

# GPU Video Panorama Stitcher

## Requirements & Installation Guide

### 1. System Requirements

Component	Minimum	Recommended
OS	Ubuntu 20.04 / Windows 10	Ubuntu 22.04 / Windows 11
Python	3.8+	3.10+
RAM	8 GB	16 GB+
GPU	NVIDIA GTX 1060 (6GB)	NVIDIA RTX 3060+ (8GB+)
CUDA	11.0+	12.0+
Storage	5 GB free	20 GB+ SSD

### 2. Python Dependencies

Package	Version	Purpose
numpy	≥1.21.0	Array operations and numerical computing
opencv-python	≥4.8.0	Computer vision (CPU version)
opencv-contrib-python	≥4.8.0	Additional CV modules (SIFT)
psutil	≥5.9.0	CPU and RAM monitoring
pynvml	≥11.5.0	NVIDIA GPU monitoring

### 3. GPU Acceleration Dependencies

Component	Purpose
NVIDIA Driver (≥525)	GPU hardware access

CUDA Toolkit (≥11.8)	GPU computing framework
cuDNN (≥8.6)	Deep learning primitives
opencv-python (CUDA build)	GPU-accelerated OpenCV
FFmpeg (with NVENC)	GPU video encoding

## 4. Installation Commands

### 4.1 Basic Python Packages (CPU)

Install core dependencies using pip:

```
# Install basic requirements
pip install numpy>=1.21.0
pip install opencv-python>=4.8.0
pip install opencv-contrib-python>=4.8.0
pip install psutil>=5.9.0
pip install pynvml>=11.5.0

# Or install all at once
pip install numpy opencv-python opencv-contrib-python psutil pynvml
```

### 4.2 NVIDIA Driver (Ubuntu)

```
# Check current driver
nvidia-smi

# Install latest driver (Ubuntu)
sudo apt update
sudo apt install nvidia-driver-535

# Reboot after installation
sudo reboot
```

### 4.3 CUDA Toolkit

```
# Download and install CUDA (Ubuntu)
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-keyring_1.1-1_all.deb
sudo dpkg -i cuda-keyring_1.1-1_all.deb
sudo apt update
sudo apt install cuda-toolkit-12-2

# Add to PATH (~/.bashrc)
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH

# Verify installation
nvcc --version
```

### 4.4 OpenCV with CUDA Support

#### Option A: Install pre-built wheel (easiest):

```
# Uninstall CPU version first
pip uninstall opencv-python opencv-contrib-python

# Install CUDA-enabled OpenCV (if available for your system)
pip install opencv-contrib-python-headless
```

#### Option B: Build from source (recommended for full GPU support):

```
# Install build dependencies
sudo apt install build-essential cmake git pkg-config
sudo apt install libjpeg-dev libpng-dev libtiff-dev
sudo apt install libavcodec-dev libavformat-dev libswscale-dev
sudo apt install libv4l-dev libxvidcore-dev libx264-dev
sudo apt install libgtk-3-dev libatlas-base-dev gfortran

# Clone OpenCV
git clone https://github.com/opencv/opencv.git
git clone https://github.com/opencv/opencv_contrib.git

# Build with CUDA
cd opencv && mkdir build && cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE \
      -D CMAKE_INSTALL_PREFIX=/usr/local \
      -D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib/modules \
      -D WITH_CUDA=ON \
      -D CUDA_ARCH_BIN=8.6 \
      -D WITH_CUDNN=ON \
      -D OPENCV_DNN_CUDA=ON \
      -D ENABLE_FAST_MATH=ON \
      -D CUDA_FAST_MATH=ON \
      -D WITH_NVCUVID=ON \
      -D BUILD_opencv_python3=ON ..

make -j$(nproc)
sudo make install
```

## 4.5 FFmpeg with NVENC

```
# Ubuntu - Install FFmpeg with NVIDIA support
sudo apt install ffmpeg

# Verify NVENC support
ffmpeg -encoders | grep nvenc

# Expected output should include:
# V....D h264_nvenc      NVIDIA NVENC H.264 encoder
# V....D hevc_nvenc      NVIDIA NVENC HEVC encoder
```

## 5. Verify Installation

```
# Create test script: test_setup.py
import cv2
import numpy as np

print(f"OpenCV version: {cv2.__version__}")
print(f"CUDA devices: {cv2.cuda.getCudaEnabledDeviceCount()}")

if cv2.cuda.getCudaEnabledDeviceCount() > 0:
    cv2.cuda.printCudaDeviceInfo(0)
    print("\n✓ CUDA is working!")
else:
    print("\n✗ CUDA not available - will use CPU")
```

```
# Test GPU matrix
try:
    gpu_mat = cv2.cuda_GpuMat()
    gpu_mat.upload(np.zeros((100, 100), dtype=np.uint8))
    print("✓ GPU memory allocation works!")
except Exception as e:
    print(f"✗ GPU error: {e}")
```

## 6. Quick Start

```
# Run the stitcher
python final.py

# Or with custom videos
python -c "
from final import stitch_videos
stitch_videos('left.mp4', 'right.mp4', 'output.mp4')
"
```

## 7. requirements.txt

Create this file for pip installation:

```
numpy>=1.21.0
opencv-contrib-python>=4.8.0
psutil>=5.9.0
pynvml>=11.5.0
```

Install with:

```
pip install -r requirements.txt
```