- Theory: Homogeneous coordinates, projective transformations
- Practice: Implement homography estimation
- Resources: Hartley & Zisserman Ch. 2

Day 5-6: Camera Models (8 hours)

- Theory: Pinhole camera, intrinsic/extrinsic parameters
- Practice: Camera calibration with OpenCV
- Project: Calibrate your phone camera

Programming Assignments

```
python

# Assignment 1: 2D transformations and image warping
# Assignment 2: Homography estimation using RANSAC
# Assignment 3: Camera calibration pipeline
```

Module 3: Signal Processing Fundamentals (Week 3)

Primary Textbooks

1. **"Digital Signal Processing" by Oppenheim & Schafer** (Chapters 1-4, 7-8)
2. **"Digital Image Processing" by Gonzalez & Woods** (Chapters 3-5)

Online Courses

1. **Signals and Systems (MIT 6.003)** - MIT OpenCourseWare
   - Focus on Fourier analysis modules
2. **Digital Signal Processing (Coursera - École Polytechnique Fédérale de Lausanne)**

Practical Resources

- **SciPy Signal Processing**: Documentation and tutorials
- **PyImageSearch**: Practical image processing tutorials

Week 3 Schedule (20 hours)

Day 1-2: Fourier Transforms (6 hours)

- Theory: DFT, FFT, frequency domain analysis
- Practice: Implement FFT-based filtering
- Resources: MIT 6.003 Lectures on Fourier analysis

Day 3-4: Convolution and Filtering (6 hours)

- Theory: Convolution theorem, filter design
- Practice: Implement various image filters
- Code: Use scipy.signal for filter design

Day 5-6: Sampling and Reconstruction (8 hours)

- Theory: Nyquist theorem, aliasing, interpolation
- Practice: Image resizing and anti-aliasing
- Project: Build a simple image editor

Programming Assignments

```python
# Assignment 1: FFT-based image filtering
# Assignment 2: Custom convolution implementation
# Assignment 3: Multi-scale image processing
```

Concrete Study Materials and Resources

Essential Software Setup

```bash
# Python environment setup
conda create -n cv_math python=3.9
conda activate cv_math
pip install numpy scipy matplotlib opencv-python jupyter
pip install sympy plotly ipywidgets  # for interactive notebooks
```

Recommended Books (Priority Order)

1. **Gilbert Strang - "Linear Algebra and Its Applications"** ($50-80)
2. **Hartley & Zisserman - "Multiple View Geometry"** ($80-120)
3. **Gonzalez & Woods - "Digital Image Processing"** ($60-100)

Free Alternatives

- **Linear Algebra**: MIT 18.06 notes (free PDF)
- **Geometry**: Computer Vision Online textbook by Szeliski (free)
- **Signal Processing**: Think DSP by Allen Downey (free online)

Online Platforms

1. **MIT OpenCourseWare** (Free)

- 18.06 Linear Algebra
- 6.003 Signals and Systems

2. **Coursera** ($39-79/month)
   - Linear Algebra for Machine Learning (Imperial College)
   - Digital Signal Processing (EPFL)
3. **edX** ($50-100 per course)
   - MIT Introduction to Computational Thinking

Interactive Tools

- **Jupyter Notebooks**: For all programming assignments
- **Desmos Graphing Calculator**: For visualizing transformations
- **GeoGebra**: For geometric intuition
- **Wolfram Alpha**: For checking mathematical calculations

Daily Study Routine

Recommended Schedule (20 hours/week)

- **Morning (2 hours)**: Theory reading and note-taking
- **Afternoon (2 hours)**: Video lectures and online content
- **Evening (1 hour)**: Programming assignments and practice

Study Techniques

1. **Active Learning**: Implement concepts immediately in code
2. **Visual Learning**: Draw diagrams for geometric concepts
3. **Spaced Repetition**: Review previous day's material each morning
4. **Project-Based**: Build small projects to reinforce concepts

Assessment and Milestones

Week 1 Checkpoint: Linear Algebra Mastery

- [ ] Implement PCA from scratch
- [ ] Understand geometric meaning of eigenvalues
- [ ] Solve image compression using SVD
- [ ] Quiz: 20 multiple-choice questions on linear algebra

Week 2 Checkpoint: Geometry Proficiency

- [ ] Calibrate a camera using checkerboard pattern

- [ ] Implement homography estimation
- [ ] Understand projective transformations
- [ ] Project: Create a simple augmented reality app

Week 3 Checkpoint: Signal Processing Skills

- [ ] Design and implement custom image filters
- [ ] Understand frequency domain analysis
- [ ] Build a noise reduction algorithm
- [ ] Project: Create a simple photo enhancement tool

Final Assessment

- **Comprehensive Project**: Build a basic image stitching application that combines all three modules
- **Theory Exam**: 50 questions covering all mathematical foundations
- **Code Review**: Submit all programming assignments for peer review

Troubleshooting Common Issues

If You're Struggling with Linear Algebra:

- Start with Khan Academy's Linear Algebra course
- Use 3Blue1Brown for visual intuition
- Practice with smaller matrices first

If Geometry Seems Abstract:

- Use physical objects to understand transformations
- Work with simple 2D examples before 3D
- Implement transformations step-by-step

If Signal Processing is Overwhelming:

- Start with 1D signals before images
- Use audio examples for intuition
- Focus on practical applications first

Next Steps Preparation

- Set up development environment for Phase 2
- Download OpenCV datasets
- Familiarize yourself with computer vision terminology
- Join computer vision communities (Reddit, Discord, Stack Overflow)
