ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2025, Vol. 86, No. 12, pp. 1089–1117.

REVIEWS

Precise Industrial Photogrammetry Methods Survey

A. V. Gudym[*,a] **and A. P. Sokolov**[*,b]

[*] *Bauman Moscow State Technical University, Moscow, Russia*
e-mail: [a] *anton.v.gudym@yandex.ru,* [b] *alsokolo@bmstu.ru*

Received January 26, 2025
Revised July 4, 2025
Accepted July 8, 2025

Abstract—The survey is about modern and classical methods that forms SOTA photogrammetric pipeline for effectively constructing high-precision 3D point clouds and determining the position of objects in space from a photo or video signal. The work pays special attention to measurement error factors of the output 3D-reconstruction. Depending on the application, the reconstructed 3D-points may correspond to contrasting features of the object's texture, landscape, or special marks applied to the object's surface. After the features matching, the bundle adjustment optimization follows to restore the 3D coordinates of points in space. The survey provides detailed overview of the algorithms and convenient and practical formulations of various camera models and their distortions for bundle adjustment process. The experimental part demonstrates the highest level of accuracy achievable in practice using the methods considered. For close-range measurements repeatability, the proposed pipeline can outperfom professional photogrammetry solution.

Keywords: precise photogrammetry, visual odometry, computer vision, feature matching, bundle adjustment

DOI: 10.7868/S1608303225120011

1. INTRODUCTION

Photogrammetry—the science of measurements from photographs—has a long history and has been actively developed both in Russia and worldwide since the late 19th century. The term "*photogrammetry*" was introduced in 1867 by the German architect Albrecht Meydenbauer (1834–1921), who had previously published his photogrammetric method for measuring buildings in 1858. The mathematical foundations of photogrammetry were laid by the German mathematician S. Finsterwalder (Sebastian Finsterwalder, 1862–1951) [1]. Significant contributions to the development of projection mathematical models, still in use today, were made by the American researcher Duane Brown [2–4]. The founders of photogrammetry in Russia are considered to be the following outstanding scientists and engineers, specialists from the Department of Photogrammetry at the Moscow State University of Geodesy and Cartography, who made significant contributions to the technology's development: Professors N.M. Alexapolsky (1890–1942), F.V. Drobyshev [5], A.N. Lobanov, L.N. Vasilyev, V.B. Dubinovsky [6], and many others.

One of the first domestic stereophotogrammetric systems was the "Talka" photogrammetric system, developed by Soviet engineer D.V. Tyukavkin in the 1960s [7].

Historically, photogrammetry was applied in the fields of geodesy and cartography (Fig. 1). However, over the past two decades, considering the rapid growth of computational power in computer systems and the capabilities of image processing methods, including those using artificial intelligence (AI), photogrammetry as a measurement technology has become widely adopted in industry (*industrial* close-range photogrammetry). The use of photogrammetry methods primarily enables the recovery or reconstruction of a 3D surface model of an object of interest from a set of

Fig. 1. Left: Polygonal mesh (3D model) of landscape surface obtained through photogrammetry. Right: Original photo (crop) where circles indicate positions of SIFT feature points for matching and triangulation in the photogrammetric optimization problem.

photographic images. Furthermore, the technology has a broad range of applications in modern manufacturing processes [8–11]:

1) Contact technologies for measuring surface shape through probe/stylus positioning (tracking) in coordinate measuring machine (CMM) mode;
2) Non-contact measurement of surface shape during laser 3D scanner tracking;
3) Monitoring the position of large-sized parts in the aerospace industry;
4) Control of precision machining of parts in mechanical engineering;
5) Tracking of robot manipulator links and the attached tool [12–15].

Industrial photogrammetry involves the application of numerous algorithms from the fields of computer vision and image analysis (pattern recognition, 3D modeling). A renowned researcher in the field of feature recognition and matching on photographic images is Yu.V. Vizilter [16].

The subject of this work is the description of methods constituting the modern photogrammetric pipeline used in industrial photogrammetry. The methods considered in this work ensure efficient and high-precision recovery of 3D point coordinates (hereinafter referred to as 3D points), matched to various characteristic features identified on the surface of observed objects from photo or video signals. The constructed 3D point cloud is used for reconstructing the object's 3D surface model.

Known alternative technologies for 3D surface reconstruction include:

1) Projection 3D scanners with structured lighting or laser lines;
2) LIDAR scanners, measuring the time-of-flight delay of a laser beam;
3) Hybrid solutions – the use of structured lighting in combination with photogrammetry.

The advantage of alternative solutions is that they do not require the presence of contrasting textural features on the object's surface. At the same time, in terms of the ratio of measurement error magnitude to object size, photogrammetry may only be inferior to LIDAR systems [9] (given the very high cost of the latter), remaining an effective tool for measurements across a wide range of sizes. For example, the same set of tools can measure objects with overall dimensions from 0.1 to 10 m. The error can be 1:100,000 [17] or even 1:200,000 [8] of the object's overall size, i.e., ten or five micrometers per 1 meter, respectively.

Within a short period, photogrammetry has become a standard tool for efficient (accurate and fast) quality control, for example, in aerospace manufacturing. Photogrammetry is often used in conjunction with projection 3D scanners for device positioning and subsequent point cloud merging into a high-resolution output polygonal mesh.

Fig. 2. Left: Example of robot's tool positioning using images from multiple stationary cameras. Right: Circular black-and-white markers (targets) used as features for 3D reconstruction.

Photogrammetry is a key tool for precise video-based navigation and a source of geodetic measurements for unmanned aerial vehicles (UAVs) [14, 18–20]. An example of 3D landscape surface reconstruction using a photogrammetric pipeline [21] with UAV aerial photography data [22] is shown in Fig. 1. The right side of the image shows the positions of textural SIFT features, which are a set of 2D coordinates and a vector of real numbers (a descriptor) that characterizes the properties of the object's texture in a small neighborhood of the feature.

A known advantage of photogrammetry compared to alternative 3D reconstruction methods is its versatility and scalability – the same algorithms and camera models (Section 6.1) are applicable both for long-range geodetic measurements and in completely different close-range scenarios when the distance between the object and the camera does not exceed a few meters. In Fig. 2, the found 3D coordinates of markers on object surfaces are used to estimate the mutual position of bodies, for example, the position of a robot manipulator's end-effector relative to a reference object, whose position is also reconstructed based on the identified positions of it's markers.

This work pays special attention to the qualitative process of feature extraction using the example of artificial circular markers, as this has a decisive impact on the accuracy of the obtained 3D coordinates. The factors determining this process are also relevant for natural textural features in such well-known algorithms as: SIFT [23], SURF [24], SuperPoint [25], and others.

2. SURVEY STRUCTURE

Any photogrammetric system represents a hardware-software system. An example of the hardware part of a photogrammetric system is presented in Section 4. The software part implements algorithms designed to solve problems of the following two types, corresponding to the stages of the photogrammetric pipeline:

1) Processing of input images (Section 5):
   - Detection of the maximum number of feature points on the surface of the observed object (Step 1);
   - Construction of a descriptor invariant to affine transformations of the marker and optical distortions of the signal (Step 2);
   - Matching of descriptors – correspondences search (Step 3);
   - Filtering of the found matches (Step 4);

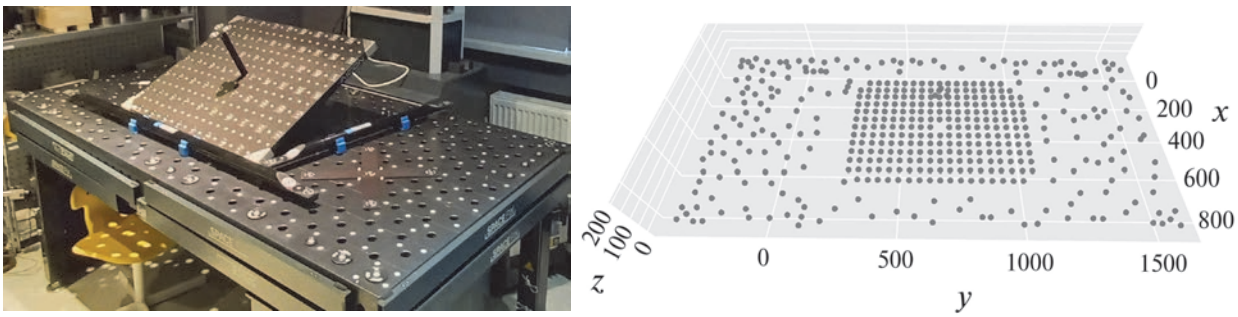

Fig. 3. Left: Test scene includes two measured objects with applied markers: steel welding table ∼2000 mm, carbon fiber calibration plate ∼800 mm, and two carbon fiber scale bars. Right: Photogrammetry result – high-precision reconstruction of markers as a 3D point cloud.

2) Solving the *bundle adjustment problem* (Section 6) for projection rays based on found correspondences:
   - Determination of initial approximations for parameters (coordinates of observed features, internal camera parameters and their positions) or simply auto-calibration (Step 5);
   - Solving the optimization problem (Step 6).

The data-processing step (Section 5) heavily depends on the application scenario. For example, in industrial measurements, simple circular or coded markers are applied to the object (Fig. 3). In aerial or satellite photography, natural contrasting features on the terrain are detected.

The optimization stage, unlike data processing, is sufficiently universal and applicable to almost any operational scenario.

Section 7 is devoted to the results of experiments with various factors of the bundle adjustment problem. It also demonstrates a high level of accuracy, comparable to professional photogrammetric products.

3. MATHEMATICAL NOTATION

Vector quantities are denoted in bold and represent column vectors of scalar quantities, e.g., $\mathbf{p} = [p_1, .., p_n]^{\mathrm{T}} \in \mathbb{R}^n$. Homogeneous coordinates, used in projective geometry and being an extension of Cartesian coordinates, are denoted by the superscript h, e.g.: $\mathbf{p}^h = [p_1^h, \ldots, p_n^h, 1/\lambda]^{\mathrm{T}} \in \mathbb{R}^{n+1}, \lambda \neq 0$. For homogeneous coordinates, the following holds: $\lambda p_i^h = p_i$, where $i \in [1..n]$, which is convenient for concise formulation of various matrix transformations.

Linear operators for coordinate system (CS) transformations typically contain notation indicating from where (subscript **bottom right**) and to where (superscript **top left**) the transition occurs. For example, to describe the position of the object CS (subscript o, $o$bject) relative to the camera CS (superscript c, $c$amera), the matrix ${}^cT_o \in \mathbf{SE3} \subset \mathbb{R}^{4\times4}, \det({}^cT_o) \neq 0$ (six degrees of freedom or *6DoF*) is used:

$$ {}^cT_o \cdot \mathbf{p}_o^h = \begin{bmatrix} {}^cR_o & {}^c\mathbf{t}_o \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} = \mathbf{p}_c^h, \tag{1} $$

where:
${}^c\mathbf{t}_o$ is the translation of the object's CS origin relative to the camera (3DoF),
$\mathbf{p}_o^h$ is the point position relative to the object,

$\mathbf{p}_c^h$ is the point position relative to the camera,
${}^cR_o \in \mathbf{SO3} \subset \mathbb{R}^{3\times3}$ is the rotation matrix (three rotation parameters are mapped to the matrix, e.g., by the Rodrigues formula [26] or using Euler angles),
SO3 is the special orthogonal group of rotations (${}^cR_o^{\mathrm{T}} = {}^cR_o^{-1}$, $\det({}^cR_o) = 1$, 3DoF),
SE3 is the Euclidean group of motion [27] or similarity transformation with unit scale, describing possible body movements in space.

The significance of the mathematical "*group*" concept for engineering applications: if it is required to calculate the change or increment of the camera position in the time interval from t_a to t_b, then multiplication by the inverse element[1] should be used:

$$ {}^bT_o \cdot {}^aT_o^{-1} \cdot \mathbf{p}_a^h = {}^bT_o \cdot {}^oT_a \cdot \mathbf{p}_a^h = {}^bT_a \cdot \mathbf{p}_a^h = \mathbf{p}_b^h, $$

where:
${}^aT_o$, ${}^bT_o$ are the positions of the object relative to the camera CS at time t_a and t_b, respectively,
${}^bT_a$ is the position of the camera at time t_a relative to the camera CS at time t_b,
$\mathbf{p}_a^h$, $\mathbf{p}_b^h$ are the positions of the 3D point relative to the camera at time t_a and t_b, respectively.

For better understanding of the formulations, one should assume the simultaneous existence of all CSs associated with the states of the moving body and consider time moments as identifiers of a particular CS. It also means that many definitions below are universal in the following – the indices a, b may correspond to either two moments in time or two different cameras at the same moment in time. $I \in \mathbf{SE3}$ is the identity matrix, describing still object relatively to the camera.

Careful notation for CS relationships in matrix indices is necessary for clarity in formulating the photogrammetric optimization problem, in particular, for describing the camera model (Subsection 6.1).

4. HARDWARE FOR INDUSTRIAL PHOTOGRAMMETRY

Figure 3 shows an experimental scene for a 3D point cloud reconstruction from circular markers (commonly called "*targets*") attached to measured objects. The scene also contains two scale bars – objects with calibrated distance between markers.[2] Scale bars allow determining absolute distance values between points identified in the scene. In the absence of scale bars, it is possible to reconstruct scene or object geometry up to scale from a set of images [26]. Scale bars are often made of carbon fiber, which provides a low coefficient of linear thermal expansion along its axis, as well as low weight and sufficient strength.

For best results, minimizing measurement noise and maximizing operational range, industrial systems often use special circular markers made of retroreflective material, e.g. "*retro-targets*". This property is extremely useful for increasing the contrast or sharpness of the marker's contour. When using a camera flash (to illuminate the object), the retroreflective material returns much more energy strictly in the direction of the source, compared to the inverse-square law of light intensity for ordinary materials.

To obtain high-quality images of key points on the surface of the measured object, a professional DSLR camera can be used. For best results, a monochrome high-contrast sensor and high-resolution optics are required. For example, the professional photogrammetric system "Hexagon DPA Pro"

[1] Matrix elements of **SE3** should not be multiplied by a scalar or subtracted – this is an invalid operation for a group element.
[2] The geometry of markers on the scale bar is measured or calibrated in advance under laboratory conditions.

from Hexagon AB includes a Canon EOS 5DS camera with a monochrome sensor $w \times h = 8700 \times 5800$ (50 Mp); lens f = 28 mm.[3]

To achieve good measurement results, specialized optics (e.g., with low distortion) are not mandatory [3]. The main factors are, naturally, those affecting signal quality (image contrast), as well as geometric stability (rigidity) of the camera-sensor optical system[4] during data acquisition [17, 28]. Geometric stability is influenced by the rigidity of the construction and lens mounting methods, mass-dimensional characteristics, flash mounting method, and the device's operating temperature regime. Thus, for high-precision measurements, careful selection of the equipment is required. Typical industrial photogrammetric system includes:

1) Calibrated carbon fiber scale bar;
2) High-resolution DSLR camera ensuring lens geometric stability;
3) Retroreflective adhesive-based markers;
4) Computational unit or PC implementing the photogrammetric pipeline stages described in the following sections.

5. FEATURE DETECTION AND MATCHING

Feature detection and matching is an extremely broad field of research [21, 29–33] that, among others, has numerous applications in photogrammetry, SfM (Structure from Motion [26]), SLAM[5], augmented reality, image retrieval, and contextual information analysis. This work focuses on the potential applications of these techniques for precise measurements.

The goal of this stage is to identify the maximum number of connections or correspondences between 3D points on the object (future measurements) and their observations – 2D points on images. Many works [25, 32, 34] allow finding correspondences between only image pairs, thus requiring additional grouping of points by their relation to a common surface point. In photogrammetry, it is the correspondence between a 2D point on an image (or projection) $(u, v) \in \mathbb{R}^2$ and a 3D point on the object $(x, y, z) \in \mathbb{R}^3$ that defines the future system of equations in the bundle adjustment problem (Section 6.5). In Fig. 4 correspondences are found by matching binary coded markers (color identify group or unique object point).

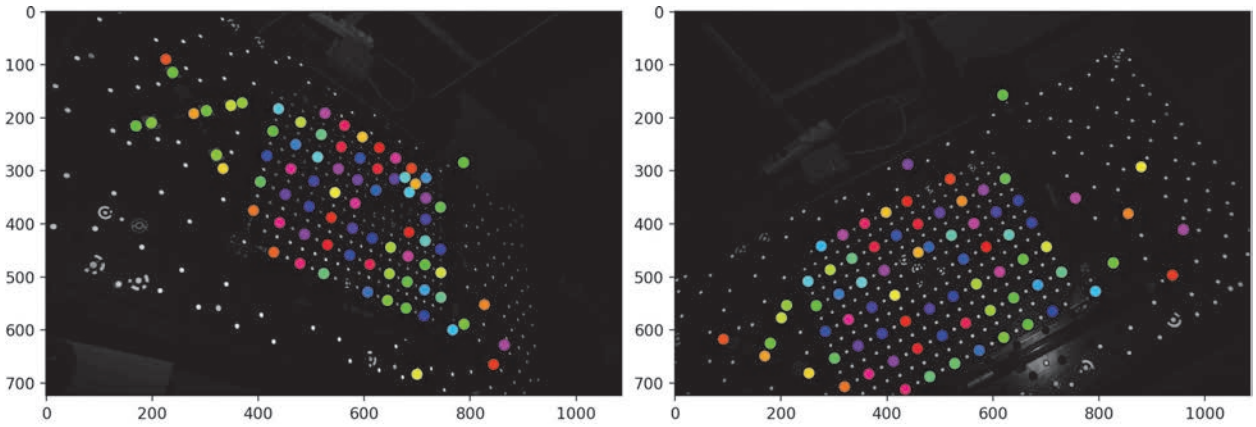

Fig. 4. Original photos of the test scene with 2D points matching results by binary descriptor. Color denotes the group or unique 3D point on the object.

[3] Field of view angles are $\Delta\varphi_{hor} = 64°$, $\Delta\varphi_{ver} = 45°$, $\Delta\varphi_{diag} = 73°$; maximum angular resolution $\approx 2\frac{\Delta\varphi_{diag}}{\sqrt{w^2+h^2}} = 0.014°$. Angular resolution is convenient for comparing sensors of different resolutions or technologies, e.g., SfM and LIDAR.

[4] Methods of dynamic sensor stabilization may be harmful in this context.

[5] SLAM (Simultaneous Localization and Mapping) – a navigation method in mobile autonomous systems.

5.1. Feature Detector

The first step of the photogrammetric pipeline (page 1091) is the extraction or detection of the maximum number N^a_{pts} of feature points $\mathcal{P}_a = \{\mathbf{p}^k_a\}_{k=1}^{N^a_{pts}}$, $\mathbf{p}^k_a = [u^k_a, v^k_a]^{\mathrm{T}} \in \mathbb{R}^2$ on the surface of the measured object, observed in image $a \in [1 \dots N_{im}]$ from a camera.[6] This step is performed by a detection algorithm or simply a detector. In general-purpose photogrammetry [21], without artificial markers, various natural features—points, circles, corners, crosses, or similar structures—are detected on the object texture using Harris [35], GFTT [36], FAST [37] detectors. Some detectors extract both the point and generalized information about the texture of the neighborhood in the form of a multidimensional descriptor vector. The point coordinates together with the descriptor are often called a *feature*. This task is solved by classical approaches SIFT [23], SURF [24] and based on machine learning (ML) SuperPoint [25], DISK [38]. In this case, special areas most suitable for subsequent determination of a stable descriptor are identified. Typically, natural features lack sufficient contrast or size for precise 2D localization and subsequent 3D reconstruction. This can be seen from the difference between the observed 2D coordinates of features and the projections (on the image) of the corresponding 3D object points. This difference is commonly called the reprojection error [26, 39]. For a high-precision measurement task, one standard deviation of the reprojection error does not exceed 0.1 pixels (Fig. 11), hence the feature must be localized even more accurately. The noise of detected coordinates for natural texture features often exceeds one pixel [25, 32, 40]. With careful calibration in [41], but using corners, the reprojection error is still ~ 0.33 pixels. Using ML for texture feature detection in [42] one observes ~ 0.5 pixels. Therefore, for industrial photogrammetric measurements, artificial black-and-white markers of circular and, less frequently, square shape are often applied to the measured object's surface (for further detection).

A circular marker is preferable to corners and similar structures. To understand the reason, let's consider the main factors affecting localization stability, dividing them into two categories for clarity:

1) **Geometric** factors determine observed feature's shape (effects of perspective projection, lens distortion, or object's shape), Fig. 5;
2) **Optical** factors affect feature's contrast, signal-to-noise ratio[7] (illumination level and surface reflectance, focusing, sensor resolution, photon leakage), Fig. 6.

Geometric factors of detection. All texture feature detectors mentioned earlier and similar ones rely on surface's continuity in the neighborhood of the point of interest. That is, the smoother, closer to planar the surface, the more stable and reproducible on other images the found feature will be. For example, the SIFT detector "suppresses" points on lines [23] in order to avoid object boundaries and possible inclusion of background information.

The assumption that the neighborhood of a feature point is a small flat surface area in 3D means that the mapping "3D surface – 2D image"[8] can be performed via an affine transformation of the observed area:

$$ {}^{o}A_a = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix}, \quad \det({}^{o}A_a) \neq 0. $$

It includes six degrees of freedom [26]: 2D translation, 2D scale, 1D diagonal shear, and 1D rotation. In a more general case (without the assumption of a small area size), a projective or homographic

[6] Several N_{im} stationary cameras or one mobile camera in N_{im} positions, depending on the application.
[7] Signal usually refers to pixel intensity or color.
[8] Similar to ray tracing in computer graphics – finding the intersection of a ray "from a pixel" and a 3D plane.

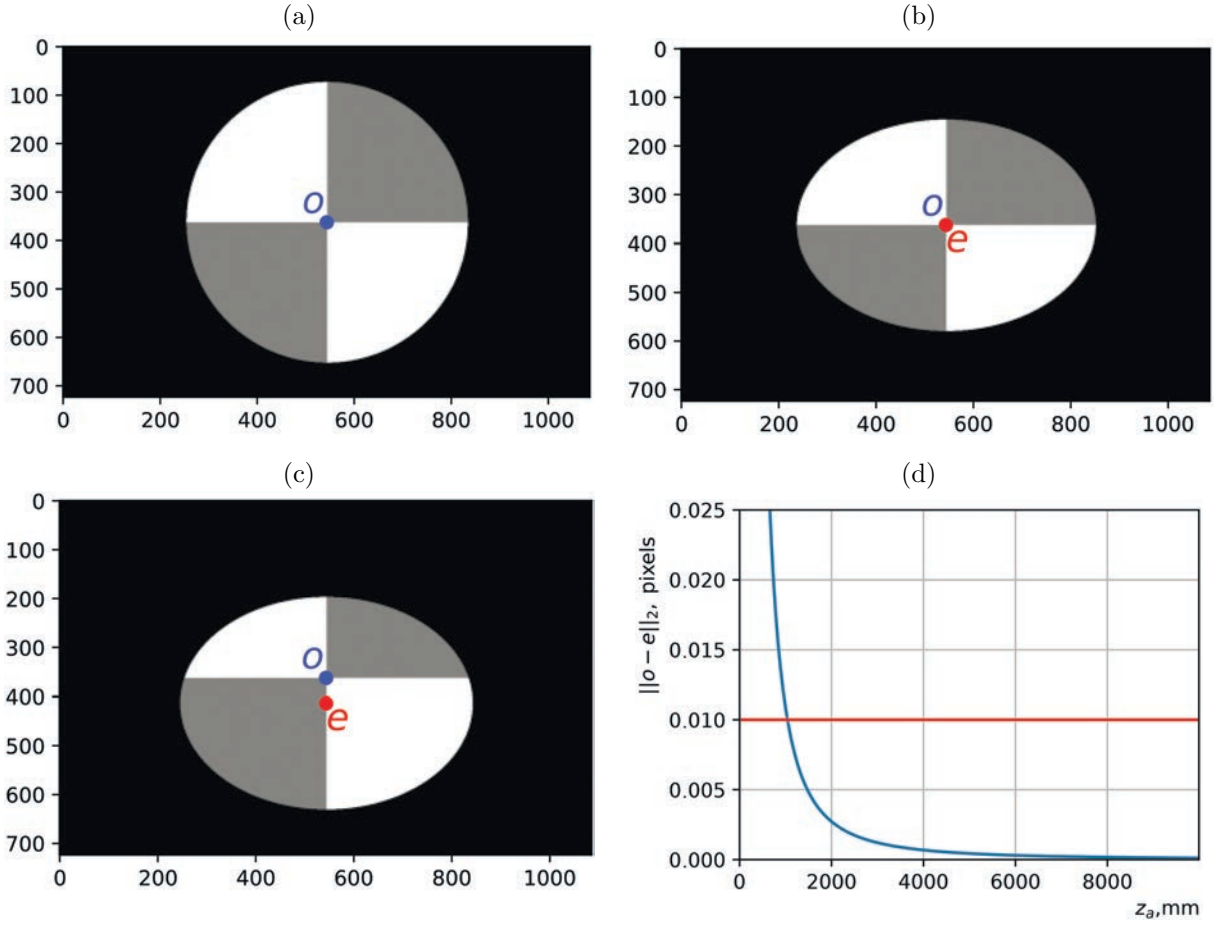

Fig. 5. (a) Circular marker on object, (b) affine projection of marker on screen, (c) perspective projection of marker, (d) dependence of perspective effect $||\mathbf{o} - \mathbf{e}||_2$ on distance to marker z_a.

transformation takes place:

$$ {}^{o}H_a = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}, \quad \det({}^{o}H_a) \neq 0. $$

In computer graphics, affine transformation ${}^{o}A_a$ corresponds to orthogonal projection, and ${}^{o}H_a$ to perspective projection. Transformation ${}^{o}A_a$ transfers a feature from a flat object directly to image a, preserving line parallelism and distance proportions along a line, unlike the more general transformation ${}^{o}H_a$. The model of these transformations in homogeneous coordinates is:

$$ \lambda \begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ 1 \end{bmatrix}, \tag{2} $$

$a, b \in [1..N_{im}]$ – image indices, u_a, v_a – point coordinates on the image (the feature itself or a point in its 3D neighborhood, point index k omitted for brevity), x_o, y_o – point coordinates on the object in the coordinate system of the feature's 3D plane ($z_o = 0$), λ – non-zero scale factor, easily eliminated from the system of linear equations (2) by substitution if ${}^{o}H_a$ needs to be found.

Consider an example. Suppose a circular marker with radius $\mathcal{R} = 5$ mm is depicted on a flat object (Fig. 5a). Under affine projection of the marker onto the screen (with observation angle

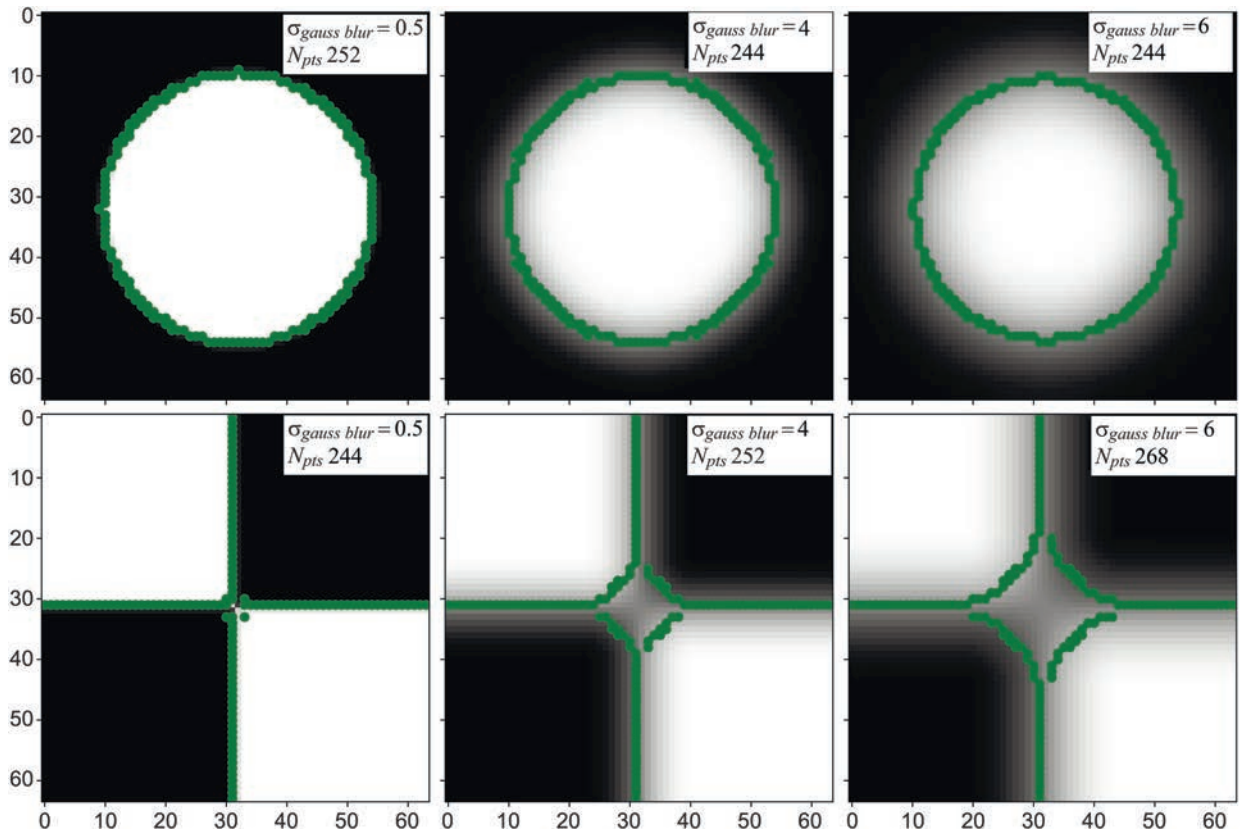

Fig. 6. Effect of sharpness reduction (left to right) for circle and square: unlike circular marker (top row), square structure (bottom row) is significantly distorted. Dots indicate feature contour – pixels with maximum signal gradient amplitude.

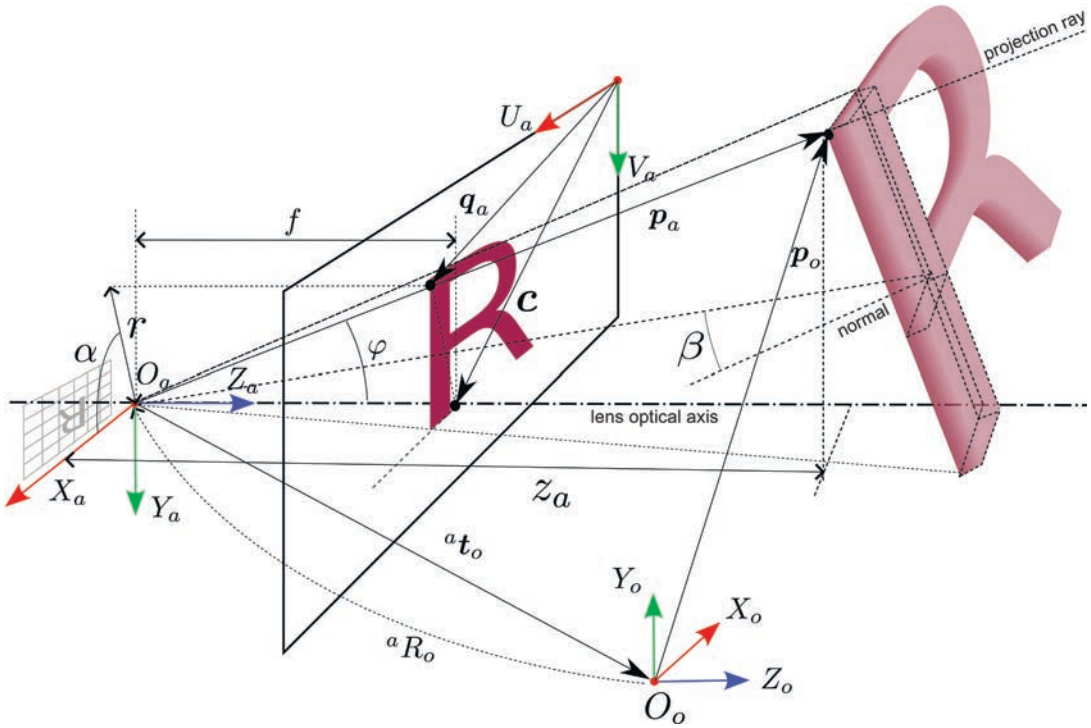

Fig. 7. Perspective or rectilinear camera model (6): 3D object relative to different coordinate systems and its projection onto screen or sensor (detailed description in text).

$\beta = 45°$, explanation in Fig. 7), the circle center $\mathbf{o} \in \mathbb{R}^2$ will perfectly correspond to the center $\mathbf{e} \in \mathbb{R}^2$ of the observed ellipse (Fig. 5b). In reality, at a very close surface distance $z_a = 15$ mm, strong projective distortion is observed (Fig. 5c), the magnitude of which depending on z_a is shown in Fig. 5d. Experimental conditions are presented in Section 7, using a Canon EOS 5DS camera.

The dependence shown in Fig. 5d is valid for a marker with radius $\mathcal{R} = 5$ mm and $\beta = 45°$. To build a similar dependence for a marker of a different radius, e.g., $\mathcal{R}' = 10$ mm, at the same observation angle β, the abscissa on the presented graph (Fig. 5c) must be multiplied (scaled) by the coefficient $\frac{\mathcal{R}'}{\mathcal{R}}$.

The difference between operators ${}^{o}A_a$ and ${}^{o}H_a$ is clearly shown in Fig. 5. When the distance to the surface is small and the area for extracting the feature center is large, an increasingly significant deviation is observed between the detected point $\mathbf{e}$ and the projection $\mathbf{o}$ of the estimated 3D point on the object. When the deviation exceeds 1:10 of the standard reprojection error in the bundle adjustment problem (Fig. 5d, red line on the graph), then not only the 3D object point but also the parameters of the surface patch orientation in space need to be optimized (Section 6).

In some applications, projective shape distortions allow reconstruction of object geometry from two or more coplanar circles [43]. Besides projective distortions described above, the deviation $||\mathbf{o} - \mathbf{e}||_2$ can be caused by much more complex nonlinear effects, e.g., camera lens distortion [44] or surface curvature. If the influence of these effects on the local feature geometry is significant, for an accurate bundle adjustment solution, one can directly minimize the residual of intensity values in each image pixel [26, 39]. This approach requires substantially more computational resources. For example, instead of two residual equations for the center of each marker (bundle adjustment problem, Section 6), there will be $O(\pi\mathcal{R}^2)$ equations[9] of color intensity differences in the feature pixels. Moreover, the number of parameters increases: for each optimized 3D point, a 3D normal[10] is added, i.e., at least five parameters per point instead of three. Such a solution is justified when working with a wide-angle camera model That significantly different from the common perspective model (Section 6.1).

Optical factors of detection. Extracting a feature from a large neighborhood creates difficulties described earlier. It seems that a simple solution would be to reduce the area size. For example, using a feature type like a cross, corner, or other line intersections on the object [45]. But under strong optical blur or other contrast loss (e.g., during noise suppression), the structure of such a feature can quickly degrade (Fig. 6).

The issue is that the high-frequency component of the signal,[11] necessary for depicting sharp edges of a corner, is gradually lost. Thus, a circle with a smooth contour structure remains the most universal marker for the most precise measurements (even if only the contour, not the center, is needed). Circular markers can be detected in real time using the following sequence of steps:

1) Coarse localization: "*blobs*" detector [35, 36];
2) Contour extraction [46, 47] or gradient extraction in the ellipse neighborhood [48];
3) Reliable and precise localization: quadratic function approximation of the ellipse as the locus of contour points or gradient field.

It is worth noting that subpixel refinement during contour extraction may not be required. The discretization noise of contour points $e_{edge} \sim \mathcal{U}(-0.5, 0.5)$, $\sigma_{edge} = \frac{1}{\sqrt{12}}$ (in pixels) is significantly averaged, so the noise in determining the marker center (mean of the random variable) is much smaller and amounts to $\sigma_{center} = \frac{\sigma_{edge}}{\sqrt{N_{pts}^{edge}}}$, N_{pts}^{edge} – number of points used to determine the center.

[9] Two equations per pixel, the number of which is proportional to the area of the observed marker.
[10] The orientation of a unit normal is defined by two angles.
[11] In computer vision, spatial signal frequency is typically discussed, unlike time function in electronics.

For example, for a circle with $\mathcal{R} = 25$ pixels, the discretization noise level of the found center coordinates σ_{center} will be $1/\sqrt{12 \cdot 2 \cdot \pi \cdot 25} = 0.02$ pixels.

The number of features does not compensate for their low quality – a general rule for precise photogrammetry. Since the parametric model of the ray (or camera) can be a high-order nonlinear function (Section 6), especially at the edges of the field of view, this inevitably leads to the overfitting.

5.2. Feature Descriptor

After key points detection a description for each point's texture neighborhood $\mathcal{D}_a = \{\mathbf{d}_a^k\}_{k=1}^{N_{pts}^a}$ is constructed (descriptor estimation, **second step** of the photogrammetric pipeline, page 1091). The key property of a descriptor is its stability or invariance to geometric transformations (due to camera or object motion) and to optical factors, e.g., due to illumination changes, defocusing, or sensor noise. That is, under the most diverse image acquisition conditions, the same object point should have identical, yet unique, descriptor. Descriptor invariance, in general, is unattainable, hence there is a wide variety of approaches effective for different scenarios.

For example, in the SfM task for large objects, affine distortions practically do not change angles in the feature structure, but rotations and scale changes are possible, similar to observations of celestial bodies [49]. In close-range photogrammetry for quality control of medium and small objects, on the contrary, noticeable projective distortions of features can occur (Fig. 5). Thus, the effectiveness of descriptors can differ significantly under different conditions [30]. This is important to consider when working with ML-based approaches [25, 31, 32, 34, 38] (when preparing the training database), or when selecting a suitable descriptor based on comparative review results [29].

The descriptor of natural texture features can be a vector of real numbers: $\mathbf{d} \in \mathbb{R}^n$ for SIFT [23], SURF [24], SuperPoint [25], DISK [38] or a binary vector $\mathbf{d} \in \{0,1\}^n$ for BRIEF [50], ORB [51], AKAZE [52]. Also, most artificial coded markers represent a binary descriptor [53–56]: QR-code, ARTag, AprilTag, ArUco, CCTag, Schneider's Coded Target (SCT), etc. SCT coded markers [53] (Fig. 3) are used in the experimental part. Markers with concentric circles (CCTag, SCT, etc.) are often used in industrial photogrammetry. "*Decoding*" markers or determining descriptor's binary sequence $\mathbf{d} \in \{0,1\}^n$ is significantly simplified by the ability to compensate for affine distortions of the feature: knowing the center, rotation angle, and magnitude of the principal axes, the five parameters of the affine transformation can be computed.

Machine learning can be effectively applied for decoding of specific key points and even arbitrary image regions [57]. Architectures based on convolutional networks can be used, e.g., Resnet [58] or Unet [59] encoding parts. As an output layer, a fully-connected bitwise classifier can be used, e.g., 12 output neurons with a sigmoid activation function for a 12-bit binary marker. By batching pixels from marker neighborhoods from several images, the decoding task can be efficiently solved on a GPU in real time.

5.3. Descriptor Matching

At this step, the found descriptor vectors of features from each image $\mathcal{D}_a = \{\mathbf{d}_a^k\}_{k=1}^{N_{pts}^a}$, $\mathcal{D}_b = \{\mathbf{d}_b^k\}_{k=1}^{N_{pts}^b}$, $a, b \in [1..N_{im}]$ are matched against each other (**third step** of the photogrammetric pipeline, page 1091), forming a set of corresponding indices: $\mathcal{M}_{a\to b} = \{(i^k, j^k) \mid i^k \in [1..N_{pts}^a], j^k \in [1..N_{pts}^b]\}_{k=1}^{N_{pairs}^{a\to b}}$. For each descriptor from the source set $\mathcal{D}_a$ with index $i \in [1..N_{pts}^a]$, a search for the nearest neighbor with index j^* in set $\mathcal{D}_b$ is performed ($l2$-norm of the difference as an example):

$$j^* = \underset{j \in [1..N_{pts}^b]}{\operatorname{argmin}} \|\mathbf{d}_a^i - \mathbf{d}_b^j\|_2. \tag{3}$$

In the simplest case, two sets $\mathcal{M}_{a\to b}$, $\mathcal{M}_{a\leftarrow b}$ are built (arrow indicates search direction) with subsequent filtering presented in Section 5.4. In the general case, optimal matching of more than two discrete descriptor sets belongs to the NP-complete transportation problem. Matching methods using machine learning [31, 32, 34, 60] effectively approximate the discrete search problem by combining local and global properties of descriptors.

If a pixel on the contour is used as a feature instead of a distinctive point (e.g., marker center), strict correspondence of 2D points between an image pair may not exist. In this case, the coordinates of the corresponding feature must be interpolated (assuming local smoothness or planarity of the surface).

As a distance function, depending on the nature of the vector, the $l2$-norm (Euclidean distance) [23–25, 38, 61] is often used, or for binary descriptors [50–52] – the number of identical bits (Hamming distance). In [61], an effective transformation of the distance function for the SIFT descriptor $\mathbf{d} = [d^1, \ldots, d^{128}]^{\mathrm{T}}$ is proposed, which significantly increases the probability of finding correct connections between images. Originally, the distance between descriptors is defined via the Euclidean norm, i.e., $\mathrm{dist}_E(\mathbf{d}_a, \mathbf{d}_b) = \|\mathbf{d}_a - \mathbf{d}_b\|_2$. However, considering that $\|\mathbf{d}_a\|_2 = \|\mathbf{d}_b\|_2 = 1$, the following holds:

$$\|\mathbf{d}_a - \mathbf{d}_b\|_2 = \sqrt{2 - 2\sum_{l=1}^{128} d_a^l \; d_b^l}.$$

The basis of SIFT is a frequency histogram or distribution function of some characteristic of the marker neighborhood. Histograms are also used when matching 3D features of point clouds, e.g., FPFH [62]. For comparing distributions similarity, instead of the $l2$-norm, it is better to use the Hellinger's f-divergence [61]. As a result of $l1$-normalization and element-wise square root, the distance between descriptors can be reduced to the corresponding form:

$$\mathrm{dist}_H(\mathbf{d}_a, \mathbf{d}_b) = \sqrt{2 - 2\sum_{l=1}^{128} \sqrt{c_a \; c_b \; d_a^l \; d_b^l}},$$

where c_a, c_b are normalization coefficients. Thus, by varying the distance function, the number of correct correspondences can be significantly increased.

Since the number of images N_{im} and the average number of found features N_{pts} per image can be large, the overall complexity of searching for similar descriptors among all images often becomes unacceptably high $O(N_{im}^2 \cdot N_{pts}^2)$ – as dozens of views (N_{im}) of a specific 3D point are needed for reliable results, and their total number of key points (N_{pts}) can easily exceed 1000. In this case, accelerating structures in the form of random trees with approximate nearest neighbor search [63][12] are primarily applied. For binary descriptors, the distance function differs from Euclidean, so methods based on hash functions [65] are used. Approximate search may allow a significant number of errors, but they can be filtered at the next stage.

Since images may not view the same object region, BoW techniques [61, 66] (abbreviation for "Bag-of-Words") are used to accelerate the selection of suitable pairs, allowing quick exclusion of non-overlapping frames and the need to match all descriptors for an image pair against each other.

5.4. Filtering of Found Correspondences

The results of the correspondence search need to be filtered (**fourth step** of the photogrammetric pipeline, page 1091) regardless of the matching methodology to minimize the number of errors or outliers.

[12] Ordinary kd-trees are inefficient – a problem known as the "curse of dimensionality," where sequential search turns out to be faster than tree traversal due to information distribution in multidimensional data structures [64].

From (3) it follows that the search direction matters and the sets $\mathcal{M}_{a\to b}$ and $\mathcal{M}_{a\leftarrow b}$ may not be identical. This is the basis for the "mutual correspondence" filter [30–32], i.e., mutually nearest descriptors or the intersection of pairs from sets $\mathcal{M}_{a\to b}$ and $\mathcal{M}_{a\leftarrow b}$ are used, while the rest are discarded. Another popular filter is based on descriptor "uniqueness" [23] – it discards match if one of the descriptors (from original image) is relatively close to multiple others (on a different image). Another common method is using repeatedly occurring descriptors forming a sequence (e.g., "tracks" in video data analysis).

These and similar heuristics by themselves are not very effective and can remove a large number of correct correspondences. This is because the local texture properties of the object, represented by descriptors, vary greatly even within a single image. Essentially, the task of such filters is to screen out the coarsest matching errors and accelerate subsequent steps.

The most effective filtering methods rely on global context [21, 30, 42], e.g., checking how well the found correspondences satisfy the geometric properties of space and the perspective camera model (Section 6.1). If the camera model significantly differs from perspective, e.g., for wide-angle lenses, or has large distortion, then fast methods considering these distortions are required [40, 67].

First, the most universal and widely used criterion for geometric consistency is epipolar geometry – all correspondences on an image pair must satisfy the following epipolar line equation:

$$\begin{bmatrix} u_a & v_a & 1 \end{bmatrix} {}^aF_b \begin{bmatrix} u_b \\ v_b \\ 1 \end{bmatrix} = 0, \tag{4}$$

u_a, v_a, u_b, v_b – coordinates of two projections of the same 3D point on images a and b respectively, ${}^aF_b \subset \mathbb{R}^{3\times 3}$ – the well-known fundamental matrix, $\operatorname{rank}({}^aF_b) = 2$. The geometric meaning of equation (4) is that matrix ${}^aF_b$ establishes a point-line correspondence between an image pair: $\begin{bmatrix} u_a & v_a & 1 \end{bmatrix} {}^aF_b$ – parameters of a line on image b, while ${}^aF_b \begin{bmatrix} u_b \\ v_b \\ 1 \end{bmatrix}$ – a line on image a. Substituting eight pairs of corresponding points into (4), a system of linear algebraic equations [68] is built for the unknown matrix ${}^aF_b$. Considering that $\operatorname{rank}({}^aF_b) = 2$, using seven point pairs, a system of nonlinear equations for finding ${}^aF_b$ can be built [26]. Thus, by repeatedly solving these equations for small groups of corresponding points, in statistical methods like RANSAC [69] or PROSAC [70], a significant portion of matching errors can be filtered out.

Second, filtering reliability can be increased if it is known that the points lie on a plane or depth variations on the object surface are significantly smaller than its dimensions. In this case, only 4 correspondences are required to define a 2D point transformation model when using methods like RANSAC or DEGENSAC [71]:

$$\lambda \begin{bmatrix} u_a \\ v_a \\ 1 \end{bmatrix} = {}^aH_b \begin{bmatrix} u_b \\ v_b \\ 1 \end{bmatrix}, \tag{5}$$

where ${}^aH_b \subset \mathbb{R}^{3\times 3}$ is the projective transformation matrix or homography between corresponding projections of a common 3D point, $\det({}^aH_b) \neq 0$.

Machine learning methods allow extraction of distinctive points and building local descriptors [25, 38]. In [31], texture features are extracted on practically homogeneous areas of natural texture. In [31, 32, 34], features are supplemented with contextual information from some image area calculated using a transformer-based neural network architecture or similar modifications for computational efficiency. As a result of pairwise matching, a feature correspondence probability matrix is obtained: $P^{a,b} \in \mathbb{R}^{N^a_{pts}\times N^b_{pts}}$, $\sum_{i=1}^{N^a_{pts}} P^{a,b}_{ij} \leqslant 1$, $\sum_{j=1}^{N^b_{pts}} P^{a,b}_{ij} \leqslant 1$. Thus, correspondences with

the required reliability can be selected, excluding points invisible from both viewpoints. In the context of precise photogrammetry, the presented ML-based solutions are excellent for finding a reliable initial approximation and constructing approximate camera positions and 3D point clouds. But unfortunately, they have low feature localization accuracy – the reprojection error in various tasks often exceeds one pixel, and the angular error in position determination is over 5°; often the training dataset is built on flat scene areas, i.e., heavily relies on (5), and matching results still require filtering [30, 32].

Practically regardless of design, the local descriptor in photogrammetry is used at the preliminary stage of finding an initial approximation in the bundle adjustment problem, in the absence of prior information about scene geometry and camera positions. When these parameters are known with sufficient accuracy, feature matching can be performed along epipolar lines obtained from (4), which significantly increases the number of correct correspondences.

6. PHOTOGRAMMETRIC OPTIMIZATION PROBLEM

6.1. Camera Model and Perspective

To solve the optimization problem in the photogrammetric pipeline, known as bundle adjustment for projection rays, it is necessary to define the key component of the pipeline. The "heart" of this technology, without exaggeration, is the camera model – a function for mapping or projecting points from the surrounding 3D space onto an image (screen or sensor). A key property of any model considered in this review is the rectilinear propagation of light; diffraction or chromatic aberrations are considered as negligible. Through any 2D point on the sensor, a ray can be drawn that will hit the corresponding 3D point on the object surface. Thus, the camera model defines the direction of projection rays based on the 2D sensor point, internal parameters, and camera position in space.

The perspective or rectilinear pinhole camera model [3, 11, 26, 41, 72, 73] is the most common in computer vision. Its characteristic feature is that 3D straight lines in object space are projected into 2D straight lines in image space. This model is often assumed (explicitly or implicitly) as the baseline in various studies. For example, it is used when finding homography (5), when determining mutual position between images [25, 31, 32, 34], or for 3D scene reconstruction [42]. This model serves as the initial approximation for more complex parameterizations of projection rays, considered further. The rectilinear camera model is defined by the projection matrix ${}^aP_o \in \mathbb{R}^{3\times4}$, $\operatorname{rank}({}^aP_o) = 3$ and establishes the following relationship:

$$\lambda \mathbf{q}_a^h = {}^aP_o\ \mathbf{p}_o^h = \begin{bmatrix} f_u & s_{uv} & u_c & 0 \\ 0 & f_v & v_c & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^aR_o & {}^a\mathbf{t}_o \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}_{4\times4} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} = \begin{bmatrix} K & \mathbf{0} \end{bmatrix}_{3\times4} {}^aT_o \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}, \tag{6}$$

$\mathbf{q}_a^h = [u_a, v_a, 1]^{\mathrm{T}}$ – 2D coordinates on image a, result of projecting 3D point $\mathbf{p}_o^h = [x_o, y_o, z_o, 1]^{\mathrm{T}}$ in the object CS;

K – upper-triangular matrix of internal camera parameters $\det(K) \neq 0$ (five degrees of freedom, in pixels), includes focal lengths $f_u,\ f_v$,[13] coordinates of the projection center $\mathbf{c} = [u_c, v_c]^{\mathrm{T}}$ – the point where the optical axis of the lens intersects the sensor (Fig. 7), and also the sensor diagonal distortion s_{uv};

${}^aT_o \in \mathbf{SE3}$ – matrix of external camera parameters (six degrees of freedom) defines the position of the object relative to the camera at the moment of capturing image a and includes rotation ${}^aR_o \in \mathbf{SO3}$ and translation of the object CS origin ${}^a\mathbf{t}_o$;

[13] In computer vision, unlike the classical optical model, two "focal lengths" are distinguished for convenience to account for the possible non-square pixel shape in these quantities.

parameter λ is set equal to z_a, where z_a – z-coordinate of the considered point or distance from the origin O_a along axis Z_a to the point (Fig. 7).[14]

Based on (6), let's define normalized homogeneous projection screen coordinates $\bar{\mathbf{q}}_a^h$, which are useful in the future. In object space, they correspond to the coordinates of 3D points on the plane for which $Z_a = 1$ in front of the camera, or projection onto a camera with internal parameters $K = I_{3\times3}$ $(f = 1)$:

$$\bar{\mathbf{q}}_a^h = \begin{bmatrix} \bar{u}_a \\ \bar{v}_a \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{x_a}{z_a} \\ \frac{y_a}{z_a} \\ 1 \end{bmatrix} = \frac{1}{z_a} \begin{bmatrix} {}^aR_o & {}^a\mathbf{t}_o \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}. \tag{7}$$

The operation scheme of the rectilinear camera model is shown in Fig. 7. For clarity in the example, one can set $f_u = f_v = f$, $s_{uv} = 0$ (often these assumptions hold for precise measurements [17]); the physical pixel size, sensor size, and focal length f value are irrelevant for the mathematical formulation, what matters is the ratio of quantities; the physical sensor (with reflected projection) is located behind the camera CS center O_a (optical center of the lens), its mathematical model is conventionally placed between the center and the object [72].

Using the structure of ${}^aP_o$ from (6) and considering that ${}^aR_o^{\mathrm{T}} = {}^aR_o^{-1}$, one can easily construct the projection ray equation for any 2D screen point and compute the 3D coordinates of point $\mathbf{p}_o$ relative to the object CS:

$$\mathbf{p}_o = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} = {}^aR_o^{\mathrm{T}}(K^{-1}\lambda\mathbf{q}_a^h - {}^a\mathbf{t}_o), \tag{8}$$

where $\mathbf{q}_a^h$ – screen point coordinates, $\lambda = z_a$ – depth, and the internal and external camera parameters are given.

The central equation of photogrammetry – the *triangulation problem* consists in determining the coordinates of a 3D point $\mathbf{p}_o$ from known projections $\mathbf{q}_a$, $\mathbf{q}_b$. For the rectilinear camera model, considering (6), a system of linear algebraic equations can be built:

$$\begin{cases} \lambda_a\mathbf{q}_a^h = \begin{bmatrix} K & \mathbf{0} \end{bmatrix} {}^aT_o \, \mathbf{p}_o^h, \\ \lambda_b\mathbf{q}_b^h = \begin{bmatrix} K & \mathbf{0} \end{bmatrix} {}^bT_o \, \mathbf{p}_o^h, \end{cases} \tag{9}$$

where ${}^aT_o$, ${}^bT_o$ – matrices defining the mutual position of the object and observer/camera, K – internal parameters of the camera (for multiple cameras scenario, the matrices may differ). In the triangulation problem, K, ${}^aT_o$, ${}^bT_o$ are known and the presence of parallax is important, i.e., $\|{}^a\mathbf{t}_b\|_2 \neq 0$. System (9) contains six equations and five unknowns (λ_a, λ_b, $\mathbf{p}_o$). Unknowns λ_a and λ_b are easily expressed, resulting in an overdetermined system of linear algebraic equations (four equations and three unknowns), which can be solved with the least squares approach (e.g., *normal equations*) for the 3D coordinates $\mathbf{p}_o$.

When solving the bundle adjustment problem (Section 6.5) in equation (9), the only known quantities are the detected coordinates $\mathbf{q}_a$, $\mathbf{q}_b$ (two residual equations for each observed 3D object point on the image), all other parameters are optimized.

[14] The three-dimensional representation of the surface is often stored in the form of depth maps relative to the optical center of the lens.

6.2. Basic Projection Models

The rectilinear camera model (6) establishes the relationship between the angle φ between the projection ray and the optical axis with the screen point coordinates $\mathbf{q}_a$ (Fig. 7):

$$\tan(\varphi) = \frac{r}{f} = \frac{||\mathbf{q}_a - \mathbf{c}||}{f} = ||\bar{\mathbf{q}}_a||.$$

In other words, this is the relationship between the angles of the ray from the surrounding space (collinear with the radius vector $\mathbf{p}_a$, entering the camera lens) and the ray falling on the sensor behind the lens; in the perspective mathematical camera model, both rays lie on the same straight line passing through the optical center of the lens O_a in Fig. 7.

In the physical lens model, the scattered light beam is collected from a surface area and undergoes a series of complex refractions; the presented mathematical models approximate this image formation process. Depending on the lens shapes and construction, at least three basic projection models can be distinguished [44]:

1) Rectilinear or perspective model: $r = f\tan(\varphi)$;
2) Stereographic model: $r = 2f\tan(\frac{\varphi}{2})$;
3) Equidistant model: $r = f\varphi$ (ideal wide-angle optics, since sensor resolution does not depend on angle φ).

In popular wide-angle cameras like Insta360 X4, two lenses with equidistant projection models and field of view exceeding 180° each are used to provide a panoramic view in video mode. An example of two projections is shown in Fig. 8.

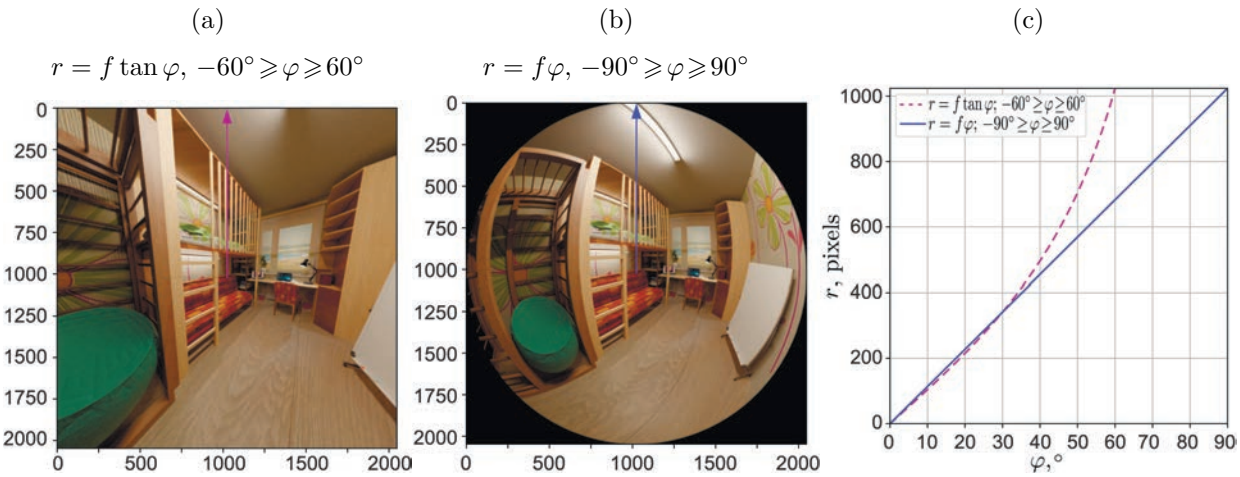

Fig. 8. Realistic 3D model of children's room (rendered in Blender 3D [84]) in two ideal projections without distortion: *a*-rectilinear, *b*-equidistant.

Let us present the equations for mapping a certain 3D point in the camera CS $\mathbf{p}_a = [x_a, y_a, z_a]^{\mathrm{T}}$ to a 2D point $\mathbf{q}_a$ on the original image for various projection models. Let φ, α be the spherical coordinates of the projection ray (3D point on a sphere), where φ is the angle between the optical axis Z_a and the projection ray, α is the rotation angle of the projection ray around axis Z_a (Fig. 7):

$$\cos(\alpha) = \frac{x_a}{\sqrt{x_a^2 + y_a^2}}, \quad \sin(\alpha) = \frac{y_a}{\sqrt{x_a^2 + y_a^2}}, \quad \tan(\varphi) = \frac{\sqrt{x_a^2 + y_a^2}}{z_a}. \tag{10}$$

As a result, the image point coordinates can be computed via polar coordinates α, $\rho(\varphi)$:

$$\mathbf{q}_a^h = K \begin{bmatrix} \rho(\varphi)\cos(\alpha) \\ \rho(\varphi)\sin(\alpha) \\ 1 \end{bmatrix}, \quad \rho(\varphi) = \begin{cases} \tan(\varphi) & \text{– for rectilinear model,} \\ 2\tan(\frac{\varphi}{2}) & \text{– for stereographic model,} \\ \varphi & \text{– for equidistant model.} \end{cases} \tag{11}$$

Substituting expressions (10) into (11), the 2D projection coordinates for any of the presented models can be obtained. It is easy to see that for the rectilinear model, (11) takes the form of (6).

There exist models combining rectilinear, stereographic, and equidistant projections. In [44], a parameter for smooth adjustment of the projection ray refraction model is introduced. This work also emphasizes the importance of accounting for the entrance pupil shift or lens optical center for wide-angle optics. Developing this idea, generalized camera's projection models with individual ray parameterization should be highlighted [45, 74, 75], where each image pixel is assigned its own ray parameters. In practice, only a subset of such pixels is selected, and parameters for the rest are interpolated. Naturally, such models easily "overfit" and require a lot of data for precise results.

In the photogrammetric pipeline, use of an appropriate projection model can significantly reduce the magnitude of distortion and decrease the total number of parameters requiring good initial approximation and careful calibration.

6.3. Distortion of Projection Models

Due to various factors, e.g., complexity of lens and optics manufacturing, sensor curvature, or camera assembly errors, the actual projection often deviates from the model, especially near the field of view or image borders. This phenomenon is called *distortion*. Often, the distortion refers to the difference between basic projection models, e.g., curvature of straight lines or shape of 3D objects in Fig. 8, since the distortion function approximates this effect [67, 76].

Typically, for the rectilinear camera model, radial and tangential distortion components are distinguished [3, 4, 17, 72]. Despite its age and significant technological development, this model works very effectively, as demonstrated in Section 7.

Radial distortion is the most significant factor distorting the rectilinear projection, with straight lines curving in a "barrel" or "pincushion" shape. It is approximated by an even-degree polynomial, as the distortion function is symmetric due to the central symmetry of lenses:

$$\delta\mathbf{r}(\mathbf{q}) = \mathbf{q}(k_r^1 r^2 + k_r^2 r^4 + k_r^3 r^6 + \ldots), \tag{12}$$

$\mathbf{q} = [u, v]^{\mathrm{T}}$ – some 2D point on the screen (distortion center at $[0, 0]^{\mathrm{T}}$);
k_r^γ – radial distortion coefficients ($\gamma \in \mathbb{N}$);
$\delta\mathbf{r}(\mathbf{q})$ – deviation of the observed 2D point from the rectilinear projection $\mathbf{q}$ (projection model (6)) due to radial distortion. The main contribution to the deviation from the rectilinear model is typically made by the first term $k_r^1 r^2$ on the right side of (12), while subsequent terms are often omitted.

Tangential distortion is caused by installation errors of the lens system and, unlike $\delta\mathbf{r}(\mathbf{q})$, creates asymmetric field distortion, formulated by [2]:

$$\delta\tau(\mathbf{q}) = \begin{bmatrix} k_\tau^2(r^2 + 2u^2) + 2k_\tau^1 uv \\ k_\tau^1(r^2 + 2v^2) + 2k_\tau^2 uv \end{bmatrix} (1 + k_\tau^3 r^2 + k_\tau^4 r^4 + \ldots), \tag{13}$$

where k_τ^γ – tangential distortion coefficients ($\gamma \in \mathbb{N}$).

Coefficients k_τ^3, $k_\tau^4, \ldots$ are typically not accounted for, or the $\delta\tau(\mathbf{q})$ factor is entirely ignored. However, the tangential distortion factor should not be completely neglected, especially in the field of precise measurements with photogrammetry: in the foundational work [3], the distortion center coordinates coincide with the projection center $\mathbf{c}$ from the camera's internal parameters K.

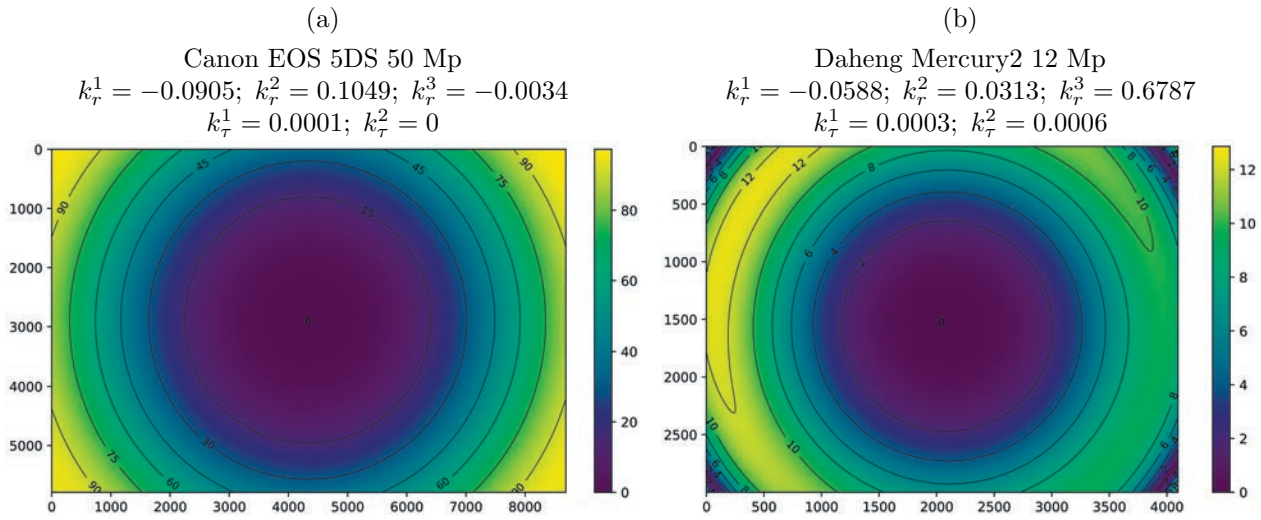

Fig. 9. Magnitude of distortion $||\delta\mathbf{r}(\mathbf{q})+\delta\tau(\mathbf{q})||_2 \cdot f$ at each frame point (in pixels). Both optical systems mainly demonstrate centrosymmetric deviation from the rectilinear model due to radial distortion.

Meanwhile, in works [67, 76, 77] neglecting tangential distortion, it is noted that $\mathbf{c}$ and the distortion center coordinates significantly deviate. Although in [3, 17] it is indicated that tangential distortion strongly correlates with the projection center position. Probably in [67, 76, 77], it is the unaccounted tangential distortion that distorts the estimate of $\mathbf{c}$.

Thus, the precise rectilinear projection model (6) accounting for distortion defines the relationship between 3D coordinates and their 2D projections as follows:

$$K^{-1}\mathbf{q}_a^h + \delta(K^{-1}\mathbf{q}_a^h) = \frac{1}{z_a}\begin{bmatrix} {}^aR_o & {}^{\mathbf{a}}\mathbf{t}_{\mathbf{o}} \end{bmatrix}\begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix} = \bar{\mathbf{q}}_a^h, \tag{14}$$

K – internal parameter matrix of the rectilinear camera model,[15]
$\mathbf{q}_a^h$ – 2D coordinates of the found feature on the original image,
$\bar{\mathbf{q}}_a^h$ – normalized 2D coordinates of the projection of the corresponding 3D point from (7),
$\delta(\mathbf{q}^h) = \begin{bmatrix} \delta\mathbf{r}(\mathbf{q}) + \delta\tau(\mathbf{q}) \\ 0 \end{bmatrix}_{3\times 1}$ – combined distortion for homogeneous coordinates.

The fundamental formula (14) is widespread in photogrammetric literature [2–4, 11, 17], where the polynomial function corrects the distortion of the observed point $\mathbf{q}_a$. This differs from the formulation in the popular category of "non-photogrammetric" works, e.g., [40, 41, 67, 72], where the polynomial function, conversely, adds the distortion effect to normalized coordinates:

$$\mathbf{q}_a^h = K[\ \bar{\mathbf{q}}_a^h + \delta(\bar{\mathbf{q}}_a^h)\ ]. \tag{15}$$

Section 7 provides a comparison of the effectiveness of models based on expressions (14) and (15). A clear example of two calibrated cameras (distortion models) is presented in Fig. 9.

For the Daheng Mercury2 camera, asymmetry is noticeable – the contribution of tangential distortion. Moreover, non-monotonicity (decrease) in the magnitude of radial distortion is observed at the frame corners, and a very large value of k_r^3 (Fig. 9b). All this is the result of instability in the optimization problem solution due to lack of observations (markers), especially in areas with

[15] Essentially, K^{-1} performs normalization of the image coordinate space.

strong distortions. In other words, the distortion is approximated by a high-order polynomial, and it incorrectly extrapolates the distortion magnitude [17].

As follows from the presented results, accounting for distortion is necessary, as its magnitude is very significant. Therefore, rectilinear camera projection models with distortion (14) and (15) are extremely widespread and found in numerous works on photogrammetry, SfM/SLAM, visual odometry, augmented reality, etc. In this work, the experimental part considers the rectilinear model with distortion (14) and (15).

6.4. Autocalibration

In photogrammetry, joint optimization of 3D coordinates of observed features, internal camera parameters, and their positions is performed (Section 6.5). To perform the nonlinear optimization procedure, an initial estimation for optimized parameters is found by the autocalibration procedure [26, 78] (**fifth step** of the photogrammetric pipeline, page 1091). Methods for 3D reconstruction with ML [42] also require such a procedure. For preliminary calibration and camera positioning in professional photogrammetry "Hexagon DPA Pro", special tools are used – groups of coded markers placed on a known cross structures. This approach has significant inconveniences. The algorithms presented below don't need any special structures in the scene for parameters estimation.

In the most common photogrammetric scenario with a single rectilinear camera (page 1104), moved against the measured object, the following optimized quantities can be distinguished:

1) 3D coordinates of observed features on the object surface: $\{\mathbf{p}_o^i \in \mathbb{R}^3\}_{i=0}^{N_{pts}}$;
2) Internal camera parameters (elements of K) and distortion coefficients: $\{f, \mathbf{c}, k_r^1, k_r^2, k_r^3, k_\tau^1, k_\tau^2, ...\}$;
3) Camera positions at different moments of time: $\{{}^aT_o \in \mathbf{SE3}\}_{a=2}^{N_{im}}$, fixing the first position ${}^1T_o = I_{4\times4}$ as the object CS.[16]

Distortion at the autocalibration step can often be neglected – the found parameters will be correct for the central area of the image (or sensor). Precise estimation of distortion parameters will be performed at the next step. For the projection center $\mathbf{c}$ – the image center is typically a good initial approximation.

If camera's distortion is too large and prevents reliable autocalibration, or the projection model differs from rectilinear due to a large field of view, then an iterative process of prior undistortion and refinement of parameters is required. In most cases, it is the rectilinear projection model that is used in autocalibration algorithms [78, 79]. In [79], autocalibration is reduced to determining the mutual camera positions ${}^aR_b$, ${}^a\mathbf{t}_b$, $a, b \in [1...N_{im}]$, using previously found correspondences, with (4) and the structure of the fundamental matrix:

$$ {}^aF_b = K^{-\mathrm{T}}\ {}^aE_b\ K^{-1} = K^{-\mathrm{T}}\ [{}^a\mathbf{t}_b]_\times\ {}^aR_b\ K^{-1}, \tag{16}$$

$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix}$ – skew-symmetric matrix of the vector product.

The translation between cameras ${}^a\mathbf{t}_b$ can be determined only up to scale (the vector specifies only direction) – this is a theoretical limitation of autocalibration. To find ${}^aR_b$, ${}^a\mathbf{t}_b$, it's sufficient to have five corresponding points to calculate residuals from (4), since ${}^aF_b$ has five degrees of freedom [26]. The minimal number of correspondences allows effective application of RANSAC [69] to improve the stability of the found solution. In [79], for autocalibration with two images (or camera positions),

[16] In the general case, the photogrammetry problem is solved up to a **SE3** transformation with an arbitrary scale of the resulting point cloud [26], thus parameters of one position can be fixed.

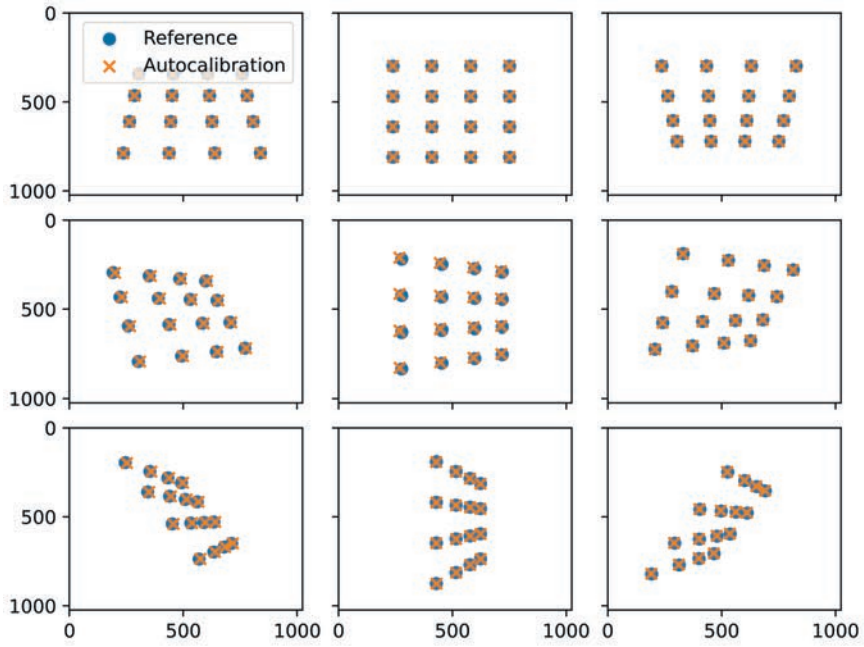

Fig. 10. Model images (1024x1024 pixels) of 9 tilted views for a flat object (circular markers) and projection of found 3D coordinates (crosses) – result of the autocalibration algorithm [78], estimating internal and external camera parameters, 3D coordinates of object points. Root mean square reprojection error according to (6) is 3.35 pixels with input observation noise amplitude of one pixel.

the internal camera parameters (matrix K) are required. At the same time, the focal length f is difficult to estimate in advance, as it can vary significantly depending on the field of view. Moreover, if the observed correspondences (for two views or camera positions) lie on a flat surface, then the internal parameters cannot be estimated [26, 79]. Thus, the most reliable solution from [79] is to use three or more images obtained from different positions.

An alternative approach [78] based on factorization of absolute quadric matrix was implemented in this work. The problem reduces to solving a nonlinear system of equations:

$$K^{-1}\, P_b\, \Omega\, P_b^{\mathrm{T}}\, K^{-\mathrm{T}} = \lambda\, I_{3\times 3} \tag{17}$$

against the elements of the quadratic form matrix $\Omega = HH^{\mathrm{T}} \in \mathbb{R}^{4\times 4}$ and the internal parameter matrix K,
λ – arbitrary scale coefficient,
$P_b = K[\, {}^bR_a | {}^b\mathbf{t}_a]_{3\times 4} H^{-1}$ – known projective transformation matrix (6) (position b, for example), which was obtained from factorization of the fundamental matrix ${}^aF_b$, and ${}^aF_b$, in turn, was found from correspondences between images a, b. Thus, knowing correspondences between several views (at least three), the relative camera position ${}^bR_a$, ${}^b\mathbf{t}_a$ and its parameters K can be determined.

At the next step, unknown 3D point coordinates are computed via the triangulation equation (9), followed by the bundle adjustment problem. Figure 10 shows the result of the autocalibration algorithm for a flat object from correspondences from nine views. Various methods for estimating relative position and camera parameters without initial approximation are also available in the open-source library [80].

6.5. Bundle Adjustment

Measurements obtained by photogrammetry are the 3D coordinates of observed 2D features, as well as the spatial position of objects (corresponding point clouds) relative to cameras. This requires

joint optimization of the multiple parameters listed above. The central problem of photogrammetry is minimizing the sum of squared residuals or reprojection errors (**sixth step** of the photogrammetric pipeline, page 1091):

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \mathcal{L}(\theta), \tag{18}$$

where θ – combined vector of optimized parameters (Section 6.4);
$\mathcal{L}(\theta)$ – minimized objective or loss function.

The geometric meaning of the problem is bringing the bundle of projection rays to converge at the corresponding 3D points by minimizing the reprojection error, hence the name "bundle adjustment".

The terms of the loss function are derived from the camera's projection model. The type of projection model is based, ideally, on the camera's lens design, e.g., (6) for a rectilinear camera without distortion (special lens design, for example), (11) for a wide-angle camera, (15) or (14) for a precise rectilinear model with radial and tangential distortion. The loss function with reprojection error from (14) has the following form:

$$\begin{aligned} \mathcal{L}(\theta) &= \sum_{a=1}^{N_{im}} \sum_{i=1}^{N_{pts}} e_{a,i}(\theta) + \ w_{scale} \sum_{(i,j,d)\in S} (||\mathbf{p}_o^i - \mathbf{p}_o^j||_2 - d)^2, \\ e_{a,i}(\theta) &= \left\| K(\theta)^{-1}\mathbf{q}_a^{m_a(i)} + \delta_\theta(K(\theta)^{-1}\mathbf{q}_a^{m_a(i)}) - \frac{1}{z_a}\ {}^aT_o(\theta)\ \mathbf{p}_o^i \right\|_2^2, \end{aligned} \tag{19}$$

where $e_{a,i}(\theta)$ – square of $l2$-norm of 2D residual or reprojection error for one point on the image;
$\mathbf{p}_o^i$ (component of vector θ) – unknown 3D point on the object surface (e.g., marker in Fig. 3), symbol h omitted for brevity;
$m_a(i)$ – mapping of the ith 3D point on the object to the index of its projection on image a (assuming, for brevity, that all object points are visible on the image);
$\mathbf{q}_a^i$ – homogeneous 2D coordinates of the feature in pixels, original observations on image a;
$S = \{(i,j,d)|i,j \in [1..N_{pts}], d \in \mathbb{R}^+\}_{i=1}^{N_{scales}}$ – scale bars (N_{scales} pieces) in the form of known points and calibrated distances between them, e.g., coded markers on two scale bars in Fig. 3;
${}^aT_o(\theta)$ – camera CS position includes rotation matrix ${}^aR_o(\theta)$ and origin ${}^a\mathbf{t}_o$ (component of vector θ) when capturing image a;
w_{scale} – weight of the scale measurement error term. The weight is needed to balance the different numbers of equations of two types. Moreover, the reprojection error can have different units: pixels for (15) and normalized coordinates for (14).

For better numerical stability, it is desirable to normalize image coordinates so that $\mathbf{q}_a \in [-1,1]^2$. This is common practice, but an important nuance can be missed – first, the origin must be shifted from the center of the first pixel to the corner of the image to decouple the coordinate system from the original image pixels. Thus, the resulting camera calibration K will not be tied to a specific resolution.

A number of key aspects of the optimization problem (18) should be noted:

1) Relatively large number of optimized parameters: in the experimental scene, 71 camera positions (six degrees of freedom each), and 430 markers on the object, i.e., at least $71 \cdot 6 + 430 \cdot 3 = 1716$ parameters;
2) Sparsity of the equation system (Jacobian): each term in (19) – reprojection error includes only one position, and not all object points are observed in every frame;
3) Presence of outliers or points with very high reprojection error: Fig. 11 shows that in almost every position there are points with residuals multiple times higher the root mean square error σ_{repr}.

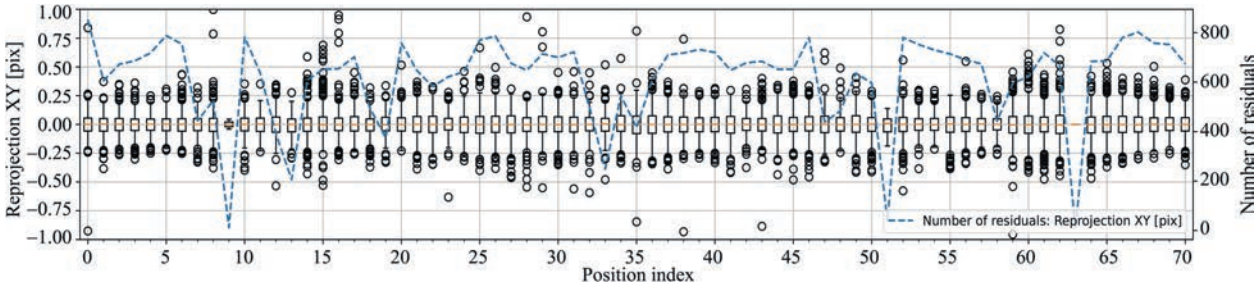

Fig. 11. Distribution of reprojection error of 3D points from (19) (left Y-axis) in pixels across 71 camera positions as box plots covering 99% of points. Circle denotes outlier (less than 1% of points). Dashed line (right Y-axis) denotes number of residuals per position – two for each observed point.

Thus, it is necessary to use optimization methods robust to noise and outliers (even after careful feature detection and filtering). As seen in Fig. 11, outliers occur more frequently than can be predicted based on a normal or Gaussian distribution, as emphasized in [39]. It proposes replacing $e_{a,i}(\theta)$ in (19) with $\rho(e_{a,i}(\theta))$, where $\rho(x)$ is a non-negative scalar function or "robust kernel": $\rho(x) > 0,\ \forall x \in \mathbb{R}, \rho'(0) = 0$, which suppresses gradients of the loss function in case of outliers due to $\rho'(x) < 1$. In the experimental part, the Huber function was used for this purpose:

$$\rho_t(x) = \begin{cases} x, & \text{if } x < t^2, \\ t^2\ (2\sqrt{x/t^2} - 1), & \text{otherwise}, \end{cases} \tag{20}$$

where $t = 0.05$ pixels – normalization coefficient.

For residual values below t, the robust kernel function has no effect on the loss function; however, in the outlier region above t, robust kernel significantly reduces outliers contribution to the loss function (19).

For reliable solution of the bundle adjustment problem (18), approximate Newton methods with regularization are used – at each algorithm's iteration, step direction and magnitude adjustment is performed [39]. The Levenberg–Marquardt optimization algorithm from this category was used in the experimental part. An efficient implementation of various optimization methods, different robust kernels and sparse matrix operations is available in [81].

7. EXPERIMENTAL PART

The experiments goal is to highlight the key factors of the optimization process (18) that affect the accuracy of the reconstructed 3D point cloud. Original data (digital images) was acquired with camera from "Hexagon DPA Pro" (HDP) hardware set, specifically, Canon EOS 5DS with a monochrome sensor $w \times h = 8700 \times 5800$ (50 Mp) and lens $f = 28$ mm. It should be noted that various professional DSLR camera models can be used for precise measurements [28].

The test scene and the reconstructed 3D point cloud of 430 markers are shown in Fig. 3. The scene includes two rigid objects – a steel welding table with attached markers (overall dimension ~2000mm) and a carbon fiber calibration plate (overall dimension ~800mm, markers applied by the manufacturer). Two scale bars of 1390mm and 790mm define metric units for the reconstructed point cloud and provide redundant accuracy verification. Photography was performed carefully with environmental temperature control (20 °C) in the lab. Two nearly identical datasets were collected with a one-day interval (74 and 79 images). Image processing for each day was performed both in the HDP software package and in the authors' own software based on the Python language and open-source mathematical libraries Jax [82], SciPy [83], OpenCV [72].

The photogrammetric pipeline of reconstruction included the following steps:

1) Photography of the test object from different positions (distance between camera and object 1.5–2 m), example in Fig. 4;

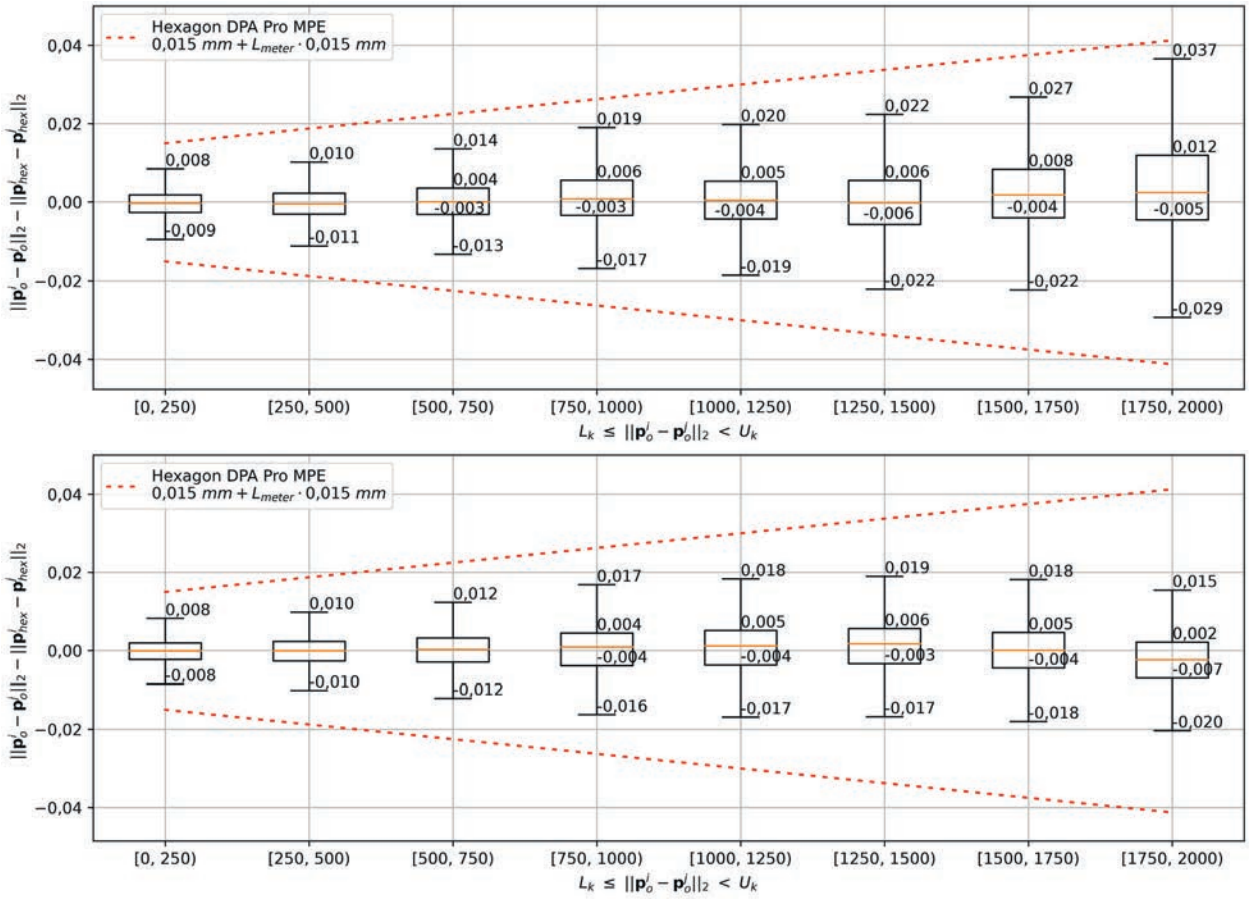

Fig. 12. Comparison of photogrammetric reconstruction results of the test scene from "Hexagon DPA Pro" and the current approach for two experiments (data from two days – upper and lower graphs). Units in **millimeters**. X-axis shows measured distance ranges.

2) Computation of circular marker parameters (coded and regular) on images and matching (grouping) of coded markers by binary code, Section 5;
3) Autocalibration estimating rough camera calibration (without accounting for distortion (12)) and its positions at different moments of time, Section 6.4;
4) Approximate bundle adjustment estimating cloud of 3D coordinates of coded markers and camera calibration 6.5;
5) Matching (grouping) of all markers using the fundamental matrix equation (4), obtained from camera calibration [26];
6) Precise bundle adjustment, result – cloud of 3D coordinates of all markers and accurate camera calibration.

Next, we consider the results – reconstructed 3D point clouds obtained for various configurations of the optimization problem (18). Let's denote Brown's projection model [3] (14) as Brown; with residual stabilization (20) as Brown & $\rho_{0.05}$. For the case without accounting for tangential distortion (13) Brown & $\rho_{0.05}$ & k^0_τ. Model (15) is denoted as OpenCV[17] and, according to previous definitions, we introduce OpenCV & $\rho_{0.05}$, OpenCV & $\rho_{0.05}$ & k^0_τ.

Figure 12 presents a comparison of reconstruction from HDP and the scheme Brown & $\rho_{0.05}$. Each graph includes $k = 8$ distributions (box plots) of distance differences between corresponding point pairs of two clouds:

[17] This formulation is given in [72].

$||\mathbf{p}_o^i - \mathbf{p}_o^j||_2 - ||\mathbf{p}_{hex}^i - \mathbf{p}_{hex}^j||_2,\ \forall i, j \in [1..N_{pts}]$, where $\mathbf{p}_{hex}^{\cdot}$ – points obtained from HDP, $\mathbf{p}_o^{\cdot}$ – result of optimization (18) with model Brown & $\rho_{0.05}$. The distribution of deviations is given for a specific interval $[L_k, U_k)$ of distances between points. The red line indicates the maximum permissible error of HDP for length measurement. It should be emphasized that HDP is a certified measurement system according to the VDI / VDE 2634, part 1 standard. This guarantees volumetric measurement accuracy within the limits indicated by the red line. Thus, the model Brown & $\rho_{0.05}$ yields results within the HDP error margin.

In HDP, both scale bars were used in the bundle adjustment problem. For all results obtained by the author's software, the loss function (19) included the length residual of only the larger bar (1390mm, $N_{scales} = 1$), the 790mm bar was used for additional verification. By increasing w_{scale} in (19), zero error of the bar length can easily be achieved at the cost of projective distortion of the resulting point cloud.

Table provides a comparison of distance deviations between point pairs for two clouds similar to Fig. 12, but with the algorithm being common and data from different days. Thus, the repeatability of results – a key characteristic of any measurement system and photogrammetry in particular – can be evaluated. The scene contained two rigid objects – a steel welding table and a carbon calibration plate; repeatability is evaluated separately for each point subset. Columns σ_{table} and σ_{plate} show the root mean squared error for table and plate points respectively. The largest error in measuring the scale bar length (790mm) over two days e_{bar} is also presented.

As seen from Table, the measurement repeatability of the small calibration plate σ_{plate} is almost identical (with one exception) and does not depend on the algorithm configuration choice. For the table, the difference can be significant – the best repeatability is achieved with optimization suppressing outliers: Brown & $\rho_{0.05}$ and OpenCV & $\rho_{0.05}$ give the best results in their groups. Moreover, without stabilization in configurations Brown and OpenCV, significant errors in scale bar measurements e_{bar} are observed. The results also confirm the necessity of accounting for tangential distortion (13) in the bundle adjustment problem even for high-class optics (Canon EOS 5DS). It's interesting to note that Brown & $\rho_{0.05}$ significantly outperforms HDP in σ_{table}. In conclusion, it should be emphasized that the main components for high-precision results are careful texture feature extraction and accounting for the detection errors in the optimization problem (18).

Repeatability of distances for 3D point clouds obtained by one algorithm from data of different days (smaller is better)

Algorithm	$\sigma_{table}, \mu m$	$\sigma_{plate}, \mu m$	$e_{bar}, \mu m$
HDP	46.5	7.2	–2.9[18]
Brown	80.4	22.5	138.7
Brown & $\rho_{0.05}$	**18.0**	7.5	–6.9
Brown & $\rho_{0.05}$ & k_τ^0	41.1	7.5	3.3
OpenCV	41.8	7.6	–154.2
OpenCV & $\rho_{0.05}$	39.8	7.6	–9.0
OpenCV & $\rho_{0.05}$ & k_τ^0	41.2	7.7	4.3

8. CONCLUSION

Within this work, a broad overview of algorithms constituting the photogrammetric pipeline has been presented. Key factors determining the accuracy of the output 3D point cloud have been demonstrated. A mathematical formulation of the photogrammetric optimization problem for various camera models has been presented. In the experimental part, high accuracy indicators for the described mathematical models on real hardware and application scenario have been confirmed.

The authors express their sincere gratitude to the employees and management of LLC "Digital Assembly" (Russian Federation, St. Petersburg) for their assistance in organizing the experiment. Thanks to the professionalism of the company's engineers, it was possible to collect high-quality data for testing various hypotheses and obtain results of world leaders level in the field of photogrammetry. The authors are ready to provide the source data (images, obtained point clouds) upon a request.

FUNDING

The work was supported by the Ministry of Science and Higher Education of Russia (theme no. FSFN-2024-008Z).

REFERENCES

1. Finsterwalder, S., Die geometrischen grundlagen der photogrammetrie, *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 1897, vol. 6, pp. 1–42.
2. Brown, D., Decentering distoriton of lenses, *Photogrammetric Engineering*, 1966, vol. 32, no. 3, pp. 444–462.
3. Brown, D., Close-range camera calibration, *Photogrammetric Engineering*, 1971, vol. 37, no. 8, pp. 855–866.
4. Fryer, J. and Brown, D., Lens distortion for close-range photogrammetry, *Photogrammetric Engineering and Remote Sensing*, 1986, vol. 52, pp. 51–58.
5. Drobyshev, F.V., Soviet stereophotogrammetric instruments, *Photogrammetria*, 1960, vol. 17, pp. 60–68.
6. Dubinovskiy, V.B., et al., A rigorous method of constructing photogrammetric networks for updating topographic maps, *Izvestiya vysshikh uchebnykh zavedenij. Geodeziya i aerofotosyemka*, 1990, no. 6, pp. 68–72.
7. Alchinov, A.I., "Talka-tdv" company and the "talka" digital photogrammetric station, *Geoprofi*, 2005, no. 1, pp. 10–11.
8. Luhmann, T., Close range photogrammetry for industrial applications, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2010, vol. 65, no. 6, pp. 558–569, iSPRS Centenary Celebration Issue.
9. Bösemann, W., Industrial photogrammetry – accepted metrology tool or exotic niche, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2016, vol. XLI-B5, pp. 15–24.
10. Altukhov, V.G., Study of the accuracy of photogrammetry as a method for determining the volume of an object, *Avtomatika i Programmnaya Inzheneriya*, 2020, vol. 32, no. 2, pp. 69–74.
11. Leizea, I., Herrera, I., and Puerto, P., Calibration procedure of a multi-camera system: Process uncertainty budget, *Sensors*, 2023, vol. 23, no. 2.
12. Balanji, H.M., Turgut, A.E., and Tunc, L.T., A novel vision-based calibration framework for industrial robotic manipulators, *Robot. Comput.-Integr. Manuf.*, 2022, vol. 73, no. C.
13. Puerto, P., et al., Analyses of key variables to industrialize a multi-camera system to guide robotic arms, *Robotics*, 2023, vol. 12, pp. 10–22.
14. Li, Z., et al., A robust camera self-calibration method based on circular oblique images, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2024, vol. X-1-2024, pp. 131–136.
15. Ulrich, M., et al., Vision-guided robot calibration using photogrammetric methods, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2024, vol. 218, pp. 645–662.
16. Visilter, Y.V., et al., Image comparison by shape using diffuse morphology and diffuse correlation, *Kompyuternaya Optika*, 2015, vol. 39, no. 2, pp. 265–274.

17. Fraser, C.S., Automatic camera calibration in close-range photogrammetry, *Photogrammetric Engineering and Remote Sensing*, 2013, vol. 79, pp. 381–388.

18. Lebedev, M.A., et al., A real-time photogrammetric algorithm for sensor and synthetic image fusion with application to aviation combined vision, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2014, vol. XL-3, pp. 171–175.

19. Knyaz, V., et al., Multi-sensor data analysis for aerial image semantic segmentation and vectorization, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2024, vol. XLVIII-1-2024, pp. 291–296.

20. Kobzev, A. and Chibunichev, A., Aerial triangulation using different time images of urban areas obtained from unmanned aerial systems, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2024, vol. XLVIII-2/W5-2024, pp. 87–93.

21. Schonberger, J.L. and Frahm, J.M., Structure-from-motion revisited, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

22. Turki, H., Ramanan, D., and Satyanarayanan, M., Mega-nerf: Scalable construction of large-scale nerfs for virtual fly-throughs, 2022.

23. Lowe, D., Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision*, 2004, vol. 60, no. 2, pp. 91–110.

24. Bay, H., Tuytelaars, T., and Van Gool, L., *Computer Vision – ECCV 2006* A. Leonardis, H. Bischof, and A. Pinz, editors, Surf: Speeded up robust features, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

25. DeTone, D., Malisiewicz, T., and Rabinovich, A., Superpoint: Self-supervised interest point detection and description, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 337–348.

26. Hartley, R. and Zisserman, A., *Multiple View Geometry in Computer Vision*, New York, NY, USA: Cambridge University Press, 2 edn., 2003.

27. Sola, J., Deray, J., and Atchuthan, D., A micro lie theory for state estimation in robotics, 2021.

28. Rieke-Zapp, D., et al., Evaluation of the geometric stability and the accuracy potential of digital cameras – comparing mechanical stabilisation versus parameterisation, *ISPRS Journal of Photogrammetry and Remote Sensing*, 2009, vol. 64, no. 3, pp. 248–258, theme Issue: Image Analysis and Image Engineering in Close Range Photogrammetry.

29. Tareen, S.A.K. and Saleem, Z., A comparative analysis of sift, surf, kaze, akaze, orb, and brisk, *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2018, pp. 1–10.

30. Jin, Y., et al., Image matching across wide baselines: From paper to practice, *Int. J. Comput. Vision*, 2021, vol. 129, no. 2, pp. 517–547.

31. Sun, J., et al., LoFTR: Detector-free local feature matching with transformers, *CVPR*, 2021, vol. 1, pp. 8918–8927.

32. Lindenberger, P., Sarlin, P.E., and Pollefeys, M., LightGlue: Local Feature Matching at Light Speed, *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 1, 2023, pp. 17581–17592.

33. Vizilter, Y., Zheltov, S., and Lebedev, M., Image and shape comparison via morphological correlation, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2021, vol. XLIV-2/W1-2021, pp. 207–211.

34. Sarlin, P.E., et al., Superglue: Learning feature matching with graph neural networks, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4937–4946.

35. Harris, C.G. and Stephens, M.J., A combined corner and edge detector, *Alvey Vision Conference*, 1988.

36. Shi, J. and Tomasi, C., Good features to track, *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2000, vol. 600.

37. Rosten, E. and Drummond, T., *Computer Vision – ECCV 2006* A. Leonardis, H. Bischof, and A. Pinz, editors, Machine learning for high-speed corner detection, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

38. Tyszkiewicz, M.J., Fua, P., and Trulls, E., Disk: Learning local features with policy gradient, 2020.

39. Triggs, B., et al., Bundle adjustment – a modern synthesis, *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV'99, Berlin, Heidelberg: Springer-Verlag, 1999, pp. 298–372.

40. Fitzgibbon, A., Simultaneous linear estimation of multiple view geometry and lens distortion, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001, pp. 1–125.

41. Zhang, Z., A flexible new technique for camera calibration, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, vol. 22, pp. 1330–1334, mSR-TR-98-71, Updated March 25, 1999.

42. Wang, J., et al., Vggsfm: Visual geometry grounded deep structure from motion, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 21686–21697.

43. Gurdjos, P., Sturm, P., and Wu, Y., *Computer Vision – ECCV 2006* A. Leonardis, H. Bischof, and A. Pinz, editors, Euclidean structure from n $\geqslant$ 2 parallel circles: Theory and algorithms, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 238–252.

44. Gennery, D., Generalized camera calibration including fish-eye lenses, *International Journal of Computer Vision*, 2006, vol. 68, pp. 239–266.

45. Schops, T., et al., Why having 10,000 parameters in your camera model is better than twelve, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, 2020, pp. 2532–2541.

46. Canny, J., A computational approach to edge detection, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 1986, vol. PAMI-8, pp. 679–698.

47. Shen, Z., et al., Combining convex hull and directed graph for fast and accurate ellipse detection, *Graphical Models*, 2021, vol. 116, pp. 101–110.

48. Ouellet, J.N. and Hebert, P., Precise ellipse estimation without contour point extraction, *Mach. Vis. Appl.*, 2009, vol. 21, pp. 59–67.

49. Mortari, D., Junkins, J., and Samaan, M., Lost-in-space pyramid algorithm for robust star pattern recognition, *Spaceflight Mechanics 2005*, 2001, vol. 120, pp. 10–20.

50. Calonder, M., et al., *Computer Vision – ECCV 2010* K. Daniilidis, P. Maragos, and N. Paragios, editors, Brief: Binary robust independent elementary features, Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 778–792.

51. Rublee, E., et al., Orb: An efficient alternative to sift or surf, *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.

52. Fernandez Alcantarilla, P., Fast explicit diffusion for accelerated features in nonlinear scale spaces, *British Machine Vision Conference (BMVC) at Bristol, UK*, 2013.

53. Schneider, C.T. and Sinnreich, K., Optical 3-d measurement systems for quality control in industry, *XVIIth ISPRS Congress Technical Commission V: Close-Range Photogrammetry and Machine Vision*, 1992, vol. 29, pp. 56–59.

54. Dos Santos Cesar, D.B., et al., An evaluation of artificial fiducial markers in underwater environments, *OCEANS 2015 – Genova*, 2015, pp. 1–6.

55. Calvet, L., et al., Detection and accurate localization of circular fiducials under highly challenging conditions, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 562–570.

56. Tushev, S., Sukhovilov, B., and Sartasov, E., Robust coded target recognition in adverse light conditions, *2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM)*, 2018, pp. 1–6.

57. Lin, T.Y., et al., Feature pyramid networks for object detection, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

58. He, K., et al., Deep residual learning for image recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

59. Ronneberger, O., Fischer, P., and Brox, T., U-net: Convolutional networks for biomedical image segmentation, *Medical image computing and computer-assisted intervention – MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

60. Caetano, T.S., et al., Learning graph matching, *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

61. Arandjelovic, R. and Zisserman, A., Three things everyone should know to improve object retrieval, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2911–2918.

62. Rusu, R.B., Blodow, N., and Beetz, M., Fast point feature histograms (fpfh) for 3d registration, *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

63. Muja, M. and Lowe, D., Fast approximate nearest neighbors with automatic algorithm configuration, *VISAPP 2009 – Proceedings of the Fourth International Conference on Computer Vision Theory and Applications, Lisboa, Portugal, February 5–8*, vol. 1, 2009, pp. 331–340.

64. Bishop, C., *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Berlin, Heidelberg: Springer-Verlag, 2006.

65. Lv, Q., et al., Multi-probe lsh: efficient indexing for high-dimensional similarity search, *Proceedings of the 33rd International Conference on Very Large Data Bases*, VLDB '07, VLDB Endowment, 2007, pp. 950–961.

66. Nister, D. and Stewenius, H., Scalable recognition with a vocabulary tree, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 2161–2168.

67. Hartley, R. and Kang, S.B., Parameter-free radial distortion correction with centre of distortion estimation, *Proceedings of IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision*, vol. 2, 2005, pp. 1834–1841.

68. Hartley, R., In defense of the eight-point algorithm, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, vol. 19, no. 6, pp. 580–593.

69. Fischler, M.A. and Bolles, R.C., Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, M.A. Fischler and O. Firschein, editors, *Readings in Computer Vision*, San Francisco (CA): Morgan Kaufmann, 1987, pp. 726–740.

70. Chum, O. and Matas, J., Matching with prosac – progressive sample consensus, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 220–226.

71. Chum, O., Werner, T., and Matas, J., Two-view geometry estimation unaffected by a dominant plane, *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 2005, pp. 772–779.

72. Kaehler, A. and Bradski, G., *Learning OpenCV, 2nd Edition*, O'Reilly Media, Inc., 2014.

73. Chibunichev, A., Govorov, A., and Chernyshev, V., Research of the camera calibration using series of images with common center of projection, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2019, vol. XLII-2/W18, pp. 19–22.

74. Grossberg, M. and Nayar, S., A general imaging model and a method for finding its parameters, *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 2001, pp. 108–115.

75. Tecklenburg, W., Luhmann, T., and Hastedt, H., Camera modelling with image-variant parameters and finite elements, *Optical 3D-Measurement Techniques V*, 2001, pp. 328–335.

76. Hughes, C., et al., Equidistant ($f\theta$) fish-eye perspective with application in distortion centre estimation, *Image and Vision Computing*, 2010, vol. 28, no. 3, pp. 538–551.

77. Bukhari, F. and Dailey, M.N., Automatic Radial Distortion Estimation from a Single Image, *Journal of Mathematical Imaging and Vision*, 2013, vol. 45, no. 1, pp. 31–45.

78. Triggs, B., Autocalibration from planar scenes, *Proceedings of the 5th European Conference on Computer Vision*, vol. 1 of *ECCV'98*, Berlin, Heidelberg: Springer-Verlag, 1998, pp. 89–105.

79. Nister, D., An efficient solution to the five-point relative pose problem, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, vol. 26, no. 6, pp. 756–770.

80. Larsson, Viktor and contributors, PoseLib – Minimal Solvers for Camera Pose Estimation, 2020.

81. Agarwal, S., Mierle, K., and The Ceres Solver Team, Ceres Solver, 2023.

82. Bradbury, J., et al., JAX: composable transformations of Python+NumPy programs, 2018.

83. Virtanen, P., et al., Scipy 1.0: Fundamental algorithms for scientific computing in python, *Nature Methods*, 2020, vol. 17, pp. 261–272.

This paper was recommended for publication by B.M. Miller, a member of the Editorial Board