Multimedia Tools and Applications (2024) 83:74535–74562
https://doi.org/10.1007/s11042-024-18573-6

Automatic 3D-display-friendly scene extraction from video sequences and optimal focusing distance identification

Tomáš Chlubna[1] · **Tomáš Milet**[1] · **Pavel Zemčík**[1,2]

Received: 28 August 2023 / Revised: 22 December 2023 / Accepted: 1 February 2024 /
Published online: 16 February 2024

Abstract
This paper proposes a method for an automatic detection of 3D-display-friendly scenes from video sequences. Manual selection of such scenes by a human user would be extremely time consuming and would require additional evaluation of the result on 3D display. The input videos can be intentionally captured or taken from other sources, such as films. First, the input video is analyzed and the camera trajectory is estimated. The optimal frame sequence that follows defined rules, based on optical attributes of the display, is then extracted. This ensures the best visual quality and viewing comfort. The following identification of a correct focusing distance is an important step to produce a sharp and artifact-free result on a 3D display. Two novel and equally efficient focus metrics for 3D displays are proposed and evaluated. Further scene enhancements are proposed to correct the unsuitably captured video. Multiple image analysis approaches used in the proposal are compared in terms of both quality and time performance. The proposal is experimentally evaluated on a state-of-the-art 3D display by Looking Glass Factory and is suitable even for other multi-view devices. The problem of optimal scene detection, which includes the input frames extraction, resampling, and focusing, was not addressed in any previous research. Separate stages of the proposal were compared with existing methods, but the results show that the proposed scheme is optimal and cannot be replaced by other state-of-the-art approaches.

Keywords 3D display · Looking glass · Frames extraction · Video analysis · Optical flow · Light field

Tomáš Milet and Pavel Zemčík contributed equally to this work.

✉ Tomáš Chlubna
ichlubna@fit.vutbr.cz

Tomáš Milet
imilet@fit.vutbr.cz

Pavel Zemčík
zemcik@fit.vutbr.cz

[1] Department of Computer Graphics and Multimedia, Faculty of Information Technology, Brno University of Technology, Božetěchova 2/1, Brno 612 00, Czech Republic

[2] Lappeenranta-Lahti University of Technology, School of Engineering Science, Yliopistonkatu 34, Lappeenranta 53850, Finland

1 Introduction

3D scenes can be projected onto the flat screen of the 2D monitor, but it is difficult to simulate the 3D perception due to the missing depth dimension. The Looking Glass Factory 3D display (LKG) and similar devices can simulate the 3D perception by displaying view-dependent content; see Fig. 1. LKG simulates two main depth cues [1] that lead to an immersive 3D experience. The first is binocular parallax caused by the distance between human eyes. The second is monocular motion parallax, where the occlusions and seeming motions of the objects depend on their distance from the observer. Since LKG does not require additional equipment apart from the screen, such as virtual reality headsets, it is used for the presentation of 3D scenes and data in educational institutions, museums, galleries, and also as a personal 3D digital picture frame. LKG was used during the evaluation of the proposed method, but the method can also be used with other 3D displays.

This paper solves a problem of optimal scene extraction from videos, ensuring the best possible viewing quality on a 3D display. This problem was not previously addressed thoroughly in research papers and no full solution exists. 3D display, such as LKG, requires an array of input images that follow defined rules. The images need to capture the same scene from positions following a horizontal trajectory. The spaces between the images need to be constant and large enough to make the 3D perception visible. The images also need to be shifted to make sure that the scene is sharp and not out of focus. Manual extraction and adjustment of the scenes from, for example, family video albums, movies, or gameplay videos, would be extremely time consuming. Figure 2 shows the common types of artifacts that appear in the result displayed on LKG if the scene is not correctly extracted and how the proposal ensures their mitigation.

The proposed method can produce a suitable scene from an arbitrary video if such scene is present there. The processing speed can reach almost 40 fps when detecting the scene in a FullHD video and about 20 fps when performing the additional resampling and focusing for the extracted scene. Manual extraction and processing takes about 17× longer even for an experienced user according to the conducted tests. The proposed method can be additionally used as an capturing assistant when the user records the scene intentionally for the 3D display. The method would be used in such case to notify the user if the recorded sequence is optimal and would be able to produce a preview of the result. Based on an input video content, the

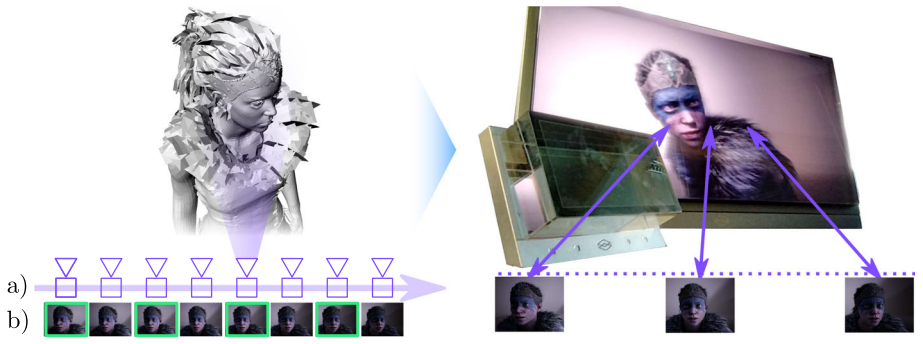


Fig. 1 The proposal where a) video frames with camera positions on a horizontal line are identified, b) desired images are selected, c) frames are converted into the proper format which can be displayed on the LKG. Different views are visible from different real-life viewing angles

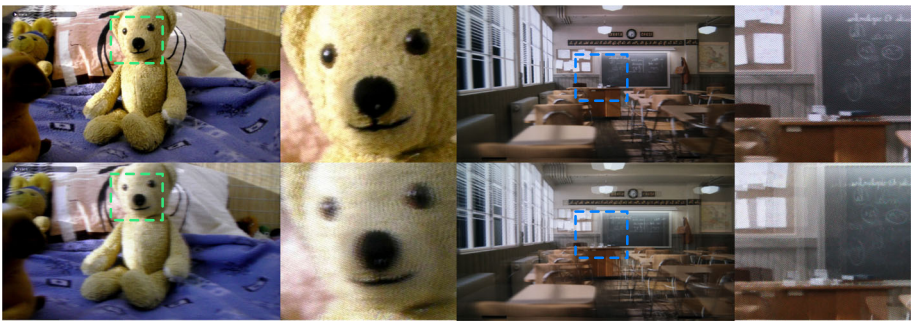


Fig. 2 The top row represents scenes processed by the proposed method. The bottom row shows scenes selected without the proposal by the user just by a quick visual evaluation. The left *Teddy* scene shows focusing artifacts and the right *Class* scene shows ghosting caused by a wrong trajectory of the input frames

proposed method can detect all suitable scenes, process them, and produce so-called *quilt*, which can be directly rendered on the 3D display.

The main scientific hypothesis of this paper is that existing computer vision methods can be adjusted and used to automatically produce an optimal input data for a state-of-the-art 3D display based on input video sequence. The proof of the hypothesis is experimental and presented in this paper. The method is evaluated using an annotated dataset. The main scientific contributions of this paper are as follows:

- Evaluation of existing computer vision methods, identification of the optimal one, and extension of the algorithm for automatic extraction of suitable frames for 3D displays.
- Novel method of resampling and stabilization of the extracted frames to reduce disparity artifacts in the result when viewed on a 3D display.
- Two novel and equally efficient focus metrics tailored for the evaluation of image sharpness on 3D displays.
- First complete novel proposal of automatic production of optimal content for 3D displays based on arbitrary sequence of input images.
- Custom annotated dataset of 7 camera motion categories containing 182 synthetic and real FullHD scenes. This dataset is annotated and designed as a benchmark for the 3D display scene detection task.

Existing state-of-the-art methods cannot be directly used for the production of suitable 3D-display-friendly scenes. Most of the similar methods focus on object tracking or extraction. Experimental evaluation and investigation of the state of the art showed that existing camera motion extraction methods are unable to detect the desired camera motion optimally for the 3D display. Existing image focus detection methods are also not suitable as a replacement of the proposed focusing methods. The input images need to be mixed together and the out-of-focus areas are different than out-of-focus blur in standard photography. 3D reconstruction methods can produce the necessary views for 3D displays by reconstructing the scene based on an input set of views. Experiments showed that such approaches produce a lot of unwanted artifacts due to the reconstruction and additional rendering of the scene. The reconstructions might also fail due to the lack of free 3D motion around the scene in input views. Such 3D motion is, however, not always necessary for 3D displays which require only specific trajectory of the input views. 3D reconstruction algorithms would, therefore, fail on many potentially suitable scenes.

2 Related work

LKG displays, as well as other similar 3D display devices, are described in this section. Related research on camera motion classification in video sequences is also summarized here.

2.1 3D display devices

Many stereoscopic devices require an equipment on the user's head. Virtual reality headsets use two small screens that display a different image of the scene per eye. The disparity of the images helps the brain perceive the depth [2]. Motion of the headset is tracked, so the virtual cameras in the scene move according to the user's head. Oculus Rift and HTC Vive are examples of the most popular devices. Samsung Gear VR is an example of a hybrid approach, where a smartphone is used with the headset as the display device. A more lightweight approach is to use polarization glasses. Each eye obtains a different picture from the screen where the image is projected in two differently polarized light streams [3].

Another approach is a volumetric projection using a rotating mirror and a projector with a high frame rate. 3D scenes can be rendered from multiple angles, and the mirror reflects each of the views in the right direction [4]. A more robust solutions are Voxon Photonics products [5]. They use a rapidly moving flat screen on which a different part of the 3D image is projected. Unlike stereoscopic devices, this approach allows the user to walk around the displayed model, not being restricted to a flat screen and specific viewing angles.

Multiview displays use the principle of viewing-angle-based ray distribution, where multiple views of the scene are displayed at the same time. The tensor display [6] uses a cascade of light-attenuation layers that modify the light coming from the source screen. Rapid temporal modulation showing different frames in a short period of time helps to widen the viewing angle, showing different parts of the scene through the synchronized layers. A so-called holographic screen where each point modifies the light rays emitted from the optical module layer in various directions is used in the HoloVizio display by Holografika [7].

2.2 Looking glass display

The 3D display by Looking Glass Factory [8, 9] belongs to the multiview display category. It is capable of simultaneously displaying 45-100 images of the scene so that the user can see a different part of the scene from a different viewing angle; see Fig. 3. No additional headsets or glasses are necessary. Three main limitations of current display models exist: the 50°-60° viewing cone, out-of-focus artifacts, and only horizontal change in views.

Fig. 3 The *classroom* scene from the testing dataset was displayed on the LKG. Three photos were taken from different angles around the display showing the 3D parallax

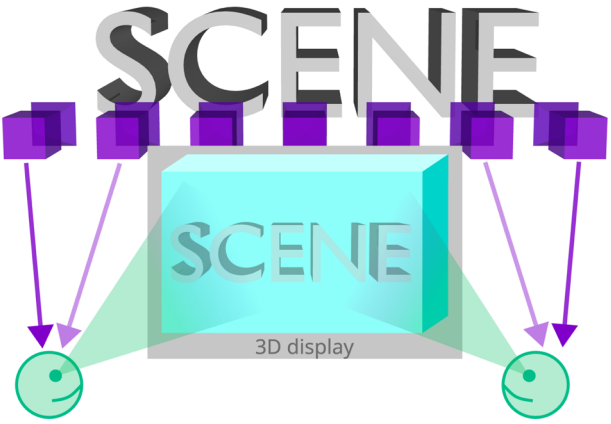


Fig. 4 The figure shows the optimal orientation of the capturing cameras which are placed in the scene according to the horizontal 3D display orientation. Users then see a combination of the captured views according to their viewing position

The input to the LKG display software is a set of images (at least 45 for their seamless transitions) capturing the scene. They represent a discrete light field approximation [10]. The images have to be placed along a horizontal straight line that is perpendicular to the optical axes of the capturing cameras and parallel to the horizontal edges of the LKG screen. The capturing cameras have the same orientation and are evenly distributed along the line; see Fig. 4. The images are then converted to an internal LKG format that is displayed directly. The process is described in Fig. 5. The optical layer, composed of lenticular lenses, distributes the rays coming from the pixels of the source display in the correct direction to simulate the 3D effect.

2.3 Camera motion evaluation

Camera motion following a horizontal straight line trajectory is called *truck* in cinematography. It is visually similar to *pan* which consists only of rotation. Dense [11] or sparse [12] optical flow can be used to detect camera motion for consecutive video frames. Another option is to use feature matching algorithms with descriptors such as SIFT [13], SURF [14], KAZE [15] or ORB [16] and find the displacement of the found features. In this paper, selected approaches are compared. The features detection is often used in tracking algorithms [17]. The method proposed in this paper uses the tracking in a reverse way to track the camera.

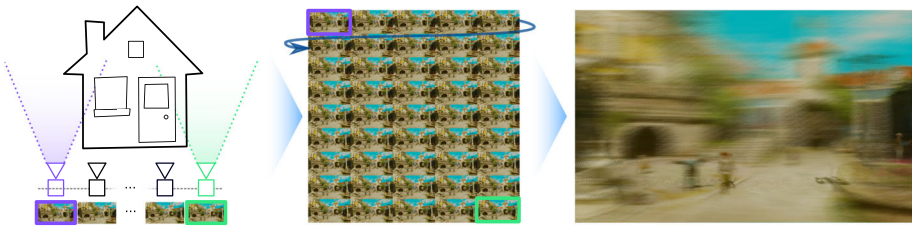


Fig. 5 Evenly distributed cameras on a horizontal line capture the scene as a set of images that are merged into one matrix (quilt). The quilt is then transformed into the internal LKG format which can be directly displayed on LKG

For a sparse optical flow, the tracking features are selected using the Shi-Tomasi corner detector [18]. The dominant camera motion in a video sequence is usually estimated by averaging optical flow vectors in predefined regions of the image/flow [19]. A set of rules is then used to estimate camera motion based on the dominant direction in each region [20, 21]. A similar optimized principle is used in the proposed method.

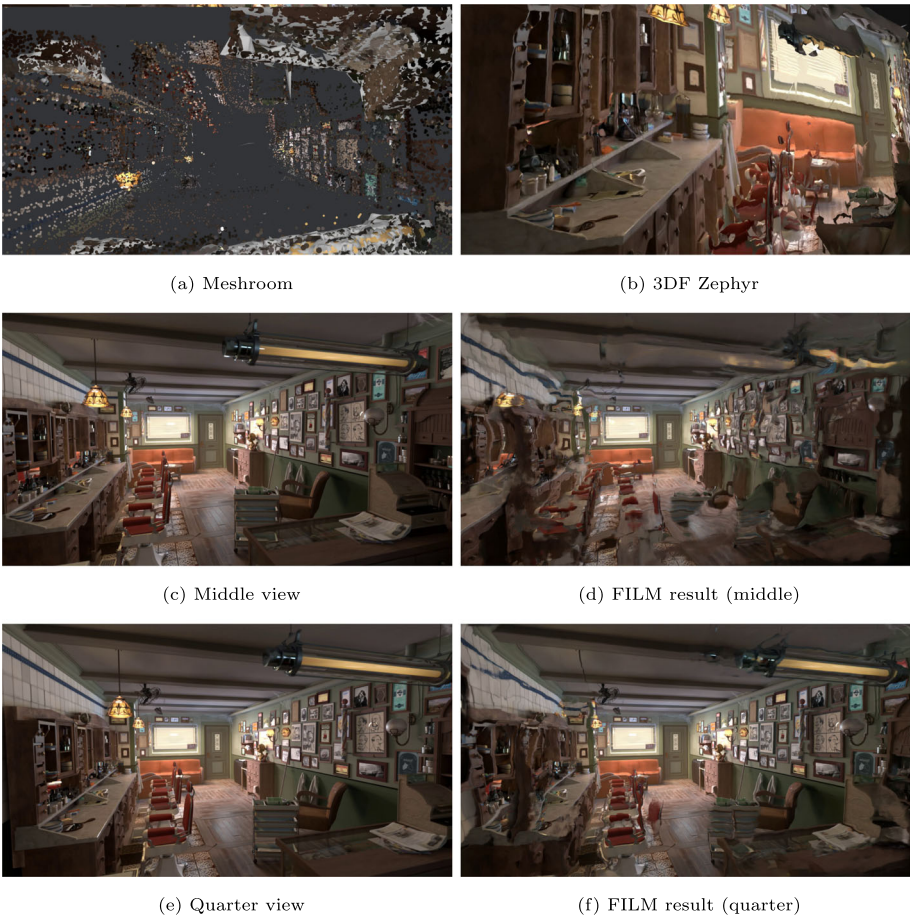
Knowing the intrinsics of the camera, a combination of multiple optical flow models can provide more accurate results [22]. A spatio-temporal derivative of the intensity in two successive frames can be used [23]. Motion vectors can be computed using temporal gradient-based block matching. The resulting motion is estimated using the motion vectors of interest, which are chosen from the field according to their significance and consistency [24]. If encoded video files are used as input data, the necessary motion vectors can be extracted from the compressed stream [25–27]. Hidden Markov Models [28] or Transferable Belief Model [29] might be more robust alternatives to solid thresholds in the camera motion estimation rules. Multiresolution least-squares methods might be used to fit the motion model even with noisy data [30]. Model-based estimations might not cover the whole range of possible videos and camera motions. Machine learning approaches to overcome this problem were also proposed [31–35] but their quality depends on the training process. The proposed direct algorithm might be more robust and lightweight.

Structure-from-motion and visual simultaneous localization and mapping techniques are frequent approaches that are used to reconstruct a point cloud of a 3D scene or to retrieve the camera trajectory from a sequence of images. ORB_SLAM3 [36] is one of the methods compared in this paper. It is based on a previous SLAM research [37, 38]. ORB_SLAM3 approach consists of tracking, mapping, relocalization, and loop closing based on ORB features extracted from the image frames. Camera pose information can be retrieved in real time for each frame after the initialization phase when the scene map is created. The quality of the result, aside from the quality of the input sequence, depends on the camera intrinsics, especially the focal length value. Similar approaches can be used to detect optical flow in deformable scenes [39].

The detection process of this paper could be improved in the future by semantic analysis of the scene, such as in preview frame detection methods [40–42]. However, the main goal of the proposed method is to select all candidate scenes from the input video so that the user can get a set of possible quilts from the input video.

2.4 Novel view synthesis

Novel views can be interpolated from different images of the same scene [43–45]. Such approaches can be used to generate the quilt for LKG. The occlusion-based parallax and reflections are the most important elements that create realistic feel of the scenes displayed on LKG. However, the results of the interpolation methods often contain visual artifacts, as information about occluded objects and reflections is missing. The artifacts might not be noticeable in computer games or dynamic scenes, but one of the main use cases for LKG is realistic simulation of objects in a real setting. For example, Looking Glass Portrait contains Raspberry Pi4 and can act as a standalone 3D photo frame. Any unrealistic artifacts can be disturbing during exploration of the scene. The frames extracted by the proposed method are guaranteed to be correct or can be even used for inter-frame interpolation between close views where artifacts are less visible. The proposed method is, therefore, an important tool for any quilt extraction task, even involving interpolations.



(a) Meshroom (b) 3DF Zephyr

(c) Middle view (d) FILM result (middle)

(e) Quarter view (f) FILM result (quarter)

Fig. 6 Zephyr produced the best result when reconstructing the scene but is still full of artifacts. The deep FILM interpolation produced distorted results for both full and halved distance between the views from *Barber* scene in the used dataset

The 3D reconstruction fails due to the lack of camera orientation change in the input views. The suitable sequences for LKG do not contain views positioned freely in the space. For example, the expected LKG-friendly *Barber* scene from the dataset was used in INGP [43], FILM [46], Meshroom[1] and 3DF Zephyr.[2] The scene was reconstructed in the two programs, but the result is not optimal, as shown in Fig. 6. INGP pre-processing of the data by Colmap[3] failed. The FILM deep learning frame interpolation produced a frame between the boundary frames from the video. Two issues appeared: the frames had to be downscaled from FullHD to HD due to insufficient resources on a high-end NVIDIA RTX 2070. The interpolated frame contained visible artifacts even when the distance between views was halved. RGBD photos can also be processed by LKG software and displayed, but the lack of parallax and missing reflections create the unwanted cardboard effect [47]. Also, rendering of

[1] alicevision.org

[2] 3dflow.net

[3] colmap.github.io

the large number of views in a high quality is problematic and requires a lot of computational resources [48].

The only reliable way to produce a good-quality quilt from videos is the proposed method with possible interpolation in a small scale. The proposed focusing metrics are still necessary for the extracted or interpolated views. The user can capture the scene in a similar way as when taking a panorama photo with a mobile phone. The only difference is that, instead of rotating the phone, the user would move to the side. Such captured footage would be automatically prepared by the proposed method for direct rendering on the 3D display.

3 Proposed methods

The proposed method consists of the following steps:

1. **Quilt extraction**: The input video stream is analyzed, and suitable frames are chosen for further processing. The frames have to capture the scene having their camera positions lying on a horizontal trajectory.
2. **Frame resampling**: The camera motion in the selected sequence can be non-linear or with unsuitable spacing between frames. Equally distanced frames are selected from the sequence to ensure the best visual quality of the rendered result.
3. **Automatic focusing**: 3D displays support only one distance where the scene is visually sharp and focused. The focusing distance is estimated to ensure that most of the scene is in focus.

Sparse optical flow, dense optical flow, feature matching, and SLAM approaches were compared in the reference implementation to estimate the camera trajectory. In the first phase, a rough camera motion estimation is performed, detecting truck and pan motion sequences. Pure truck sequences are ideal for LKG. The camera can also do a combination of truck and pan which might still be acceptable. The panning motion reduces the parallax effect. The second phase determines the amount of pan motion in the sequence. Additional checks are performed, such as motion blur and shakiness detection. A score that evaluates the suitability of the sequence is the result of the second phase. An approach using ORB_SLAM3 does both tasks in one phase.

3.1 Horizontal sequence detection

Pairs of pixels blocks are identified as belonging to the same spot in the scene in two consecutive video frames. The field of motion vectors (optical flow) is then known for each frame pair. The mean of these vectors is a good guide to determine the overall motion of the camera. Monocular visual SLAM methods also exploit a similar concept of feature matching in two frames, and based on them, the camera pose or scene map is estimated. Figure 7 shows a simple scenario in which the ideal horizontal quilt sequence is mapped to a subset of suitable video frames.

Possible scene change can be detected by a simple histogram comparison or by setting a threshold for the estimated camera pose change. The result of the analysis is an array of estimated camera pose differences between frames. The analysis phase is the most computationally expensive part of the processing.

In many cases, the mean value of these vectors can be used directly to decide whether the truck or pan motion is dominant in the shot. When the truck motion is combined with pan, the parallax can be inverted, depending on the distance of the objects in the scene from the

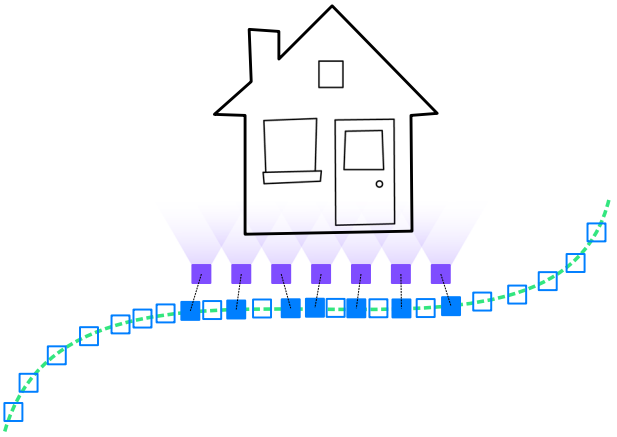


Fig. 7 The camera positions on the trajectory belong to frames in the input video. The required ideal horizontal sequence capturing the scene from a desired spot and with a given spacing is then mapped on the closest input frames

camera. To solve this issue, the direction of the sequence can be ignored, working only with the magnitude of horizontal motion. The method accepts ith frame f_i in the sequence seq_i until the vertical velocity vel_{vert} is above a defined threshold t_{vert} or the horizontal velocity vel_{hor} is close to zero; see Algorithm 1 and (1).

$$seq_i = \begin{cases} accept & \text{if } vel_{hor}(f_i, f_{i-1}) > 0 \wedge \\ & vel_{vert}(f_i, f_{i-1}) \leq t_{vert} \\ reject & \text{otherwise} \end{cases} \tag{1}$$

In the reference implementation, 30% of the extreme vectors are dropped, 15% from each side of the sorted array to eliminate outliers. The distortions caused by, e.g., very close objects

Algorithm 1 The algorithm iterates over all consecutive frame pairs and adds frames into the horizontal sequence.

```
Data: Video frames sequence fr
Result: Clips suitable for LKG quilt clips
motions = [];
foreach two consecutive frames ∈ fr do
    insert(motions, motion(prev, next));
clips = [];
insert(clips, emptyClip);
foreach (id, motion) ∈ motions do
    x, y = |motion_xy|;
    if x > x_min and y < y_max then
        insert(last(clips), id);
    else
        if sufficientLength(last(clips)) then
            insert(clips, emptyClip);
        else
            clear(last(clips));
```

Fig. 8 The apparent motion of the pixels is demonstrated in the images for truck, pan and combined camera motion. Frames from the testing sequences were blended together and image gradient of the result was highlighted

that have high apparent velocity or poorly matched features, are then reduced. The vertical motion threshold was set in the implementation to $1.5px$ in both directions. The value was determined according to the acceptable limits discovered in a previous study [49].

3.2 Pan elimination

Figure 8 shows images composed of multiple video frames. The pan shots are very similar to the truck ones, except for the distortion in the corners of the field. The angles between the motion vectors and the horizontal axis are higher than zero; see Fig. 9. Dense optical flow can be computed for each corner of the image pair. The mean vector should point down on one side and up on the other side of the image or vice versa, according to the camera motion direction. Each fifth pair is tested in the reference implementation for performance reasons and shows no quality degradation compared to testing every pair.

A score is calculated at the end of the second phase for each processed sequence; see (2). The score s takes into account the overall amount of motion $blur$ in the sequence, $shake$ (amount of vertical motion), the average presence of the pan pattern, and the difference $dist$ between the average amount of vertical motion at the center of the image y_{cent} and at the corners y_{cor} normalized by the maximal allowed limit $maxDiff$; see (3). The amount of blur is measured using the variance of Laplacian method [50, 51]. Lower-scored sequences are assumed to be more suitable for LKG. The weights w_i can be adjusted depending on specific quilt requirements. In the measurements, the weights were set to 0.2, 0.5, 0.1, 0.2 in this

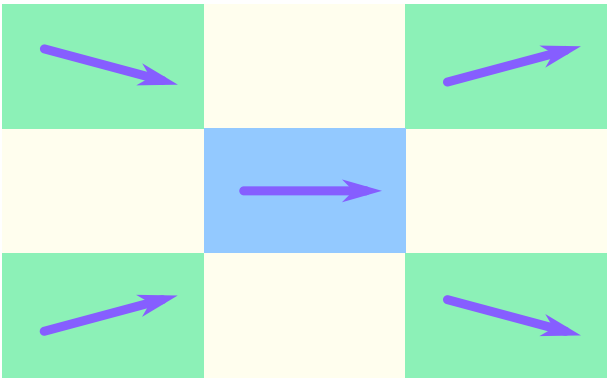


Fig. 9 Motion field model, used to identify pan in the second phase of the extraction, is depicted in the figure

order. The values were determined according to a previous study on user experience related to camera trajectory distortions on LKG [49]. It shows that the pan pattern can negatively affect the result quickly, while shaking can be tolerated to a certain extent.

$$s = w_1 \cdot dist + w_2 \cdot pan + w_3 \cdot shake + w_4 \cdot blur \tag{2}$$

$$dist = min\left(1, \frac{y_{cor} - y_{cent}}{maxDiff}\right) \tag{3}$$

In the case of ORB_SLAM3 analysis, the camera pose matrix is decomposed into translation and rotation. The translation difference can be simply obtained by subtracting the translation vectors. The rotation difference $diff_{rot}$ is estimated by comparing the rotational quaternions $q1$ and $q2$ according to (4). Pure pan motion is not suitable for SLAM reconstruction and is rejected automatically.

$$diff_{rot} = 1 - \left|\frac{acos(q_1 \cdot q_2)}{\pi/2} - 1\right| \tag{4}$$

3.3 Non-linear camera velocity and noise

The motion vectors are available from the analysis phase and can be used to improve the quality of the accepted sequence. The vertical component of the average motion vector can be used to reduce the shaking of the camera on the vertical axis. The camera motion perpendicular to the viewing plane (dolly) can be used to change the scale of the images to reduce the noise on the depth axis.

For the best viewing experience on LKG, equally distanced views of the scene are optimal [49]. This requires a constant camera velocity and frame rate, which is not guaranteed. If the frames are not equally spatially distanced, the sequence has to be sampled non-uniformly. The frame distance can be represented by a horizontal component of the average motion vector between frames. Horizontal position changes are accumulated in acc frame-by-frame, with increasing index i. The nearest neighbor can be accepted as the nth frame in the resampled sequence res_n when the accumulated motion exceeds the maximal allowed value m_{max} which is computed by their difference d in (5). Algorithm 2 and (6) describe the conditions which decide the index of the accepted frame which is closer to the desired position.

$$d = acc - m_{max} \tag{5}$$

$$res_n = \begin{cases} i & \text{if } d < |d - |m_{i-1}|| \\ i-1 & \text{otherwise} \end{cases} \tag{6}$$

The distances between frames have to be large enough to create sufficient 3D perception. If the frames are too close, the LKG result looks flat without significant parallax. On the other hand, too large distance reduces the amount of focused area in the scene. The sequence can be resampled according to the optimal frame distance; see Fig. 10. Equation (7) describes the computation of the best alignment offset o_{best} from the range (o_s, o_e) of the detected sequence. The $\boldsymbol{s_{optimal}}$ is a vector that contains the positions of the views in the optimal sequence with constant spacing. The $\boldsymbol{s_{detected}}$ is a vector that contains the positions of the views in the sequence that was detected in the previous phase. The optimal sequence is shifted by the offset o and the error between the sequences is computed, looking for its minimum.

Algorithm 2 The sequence is resampled in the algorithm to ensure a constant camera position difference between the frames.

Data: X axis motion values for a sequence of selected frames $clip_x$
Result: Sequence of frame indices with linearized motion $clip_{lin}$
$motion_{max} = \max(clip_x)$;
$acc = |\text{first}(clip_x)|$;
$clip_{lin} = []$;
foreach *two consecutive (id, motion)* $\in clip_x$ **do**
  $acc \mathrel{+}= |motion_{next}|$;
  if $acc >= motion_{max}$ **then**
    $delta = acc - motion_{max}$;
    if $delta < |delta - |motion_{prev}||$ **then**
      $\text{insert}(clip_{lin}, id_{next})$;
    else
      $\text{insert}(clip_{lin}, id_{prev})$;
    $acc \mathrel{-}= motion_{max}$;

The vector and scalar addition is defined as the addition of the scalar value to all values in the vector.

$$o_{best} = \underset{o \in [o_s, o_e]}{\arg\min} \left\{ \left| (\boldsymbol{s_{optimal}} + o) - \boldsymbol{s_{detected}} \right| \right\} \tag{7}$$

Equation (7) is implemented in practice by Algorithm 3 that describes a quilt window with the desired distance. This window is sliding over the extracted frames, and the error is computed from the distance between the actual frames and the positions in the window. The window position with the lowest error marks the best sequence according to the given requirement. The optimal distance depends on the scene content and the users' preferences.

The resampling algorithm can be used to also export a quilt with different number of views for other multiview devices such as 3D tablet Lume Pad.

3.4 Focusing

The display blends multiple frames to avoid discrete frame changes when users change viewing position. The blending can be simulated by (8) which is implemented by the microlens array on the display. A pixel at the coordinates x, y of the resulting view v, focused at a focusing distance f is calculated as the sum of pixels of the total N images of the quilt where

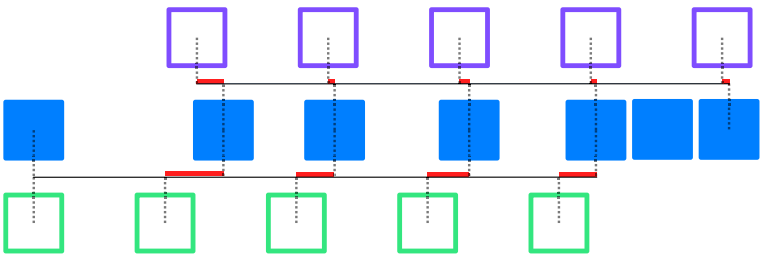


Fig. 10 The middle row depicts the input frames. A quilt window with fixed spacing is placed at the beginning in the bottom row and at a better position in the top row. The distances between the positions in the sampling windows and the positions of the nearest input frames are defining the overall error

Algorithm 3 A quilt window is sliding along the sequence, computing the distance between the closest samples and optimal positions. The window position with the lowest error is then selected.

Data: Sequence of frame positions pos; desired spacing between the views $space$; location of interest in the input sequence loc; searching distance around the defined position $dist$
Result: An offset of the window with desired parameters with the lowest error $clip_{best}$

```
clips = [];
step = (2 · dist)/RESOLUTION;
range = vec2(loc − dist, loc + dist);
for offset in range with step do
    error = 0;
    window = [];
    for i from 0 to QUILT_SIZE do
        sample = offset + i · space;
        nearest = findNearest(pos, sample);
        diff = |nearest_pos − sample|;
        error += diff^2;
        insert(window, nearest_id);
    // if duplicates are not allowed
    if hasNoDuplicates(window) then
        clip = newClip(error, offset);
        insert(clips, clip);
clip_best = lowestErrorClip(clips);
```

the ith image is sampled by function $I(i, x, y)$ at the desired position. The sum is weighted and the weight w_i for the tth image is calculated according to the position p that defines where the resulting view is in the quilt as a rational number in the range $(1, N)$.

$$v(x, y, p, f) = \frac{\sum_{i=1}^{N} I(i, x + (1 - 2(i/N))f, y) \cdot w_i(p))}{\sum_{i=1}^{N} w_i(p))} \tag{8}$$

Scene content positioned at *zero-parallax plane* is always sharp and maintains its screen space coordinates; see Fig. 11. The further the content is from the plane, the more parallax is present along with the out-of-focus artifacts. The zero-parallax plane position can be changed by shifting the views so that the same area of the scene lies on the same screen-space coordinates. The position of the zero-parallax plane is defined as the focusing parameter f in (8).

The positions of the start and end points of the acceptable focusing range, where at least parts of the scene are focused, depend on the depth range of the captured scene; see Fig. 12.

Fig. 11 The first picture shows the 3D scene with two zero parallax plane positions. The second and third pictures are the actual results displayed on the LKG showing how the plane position affects the focusing in the rendered image

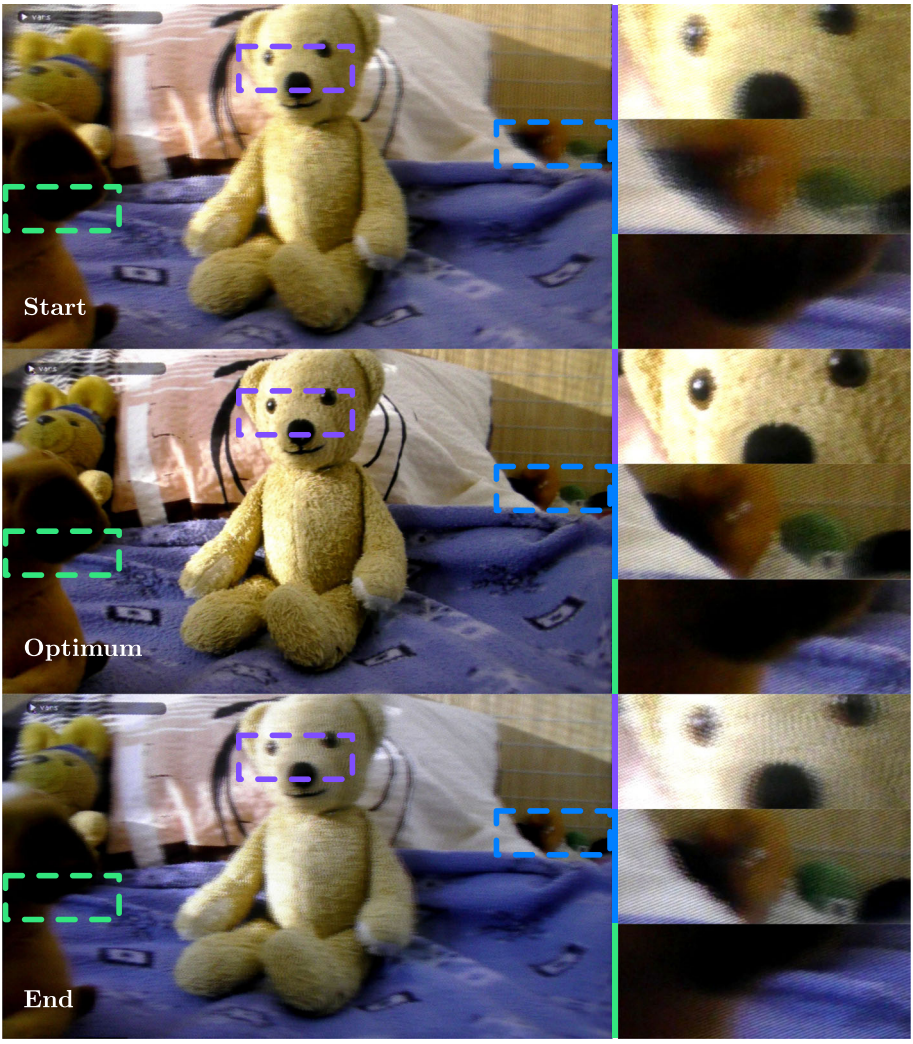


Fig. 12 Scene is focused at the borders of the focusing range and in the optimal distance where most of the scene appears to be in focus

The proposed method finds the optimal zero-parallax plane position that makes most of the scene in focus.

The unique internal image format, displayed directly on the LKG screen, where all quilt views are combined into one image, is analyzed. The combination of the images from the quilt into the internal format is performed according to (9), where h is the index of the RGB channel ($R = 0, G = 1, B = 2$) and internal format access function I is extended by this parameter to index the specific channel. The calibration parameters defined by the LKF manufacturer are defined as follows: s is subpixel shift, t is tilt, p is pitch, v is view portion, and c is center shift. The total number of views is defined as N and $\lfloor\rfloor$ means floor operation and $\{\}$ is the extraction of fractional part. Each physical LKG device comes with a different set of these parameters. The equation allows to directly render the quilt on the

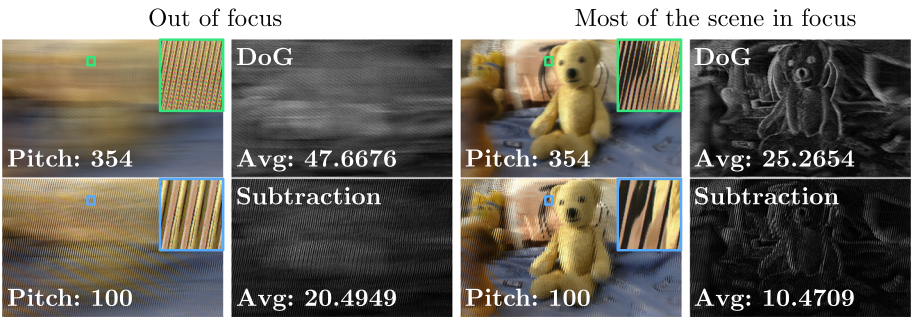


Fig. 13 Focused and out-of-focus images are shown. Left ones are in the LKG internal format with two different display calibration settings. The right images show the energy of the images processed by the proposed methods

display without any proprietary software as it distributes the pixels according to the structure of the microlens array on LKG. The focusing can be achieved by the transformation of x, y coordinates similarly to (8).

$$I_{lkg}(x, y, h) = I(\lfloor\{(x + sh + yt)p - c\}N\rfloor, vx, vy) \tag{9}$$

The areas of the image outside the zero-parallax plane are distorted by a diagonal blur-like pattern. The more the pattern is visible, the less focused the given part of the image is on LKG. Two focus estimation metrics are proposed.

The first is a difference of Gaussians (DoG) edge detection on the internal LKG format image. The second performs a subtraction of two internal LKG images with different display calibration settings. The diagonal blur pattern is different for each calibration. In both cases, the out-of-focus pixels in the result contain high values. A similar principle was used in all-focused light field research [52]. Figure 13 shows both metrics and their results. The algorithm then iterates over a wide focus range and searches for the minimum of the given metric, as shown in Fig. 14. The principle is the opposite of the standard blur measurement methods [53]. The Gaussian metric is described by (10) and subtraction by (11). The optimal focusing value f_o^{DoG} and f_o^{sub} searched as f in interval (f_s, f_e) is computed, for the first metric, as a difference of internal LKG format images I_{lkg} with calibration parameter c_1,

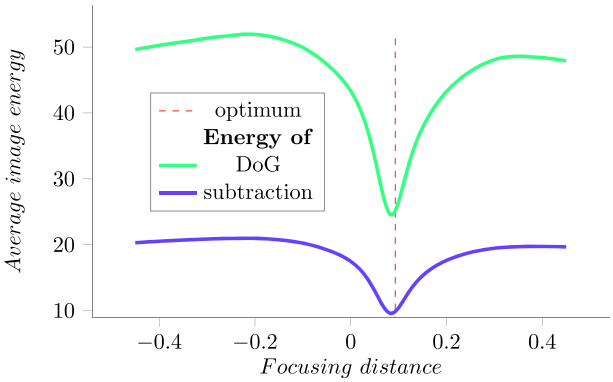


Fig. 14 Two focusing metrics are plotted with manually marked optimum distance where most of the image is focused in *Teddy* scene. The location of the global minimum of the energy marks the best focus

processed by Gaussian filter G with filter parameters p_1 and p_2. The second metric subtracts two different internal images with calibration parameters c_1 and c_2. The DoG parameters used in the measurement were $radius = 10, \sigma_1 = 0, \sigma_2 = 10$ but different reasonable parameters showed the same results. The calibration parameters, used in (9), c_1 were set to $t = -0.1153$, $p = 354.42108$, $c = 0.04239$, $v = 0.99976$ according to a physical device used in the tests and the parameters c_2 were changed as $t = 0.05$, $p = 150$ which showed the most visible changes after subtraction.

$$f_o^{DoG} = \underset{f \in [f_s, f_e]}{\arg\min} \{G(p_1, I_{lkg}(c_1, f)) - G(p_2, I_{lkg}(c_1, f))\} \tag{10}$$

$$f_o^{sub} = \underset{f \in [f_s, f_e]}{\arg\min} \{I_{lkg}(c_1, f) - I_{lkg}(c_2, f)\} \tag{11}$$

4 Experimental evaluation

The overall quality of the detector was evaluated in terms of efficiency and performance (see Table 1) in the following experiments:

1. **Quilt extraction**. The aim of the experiment was to measure the accuracy of automatic quilt extraction from a video. The classification is evaluated using a custom annotated dataset.
2. **Frame resampling** experiment which proves that frames can be sampled in an optimal way, resulting in a lower error compared to a simple first frame alignment.

Table 1 The table contains important overall results of the three main experiments

Horizontal sequence extraction		
Analysis method	True [%]	False [%]
ORB_SLAM3	**79.7**	**20.3**
ORB	75.6	24.4
Optimal window positioning		
Scene	Squared error	
	First frame aligned	Optimized
Teddy	382.988	**251.594**
Hut	502.925	**300.505**
Class	115.437	**82.4649**
Pavilion	88.3094	**72.9947**
Focusing distance detection		
Method	Absolute error	
	DoG	Subtraction
Average	0.0475	**0.0441**
Min	0	**0**
Max	0.549	**0.549**

True values are considered $TN + TP$ and false $FN + FP$ according to the custom annotated dataset
The bold entries mark the best results

Fig. 15 The figure contains three previews of synthetic and three of real scenes from the original dataset

3. **Automatic focusing** of the resulting quilt. The experiment proves that the optimal focusing distance where most of the scene lies on the zero-parallax plane can be found automatically, which makes the resulting quilt ready to display.

Although LKG is used as the main experimental device in this paper, the resulting quilts were also tested on a N080 UHD Glasses-free 3D Network Multimedia Advertising Machine by Smarter Instruments and a Lume Pad 3D tablet by Leia Inc. The proposed method is suitable for other 3D display devices that use the same principle of horizontal multiview imaging.

OpenCV implementations of the Lucas-Kanade, Farneback, and ORB methods were used along with the original ORB_SLAM3 implementation for the quilt extraction. All experiments were executed on a machine equipped with Nvidia GeForce RTX 2070 GPU and Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz CPU, running Arch Linux.

4.1 Dataset

An original dataset is used in this paper and also published for other researches. The dataset consists of 182 FullHD, 25 fps, synthetic and handheld camera scenes encoded in H.265 format with standard YUV420 colorspace. The scenes are divided into 7 categories according to the dominant camera motion: chaotic, dolly, pan, pan and truck combined, pedestal, tilt, and truck. The scenes were manually annotated and sorted. The two categories which are considered to be suitable for 3D display are truck and truck with pan. The scenes from these categories were tested on LKG to make sure that they contain enough 3D information and are aligned enough with the optimal horizontal trajectory. The dataset is specifically designed to contain the ideal scenes for 3D displays along with other combinations of unwanted camera motions, which serves as the best benchmark for the given task. The duration of the videos is from 2 to 13 seconds. The videos were captured to cover all types of scenes with various depths, close and far objects of interest, etc. The synthetic shots were rendered in Blender with premade scenes from Blender Demo Files.[4] The synthetic videos are in two versions, with ideal smooth camera motion and with additional shaking and motion blur. The real-life videos were captured by Panasonic HC-VX980 Camcorder without any special equipment in both indoors and outdoors settings. Figure 15 shows examples of the scenes in the dataset.

4.2 Quilt extraction

The desired point in Fig. 16 marks the maximal vertical motion between two frames that is accepted as valid. The position was experimentally identified by tracking the amount of artifacts on LKG. An experiment was conducted in which identical frames with a simple pattern were vertically shifted with increasing offset. The first kind of artifact is caused by the mixing of the frames, which occurs throughout the displaying range to simulate the binocular parallax.

[4] blender.org/download/demo-files/

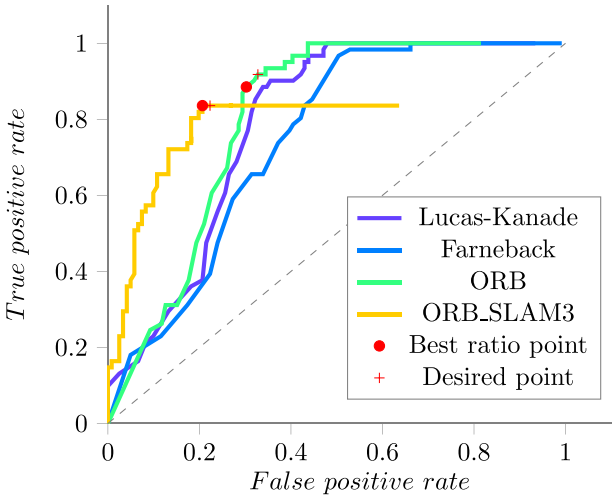


Fig. 16 The threshold parameter is the limit of vertical motion. The desired point marks 1px vertical motion tolerance which makes the artifacts visible. The best ratio of 0.9px tolerance was marked according to the used metrics

Figure 17 shows the difference between the vertically displaced quilt and the aligned one. These artifacts start to occur with displacement around 1.5px. The second type is visible in between two frame positions in the monocular motion parallax when watching the display from a close distance. This kind of artifact is more sensitive to displacement and was detected starting at 0.5px displacement. A value between these two, 1px, was selected as desired; see Fig. 18.

Three metrics (Youden index, closest to [0, 1] and maximum area) were used to identify the best ratio threshold in the Receiver Operating Characteristic (ROC) curves [54], which corresponds to a vertical motion tolerance of 0.9px. Figure 19 shows the distribution of negative and positive results among the categories in the dataset. Figure 20 shows the accuracy and precision results.

ORB-based feature detection method outperforms the optical flow approach. ORB_SLAM3 shows better results than all other approaches, but is limited by its inability to analyze all possible sequences. Most of the problematic sequences are true negatives, but some positive sequences were also refused. That is the reason why its ROC curve does not reach the right top corner as the rest.

An unknown focal length is assumed in the measurements, according to the use case, where the user can provide a random video without further information. The focal length was by Meshroom photogrammetry software, as it is a necessary parameter for ORB_SLAM3 method.

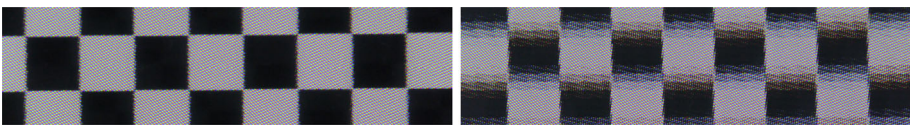


Fig. 17 The first picture shows a checkerboard quilt with views that are aligned in one horizontal line. The second picture shows the same quilt with vertical displacement between views where artifacts are visible

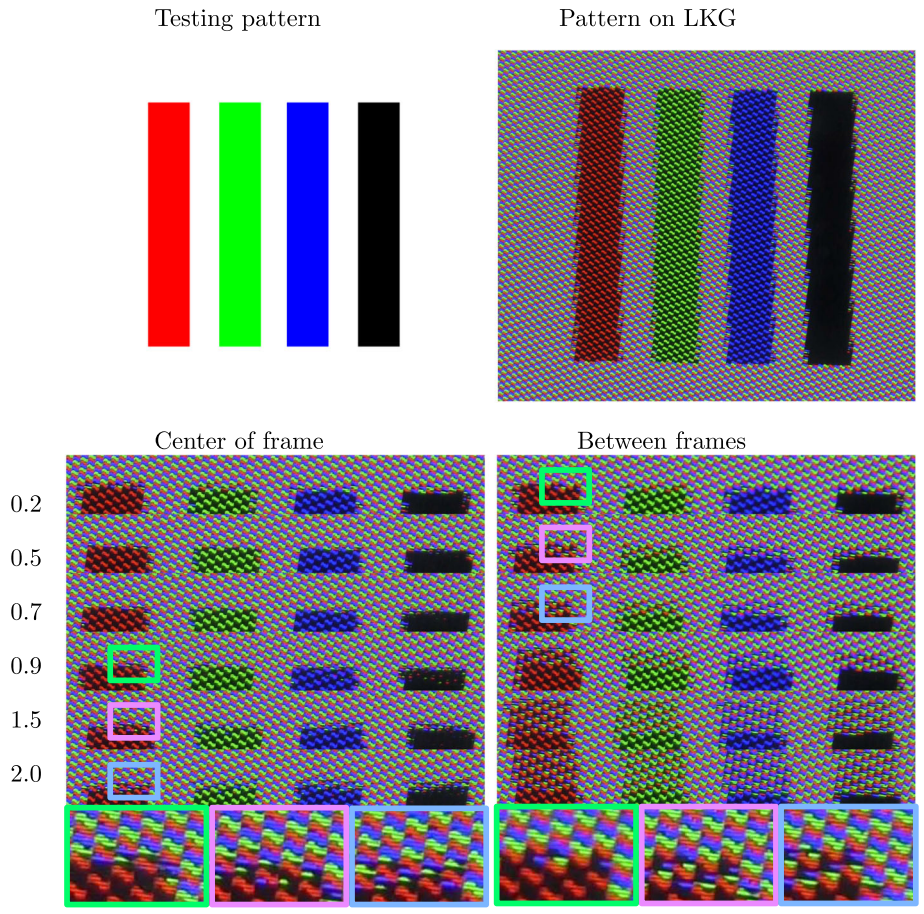


Fig. 18 A simple pattern was chosen to measure how vertical motion limit affects the quality of the result. The border of the pattern is getting distorted with the increasing vertical displacement (values on the left)

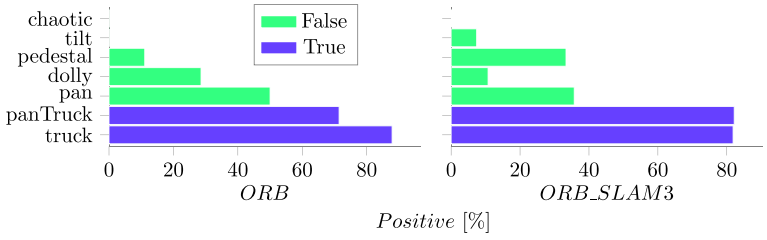


Fig. 19 The amount of positive ORB and ORB_SLAM3 results in each category of the dataset is depicted. Only pure truck or truck-and-pan combined camera motions are true positives

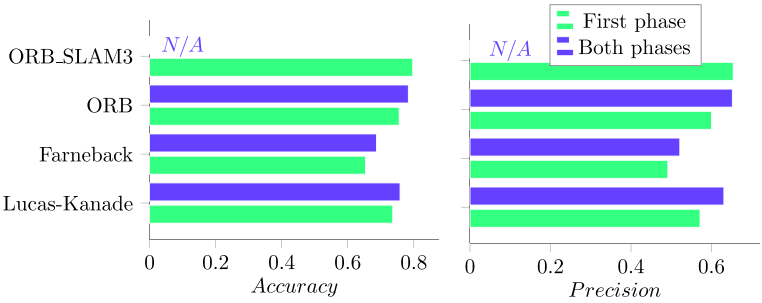


Fig. 20 The figure shows precision and accuracy values for all methods. The vertical motion limit is set to 1px. ORB_SLAM3 does both phases merged together

The results of the computational time measurements are presented in Fig. 21. ORB might be more versatile and robust, while ORB_SLAM3 is more accurate in classification. However, the efficiency of ORB_SLAM3, depends on the focal length estimation, which might be problematic in scenes shot by multiple different cameras or shots containing zooming.

4.3 Frame resampling

An experiment was performed to evaluate the Algorithm 3. *Teddy* sequence from the *Truck* category in the dataset was chosen because it is recorded with a handheld camera and thus is bound to have different spaces between the frames. Figure 22 shows how the algorithm selects the optimal offset within the defined range to minimize error.

4.4 Automatic focusing

The focusing metrics were evaluated on the quilts created from the truck sequences. The range and optimal value were manually annotated, evaluating the result on LKG. The absolute error of the estimated optimal value and the annotated one was calculated and normalized by the whole scanning range. SMAPE was used as a second evaluation metric. The results of the two proposed focusing metrics do not differ significantly, as shown in Fig. 23. The optimal focusing value is in all quilts very close to the metric minimum.

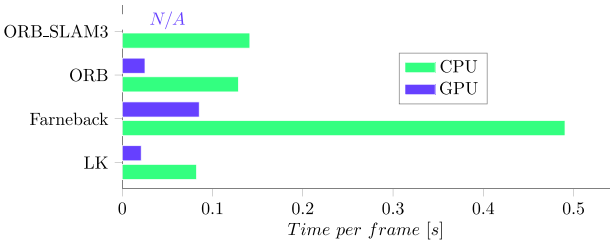


Fig. 21 The chart compares the computation times of both CPU and GPU implementations of the analysis methods. Times were measured and averaged over the whole dataset. The resolution of the processed sequences is 1920 × 1080

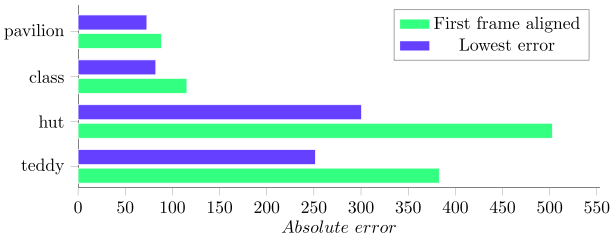


Fig. 22 The chart shows the results if the adjustment of the quilt window in *truck* scenes. The distance between the frames was twice the average distance in the sequence to avoid duplicates. The search range was set to one quarter of the distance

4.5 Comparison to existing methods

To prove the novelty of the proposal, the method was compared experimentally with similar methods. There are no existing alternative full quilt detection methods for 3D displays, so only the subtasks of the proposal were compared in standalone tests. The results show that the proposal cannot be replaced by existing standard approaches. Table 2 shows the explored works with description regarding their suitability for the 3D display scene detection.

4.5.1 Quilt extraction

3D reconstruction software such as Meshroom can be used to estimate the camera poses but it takes minutes to process one frame compared to near real-time processing proposed in this paper. It would also be problematic to process a video file not knowing how many frames are necessary to include in the computation. The proposal is focused on processing of the two subsequent frames, holding the context of the previous results. Two frames are in most cases not enough for a successful 3D reconstruction.

Panorama stitching software is another option as it detects the pan motion with possible shaking when the capturing device is handheld. Experiments with program Autostitch showed that pure truck motion shots can be properly detected as a suitable sequence and processed by the software. However, the combination of pan and truck does not yield acceptable results and this category of potentially suitable quilts would be rejected. Simple averaging of motion vectors from optical flow would also make the distinction between small amount of camera vertical shake, pan, and truck impossible.

Recent research in video analysis [55] usually focuses only on the general classification of the shot type such as *moving* or *static* and these approaches cannot be used for the quilt detection. CAMHID [24] method can be used to classify camera motion but the results presented by the authors mention only static, pan, tilt, and zoom categories without the necessary distinction between pan and truck which is necessary for the proposed method. Similar categories translation(pan and/or tilt), zoom, and static are defined in other works [29, 56–58] which is not sufficient for the scene detection. Other existing methods often focus on extraction or tracking of objects [60] and would be problematic to adjust for the given task.

Truck motion is distinguished from pan in a novel deep learning approach of camera motion detection for story and multimedia information convergence [59] where 8 frame intervals are processed. An advanced motion vector extractor preprocesses the shots, and ResNet is used to classify the resulting sequences.

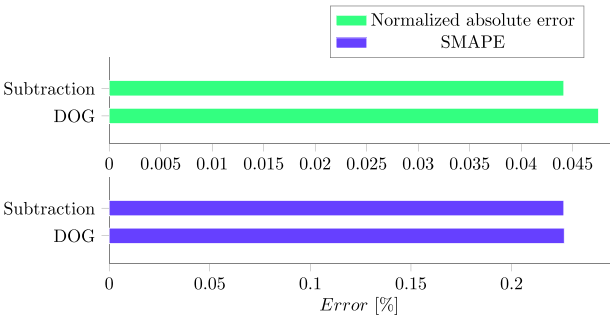


Fig. 23 Bars are showing the errors of both focus estimation metrics. The differences are not significant

The available implementation of the network was trained with the same data that the authors provided. The network was then used to measure its accuracy on the data tested in this paper to determine if the network can be used instead of the proposal. The network can classify a shot into pan, pedestal, tilt, truck, zoom in, and zoom out categories. This classification is not detailed enough for the quilt detection as truck with a small amount of pan is also acceptable and is not distinguishable in the network. The measurement was slightly adjusted in favor of the network approach, and any shot from the combined truck and pan category classified by the network as either standalone pan or truck is considered to be successful true positive, and false classification between other categories is not penalized. Table 3 shows that the proposal in this paper is more suitable for the task than the alternative deep learning network.

Table 2 The table sums up possible alternatives for the subtasks in the proposal and the reasons why they are not suitable

Method	Type	Usability
Meshroom	Scene detection	Camera estimation can fail due to lack of motion or frames that cannot be processed by pairs
Autostitch	Scene detection	Unable to detect pan and truck motion combination
avg optical flow	Scene detection	Unable to detect pan and truck motion combination
[55]	Scene detection	Cannot detect truck motion
[24]	Scene detection	Cannot detect truck motion
[29]	Scene detection	Cannot detect truck motion
[56]	Scene detection	Cannot detect truck motion
[57]	Scene detection	Cannot detect truck motion
[58]	Scene detection	Cannot detect truck motion
[59]	Scene detection	Can be used but is not as accurate as proposal
[60]	scene detection	Cannot directly extract the desired motion
[61]	Focusing	Can be used but is not as accurate as proposal
[62]	Focusing	Results are noisy and not accurate
PSNR	Focusing	Results are not accurate
SfM	Novel views	Contains artifacts and might fail without large number of frames

Novel views category means that the quilt can be rendered from the reconstructed 3D scene based on the frames

Table 3 The table compares alternative existing detection method with the proposal

Analysis method	True [%]	False [%]
Proposed	**79.7**	**20.3**
Deep [59]	68.1	46.9

True values are considered $TN + TP$ and false $FN + FP$ according to the custom annotated dataset
The bold entries mark the best results

The frames containing the motion vectors need to be downsampled to resolution of 600 × 300 compared to 1920 × 1080 in this paper's proposal. The time of the classification for one frame in the network is 0.006 s and the time of optical flow preprocessing is 0.1 s compared to 0.025 s for ORB and 0.14 s for ORB_SLAM3 in the proposal of this paper. The proposal is not a real-time solution and both compared methods can be labeled as the same speed category since the difference is not significant. If the same resolution frames were used as in the proposal, the deep classification time would be at least 0.07 s and the preprocessing would be 1.15 s due to the processing of 11.52× more data.

4.5.2 Focusing

To prove that a novel focusing metric is necessary for the quilt detection task, an attempt to identify focusing level with three other methods was conducted. First is a standard contrast-based method [61] with root mean square (RMS) contrast metric [63]. Second is a no-reference image quality metric LIQE [62] which should be able to detect various artifacts and defocused areas. Both metrics were used on the internal LKG format like in the proposal and on the blended views according to (8) with all weights set to the same constant, resulting in averaging of the refocused images. Third is a PSNR comparison of the refocused views according to the central one. Each view is shifted according to the focus value, compared with the central one with the PSNR metric, and the average of the PSNR values is taken as the metric. The results are shown in Fig. 24. The maximal values of both metrics on the internal LKG format are close to the optimum but not as close as proposed metrics in Fig. 14. The error of the proposed metrics on *Teddy* scene is only $15\,px$ in the focusing displacement while

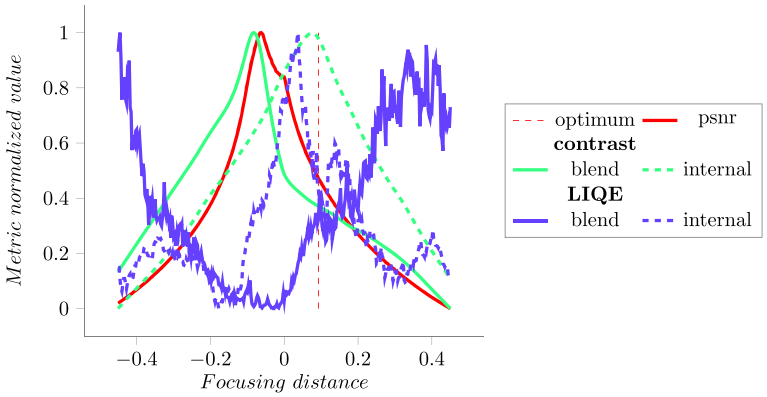


Fig. 24 Alternative existing metrics that can be used to measure focus in image were compared on *Teddy* scene. The values were transformed to the range of (0,1). The results show that these metrics are not suitable for 3D displays. The metrics were tested on internal LKG format and blended views

the contrast metric error is $36px$ which is more than $2\times$ higher than in the proposal. The charts of the proposed metrics (Fig. 14) also show steeper curve towards the minimum which identifies the area of potential other focusing distances that the user might want to choose. The slope of the alternative methods is either noisy or not that steep so such identification would be more problematic. The idea of evaluation of the internal format is proposed in this paper, so the internal versions of the metrics are only semi-alternative. The versions with blending, which is often used in light field rendering [64] to simulate focusing, and can be considered as an alternative approach, are not suitable at all. The conventional methods cannot be used in this case.

5 Conclusions

The proposed method can be used to automatically extract and process sequences from arbitrary video sources. This is the first study to explore the capture process and the final visual quality of the result on the state-of-the-art 3D display and fills the niche in this field. The proposed method can significantly reduce the time necessary for the user to choose suitable scenes from, e.g., a movie or a set of personal videos where many various scenes might be present. Manual selection would be time-consuming, and the best resulting viewing comfort would be hard to achieve by the user's trial-and-error approach. The measurements showed that the proposed method can produce artifact-free scenes in a shorter time and more efficiently than a human user. The proposal is robust and can produce suitable scenes from various kinds of real and synthetic videos. The method does not depend on the training process, and any video can be directly analyzed. The method was able to correctly identify almost 80% of the sequences in the created dataset. The main stages of camera motion classification and focus detection in the proposal were compared with state-of-the-art methods. The proposal proved to be the best choice for the defined problem of 3D display scene detection and cannot be replaced by conventional methods. Each stage of the proposal is accelerable on GPU.

The current reference implementation can extract the scenes from a FullHD video in about 7 fps including all subtasks. Note that the focusing and resampling algorithms can be accelerated on GPU and the whole proposal would most likely be able to process the inputs in real-time framerates. The optimal GPU acceleration of all tasks is out of scope of this paper. The computational complexity of the proposal is proportional to the resolution of the input image and the length of the input video. The optical flow computation matches the pixel blocks in pairs of subsequent frames, and a straightforward mixing and filtering of the images is used in the proposed focusing metrics. No preprocessing or training phases are necessary, and the input videos can be directly used with the proposed method.

Optical flow, feature matching, and SLAM-based methods were used in the video analysis phase and compared in terms of visual quality and performance. ORB feature detection and matching and ORB_SLAM3 methods showed the best results and are most suitable for the quilt extraction task. However, SLAM-based methods depend on focal length estimation and might not be able to process all possible scenes with the same robustness as the feature-matching ones. These results may be important for future studies in this field as the analysis method selection is not straightforward.

The results can be improved by scene depth and content analysis. An extensive study with a large number of human annotators might discover the relations between the camera motion properties (velocity, amount of rotation, etc.) and the visual experience of a human user. Certain camera trajectories might be more visually appealing to human perception than

others. This would also depend on the content of the scene and the distribution of its objects. However, such a study is beyond the scope of this paper and will be addressed in future work. Future work will also cover transformation methods to obtain an ideal quilt even from distorted data (non-uniform camera orientation and position between frames. The code and data are publicly available.[5]

Acknowledgements The authors would like to thank the reviewers for their valuable feedback.

Author Contributions All authors approved the final manuscript and contributed to the proposals.

Funding Open access publishing supported by the National Technical Library in Prague. This work was supported by the KDT JU project AIDOaRt, grant agreement No 101007350.

Availability of supporting data The code, dataset, and converted quilts are available and free to use.

Declarations

Competing interests All authors declare that they have no conflicts of interest.

References

1. Mehrabi M, Peek EM, Wuensche BC, Lutteroth C (2013) Making 3D work: a classification of visual depth cues, 3D display technologies and their applications. AUIC '13, pp 91–100. Australian Computer Society, Inc., AUS
2. El Jamiy F, Marsh R (2019) Survey on depth perception in head mounted displays: distance estimation in virtual reality, augmented reality, and mixed reality. IET Image Process 13(5):707–712. https://doi.org/10.1049/iet-ipr.2018.5920. https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/iet-ipr.2018.5920
3. Moorthy AK, Bovik AC (2013) A survey on 3D quality of experience and 3D quality assessment. In: Human vision and electronic imaging XVIII, vol 8651. International Society for Optics and Photonics, pp 86510. https://doi.org/10.1117/12.2008355
4. Jones A, McDowall I, Yamada H, Bolas M, Debevec P (2007) Rendering for an interactive 360° light field display. ACM Trans Graph 26(3):40. https://doi.org/10.1145/1276377.1276427
5. Keane SF, Jackson A, Smith GF, Tamblyn WJ, Silverman K (2019) Volumetric 3D display. Google Patents. US Patent 10401636
6. Hirsch M, Lanman D, Wetzstein G, Raskar R (2012) Tensor displays. In: ACM SIGGRAPH 2012 emerging technologies. ACM, pp 24. https://doi.org/10.1145/2185520.2185576
7. Balogh T, Kovacs PT, Barsi A (2007) Holovizio 3D display system. In: 2007 3DTV conference. pp 1–4. https://doi.org/10.1109/3DTV.2007.4379386
8. Frayne S, Lee SP, Fok TY, Hornstein A, Hwang A, Appelgate K (2018) Advanced retroreflecting aerial displays. Google Patents. US Patent App. 10/012841
9. Frayne S, Fok TY, Lee SP (2019) Superstereoscopic display with enhanced off-angle separation. Google Patents. US Patent 10298921

[5] fit.vutbr.cz/~ichlubna/research

10. Levoy M, Hanrahan P (1996) Light field rendering. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques. SIGGRAPH '96. Association for Computing Machinery, New York, USA, pp 31–42. https://doi.org/10.1145/237170.237199
11. Farneback G (2000) Fast and accurate motion estimation using orientation tensors and parametric motion models. In: Proceedings 15th international conference on pattern recognition. ICPR-2000, vol 1, pp 135–1391. https://doi.org/10.1109/ICPR.2000.905291
12. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th international joint conference on artificial intelligence - vol 2. IJCAI'81, pp 674–679. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. https://doi.org/10.5555/1623264.1623280
13. Arya S, Mount DM, Netanyahu NS, Silverman R, Wu AY (1998) An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J ACM (JACM) 45(6):891–923. https://doi.org/10.1145/293347.293348
14. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: Leonardis A, Bischof H, Pinz A (eds) Computer vision - ECCV 2006. Springer, Berlin, Heidelberg, pp 404–417
15. Alcantarilla PF, Bartoli A, Davison AJ (2012) Kaze features. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C (eds) Computer vision - ECCV 2012. Springer, Berlin, Heidelberg, pp 214–227
16. Rublee E, Rabaud V, Konolige K, Bradski G (2011) ORB: an efficient alternative to SIFT or SURF. In: 2011 International conference on computer vision, pp 2564–2571. Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICCV.2011.6126544
17. Singh PP, Ramchiary P, Bora JI, Bhuyan R, Prasad S (2023) An ensemble approach for moving vehicle detection and tracking by using Ni vision module. In: Gupta D, Bhurchandi K, Murala S, Raman B, Kumar S (eds) Computer vision and image processing. Springer, Cham, pp 712–721
18. Shi J, Tomasi (1994) Good features to track. In: 1994 Proceedings of IEEE conference on computer vision and pattern recognition, pp 593–600. https://doi.org/10.1109/CVPR.1994.323794
19. Younis O, Al-Nuaimy W, Rowe F, Alomari MH (2018) Real-time detection of wearable camera motion using optical flow. In: 2018 IEEE congress on evolutionary computation (CEC), pp 1–6. https://doi.org/10.1109/CEC.2018.8477783
20. Xiong W, Lee JC-M (1998) Efficient scene change detection and camera motion annotation for video classification. Comput Vis Image Underst 71(2):166–181. https://doi.org/10.1006/cviu.1998.0711
21. Lee S, Hayes MH (2002) Real-time camera motion classification for content-based indexing and retrieval using templates. In: 2002 IEEE international conference on acoustics, speech, and signal processing, vol 4. pp 3664–3667. https://doi.org/10.1109/ICASSP.2002.5745450
22. Park S-C, Lee H-S, Lee S-W (2004) Qualitative estimation of camera motion parameters from the linear composition of optical flow. Pattern Recognit 37(4):767–779. https://doi.org/10.1016/j.patcog.2003.07.012. Agent Based Computer Vision
23. Bouthemy P, Gelgon M, Ganansia F (1999) A unified approach to shot change detection and camera motion characterization. IEEE Trans Circuits Syst Video Technol 9(7):1030–1044. https://doi.org/10.1109/76.795057
24. Hasan MA, Xu M, He X, Xu C (2014) CAMHID: camera motion histogram descriptor and its application to cinematographic shot classification. IEEE Trans Circuits Syst Video Technol 24(10):1682–1695. https://doi.org/10.1109/TCSVT.2014.2345933
25. Almeida J, Minetto R, Almeida TA, da S Torres R, Leite NJ (2009) Robust estimation of camera motion using optical flow models. In: Advances in visual computing. Springer, Berlin, Heidelberg, pp 435–446
26. Weng Y, Jiang J (2011) Fast camera motion estimation in mpeg compressed domain. IEEE Trans Consum Electron 57:1329–1335. https://doi.org/10.1109/TCE.2011.6018891
27. Tiburzi F, Bescos J (2007) Camera motion analysis in on-line MPEG sequences. In: Eighth international workshop on image analysis for multimedia interactive services (WIAMIS '07), pp 42–42. https://doi.org/10.1109/WIAMIS.2007.27
28. Naito M, Matsumoto K, Hoashi K, Sugaya F (2006) Camera motion detection using video mosaicing. In: 2006 IEEE international conference on multimedia and expo, pp 1741–1744. https://doi.org/10.1109/ICME.2006.262887
29. Guironnet M, Pellerin D, Rombaut M (2006) Camera motion classification based on transferable belief model. In: 2006 14th European signal processing conference, pp 1–5
30. Odobez JM, Bouthemy P (1995) Robust multiresolution estimation of parametric motion models. J Vis Commun Image Represent 6(4):348–365. https://doi.org/10.1006/jvci.1995.1029
31. Duan LY, Jin JS, Tian Q, Xu CS (2006) Nonparametric motion characterization for robust classification of camera motion patterns. IEEE Trans Multimed 8(2):323–340. https://doi.org/10.1109/TMM.2005.864344

32. Liu L, Zhang R, Fan L (2010) Camera motion classification based on SVM. In: 2010 3rd International congress on image and signal processing, vol 1. pp 392–394. https://doi.org/10.1109/CISP.2010.5648012
33. Chen CLP, Bhumireddy C, Darvemula PK (2004) Camera motion classification using a genetic functional-link neural network. In: 2004 IEEE/RSJ international conference on intelligent robots and systems (IROS) (IEEE Cat. No.04CH37566), vol 3. pp 2343–23483. https://doi.org/10.1109/IROS.2004.1389759
34. Chang H-C, Lai S-H (2004) Robust camera motion estimation and classification for video analysis. Proc SPIE 5308. https://doi.org/10.1117/12.527698
35. Geng Y, Xu D, Feng S, Yuan J (2006) A robust and hierarchical approach for camera motion classification. In: Yeung D-Y, Kwok JT, Fred A, Roli F, Ridder D (eds) Structural, syntactic, and statistical pattern recognition. Springer, Berlin, Heidelberg, pp 340–348
36. Campos C, Elvira R, Rodríguez JJG, M Montiel JM, D Tardós J (2021) ORB-SLAM3: an accurate open-source library for visual, visual-inertial, and multimap SLAM. Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/TRO.2021.3075644
37. Mur-Artal R, Montiel JMM, Tardos JD (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot 31(5):1147–1163. https://doi.org/10.1109/tro.2015.2463671
38. Mur-Artal R, Tardós JD (2017) ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D cameras. IEEE Trans Robot 33(5):1255–1262. https://doi.org/10.1109/TRO.2017.2705103
39. Recasens D, Oswald MR, Pollefeys M, Civera J (2023) The drunkard's odometry: estimating camera motion in deforming scenes. arXiv:2306.16917
40. Zhang B, Wang Z, Tao D, Hua X-S, Feng DD (2015) Automatic preview frame selection for online videos. In: 2015 International conference on digital image computing: techniques and applications (DICTA), pp 1–6. https://doi.org/10.1109/DICTA.2015.7371237
41. Ren J, Shen X, Lin Z, Měch R (2020) Best frame selection in a short video. In: 2020 IEEE winter conference on applications of computer vision (WACV), pp 3201–3210. https://doi.org/10.1109/WACV45572.2020.9093615
42. Yan X, Gilani SZ, Feng M, Zhang L, Qin H, Mian A (2020) Self-supervised learning to detect key frames in videos. Sensors 20(23). https://doi.org/10.3390/s20236941
43. Müller T, Evans A, Schied C, Keller A (2022) Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans Graph 41(4):102–110215. https://doi.org/10.1145/3528223.3530127
44. Schönberger JL, Frahm J-M (2016) Structure-from-motion revisited. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 4104–4113. https://doi.org/10.1109/CVPR.2016.445
45. Reda F, Kontkanen J, Tabellion E, Sun D, Pantofaru C, Curless B (2022) Film: frame interpolation for large motion. In: Avidan S, Brostow G, Cissé M, Farinella GM, Hassner T (eds) Computer vision - ECCV 2022. Springer, Cham, pp 250–266
46. Reda F, Kontkanen J, Tabellion E, Sun D, Pantofaru C, Curless B (2022) Tensorflow 2 Implementation of "FILM: Frame Interpolation for Large Motion". GitHub
47. Yamanoue H, Okui M, Yuyama I (2000) A study on the relationship between shooting conditions and cardboard effect of stereoscopic images. IEEE Trans Circuits Syst Video Technol 10(3):411–416. https://doi.org/10.1109/76.836285
48. Shen S, Xing S, Sang X, Yan B, Chen Y (2023) Virtual stereo content rendering technology review for light-field display. Displays 76:102320. https://doi.org/10.1016/j.displa.2022.102320
49. Chlubna T, Milet T, Zemčík P (2023) How capturing camera trajectory distortion affects user experience on looking glass 3D display. Multimed Tools Appl - Major Revision. https://doi.org/10.1007/s11042-023-16350-5
50. Pech-Pacheco JL, Cristobal G, Chamorro-Martinez J, Fernandez-Valdivia J (2000) Diatom autofocusing in brightfield microscopy: a comparative study. In: Proceedings 15th international conference on pattern recognition. ICPR-2000, vol 3. pp 314–3173. https://doi.org/10.1109/ICPR.2000.903548
51. Pertuz S, Puig D, Garcia MA (2013) Analysis of focus measure operators for shape-from-focus. Pattern Recognit 46(5):1415–1432. https://doi.org/10.1016/j.patcog.2012.11.011
52. Kubota A, Takahashi K, Aizawa K, Chen T (2004) All-focused light field rendering. In: Eurographics workshop on rendering. The Eurographics Association. https://doi.org/10.2312/EGWR/EGSR04/235-242
53. Elder JH, Zucker SW (1998) Local scale control for edge detection and blur estimation. IEEE Trans Pattern Anal Mach Intell 20(7):699–716. https://doi.org/10.1109/34.689301
54. Dong T, Attwood K, Hutson A, Liu S, Tian L (2015) A new diagnostic accuracy measure and cut-point selection criterion. Stat Methods Med Res 26. https://doi.org/10.1177/0962280215611631
55. Rao A, Wang J, Xu L, Jiang X, Huang Q, Zhou B, Lin D (2020) A unified framework for shot type classification based on subject centric lens. In: Vedaldi A, Bischof H, Brox T, Frahm J-M (eds) Computer vision - ECCV 2020. Springer, Cham, pp 17–34

56. Delibasoglu I, Kosesoy I, Kotan M, Selamet F (2022) Motion detection in moving camera videos using background modeling and FlowNet. J Vis Commun Image Represent 88:103616. https://doi.org/10.1016/j.jvcir.2022.103616
57. Wang S, Jiang S, Huang Q, Gao W (2008) Shot classification for action movies based on motion characteristics. In: 2008 15th IEEE international conference on image processing. pp 2508–2511. https://doi.org/10.1109/ICIP.2008.4712303
58. Sandula P, Kolanu HR, Okade M (2022) CNN-based camera motion classification using HSI color model for compressed videos. Signal Image Video Process 1–8. https://doi.org/10.1007/s11760-021-01964-9
59. Bak H-Y, Park S-B (2023) Camera motion detection for story and multimedia information convergence. Pers Ubiquitous Comput 27(3):1221–1231. https://doi.org/10.1007/s00779-021-01585-6
60. Ye V, Pavlakos G, Malik J, Kanazawa A (2023) Decoupling human and camera motion from videos in the wild. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp 21222–21232
61. Shih L (2007) Autofocus survey: a comparison of algorithms. In: Martin RA, DiCarlo JM, Sampat N (eds) Digital Photography III, vol 6502. pp 65020. SPIE https://doi.org/10.1117/12.705386. International Society for Optics and Photonics
62. Zhang W, Zhai G, Wei Y, Yang X, Ma K (2023) Blind image quality assessment via vision-language correspondence: a multitask learning perspective. In: IEEE conference on computer vision and pattern recognition. https://doi.org/10.48550/arXiv.2303.14968
63. Kukkonen H, Rovamo J, Tiippana K, Näsänen R (1993) Michelson contrast, RMS contrast and energy of various spatial stimuli at threshold. Vis Res 33(10):1431–1436. https://doi.org/10.1016/0042-6989(93)90049-3
64. Chlubna T, Milet T, Zemčík P, Kula M (2023) Real-time light field video focusing and GPU accelerated streaming. J Signal Process Syst 1–17. https://doi.org/10.1007/s11265-023-01874-8