

# A Computer Vision Toolbox

Kalle Åström, Anders Heyden, Fredrik Kahl, Rikard Berthilsson, Gunnar Sparr

Dept of Mathematics, Lund University

Box 118, S-221 00 Lund, Sweden

email: {kalle,heyden,fredrik,rikard,gunnar}@maths.lth.se

## Abstract

*Computer vision is a hot research topic. The goal is to empower a computer or robot with vision, similar to the human visual system. Some of the typical subproblems are: reconstruction of a three-dimensional scene from a set of images, recognition of previously seen objects, and calculation of camera motion relative to the scene. This has proved to be an extremely difficult task. However, some of these problems can be solved in simple cases. As an example, the calculation of three-dimensional structure and camera motion relative to the scene, the so called structure and motion problem, is more or less solved for the case of point configurations. We will present different tools, implemented in MATLAB, that performs a number of computer vision tasks. The paper also includes examples of experiments and point out possible applications.*

## 1 Introduction

Computer vision is a rapidly growing research field, with activities all over the world. The field is of importance for such various applications as autonomous vehicles, navigating with the help of images, captured by a mounted camera, and high precision measurements using images, taken by calibrated cameras. In this paper, we will present a number of numerical routines, implemented in MATLAB, that are useful in a variety of computer vision applications. The collection of routines will be called the Computer Vision Toolbox.

One of the main problems in Computer Vision is to calculate the 3D-structure of the scene and the motion of the camera from measurements in the images taken from different view-points in the scene. This problem is called *structure and motion*, referring to the fact that both the structure of the scene and the motion of the camera are calculated from image measurements only. A number of different subproblems, arising from different knowledge of the intrinsic properties of the camera (called *camera calibration*) appear.

Other important problems are to calculate the structure of the 3D-scene given the motion of the camera and to calculate the motion of the camera given the structure of the

scene. These problems are somewhat simpler than the general structure and motion problem, but are nevertheless important for navigation and obstacle avoidance. A related problem is to calibrate a camera, i.e. calculate the focal distance, the principal point etc., from images of a known scene. This calibration may be used to make a more precise reconstruction.

The problem of estimating structure and motion from image sequences are closely related to the field of closed range photogrammetry, [1]. In this field images are used to make precise measurements of three-dimensional objects from a number of images, often taken by calibrated cameras. The main difference is that in computer vision, the focus is not on accuracy, but instead on reliability and speed of calculation.

In all these problems different kind of so called *features* can be used. The most simple feature is points that are easily detected in the sequence, such as corners or painted marks. Other types of features are lines and curves, which are easier to detect and track but more difficult to use in structure and motion algorithms. In order to use our computer vision routines these features must be detected in advance and the correspondence between different images must be established. The main emphasis of our computer vision toolbox is to use these detected features to solve for structure and motion, although some simple routines for feature extraction and correspondence are included, [4].

At the department of mathematics at Lund University we are interested in the geometry and algebra of computer vision. Most of our research implementations and experiments are conducted within MATLAB. This has over the years resulted in numerous tools for analysing image geometry. We present the Computer Vision Toolbox based on recent research. Some of the topics included are

- Extraction of feature points (corners) in an image.
- Extraction of edge curves in an image.
- Intersection: calculation of 3D point coordinates assuming known camera positions.

- Resection: calculation of camera position from images of points with known 3D-coordinates.
- Calculation of multi-camera geometry.
- Calculation of structure and motion in ways that are independent of camera coordinate systems, image and point ordering.
- Calculation of so called projective structure and motion from corresponding points in two or more uncalibrated images, see Fig. 1.
- Calculation of structure and motion from corresponding curves.
- Bundle adjustment, or iterative refinement, of structure and motion estimates using non-linear optimization.
- Automatic calibration of cameras, also known as self-calibration.

The paper is organised as follows: In Section 2, we present the basic notation, which are needed to understand the algorithms. A more detailed presentation of the different algorithms we have implemented will be given in Section 3. In Section 4 two examples will be given, one showing the reconstruction using feature points and another showing the reconstruction of curves. Finally, in Section 5 some conclusions will be given. For convenience an appendix listing all algorithms are added at the end.

## 2 The pinhole camera

In this section we will briefly describe how the camera is modeled and introduce some basic terminology, which is needed in order to comprehend and utilize the toolbox.

One of the most frequently used camera models in computer vision is the so-called *pinhole camera*. It can be viewed as an perspective projection of the 3D-scene onto a plane, referred to as the image. The pinhole camera provides a good approximation to the manner in which an image is formed by viewing the 3D-world and it will be the only model we consider in this paper.

A point  $U$  in space with coordinates  $(X, Y, Z)$  can be represented in homogenous coordinates as  $U = [X \ Y \ Z \ 1]^T$ . Vectors are considered to be equal in homogenous coordinates if they are a non-zero multiple of each other. Similarly in the image, a point  $u$  with coordinates  $(x, y)$  is represented in homogenous coordinates as  $u = [x \ y \ 1]^T$ . The point  $U$  in the scene with homogenous coordinates is projected to the point  $u$  in the image according to

$$\lambda u = PU \ , \quad \lambda \neq 0 \ , \quad (1)$$

where  $P$  is a  $3 \times 4$  matrix called the camera matrix and  $\lambda$  is a scale factor. The camera projection matrix can be factorised as

$$P = K [R | (-Rc)] \ , \quad (2)$$

where  $K$  contains information about the internal calibration of the camera,  $R$  is a rotation matrix representing camera orientation and  $c$  is the camera center or focal point. A camera is said to be calibrated if  $K$  is known.

Equation (1) is fundamental in computer vision. It is a linear mapping between  $u$  and  $U$ , except for the scale factor  $\lambda$ . The image of other geometric features, like lines, conics and curves, can easily be derived from this equation. If we observe several points  $U_1, \dots, U_n$  in several images, the following notation is used for point  $j$  in image  $i$ :

$$\lambda_{i,j} u_{i,j} = P_i U_j \ , \quad \lambda_{i,j} \neq 0 \ , \quad (3)$$

As we shall see in the next section, most of our problems can be related to this equation.

## 3 The Toolbox

The computer vision toolbox presented in this paper contains a package of algorithms that solve some of the most frequently encountered problems in computer vision. The complete list of routines can be found in the appendix, which is in fact the 'Contents.m' file in Matlab. The main tasks can be divided into three problems, *intersection*, *resection* and *structure and motion*. We restrict the problem formulations to the case of points. However, in the toolbox it is also possible to use other features, such as lines, conics, curves or a combination of different features.

**Problem 1. Intersection.** Given  $m$  images of  $n$  points  $u_{1,1}, u_{1,2}, \dots, u_{m,n}$  and given the camera matrices  $P_1, P_2, \dots, P_m$ , find the 3D points in the scene  $U_1, U_2, \dots, U_n$  with corresponding scale factors  $\lambda_{1,1}, \lambda_{1,2}, \dots, \lambda_{m,n}$  that satisfy the camera equation (3).

This problem is also known as *structure from motion*. It can be solved by least-squares, since the unknowns  $U_i$  and  $\lambda_{i,j}$  are linear in equation (3).

The next problem concerns the case when the scene structure is known as well as images of it. The unknown is the camera motion.

**Problem 2. Resection.** Given  $n$  3D points in the scene  $U_1, U_2, \dots, U_n$  and  $m$  images of these points  $u_{1,1}, u_{1,2}, \dots, u_{m,n}$ , find the camera matrices  $P_1, P_2, \dots, P_m$  that satisfy the camera equation (3).

The problem is also referred to as *absolute orientation* or *motion from structure*. It can be further divided into two subgroups, the first with known camera calibration and the

second with unknown. With known calibration, it is possible to obtain solutions with less number of points than for the uncalibrated case. For example, with a calibrated camera the minimum number of points necessary in order to determine the camera orientation is 3 and only one image is needed. However, there are up to 4 possible solutions in this case.

**Problem 3. Structure and Motion.** Given  $m$  images of  $n$  points  $u_{1,1}, u_{1,2}, \dots, u_{m,n}$ , find the camera matrices  $P_1, P_2, \dots, P_m$ , the 3D points in the scene  $U_1, U_2, \dots, U_n$  and the corresponding scale factors  $\lambda_{1,1}, \lambda_{1,2}, \dots, \lambda_{m,n}$  that satisfy the camera equation (3).

This problem may at first glance look trivial, but it is only recently that robust algorithms have been presented for the general case. For minimal cases, i.e. the minimal number of points required in order to calculate the structure and motion, there may exist several solutions. In general, to get a unique solution one more point than in the minimal case is needed. These specific cases are handled with specialized algorithms. Furthermore, by assuming constant internal camera parameters between the imaging instants, it is possible to recover the intrinsic parameters of the camera as well. This is known as self-calibration.

In addition, the toolbox contains some simple routines for feature extraction and correspondence.

## 4 Examples

The use of some of the routines in the toolbox is illustrated in two examples. The first example illustrates the reconstruction of point-like features like corners. The second illustrates the reconstruction of three-dimensional curves.

### Reconstruction of points

The Computer Vision Toolbox can be used to calculate camera motion and three dimensional structure of unknown scenes. A short image sequence of 21 images is shown in Figure 1. The images are taken of an unknown, but rigid, scene. The camera motion and the internal calibration (focal length, principal point, spherical aberration, etc.) is unknown.

Special points with high local energy are extracted from each image. These points, called feature points, are often images of corners. In the top of Figure 2 the feature points in image 1 are marked. The middle diagram of Figure 2 illustrate all feature points of the first four images. The  $z$ -coordinate is used as image number.

The feature points in consecutive images contain several points that correspond to the same physical points in the scene. A preliminary correspondence is made between points in different images, [4]. This is illustrated in the lower diagram of Figure 2. Corresponding feature points have been joined with a line in the diagram. These are called feature point tracks.



Figure 1: A short image sequence containing 21 images. Nothing is known about the camera motion relative to the scene or about the shape and contents of the objects in the scene.

The routines of the Computer Vision Toolbox can now be used to calculate the motion of the camera and the three-dimensional structure of the physical points that correspond to the feature points.

The so called structure and motion problem can be solved with a variety of methods. One popular method is to study at least seven corresponding feature points in 2 images and solve for the so called *epipolar geometry* or *bilinear constraint*. This is essentially a mathematical object that contains the relative motion of the camera between two instants.

Another method which is more stable to measurement errors uses several points in many images. Both methods are, however, very sensitive to gross measurement errors. If one or several of the correspondences in forming the feature point tracks are wrong, this can be detected by the structure and motion routines.

Typically, one would select a group of feature point tracks, calculate structure and motion and then verify if these tracks are correct. If not, then another group of feature point tracks are selected until a plausible result is found.

The result can then be improved by non-linear optimization routines, so called bundle adjustment techniques.

The result after analysing some of the feature point tracks for some of the images is the three-dimensional co-

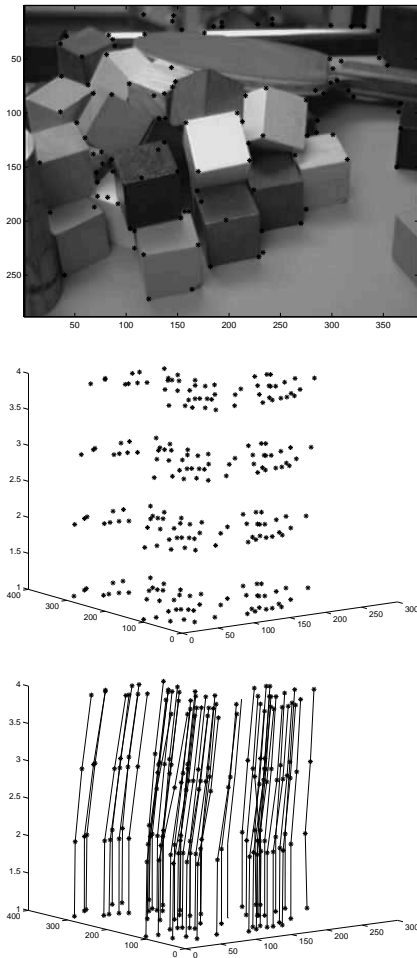


Figure 2: A so called corner detector finds interesting points in each image. These points are often corners or other specific points, that can be found in several images. The second figure shows all points from images 1, 2, 3 and 4 (the  $z$ -coordinate is image number). The program tries to find points in different images that correspond to the same physical point in the scene. This is illustrated in the last image with lines drawn between corresponding points.

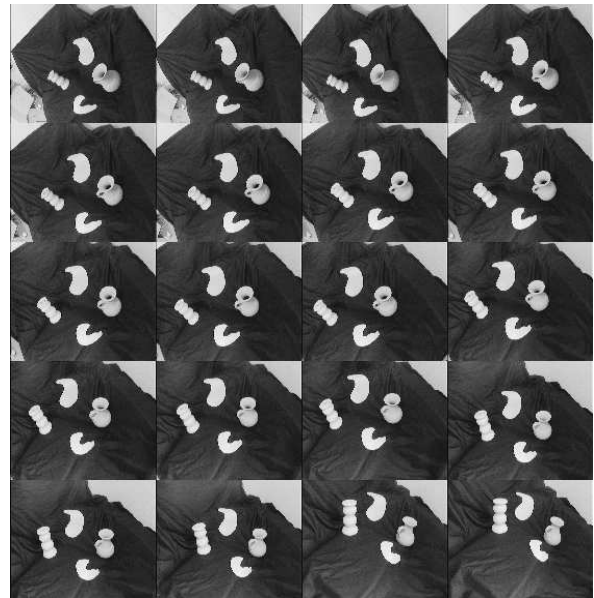


Figure 3: A short image sequence containing 20 images containing two curves and two solid objects. Nothing is known about the camera motion relative to the scene or about the shape and contents of the objects in the scene.

ordinates of these points and the relative motion of the camera for these images. The three-dimensional coordinates of the rest of the points are then calculated using intersection. The relative position of the camera for the remaining images are calculated using resection techniques.

When the full camera motion has been calculated it is possible to analyse the data in order to calibrate the camera. It is thus possible to calculate the focal length of the camera lens system and correct for non-linear distortion like spherical aberration etc.

### Reconstruction of curves

An image sequence, shown in Figure 3, contains images of two three-dimensional curves. The curves are extracted from the images using a so called edge detector. The first five images of the two curves are shown in Figure 4, where the  $z$ -coordinate is used to denote image number. It is relatively straight forward to find the correspondences between curves, i.e. to decide which curves correspond to the same space curve. It is, however, difficult to find the correspondences between individual points along the curve.

A solution to the structure and motion problem for curves is given in the computer vision toolbox. The routine calculates the three-dimensional structure of the curves and the relative motion of the camera. The algorithm is fast and gives reasonably good estimates.

Improved estimates can be obtained using a non-linear

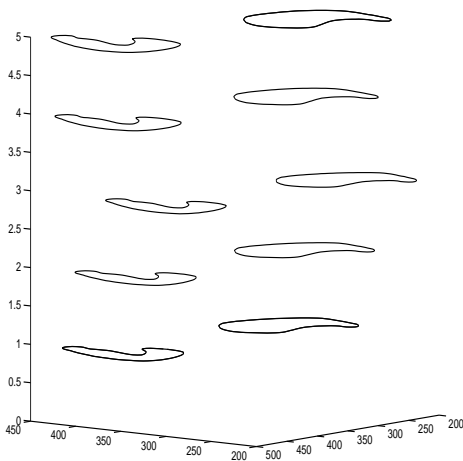


Figure 4: A so called edge detector finds high contrast edges in each image. The corresponding points along these edges are not known.

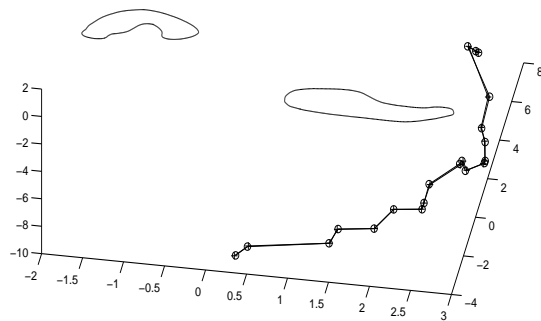


Figure 5: The computer vision toolbox has been used to calculate the three-dimensional shape of the curves and the relative motion of the camera using the curves.

optimisation routine, which is also implemented in the toolbox.

As in the previous example it is possible to analyse the result and thereby obtain estimates of camera calibration. This can be used to further improve the estimates.

Figure 5 shows the reconstructed three-dimensional curves and the position of the focal point where the 20 different images were taken.

Figure 6 show two of five images and the reconstruction of both points, curves and camera motion as obtained from the Computer Vision Toolbox.

## 5 Conclusions

In this paper we have presented a number of different routines, written in MATLAB, for solving a variety of computer vision problems. They span a wide range of applications, from extracting points and determining correspondences to structure and motion algorithms. These routines are collected in a toolbox, called the Computer Vision Toolbox. The field of computer vision is rapidly growing, both in theoretical and practical aspects and new algorithms are presented every now and then. Thus the toolbox is not static and will change over time when new algorithms are implemented.

## 6 Appendix

All routines available in the toolbox are listed below. This is a copy of the 'Contents.m' file.

```
% Computer Vision Toolbox.
% Mathematical Imaging Group,
% Dept of mathematics,
% Lund University,
% Sweden,
% Last-Edited 97-10-10.
%
% 0. Help on datastructures.
% cvnotations.m - Explanation of notations used in toolbox.
%
% 1. Intersection or structure from motion is the problem
% of calculating scene structure when the camera motion is known
% intsec1 - points, linear method
% intsec2 - points, non-linear minimisation.
% intsec3 - lines, linear method.
% intsec4 - conics.
% intsec5 - curves.
% intsec6 - use intersection to find and/or refine
% the estimates of all object positioins.
%
% 2. Resection or absolute orientation or motion from structure
% is the problem of calculating camera position orientation
% and internal calibration when something is known about
% the structure in the scene.
% a) Resection: Calibrated.
% resec1 - 3 points.
% resec2 - 3-k points and k, lines.
% resec5 - resection for each camera in a list.
% resec7 - resection adjustment for points.
%
% b) Resection: Uncalibrated.
% resec3 - 6-k points and k, lines.
% resec4 - resection for each camera in a list.
%
% 3. Structure and motion or relative orientation is the
% problem of calculating both scene structure and
% the relative positions of the cameras at the same time.
% a) Assuming known interior calibration:
% sm1 - 6 non-coplanar points in 2 images.
% sm2 - 4 coplanar points in 2 images.
%
% b) Assuming unknown and possibly varying interior calibration.
% sm3 - 7 non-coplanar points in 2 images
% sm4 - 6 non-coplanar points in 3 images
% sm5 - 8 non-coplanar points in 2 images
% sm6 - 7 non-coplanar points in 3 images
% sm7 - 6 non-coplanar points in 4 images
% sm8 - shape based factorisation method for points
% sm9 - shape based factorisation method for curves
```



Figure 6: The Computer Vision Toolbox has been used to calculate the three-dimensional structure of the points and the curve, and the relative motion of the camera using the point and curve tracks.

```

%
% 4. Bundle adjustment are routines for optimising both
% structure and motion with respect to image measurements.
% Bundle:
% bundle1 - Calbrated bundle adjustment for points
% bundle2 - Uncalibrated bundle adjustment for points
% bundle3 - Calibrated bundle adjsuement with camera distorsion for points
% bundle4 - Calbrated bundle adjustment for curves
% bundle5 - Uncalibrated bundle adjustment for curves
%
% 5. Selfcalibration or autocalibration is the
% problem of determining the interior calibration of
% the camear assuming some knowledge, e.g. that
% the intrinsic parameters are constant.
% Autocalibration
% autocalib1 - points
% autocalib2 - curves
%
% 6. Multilinear geometry.
% essentialm.m - Create the essential matrix from t and R.
% invessentialm.m - Factorize the essentiell matrix into t and R.
% fundamentalm.m - Create the fundamental matrix from R, t, K_1 och K_2.
% invfundamental.m - Factorize the fundamental matrix
% bitensor.m - (pd,pind) -> F_ij (2-image constraints)
% tritensor.m - (pd,pind) -> T_ij (3-image constraints)
% quadritensor.m - (pd,pind) -> Q_ij (4-image constraints)
% invbitensor.m - F_ij -> (pd,pind)
% invtritensor.m - T_ij -> (pd,pind)
% invquadritensor.m - Q_ij -> (pd,pind)
%
% Matrix generation and manipulation
% eulerian.m - Rotation matrix from eulerian angles?.
% randrot.m - Create a random rotation matrix.
% focalpoints.m - Calculate camera positions from camera matrices.
%
% Construct simulated data.
% simimages.m - Create images of a point configuration
%
% General routines
% pflat.m - Normalise homogeneous coordinates (Z=1)
% psphere.m - Normalise homogeneous coordinates (sphere)
% rnorm.m - Normalise shape representation
% shape.m - Calculate a basis for shape-space
% shape2.m - Calculate a basis for the depth-space
% rita2.m - Plot twodimensional point sets
% rita3.m - Plot threedimensional point sets
% fillones.m - Transform to homogeneous coordinates
% removeones.m - Transform from homogeneous coordinates
%
% Local routines
% hough - Houghtransform
% readpgm - Read a pgm image
% savepgm - Write a pgm image
% scale - Scale images
% show - Show an image
% showsc - Show a scaled image
%
% Mathematical Imaging Group at the department of mathematics,
% Lund University, Lund, Sweden.

```

## References

- [1] K. B. Atkinson, editor. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, 1996.
- [2] R. Berthilsson and K. Åström. Reconstruction of 3d-curves from 2d-images using affine shape methods for curves. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 476–481. IEEE Computer Society Press, 1997.
- [3] R. Berthilsson, K. Åström, and A. Heyden. Projective reconstruction of 3d-curves from its 2d-images using error models and bundle adjustments. In *Proc. 1st Scandinavian Conf. on Image Analysis, Lappeenranta, Finland*, pages 581–588, 1997.
- [4] P. Hallberg and A. Vegh. Automatic feature detection, tracking and iterative reconstruction from image sequences. Master's thesis, Dept. of Mathematics, Lund University, okt. 1997. To appear.
- [5] A. Heyden. Projective structure and motion from image sequences using subspace methods. In *Proc. 1st Scandinavian Conf. on Image Analysis, Lappeenranta, Finland*, pages 963–968, 1997.
- [6] A. Heyden and K. Åström. Euclidean reconstruction from constant intrinsic parameters. In *Proc. International Conference on Pattern Recognition, Vienna, Austria*, volume 1, pages 339–343. IEEE Computer Society Press, 1996.
- [7] A. Heyden and K. Åström. Algebraic properties of multilinear constraints. *Mathematical Methods in the Applied Sciences*, 20:1135–1162, 1997.
- [8] A. Heyden and K. Åström. Simplifications of multilinear forms for sequences of images. *Image and Vision Computing*, 1997. To appear.
- [9] F. Kahl and A. Heyden. Using conic correspondences to estimate the epipolar geometry. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, 1998. To appear.
- [10] F. Kahl and K. Åström. Motion estimation in image sequences using the deformation of apparent contours. In *Proc. 6th Int. Conf. on Computer Vision, Mumbai, India*, 1998. To appear.
- [11] C. C. Slama. *Manual of Photogrammetry*. American Society of Photogrammetry, Falls Church, VA, 1980.
- [12] G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *Proc. International Conference on Pattern Recognition, Vienna, Austria*, volume 1, pages 328–333. IEEE Computer Society Press, 1996.