Computer Vision System Toolbox

Design and simulate computer vision and video processing systems

Computer Vision System Toolbox™ provides algorithms, functions, and apps for the design and simulation of computer vision and video processing systems. You can perform object detection and tracking, feature detection and extraction, feature matching, stereo vision, camera calibration, and motion detection tasks. The system toolbox also provides tools for video processing, including video file I/O, video display, object annotation, drawing graphics, and compositing. Algorithms are available as MATLAB® functions, System objects™, and Simulink® blocks.

For rapid prototyping and embedded system design, the system toolbox supports fixed-point arithmetic and automatic C-code generation.

**Key Features**

- Object detection, including Viola-Jones and other pretrained detectors
- Object tracking, including Kanade-Lucas-Tomasi (KLT) and Kalman filters
- Feature detection, extraction, and matching, including FAST, BRISK, MSER, and HOG
- Camera calibration for single and stereo cameras, including automatic checkerboard detection and an app for workflow automation
- Stereo vision, including rectification, disparity calculation, and 3D reconstruction
- Support for C-code generation and fixed-point arithmetic with code generation products
- Video processing, object annotation, video file I/O, video display, graphic overlays, and compositing

**Feature Detection, Extraction, and Matching**

Computer Vision System Toolbox provides a suite of feature detectors and descriptors. Additionally, the system toolbox provides functionality to match two sets of feature vectors and visualize the results.

When combined into a single workflow, feature detection, extraction, and matching can be used to solve many computer vision design challenges, such as image registration, stereo vision, object detection, and tracking.

**Feature Detection and Extraction**

A feature is an interesting part of an image, such as a corner, blob, edge, or line. Feature extraction enables you to derive a set of feature vectors, also called descriptors, from a set of detected features. Computer Vision System Toolbox offers capabilities for feature detection and extraction that include:

- Corner detection, including Shi & Tomasi, Harris, and FAST methods
- **BRISK, MSER, and SURF detection** for blobs and regions
- Extraction of BRISK, FREAK, SURF, and simple pixel neighborhood descriptors
- Histogram of Oriented Gradients (HOG) feature extraction
- Visualization of feature location, scale, and orientation
**Feature Matching**

Feature matching is the comparison of two sets of feature descriptors obtained from different images to provide point correspondences between images. Computer Vision System Toolbox offers functionality for feature matching that includes:

- Configurable matching metrics, including SAD, SSD, and normalized cross-correlation
- Hamming distance for binary features
- Matching methods including Nearest Neighbor Ratio, Nearest Neighbor, and Threshold
- Multicore support for faster execution on large feature sets

Statistically robust methods like RANSAC can be used to filter outliers in matched feature sets while estimating the geometric transformation or fundamental matrix, which is useful when using feature matching for image registration, object detection, or stereo vision applications.
**Feature-Based Image Registration**

Image registration is the transformation of images from different camera views to use a unified co-coordinate system. Computer Vision System Toolbox supports an automatic approach to image registration by using features. Typical uses include video mosaicking, video stabilization, and image fusion.

Feature detection, extraction, and matching are the first steps in the feature-based automatic image registration workflow. You can remove the outliers in the matched feature sets using RANSAC to compute the geometric transformation between images and then apply the geometric transformation to align the two images.

![Feature-based registration, used for video stabilization. The system toolbox detects interest points in two sequential video frames using corner features (top); the putative matches are determined with numerous outliers (bottom left), and outliers are removed using the RANSAC method (bottom right).](image)

**Object Detection and Recognition**

Object detection and recognition are used to locate, identify, and categorize objects in images and video. Computer Vision System Toolbox provides a comprehensive suite of algorithms and tools for object detection and recognition.

**Object Classification**

You can detect or recognize an object in an image by training an object classifier using pattern recognition algorithms that create classifiers based on training data from different object classes. The classifier accepts image data and assigns the appropriate object or class label.
Motion-Based Object Detection

Motion-based object detection algorithms use motion extraction and segmentation techniques such as optical flow and Gaussian mixture model (GMM) foreground detection to locate moving objects in a scene. Blob analysis is used to identify objects of interest by computing the blob properties from the output of a segmentation or motion extraction algorithm such as background subtraction.

Feature-Based Object Detection

Feature points are used for object detection by detecting a set of features in a reference image, extracting feature descriptors, and matching features between the reference image and an input. This method of object detection can detect reference objects despite scale and orientation changes and is robust to partial occlusions.
Training Object Detectors and Classifiers

Training is the process of creating an object detector or classifier to detect or recognize a specific object of interest. The training process utilizes:

- Positive images of the object of interest at different scales and orientations
- Negative images of backgrounds typically associated with the object of interest
- Nonobjects similar in appearance to the object of interest

The system toolbox provides an [app](#) to select and assign regions of interest (ROI) and label training images.

The system toolbox provides functions to [train a Viola-Jones object detector](#) to locate any object of interest. An [app to train a detector](#) is available on File Exchange.
Process of training a cascade object detector.

Object Tracking and Motion Estimation

Computer vision often involves the tracking of moving objects in video. Computer Vision System Toolbox provides a comprehensive set of algorithms and functions for object tracking and motion estimation tasks.

Object Tracking

Computer Vision System Toolbox provides video tracking algorithms, such as continuously adaptive mean shift (CAMShift) and Kanade-Lucas-Tomasi (KLT). You can use these algorithms for tracking a single object or as building blocks in a more complex tracking system. The system toolbox also provides a framework for multiple object tracking that includes Kalman filtering and the Hungarian algorithm for assigning object detections to tracks.

KLT tracks a set of feature points from frame to frame and can be used in video stabilization, camera motion estimation, and object tracking applications.

Detected feature points (left) and tracked object using KLT (right).

CAMShift uses a moving rectangular window that traverses the back projection of an object’s color histogram to track the location, size, and orientation of the object from frame to frame.

Multiple Object Tracking Framework

Computer Vision System Toolbox provides an extensible framework to track multiple objects in a video stream and includes the following to facilitate multiple object tracking:
- **Kalman filtering** to predict a physical object’s future location, reduce noise in the detected location, and help associate multiple objects with their corresponding tracks
- Hungarian algorithm to assign object detections to tracks
- Moving object detection using blob analysis and foreground detection
- Annotation capabilities to visualize object location and add object label

Multiple objects tracked using the Computer Vision System Toolbox multiple object tracking framework. The trails indicate trajectories of tracked objects.

**Motion Estimation**

Motion estimation is the process of determining the movement of blocks between adjacent video frames. The system toolbox provides a variety of motion estimation algorithms, such as optical flow, block matching, and template matching. These algorithms create motion vectors, which relate to the whole image, blocks, arbitrary patches, or individual pixels. For block and template matching, the evaluation metrics for finding the best match include MSE, MAD, MaxAD, SAD, and SSD.

Detecting moving objects using a stationary camera. Optical flow is calculated and detected motion is shown by overlaying the flow field on top of each frame.

**Camera Calibration**

Camera calibration is the estimation of a camera’s intrinsic, extrinsic, and lens-distortion parameters. Typical uses of a calibrated camera are correction of optical distortion artifacts, estimating distance of an object from a camera, measuring the size of objects in an image, and constructing 3D views for augmented reality systems.
Computer Vision System Toolbox provides an app and functions to perform all essential tasks in the camera calibration workflow:

- Fully automatic detection and location of checkerboard calibration pattern including corner detection with subpixel accuracy
- Estimation of all intrinsic and extrinsic parameters including axis skew
- Calculation of radial and tangential lens distortion coefficients
- Correction of optical distortion
- Support for single camera and stereo calibration

The Camera Calibrator app is used to select and filter calibration images, choose the number and type of radial distortion coefficients, view reprojection errors, visualize extrinsic parameters, and export camera calibration parameters.

Camera Calibration app. You can add or remove calibration images (left), view detected corners and reprojected points (center), plot reprojection errors (top right), and visualize extrinsic parameters (bottom right).
Diameter of various objects measured using a single calibrated camera. A calibrated camera can be used to measure planar objects in world units; an example of this application is measuring the size of parts on a conveyor belt.

Camera Calibration with MATLAB
Explore camera calibration capabilities in MATLAB®. Calibrate a camera using the camera calibrator app, perform image undistortion, and measure the actual size of an object using a calibrated camera.

Stereo Vision
Stereo vision is the process of extracting the 3D structure of a scene from multiple 2D views.

Computer Vision System Toolbox provides functions and algorithms to complete the following steps in the stereo vision workflow:

- Stereo calibration
- Stereo image rectification
• Disparity map computation
• 3D scene reconstruction

**Stereo Calibration**

Stereo calibration is the process of finding the intrinsic and extrinsic parameters of a pair of cameras, as well as the relative positions and orientations of the cameras. Stereo calibration is a precursor to calibrated stereo rectification and 3D scene reconstruction. Computer Vision System Toolbox provides algorithms and functions to calibrate a pair of stereo cameras using a checkerboard calibration pattern.

Visualizing the extrinsic parameters of a pair of stereo cameras.

**Stereo Image Rectification**

Stereo image rectification transforms a pair of stereo images so that a corresponding point in one image can be found in the corresponding row in the other image. This process reduces the 2-D stereo correspondence problem to a 1-D problem, and it simplifies how to determine the depth of each point in the scene. Computer Vision System Toolbox provides functionality for stereo rectification that includes:

- Uncalibrated stereo rectification using feature matching and RANSAC to estimate the projective transform between cameras
- Calibrated stereo rectification using stereo calibration to compute the fundamental matrix
Results from uncalibrated stereo image rectification. Non-overlapping areas are shown in red and cyan.

**Disparity Computation and 3D Scene Reconstruction**

The relative depths of points in a scene are represented in a stereo disparity map which is calculated by matching corresponding points in a pair of rectified stereo images. The system toolbox provides algorithms for disparity calculation including:

- Semi-global matching
- Block matching

*Stereo disparity map (right) representing the relative depths of points in a scene (left).*

You can reconstruct the 3D structure of a scene by projecting the 2D contents of a scene to three dimensions using the disparity map and information from stereo calibration.
Reconstructing a scene using a pair of stereo images. To visualize the disparity, the right channel is combined with the left channel to create a composite (top left); also shown are a disparity map of the scene (top right) and a 3D rendering of the scene (bottom).

**Video Processing, Display, and Graphics**

Computer Vision System Toolbox provides algorithms and tools for video processing. You can read and write from common video formats, apply common video processing algorithms such as deinterlacing and chroma-resampling, and display results with text and graphics burnt into the video. Video processing in MATLAB uses System objects™, which avoids excessive memory use by streaming data for processing one frame at a time.

Video deinterlacing in MATLAB.
Video I/O

Computer Vision System Toolbox can read and write multimedia files in a wide range of formats, including AVI, MPEG, and WMV. You can stream video to and from MMS sources over the Internet or a local network. You can acquire video directly from web cameras, frame grabbers, DCAM-compatible cameras, and other imaging devices using Image Acquisition Toolbox™. Simulink users can use the MATLAB workspace as a video source or sink.

Video Display

The system toolbox includes a video viewer that lets you:

- View video streams in-the-loop as the data is being processed
- View any video signal within your code or block diagram
- Use multiple video viewers at the same time
- Freeze the display and evaluate the current frame
- Display pixel information for a region in the frame
- Pan and zoom for closer inspection as the simulation is running
- Start, stop, pause, and step through Simulink simulations one frame at a time

Model with viewers for four videos: (from left) original, estimated background, foreground pixels, and results of tracking.

Graphics

Adding graphics to video helps with visualizing extracted information or debugging a system design. You can insert text to display the number of objects or to keep track of other key information. You can insert graphics,
such as markers, lines, and polygons to mark found features, delineate objects, or highlight other key features. The system toolbox fuses text and graphics into the image or video itself rather than maintaining a separate layer. You can combine two video sources in a composite that can highlight objects or a key region.

Images with text and graphics inserted. Adding these elements can help you visualize extracted information and debug your design.

Fixed Point and Code Generation

Computer Vision System Toolbox supports the creation of system-level test benches, fixed-point modeling, and code generation within MATLAB and Simulink. This support lets you integrate algorithm development with rapid prototyping, implementation, and verification workflows.

Code Generation Support

Most System objects, functions, and blocks in Computer Vision System Toolbox can generate ANSI/ISO C code using MATLAB Coder®, Simulink Coder®, or Embedded Coder®. You can select optimizations for specific processor architectures and integrate legacy C code with the generated code to leverage existing intellectual property. You can generate C code for both floating-point and fixed-point data types. The system toolbox ships with an example that shows how to convert an algorithm created in MATLAB to C code using code generation.

Fixed-Point Modeling

Many real-time systems use hardware that requires fixed-point representation of your algorithm. Computer Vision System Toolbox supports fixed-point modeling in most blocks and System objects, with dialog boxes and object properties that help you with configuration.

System toolbox support for fixed point includes:

- Word sizes from 1 to 128 bits
- Arbitrary binary-point placement
- Overflow handling methods (wrap or saturation)
- Rounding methods, including ceiling, convergent, floor, nearest, round, simplest, and zero
Simulink model designed to create code for a specific hardware target. This model generates C code for a video stabilization system and embeds the algorithm into a digital signal processor (DSP).

Image Processing Primitives

Computer Vision System Toolbox includes image processing primitives that support fixed-point data types and C-code generation. These System objects and Simulink blocks include:

- 2-D spatial and frequency filtering
- Image preprocessing and postprocessing algorithms
- Morphological operators
- Geometric transformations
- Color space conversions

Resources

<table>
<thead>
<tr>
<th>Product Details, Examples, and System Requirements</th>
<th>Online User Community</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trial Software</td>
<td>Training Services</td>
</tr>
<tr>
<td><a href="http://www.mathworks.com/trialrequest">www.mathworks.com/trialrequest</a></td>
<td><a href="http://www.mathworks.com/training">www.mathworks.com/training</a></td>
</tr>
<tr>
<td>Sales</td>
<td>Third-Party Products and Services</td>
</tr>
<tr>
<td>Technical Support</td>
<td>Worldwide Contacts</td>
</tr>
<tr>
<td><a href="http://www.mathworks.com/support">www.mathworks.com/support</a></td>
<td><a href="http://www.mathworks.com/contact">www.mathworks.com/contact</a></td>
</tr>
</tbody>
</table>