripe

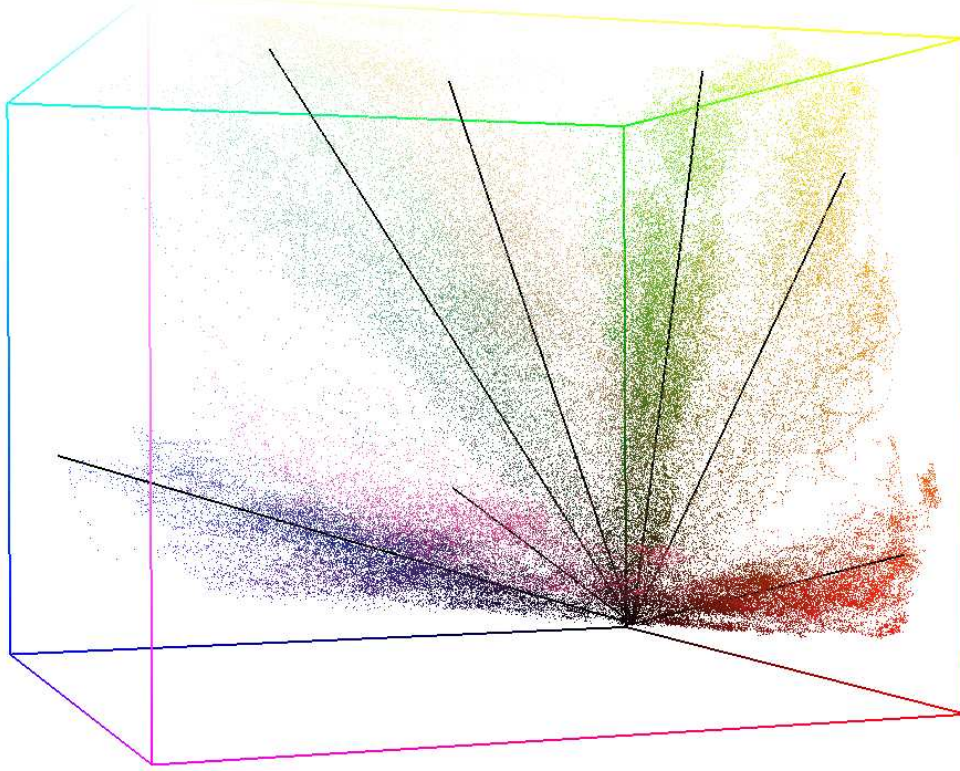


Figure 7 *Prototype lines found by adapted KMeansLineFit in RGB space*

the distance to the different prototype lines is defined. Fig. 7 shows the results of an example. In all experiments only negligible changes occurred after the tenth iteration.

4.4 Probability of Colour Assignments

The distance measurements are transformed into likelihoods by utilising a softmax-like function. The distance measurements are inverted and normalised to the range of $[0..1]$ with the sum of the reciprocal of all distances:

$$p_{i,color} = \frac{\left[d(\mathbf{p}_i, g_{color}) + \epsilon \right]^{-1}}{\sum_{c=1}^C \left[d(\mathbf{p}_i, g_c) + \epsilon \right]^{-1}} \quad (5)$$

The positive small value of ϵ has been introduced to overcome the division-by-zero problem for RGB pixels lying on (or close to) the prototype line.

With the presented method we have a soft colour classifier which assigns probabilities in contrast to absolute values. Additionally by utilising the proposed non-parametric KMeansLineFit method, the soft colour classifier is adapted to the characteristics of the given input image I_{input} in terms of colour cross talk, albedo etc. without explicitly modelling these effects. As a result the adaptation to the colour statistics implicitly implies a colour calibration (e.g. colour cross talk). Standard colour calibration would account for the displacement of the prototype lines which are caused by just the

systems inherent characteristics. An explicit colour calibration is unnecessary, since the difference between the detected colours from the projected ones, which are caused significantly by the scenes albedo, are directly handled by means of adaptation to the colour statistics.

5 Stripe Matching

During the previous steps a set has been computed containing all the detected stripe candidates. Each one is specified by a location (scan column index and position along that one), a likelihood of being correctly recovered and the likelihoods of being projected with the pattern colours. The current task is to establish correspondences between projected and detected stripes and to skip all invalid stripe candidates which have emerged due to non-optimal light, skin and sensor conditions. This constitutes the probabilistic global optimisation.

This matching is a typical combinatorial optimisation problem (COP) finding a combination of correspondences that fits best. We follow the common way to solve such tasks by modelling it as a Markov random field and setting up an objective function that has to be maximised in order to find the best combination. The objective function takes all the available information for all stripe candidates into account, which is:

- the likelihood of being a valid stripe ($p_{i,valid}$),
- the likelihood of being projected with the different pattern colours ($p_{i,colour}$) and
- the deviation of the detected sequence from the projected pattern ($p_{i,sequence}$).

This problem is solved for each vertical scan column separately. The objective function we have developed is the product over all the available probabilistic weights. We distinguish between the two cases of stripe candidates being matched ($\mathbf{p} \in M$) and skipped ($\mathbf{p} \notin M$)

$$L = \prod_{\forall i \in M} p_{i,colour} \cdot p_{i,valid} \cdot p_{i,sequence} \cdot \prod_{\forall i \notin M} 1 - p_{i,valid} \quad (6)$$

The global likelihood L contains the probability of being invalid for every rejected stripe candidate. For every successfully matched stripe it contains the product of the likelihood of being a valid stripe, the likelihood that this stripe was projected with the corresponding pattern colour and the likelihood that this colour occurs in this sequence in the projected pattern. The latter term is often called a jump weight because it assigns good scores for stripes being in order with the pattern and bad scores for incoherent sequences.

This COP is solved efficiently with the dynamic programming method [25] to find the global optimum. The typical dynamic programming approach is to set up a table containing scores for the assignments and traversing through it. Afterwards, the best score achieved is traced back and all the encountered correspondences are found and the prospective stripes marked as invalid are skipped.

A more comprehensive discussion on dynamic programming and Markov random fields in general (among others) can be found again in the text book [4]. Since we are dealing with faces that just contain occlusions of the type that some projected stripes are missing in the captured image, the ordinary dynamic programming method is sufficient. In the case, that in the captured image also the order of the projected stripes can change, the multi-pass dynamic programming method [29] can be used.

To reduce the processing time the concept of Epipolar segments have been adopted [28]. This means that certain combinations of input and output stripes are not considered for matching, because the resulting vertices would lie outside the region where the face is expected to be. In the same way the dynamic programming is restricted to check only for matches that result in surfaces without discontinuities by limiting the considered pattern stripe candidates for each detected camera stripe.

6 Experimental Results

The final depth of every correspondence is evaluated by triangulating the 3D point cloud. Therefore the projection matrices of the camera and projector are needed, which we obtain by calibration [8]. The depth calculation is performed by intersecting the two lines of sight through the focal points and the image points of the camera respective projector [12].

In our experiments we use a DLP projector "Projection Design F1+" with a SXGA+ resolution (1400 x 1050) and a camera "Canon EOS 20D" with a resolution of 8.2 mega pixels (3522 x 2348). The pattern contains stripes with a width of two pixels and three pixels intersection.

The C++ running time for generating a 3D model of a face lies in the range of a minute on a 3 GHz Pentium-4 computer without exploiting any potential for parallel execution on multi-core processors. The running time depends on the number of detected vertices. In our experiments the KMeansLineFit algorithm converged in less than ten iterations until being fully adapted to the light conditions respective reflectance characteristics found in the captured image.

Many experiments have been performed with the 3D face scanner. Fig. 1 shows the results of a typical scenario. Two pictures have been taken, one with regular white light and one with the structured light pattern shown in Fig. 3. After selecting the region of interest, the system has set up a colour classifier suitable for the given scene as shown in Fig. 7. After classifying the detected stripes and establishing correspondences the 3D model of the face is calculated and optionally presented as wire frame model, surface or textured 3D face. The 3D scan of the face presented in Fig. 8 illustrates the richness in detail. It contains 126 544 triangles with 63 644 vertices.

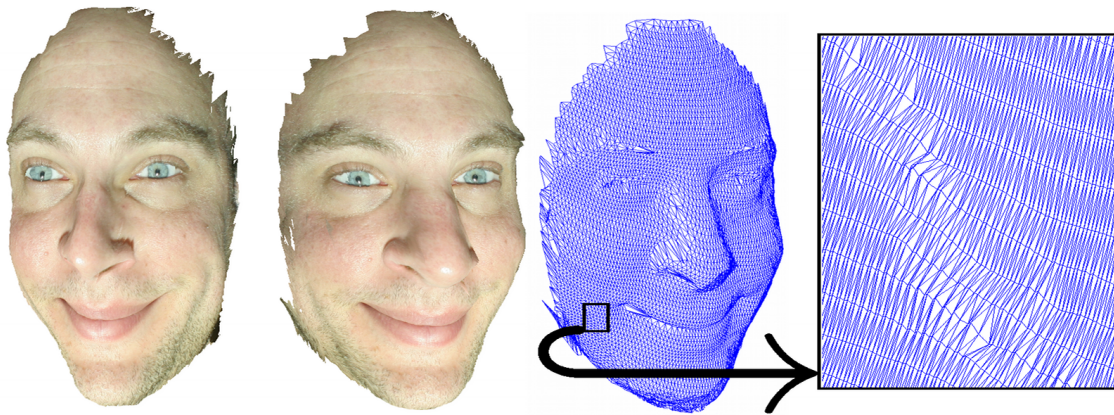


Figure 8 *Result of 3D scan*

In Fig. 9, a region around the mouth is depicted to demonstrate the main stripe matching including stripe detection and colour classification. Here, no post processing has been performed which is normally done to remove or align outliers. A simple colour classifier is used as reference method [10]. It is a pixel-based method which considers the ratios between the colour channels: $\frac{r}{g}, \frac{r}{b}, \frac{g}{r}, \frac{g}{b}, \frac{b}{r}, \frac{b}{g}$. Despite of low-quality and over- and under-exposed pixels in the input image the proposed system produces good results.

In Fig. 10, we show a quantitative comparison of our results with the KMeansLineFit colour classifier and the colour ratio-based method [10] mentioned above. It is clearly visible that the KMeansLineFit colour classifier drastically improves the number of correctly labelled stripe pixels.

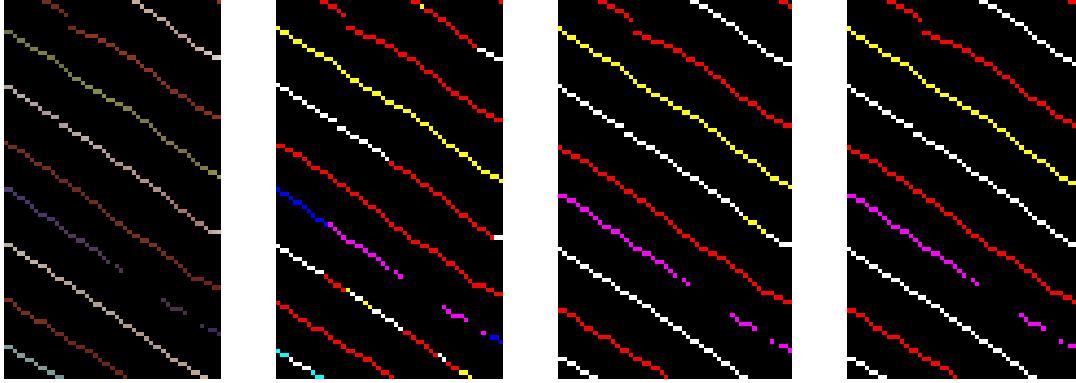


Figure 9 Results of stripe detection: *most left: detected stripes input, middle left: ratio based colour classification, middle right: KMeansLineFit colour classification, most right: ground truth*

7 Conclusion and Future Work

We have presented a system for high-resolution 3D face scanning based on single captured images. The system generates high-accuracy 3D models by exploiting specialised low-level algorithms performing on raw photo sensor data. Additionally, the 3D face scanner has been made robust in terms of lighting conditions, skin, colour-cross talk. This is achieved by adapting the colour classification to the characteristics of the captured image utilising the proposed non-parametric KMeansLineFit algorithm without the need to explicitly model any of these perturbing effects. Experiments with scanned faces under non-ideal light conditions are presented to demonstrate the systems performance. The simple setup and its easy usage make the presented system ideal suited for various 3D model creation scenarios, for example virtual environments like 3D games or human machine interfaces.

An interesting enhancement could be to explicitly adapt the colour classifier to local regions instead of the entire image. Additionally, enhanced accuracy may be achieved by calculating the correspondences with consideration of the whole input image instead of executing it for each scan column separately.

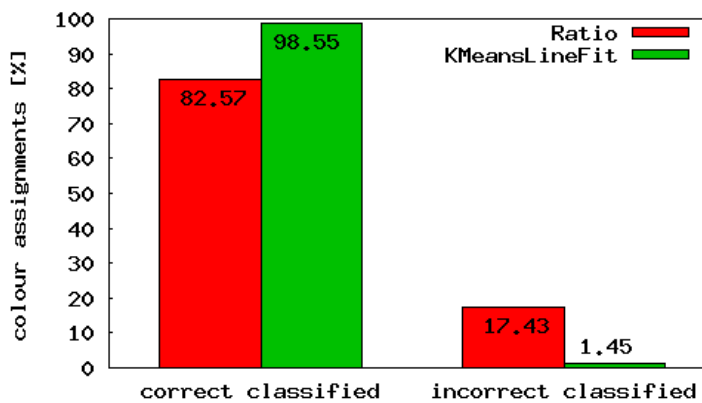


Figure 10 Quantitative comparison of KMeansLineFit colour classifier

8 Acknowledgement

The work presented in this paper has been developed with the support of the European Network of Excellence VISNET II (Contract IST-1-038398).

References

- [1] C. Albitar, P. Graebbling, and C. Doignon. Design of a Monochromatic Pattern for a Robust Structured Light Coding. In *IEEE International Conference on Image Processing (ICIP 2007)*, pages 529–532 (VI), San Antonio, Texas, USA, 16-20 September 2007. [2](#)
- [2] C. Albitar, P. Graebbling, and C. Doignon. Robust Structured Light Coding for 3D Reconstruction. In *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1 – 6, Rio de Janeiro, Brazil, 14-21 October 2007. [2](#)
- [3] D. Arthur and S. Vassilvitskii. k-means++: the advantages of careful seeding. In *Proceedings of the 18th annual ACM-SIAM Symposium on Discrete Algorithms (SODA2007)*, pages 1027 – 1035, New Orleans, Louisiana, USA, 7-9 January 2007. [8](#)
- [4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006. [8](#), [11](#)
- [5] F. Blais, M. Picard, and G. Godin. Recursive Model Optimization Using ICP and Free Moving 3D Data Acquisition. In *Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling (3DIM 2003)*, pages 251– 258, Banff, Alberta, Canada, 6-10 October 2003. [3](#)
- [6] K. L. Boyer and A. C. Kak. Color-Encoded Structured Light for Rapid Active Ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9:14 – 28, January 1987. [2](#)
- [7] D. Caspi, N. Kiryati, and J. Shamir. Range Imaging With Adaptive Color Structured Light. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 20, Issue 5:470–480, May 1998. [2](#)
- [8] P. Eisert. Model-based camera calibration using analysis by synthesis techniques. In *Proc. 7th International Workshop Modeling, and Visualization 2002*, pages 307–314, Erlangen, Germany, November 2002. [12](#)
- [9] P. Fichteler and P. Eisert. Adaptive Color Classification for Structured Light Systems. In *Proceedings of the 15th International Conference on Computer Vision and Pattern Recognition (CVPR2008) - Workshop on 3D Face Processing (3DFP)*, pages 1–7, Anchorage, Alaska, USA, 27th June 2008. [2](#)
- [10] P. Fichteler, P. Eisert, and J. Rurainsky. Fast and High Resolution 3D Face Scanning. In *Proceedings of the International Conference on Image Processing (ICIP 2007)*, volume 3, pages 81–84, San Antonio, Texas, USA, September 2007. [2](#), [12](#)
- [11] H. Fredricksen. The lexicographically least debruijn cycle. *Journal of Combinatorial Theory*, 9:509–510, 1970. [4](#)
- [12] R. Goldman. *Intersection of Two Lines in Three Space*, page 304. Academic Press, 1990. [12](#)
- [13] C. Je, S. W. Lee, and R.-H. Park. High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging. In *Proceedings of 8th European Conference on Computer Vision (ECCV2004)*, volume 3021/2004, pages 95–107, Prague, Czech Republic, May 11-14 2004. [4](#)
- [14] T. P. Koninckx, I. Geys, T. Jaeggli, and L. V. Gool. A Graph Cut based Adaptive Structured Light approach for real-time Range Acquisition. In *Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT 2004)*, pages 413– 421, Thessaloniki, Greece, 6-9 September 2004. [2](#)
- [15] T. P. Koninckx, P. Peers, P. Dutre, and L. V. Gool. Scene-Adapted Structured Light. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, volume Volume 2, pages 611 – 618, San Diego, CA, USA, June 20-25 2005. [3](#)
- [16] U. Kthe. Edge and Junction Detection with an Improved Structure Tensor. In *Pattern Recognition, Proc. of 25th DAGM Symposium*, pages 25–32, Magdeburg, 2003. [5](#)
- [17] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, 1967. [8](#)
- [18] T. Monks, J. N. Carter, and C. Shadle. Colour-Encoded Structured Light for Digitisation of Real-Time 3D Data. In *International Conference on Image Processing and its Applications (ICIPA1992)*, pages 327 – 330, 7-9 April 1992. [2](#)

- [19] S. G. Narasimhan, S. K. Nayar, B. Sun, and S. J. Koppal. Structured Light in Scattering Media. In *Proceedings of the 10th IEEE International Conference on Computer Vision (ICCV2005)*, volume 1, pages 420 – 427, 17-21 October 2005. [2](#)
- [20] S. K. Nayar, G. Krishnan, M. D. Grossberg, and R. Raskar. Fast Separation of Direct and Global Components of a Scene using High Frequency Illumination. In *Proceedings of ACM Transactions on Graphics (SIGGRAPH 2006)*, volume 25, pages 935 – 944, New York, NY, USA, July 2006. [3](#)
- [21] J. Pags, J. Salvi, C. Collewet, and J. Forest. Optimised De Bruijn patterns for one-shot shape acquisition. *Image and Vision Computing*, 23:707–720, 2005. [4](#)
- [22] J. Salvi, J. Pages, and J. Battle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37:827–849, 2004. [2](#)
- [23] D. Scharstein and R. Szeliski. A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. *International Journal of Computer Vision (IJCV)*, 47(1/2/3):7–42, April-June 2002. [1](#)
- [24] D. Scharstein and R. Szeliski. High-Accuracy Stereo Depth Maps Using Structured Light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2003)*, volume 1, pages 195–202, Madison, WI, June 2003. [2](#)
- [25] D. B. Wagner. Dynamic Programming. *The Mathematica Journal*, 5(4):42–51, 1995. [11](#)
- [26] Y. Xu and D. G. Aliaga. Robust Pixel Classification for 3D Modeling with Structured Light. In *Proceedings of Graphics Interface (GI2007)*, volume 234 of *ACM International Conference Proceeding Series*, pages 233 – 240, Montreal, Canada, May 2007. [3](#)
- [27] Y. Xu and D. G. Aliaga. Photogeometric Structured Light: A Self-Calibrating and Multi-Viewpoint Framework for Accurate 3D Modeling. In *IEEE Computer Vision and Pattern Recognition (CVPR2008)*, pages 1 – 8, Anchorage, Alaska, USA, June 24-26 2008. [2](#)
- [28] M. Young, E. Beeson, J. Davis, S. Rusinkiewicz, and R. Ramamoorthi. Viewpoint-Coded Structured Light. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2007)*, pages 1–8, Minneapolis, Minnesota, USA, 18-23 June 2007. [11](#)
- [29] L. Zhang, B. Curless, and S. M. Seitz. Rapid Shape Acquisition Using Color Structured Light and Multi-pass Dynamic Programming. In *The 1st IEEE International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT2002)*, pages 24–36, Padova, Italy, June 2002. [2](#), [4](#), [11](#)
- [30] L. Zhang, B. Curless, and S. M. Seitz. Spacetime Stereo: Shape Recovery for Dynamic Scenes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR2003)*, volume 2, pages 367–374, Madison, Wisconsin, 18-20 June 2003. [2](#)
- [31] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz. Spacetime Faces: High-Resolution Capture for Modeling and Animation. In *Proceedings of ACM Annual Conference on Computer Graphics (SIGGRAPH2004)*, pages 548–558, Los Angeles, CA, August 2004. [2](#)
- [32] S. Zhang and P. S. Huang. High-resolution Real-time 3-D Shape Measurement. *Optical Engineering*, 45(12), December 2006. [2](#)