

MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS EDUCATIONAL
INSTITUTION OF HIGHER EDUCATION
NORTH CAUCASUS FEDERAL UNIVERSITY

Methodical Guidelines

on Laboratory Works

on the discipline

**DIGITAL IMAGE PROCESSING AND MACHINE VISION / ЦИФРОВАЯ
ОБРАБОТКА ИЗОБРАЖЕНИЙ И МАШИННОЕ ЗРЕНИЕ**

for students of the Master's program

02.04.01 Mathematics and Computer Sciences

profile: Training Program for Electronic Information

Stavropol, 2026

CONTENT

Introduction

Laboratory Work №1: Image Fundamentals and Spatial Operations

Laboratory Work №2: Frequency Domain Analysis: Fourier Transform & Filtering

Laboratory Work №3: Wavelet Transforms and Multiscale Analysis

Laboratory Work №4: Morphological Processing and Image Segmentation

Laboratory Work №5: Edge and Keypoint Detection

Laboratory Work №6: Descriptor Construction and Image Matching

Laboratory Work №7: Single and Two-View Geometry: Calibration & Epipolar Geometry

Laboratory Work №8: End-to-End Processing Pipeline (Project Workshop)

Guidelines for Independent Work

Evaluation Criteria and Competency Assessment

List of Recommended Literature

INTRODUCTION

1. Discipline Purpose and Position in the Educational Program

The discipline "Digital Image Processing and Machine Vision / Цифровая обработка изображений и машинное зрение" (Index: Б1.О.09) belongs to Block 1, Mandatory Part of the Master's educational program 02.04.01 Mathematics and Computer Sciences. It is implemented in Semester 2 and carries 4 credit units (144 academic hours: 16 h lectures, 16 h laboratory sessions, 58 h independent work, 54 h exam preparation).

Purpose of the Laboratory Component:

To develop students' practical skills in implementing classical image processing and computer vision algorithms from first principles, ensuring deep understanding of mathematical foundations, numerical stability, and computational trade-offs before transitioning to high-level library abstractions or deep learning frameworks.

List of planned learning outcomes in the discipline (module) correlated with the planned outcomes of the educational program

Competency code, description	Indicator code, description	Planned learning outcomes in the discipline (module) characterizing the stages of competencies and indicators formation
UC-1 Able to conduct critical analysis of information, apply a systematic approach to solve research and professional tasks	UC-1.1 Identifies and formulates professional problems, selects appropriate mathematical and algorithmic tools for their solution	Applies spatial and frequency domain representations, variational principles, and geometric vision models to design robust image processing pipelines for real-world visual data analysis.
OPC-1 Able to apply fundamental mathematical knowledge and modern programming technologies to solve professional problems in the field of mathematics and computer science	OPC-1.1 Develops and implements efficient algorithms for processing and analyzing visual data	Implements core image processing and computer vision algorithms from scratch using Python, NumPy, and OpenCV, demonstrating understanding of numerical stability, computational complexity, and algorithmic assumptions.
OPC-2 Able to conduct independent research using modern methods, tools, and information technologies in the chosen professional field	OPC-2.1 Designs and validates experimental pipelines for visual information processing, analyzing results against theoretical models	Designs complete processing workflows (filtering → segmentation → feature extraction → 3D reconstruction), critically evaluates parameter sensitivity and noise robustness, and justifies the integration of classical methods with hybrid AI architectures.
PC-1 Able to develop and apply mathematical models and software systems for processing electronic information and intelligent data analysis	PC-1.2 Applies geometric and algebraic methods for camera calibration, stereo vision, and structure-from-motion	Solves multi-view geometry problems, performs intrinsic/extrinsic camera calibration, and reconstructs 3D scenes from 2D image sequences using projective geometry, epipolar constraints, and optimization on manifolds.

2. Structure of Laboratory Sessions

Each laboratory session follows a 2-hour format:

- **First 30 minutes:** Brief theoretical recap and task clarification
- **Next 75 minutes:** Hands-on implementation with instructor consultation
- **Final 15 minutes:** Results demonstration and preliminary discussion

All assignments are submitted via the university LMS as Jupyter notebooks containing:

- Executable code with inline mathematical comments
- Visualizations of intermediate processing stages
- Quantitative evaluation metrics (PSNR, SSIM, reprojection error, etc.)
- Critical analysis of limitations and failure modes

3. Technical Requirements

- Python 3.9+ with NumPy, SciPy, scikit-image, OpenCV-Python, Matplotlib
- JupyterLab environment (provided via university cloud workspace)
- Access to curated datasets: Middlebury (stereo), KITTI (visual odometry), COCO subsets (keypoints)
- Version control via Git for project submissions

LABORATORY WORK №1

Image Fundamentals and Spatial Operations

Objective:

Master basic image representation, spatial filtering, and point operations by implementing convolution, histogram manipulation, and geometric transformations from scratch.

Competencies Assessed: OPC-1.1, UC-1.1

Theoretical Background:

- Discrete image representation as 2D arrays; color spaces (RGB, HSV, LAB)
- Convolution theorem: spatial filtering as weighted neighborhood operations
- Separable kernels for computational efficiency
- Interpolation methods for geometric transformations (nearest, bilinear, bicubic)

Tasks:

1. Custom Convolution Implementation

- o Implement 2D convolution without using `scipy.signal.convolve2d` or `cv2.filter2D`
- o Support arbitrary kernel sizes, boundary handling modes (zero-padding, replicate, reflect)
- o Validate against reference implementation on test images (impulse response, step edge)

2. Histogram Equalization from First Principles

- o Compute cumulative distribution function (CDF) of image intensity
- o Map original intensities using CDF normalization
- o Compare results with `cv2.equalizeHist`; analyze contrast enhancement metrics

3. Affine Transformation Pipeline

- o Implement forward and inverse mapping for rotation, scaling, shearing
- o Compare nearest-neighbor vs. bilinear interpolation in terms of visual quality and PSNR
- o Document edge artifacts and propose mitigation strategies

Questions for Self-Check:

1. Why is convolution commutative in the spatial domain but not always in practice with finite images?
2. How does histogram equalization affect the signal-to-noise ratio of an image?
3. What are the computational complexities of forward vs. inverse mapping for geometric transformations?

Report Requirements:

- Code with mathematical annotations for each algorithmic step
- Side-by-side visual comparisons of custom vs. library implementations
- Quantitative table: execution time, memory usage, PSNR/SSIM for each operation
- Critical discussion: when would you prefer custom implementation over library function?

Recommended Literature:

- Gonzalez & Woods, Ch. 3 (Intensity Transformations and Spatial Filtering)
- Forsyth & Ponce, Sec. 2.1–2.3 (Image Formation and Basic Operations)
- OpenCV Documentation: `cv2.warpAffine`, `cv2.filter2D` (for verification only)

LABORATORY WORK №2

Frequency Domain Analysis: Fourier Transform & Filtering

Objective:

Understand the relationship between spatial and frequency representations; implement DFT-based filtering and analyze spectral properties of images.

Competencies Assessed: OPC-1.1, UC-1.1, OPC-2.1

Theoretical Background:

- 2D Discrete Fourier Transform: definition, properties, shift theorem
- Frequency domain filtering: ideal, Butterworth, Gaussian low/high-pass filters
- Ringing artifacts and windowing techniques
- Phase vs. magnitude: perceptual importance in image reconstruction

Tasks:

1. **Manual 2D DFT Implementation**
 - o Compute DFT using direct summation formula (no `np.fft`) for small images ($\leq 64 \times 64$)
 - o Verify linearity, shift, and conjugate symmetry properties
 - o Visualize magnitude spectrum with logarithmic scaling and centering
2. **Frequency-Domain Filtering Pipeline**
 - o Design Butterworth low-pass filter with adjustable order and cutoff
 - o Apply filter in frequency domain; reconstruct image via inverse DFT
 - o Compare with spatial-domain Gaussian filtering: visual quality, computational cost, edge preservation
3. **Phase-Magnitude Experiment**
 - o Swap phase/magnitude between two images; reconstruct and analyze results
 - o Quantify perceptual similarity using SSIM; discuss implications for compression and restoration

Questions for Self-Check:

1. Why does zero-padding an image before DFT reduce spectral leakage?
2. How does the order of a Butterworth filter affect the trade-off between sharp cutoff and ringing?
3. What does the phase spectrum encode that makes it more critical than magnitude for image recognition?

Report Requirements:

- Mathematical derivation of filter transfer functions
- Spectral visualizations with annotated frequency axes
- Quantitative comparison: PSNR, SSIM, runtime for spatial vs. frequency filtering
- Analysis of phase-magnitude experiment with perceptual examples

LABORATORY WORK №3

Wavelet Transforms and Multiscale Analysis

Objective:

Master multiresolution image representations by implementing wavelet decomposition, analyzing coefficient sparsity, and applying thresholding-based denoising techniques.

Competencies Assessed: OPC-1.1, OPC-2.1

Theoretical Background:

- Discrete wavelet transform (DWT) vs. continuous wavelet transform
- Mallat's fast wavelet algorithm: filter bank implementation, approximation and detail coefficients
- Haar and Daubechies wavelet families: regularity, vanishing moments, and compact support
- Sparse representation of piecewise-smooth signals in wavelet domain
- Thresholding strategies: hard vs. soft thresholding, universal threshold (VisuShrink), SURE-based adaptive thresholding

Tasks:

1. **2D Haar Wavelet Implementation**
 - o Implement forward and inverse 2D Haar DWT from scratch using separable filtering and subsampling
 - o Visualize approximation (LL) and detail (LH, HL, HH) coefficients at multiple decomposition levels
 - o Verify perfect reconstruction by computing L2 error between original and reconstructed image
2. **Wavelet Thresholding Denoising**
 - o Add controlled Gaussian and salt-and-pepper noise to test images
 - o Implement soft and hard thresholding on detail coefficients using global and level-dependent thresholds
 - o Compare PSNR/SSIM against spatial-domain Gaussian and median filters; analyze edge preservation vs. smoothing trade-off
3. **Wavelet Packet Analysis**
 - o Extend DWT to wavelet packet decomposition (splitting both approximation and detail nodes)
 - o Select optimal basis using entropy criterion; apply to texture-rich images
 - o Document computational complexity and memory footprint compared to standard DWT

Questions for Self-Check:

1. Why do wavelets provide superior compression and denoising performance for piecewise-smooth images compared to Fourier basis?
2. How does the number of vanishing moments in a Daubechies wavelet affect its ability to represent polynomial trends?
3. What is the mathematical justification for soft thresholding promoting sparsity while maintaining continuity?

Report Requirements:

- Mathematical derivation of filter coefficients for Haar and DB2 wavelets
- Coefficient visualization at each decomposition level with energy distribution analysis

- Quantitative comparison table: PSNR, SSIM, runtime for wavelet vs. spatial filters under varying noise levels
- Critical discussion: limitations of fixed-threshold methods, potential for adaptive Bayesian thresholding

Recommended Literature:

- Mallat, S. *A Wavelet Tour of Signal Processing*. 3rd Edition. Academic Press, 2008.
- Gonzalez & Woods, Ch. 7 (Wavelets and Multiresolution Processing)
- Szeliski, Sec. 3.2 (Image Filtering and Multiscale Representations)

LABORATORY WORK №4

Morphological Processing and Image Segmentation

Objective:

Develop practical skills in structuring element operations, region-based segmentation, and boundary refinement techniques.

Competencies Assessed: OPC-2.1, PC-1.2

Theoretical Background:

- Binary and grayscale morphology: erosion, dilation, opening, closing, hit-or-miss transform
- Morphological gradient, top-hat, and black-hat operators for feature extraction
- Distance transform and skeletonization
- Watershed algorithm: marker-controlled segmentation, over-segmentation mitigation
- Graph-based and variational segmentation: active contours (snakes), level sets, graph-cut formulation

Tasks:

1. **Custom Morphological Operators**
 - o Implement erosion and dilation using arbitrary structuring elements (disk, line, cross)
 - o Build opening/closing pipelines; demonstrate noise removal and shape smoothing on binary and grayscale images
 - o Compare performance of flat vs. non-flat structuring elements on medical/textural data
2. **Watershed Segmentation Pipeline**
 - o Compute gradient magnitude image and apply morphological reconstruction to suppress local minima
 - o Implement marker-controlled watershed using distance transform and regional maxima
 - o Evaluate segmentation accuracy against manual annotations using Intersection-over-Union (IoU) and Boundary F1 score
3. **Active Contour Refinement**
 - o Implement a simplified Chan-Vese active contour model for region-based segmentation
 - o Compare watershed vs. active contours on images with weak boundaries and intensity inhomogeneity
 - o Analyze parameter sensitivity (weight between image energy and regularization)

Questions for Self-Check:

1. Why does morphological opening remove small bright objects while preserving shape and size of larger structures?
2. How does the distance transform help generate reliable markers for watershed segmentation?
3. What are the computational and numerical stability challenges in level-set vs. parametric active contours?

Report Requirements:

- Structuring element design rationale and visual impact on different image classes
- Segmentation overlay comparisons with quantitative metrics (IoU, boundary accuracy)

- Parameter sweep plots showing segmentation stability under threshold/regularization changes
- Discussion of failure cases: touching objects, low contrast, intensity drift

Recommended Literature:

- Gonzalez & Woods, Ch. 9 (Morphological Image Processing)
- Sonka, Hlavac & Boyle, Ch. 10 (Segmentation)
- Davies, *Computer and Machine Vision*, Sec. 6.3–6.5 (Watershed and Active Contours)

LABORATORY WORK №5

Edge and Keypoint Detection

Objective:

Implement gradient-based edge detectors and second-derivative corner detectors, evaluating their geometric invariance properties.

Competencies Assessed: OPC-1.1, OPC-2.1

Theoretical Background:

- First-order gradient operators: Sobel, Prewitt, Scharr
- Canny edge detector pipeline: Gaussian smoothing, gradient computation, non-maximum suppression, hysteresis thresholding
- Second-order derivative methods: Laplacian of Gaussian (LoG), Difference of Gaussians (DoG)
- Harris corner detector: structure tensor, response function, eigenvalue analysis, rotation invariance proof
- Keypoint localization: sub-pixel refinement, scale-space extrema detection basics

Tasks:

1. **Canny Edge Detector from Scratch**
 - o Implement full Canny pipeline without cv2.Canny; include anisotropic Gaussian smoothing and directional gradient computation
 - o Design adaptive hysteresis thresholds based on gradient histogram percentiles
 - o Evaluate edge completeness and thinness against ground-truth boundaries
2. **Harris Corner Detector**
 - o Compute structure tensor using Sobel gradients; implement Gaussian windowing and determinant/trace response calculation
 - o Apply non-maximum suppression to extract stable corner points
 - o Visualize response maps and overlay detected keypoints on original images
3. **Invariance Analysis**
 - o Apply affine transformations (rotation, scaling, shear) and illumination changes to test images
 - o Compute repeatability rate and localization error for Harris vs. Canny-derived corners
 - o Analyze failure modes under perspective distortion and motion blur

Questions for Self-Check:

1. Why is non-maximum suppression critical for edge thinning, and how does it interact with gradient direction quantization?
2. How does the Harris response function achieve rotation invariance while remaining sensitive to intensity changes?
3. What are the theoretical limitations of first-order detectors in low-texture or highly noisy regions?

Report Requirements:

- Step-by-step pipeline visualization with intermediate gradient/response maps
- Repeatability vs. transformation plots (angle/scale vs. detected point overlap)
- Precision-recall analysis for edge detection under varying SNR
- Critical comparison: computational cost, parameter sensitivity, suitability for downstream matching

Recommended Literature:

- Forsyth & Ponce, Sec. 4.1–4.3 (Edge and Corner Detection)
- Gonzalez & Woods, Ch. 10 (Image Segmentation and Feature Extraction)
- Lowe, D. *Distinctive Image Features from Scale-Invariant Keypoints* (for historical context)

LABORATORY WORK №6

Descriptor Construction and Image Matching

Objective:

Build local feature descriptors, implement robust matching strategies, and evaluate geometric verification pipelines.

Competencies Assessed: OPC-2.1, PC-1.2

Theoretical Background:

- Descriptor design principles: gradient orientation histograms, spatial binning, normalization
- SIFT/SURF vs. binary descriptors (BRIEF, ORB): accuracy vs. speed trade-offs
- Feature matching metrics: Euclidean distance, Hamming distance, ratio test
- Geometric verification: fundamental matrix estimation, RANSAC framework, inlier consensus
- Matching evaluation: precision, recall, ROC curves, epipolar consistency

Tasks:

1. **ORB Descriptor Implementation**
 - o Implement oriented BRIEF descriptor using FAST keypoints and intensity comparison tests
 - o Apply rotation normalization using intensity centroid method
 - o Compare descriptor distinctiveness against SIFT and SURF on controlled image pairs
2. **Matching Pipeline with Ratio Test and RANSAC**
 - o Implement cross-check ratio test to filter ambiguous matches
 - o Estimate fundamental matrix using normalized 8-point algorithm + RANSAC
 - o Visualize inliers/outliers and compute epipolar error distribution
3. **Descriptor Evaluation Under Degradations**
 - o Test matching performance on images with blur, JPEG compression, viewpoint change, and illumination shift
 - o Plot ROC curves for different matching thresholds; compute area under curve (AUC)
 - o Conduct ablation study: impact of descriptor length, bin count, and RANSAC iteration budget

Questions for Self-Check:

1. Why does the ratio test significantly reduce false matches in texture-rich scenes?
2. How does coordinate normalization improve numerical stability in the 8-point algorithm?
3. What are the failure conditions for RANSAC when inlier ratio drops below 10%?

Report Requirements:

- Descriptor histogram visualizations and Hamming/Euclidean distance distributions
- Matching visualization with color-coded inliers/outliers
- ROC curves and AUC metrics across degradation types
- Critical analysis: when to prefer binary vs. floating-point descriptors, RANSAC tuning guidelines

Recommended Literature:

- Hartley & Zisserman, Ch. 4 (Two-View Geometry), Ch. 13 (Robust Estimation)
- Rublee et al., *ORB: an efficient alternative to SIFT or SURF* (ICCV 2011)
- Szeliski, Sec. 4.1–4.3 (Feature Matching and Geometric Verification)

LABORATORY WORK №7

Single and Two-View Geometry: Calibration & Epipolar Geometry

Objective:

Master camera calibration, intrinsic/extrinsic parameter estimation, and epipolar constraint validation.

Competencies Assessed: PC-1.2, UC-1.1

Theoretical Background:

- Pinhole camera model: projection matrix, homogeneous coordinates, scale ambiguity
- Intrinsic parameters: focal length, principal point, skew, aspect ratio
- Lens distortion: radial and tangential models, undistortion pipeline
- Epipolar geometry: essential vs. fundamental matrix, epipolar lines, epipoles, coplanarity constraint
- Calibration methods: Zhang's planar pattern approach, direct linear transform (DLT), bundle adjustment basics

Tasks:

1. **Zhang's Calibration Implementation**
 - o Extract chessboard corner coordinates from multiple views using `cv2.findChessboardCorners`
 - o Compute homography between pattern and image plane; decompose to obtain initial intrinsics/extrinsics
 - o Refine parameters via Levenberg-Marquardt optimization; report reprojection error
2. **Fundamental Matrix Estimation**
 - o Implement normalized 8-point algorithm with coordinate centering and scaling
 - o Apply RANSAC to handle outliers; enforce rank-2 constraint via SVD
 - o Visualize epipolar lines and verify coplanarity constraint on matched point pairs
3. **Distortion Correction and Validation**
 - o Estimate radial/tangential distortion coefficients; generate undistorted image grid
 - o Compare straight-line preservation before/after correction using vanishing point analysis
 - o Quantify calibration accuracy using held-out views and ground-truth measurements

Questions for Self-Check:

1. Why is coordinate normalization critical before applying the 8-point algorithm?
2. How does the rank-2 constraint on the fundamental matrix enforce geometric consistency?
3. What are the degeneracy conditions in planar calibration, and how to avoid them in practice?

Report Requirements:

- Calibration matrix results with physical interpretation (focal length in pixels/mm, principal point offset)
- Distortion correction overlays and vanishing point convergence analysis
- Epipolar line visualization with error histograms (algebraic vs. geometric distance)
- Critical discussion: calibration stability under limited viewpoints, impact on downstream 3D tasks

Recommended Literature:

- Hartley & Zisserman, Ch. 2 (Single View Geometry), Ch. 4 (Two-View Geometry), Ch. 11 (Calibration)
- Forsyth & Ponce, Sec. 5.1–5.4 (Camera Models and Calibration)
- Zhang, Z. *A Flexible New Technique for Camera Calibration* (IEEE TPAMI, 2000)

LABORATORY WORK №8

End-to-End Processing Pipeline (Project Workshop)

Objective:

Integrate classical image processing and geometric vision modules into a complete, reproducible pipeline for a real-world task.

Competencies Assessed: OPC-2.1, PC-1.2

Theoretical Background:

- Pipeline architecture design: data flow, error propagation, modular abstraction
- Hybrid classical/deep approaches: geometric priors for neural networks, learned feature extraction with classical verification
- Reproducibility practices: version control, environment specification, automated testing, documentation standards
- Performance evaluation: benchmark datasets, ablation studies, failure mode analysis

Tasks:

Select one project track and implement a complete pipeline:

- **Track A: Structure-from-Motion (SfM)**
Feature matching → fundamental matrix estimation → triangulation → incremental bundle adjustment → sparse 3D reconstruction
- **Track B: Variational Image Restoration**
ROF model implementation → adaptive regularization based on local gradient statistics → parameter optimization → quantitative benchmarking on medical/noisy datasets
- **Track C: Hybrid Image Registration**
Classical geometric feature extraction → neural descriptor refinement → RANSAC verification → transformation estimation → alignment quality assessment

All tracks require: modular code design, configuration management, automated metric computation, and comprehensive documentation.

Questions for Self-Check:

1. How does error accumulation in multi-stage pipelines affect final reconstruction/registration accuracy?
2. When should classical geometric constraints be preferred over end-to-end learning, and vice versa?
3. What are the minimum requirements for a reproducible computer vision experiment in a collaborative environment?

Report Requirements:

- Pipeline architecture diagram with data flow and dependency mapping
- Quantitative results on benchmark datasets (reprojection error, PSNR/SSIM, registration accuracy)
- Ablation study: impact of removing/modifying individual modules
- Failure mode analysis: degenerate configurations, noise sensitivity, computational bottlenecks
- Discussion of hybrid extension potential: where classical methods outperform or complement deep learning

Recommended Literature:

- Szeliski, *Computer Vision: Algorithms and Applications* (Full pipeline chapters)
- Hartley & Zisserman, Ch. 18–19 (Structure from Motion, Multi-View Geometry)
- Course textbooks (Gonzalez & Woods, Forsyth & Ponce) for foundational references
- Selected arXiv papers on hybrid classical/neural pipelines (to be provided via LMS)

List of references recommended for use on this topic

Essential References:

1. Gonzalez, R. C., & Woods, R. E. *Digital Image Processing*. 4th Edition. Pearson, 2018.
2. Forsyth, D., & Ponce, J. *Computer Vision: A Modern Approach*. 3rd Edition. Pearson, 2019.
3. Hartley, R., & Zisserman, A. *Multiple View Geometry in Computer Vision*. 2nd Edition. Cambridge University Press, 2004.

Supplementary References:

1. Szeliski, R. *Computer Vision: Algorithms and Applications*. 2nd Edition. Springer, 2022. (Open access)
2. Mallat, S. *A Wavelet Tour of Signal Processing*. 3rd Edition. Academic Press, 2008.
3. Sonka, M., Hlavac, V., & Boyle, R. *Image Processing, Analysis, and Machine Vision*. 4th Edition. Cengage, 2014.

Online Resources:

- OpenCV Documentation: <https://docs.opencv.org> (reference only)
- scikit-image Gallery: https://scikit-image.org/docs/stable/auto_examples
- Middlebury Stereo Dataset: <https://vision.middlebury.edu/stereo>
- KITTI Vision Benchmark: <http://www.cvlibs.net/datasets/kitti>