Basic Image Analysis applied to the Visual Interaction Platform

Jean-Bernard Martens

This paper starts with a brief characterization of the field of image analysis and defines some basic terminology[a]. Next, a specific application in the field of user interface design is discussed in somewhat more detail. This latter application is of interest in its own right, but also serves to illustrate how the general characterization of the field of image analysis applies to a specific example.

[a]The first part of this paper has previously been published in Dutch as part of the ICT-Zakboekje Informatie- en Communicatietechnologie, "Patroonherkenning en beeldbewerking", p. 437-457, Koninklijke PBNA: Arnhem, 1999.

Introduction

Image analysis is concerned with the recognition of patterns in still images or image sequences. As such, it is a specific instance of **pattern recognition**, which can be defined as: the recognition of shapes, models or configurations by fully automated means (i.e., without the intervention of a human operator).

Humans are not only very good at recognizing objects in images, but can quite effortlessly also determine the three-dimensional shape and position of these objects. The complexity of these tasks becomes apparent when one tries to perform them with automated systems. Especially the ability of humans to perform pattern recognition in very diverse situations is far beyond the reach of existing systems. It should therefore be realized from the start that the current state of technology in image analysis only allows robust pattern recognition in very controlled circumstances. Some of the typical boundary conditions that allow to drastically simplify the pattern recognition are: the objects to be recognized are known a priori, and the image acquisition can be controlled such that the objects have a high contrast with respect to the background. The example that we discuss later in this paper will illustrate the importance of such boundary conditions.

In many cases where fully automated pattern recognition is not currently feasible (amongst others, many medical applications), some of the tools available within the field of image analysis (such as tools for image processing) may still be used as valuable support for a human operator (for example, to enhance the contrast and/or to reduce the noise).

Definitions and subfields

It is useful for the general discussion in this section to subdivide pattern recognition into a number of distinct but connected subfields, as shown in Figure 1. The classification in Figure 1 is sufficiently general to cover most applications of image analysis and may therefore serve as a guideline for identifying the major components in such applications. It should however be realized that not all steps mentioned in Figure 1 do necessarily occur (in a non-trivial way) in any specific application. Specialized treatments of most of the mentioned subfields can for instance be found in some of the referenced handbooks [3, 7, 1].

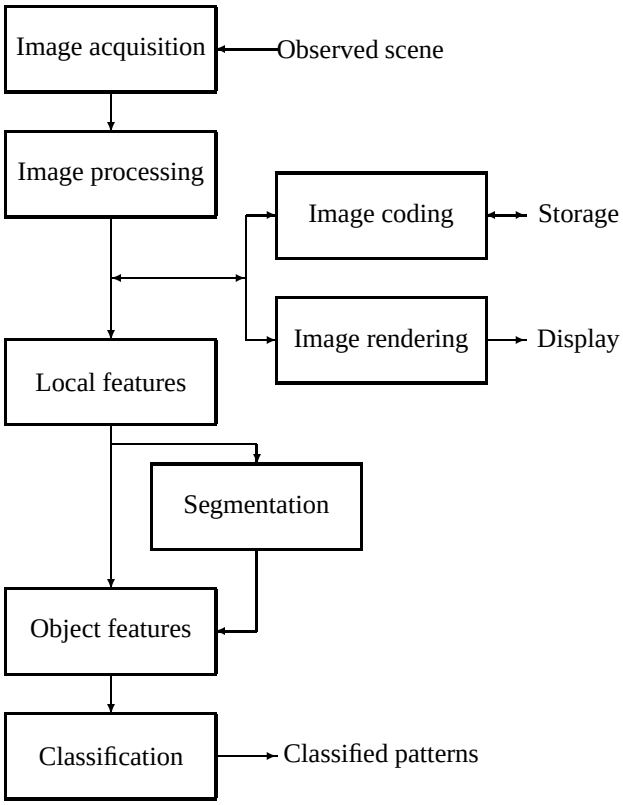


Figure 1: Relationships between important subfields of image analysis.

Image analysis starts with **image acquisition**. This involves all aspects that have to be addressed in order to obtain images, or image sequences, of the objects of interest. The selection of radiation (light) sources and sensors (such as cameras), including the choice of a suitable wavelength region for the radiation, has to be considered very carefully. Very often, this requires some in-depth understanding of the physics of the image-generating process. The geometry of the viewing situation, i.e., the relative positioning of sources and sensors with respect to the objects of interest, usually also has a major impact on the contrast between these objects and their background. Therefore, image acquisition is the most critical factor in any image analysis application. Any improvements that can be made at this stage are usually more than offset by the reduced complexity and increased robustness in successive stages.

Nowadays, images are mostly processed digitally, so that image acquisition also involves analog-to-digital conversion of the images. This means that all variations in the images in time, space and intensity (or color) are recorded with a limited **resolution** or accuracy. As a result, image sequences are subdivided into a discrete number of **frames**, where each frame consists of discrete **pixels** (picture elements). The color of these pixels is described by integer numbers with a finite number of bits. For example, digital TV signals used in broadcasting contain 25 frames per second. Each frame consists of 575 lines with 720 pixels per line for the luminance (black-white) component of the color and 360 pixels per line for each of the two color-difference components (red-green and yellow-blue). The three color components of a pixel are coded with one byte each.

In case of **tomographic** imaging techniques, the image acquisition is more complex and partly performed by a computer algorithm. Two- or three-dimensional images are reconstructed, based on one-dimensional signals or two-dimensional images from the sensors and an underlying mathematical model of the image formation. These tomographic techniques were originally developed for medical applications, but are now also increasingly being used in industrial applications.

The acquired digital images often have to be stored or processed before being used for image analysis. In some cases, (part of) the images also have to be inspected by a human observer.

Image rendering involves all aspects that need to be taken care of when converting images from digital to analog. The analog output medium may be film, paper or a display (such as a CRT, Cathode Ray Tube, or an LCD, Liquid Crystal Display). Image rendering usually results in two-dimensional images on a flat display surface, so that this is only a partial inversion of the image acquisition process, where most often objects are observed in a three-dimensional surrounding. Stereographic displays are sometimes used to partly compensate for this.

Alternative algorithms for the economical storage and transmission of digital images are developed within the field of **image coding**. An important distinction has to be made between **lossless** and **lossy** encoding of images. In case of lossless coding, the original and decoded image are identical. The encoding is based on the **redundancy** in the images, i.e., the observation that the intensity of a typical pixel can be predicted quite accurately based on the intensities of surrounding pixels. Lossy encoding, on the other hand, is based on the fact that not all image information is **relevant** to a human observer.

Recent image coding standards, such as JPEG (Joint Picture Expert Group) [4], for static images, and MPEG (Motion Picture Expert Group) [1], for image sequences, are designed such that the original and the encoded image look (almost) identical to a human observer. Much higher data reductions can obviously be accomplished with lossy encoding than with lossless encoding. Lossy encoding must however be applied with care when the images are intended for subsequent image analysis. The distortions introduced by the coding may not be objectionable to a human observer, but may nevertheless seriously hamper the feature extraction and object recognition.

In **image processing**, a digital input image is converted into a digital output image. Image processing can be used for **image enhancement** (for example, to selectively increase contrasts), for **image restoration** (for example, to correct for geometrical distortions and non-uniform lighting in the image acquisition) and for **feature extraction**. This feature extraction is often the first stage of the image analysis and is discussed in more detail below.

The detection and identification of objects in an image is a **bottom-up** process that usually starts by creating **feature images**. These feature images are mostly derived from the original image by means of **local image transformations**. This implies that the value of an output pixel is only determined by pixel values in the original image that are situated within a (small) surround of this output position. The purpose of these local image transformations is to make specific features in the image (such as edges, corners, etc) explicit. In this way, an easier distinctions between objects and the background, as well as between different objects, is pursued. An important class of local features is based on the concept of **local dimensionality**. Areas within the image for which the color is approximately constant have dimensionality zero. The average color over an area with dimensionality zero is hence a reliable feature (for example, to identify red objects in a collection of objects with different colors). Features of dimensionality one typically arise at the boundary between objects, where one of the objects may be the background. They are characterized by the fact that the image varies locally in only one direction and is constant in the orthogonal direction(s). These features can be characterized by their **contrast**, i.e., the difference between colors at both sides of the boundary, and their **orientation** (in case of image sequences, this orientation also includes information about the velocity with which the feature moves). Regions in the image that have higher dimensionality are indicative of more complex features, such as corners, junctions and textures. Feature images can be combined or processed **recursively** by means of local image transformations. This allows to derive more complex features from simple features, and is especially useful at positions where the local dimensionality is higher than one.

The feature images must allow to distinguish objects from each other and from the background. In this image **segmentation** or **labeling** stage, the image is partitioned in (non-overlapping) areas that are given the same label. Some form of **regularization** is usually required to guarantee that the segmentation is not only determined by local features but also by global, a priori determined, boundary conditions. Pixels that have approximately the same local feature values are very likely to belong to the same objects, so that a pixel-to-pixel comparison of local features is the obvious basis for segmentation. However, there are several reasons why a segmentation that is solely based on local features is often not possible or desirable. There may be *missing information* in the sense that some features (such as the local orientation) can for instance not be determined (reliably) at all pixel positions. There may also be *conflicting information* in the sense that local features at one pixel position can deviate substantially from features values in neighbouring pixels (for instance, due to noise). The global boundary conditions specify how such missing or conflicting information is to be handled. Very often segmentation is conceptually the most difficult stage in the image analysis. Several interesting mathematical approaches have been proposed, but important future developments in this area are certainly to be expected.

Once labeled areas in an image have been identified, **object features** can be determined. Some possible examples are the area and perimeter of the labeled region, average values of local features over the labeled area, etc. These object features are subsequently used to **classify** the areas. It depends on the specific application how many different classes of objects need to be distinguished, and which **interpretation** should be assigned to each of these

classes. The classification and interpretation is often derived from a **training set**. A human **supervisor** assigns the object features to different classes for a number of example images. The system derives classification rules from these examples that are subsequently applied to images when there is no human supervisor.

In the next section we will illustrate some of the stages in this general pattern classification scheme for a specific example.

Visual Interaction Platform

Platform description

Our laboratory has been involved in developing and testing interfaces that go beyond the widely used desk-top environment. The goal is to create interfaces that are more efficient and pleasant to use because they build on human skills of real world object manipulation, rather then on acquired skills of typing and mouse manipulation. I shortly describe one such possible system, called the Visual Interaction Platform (VIP), that has been realized using currently available hardware technology and image analysis software (see http://www.ipo.tue.nl/vip3 for more details).

The hardware configuration of the VIP is similar to the hardware configuration of the commercially available BUILD-IT system [5] and is shown schematically in Figure 2. A single Intel Pentium II PC operates all components in the system. The VIP uses a video projector to create a large computer workspace on the horizontal surface of a table. This horizontal workspace is called the **action-perception space**. Instead of using the traditional keyboard and mouse for interaction the user can interact (perform his/her actions) with the VIP system using physical objects such as small bricks. These bricks are coated with infrared-reflecting material and there is an infrared light source located above the table. A camera located next to the infrared light source tracks the movements of the interaction elements. The user interacts with the system by modifying the location(s) and orientation(s) of these brick(s). Unlike in the current desktop environment, where the mouse actions and the cursor movements occur at separate positions, visual feedback in the VIP system occurs at the positions occupied by the bricks. Therefore, the action and perception space [6] of the user coincide much more closely. Apart from this horizontal action-perception space, the VIP can also project a second image on a (vertically oriented) back-projection screen. This optional second image is most often used to supply the user with more extensive visual feedback for increased spatial awareness, or to communicate with remote participants. It is therefore usually referred to as the **communication space**.

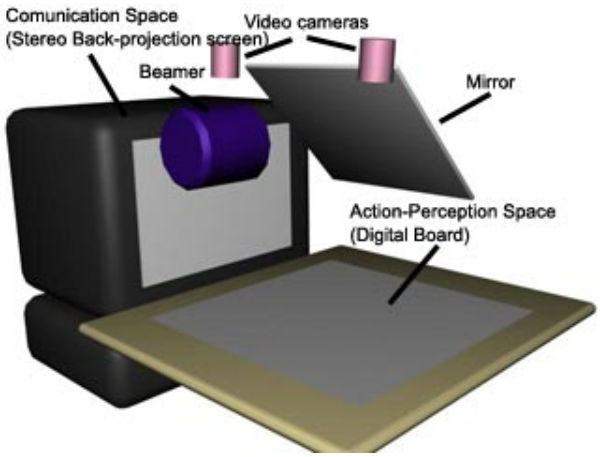


Figure 2: **VIP** Visual Interaction Platform (VIP).

The main features of the VIP are:

1. the action and perception spaces coincide;
2. two-handed interaction is possible;
3. multiple users can collectively interact at the same time, using separate interaction elements, thereby promoting group work;
4. easy-to-learn interaction style that requires little or no computer skills;
5. the users do not have to wear intrusive devices like head-mounted displays;
6. no messy wires or other system components to hinder user movements.

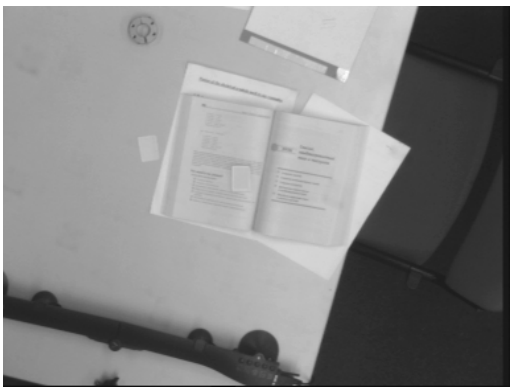
The VIP++ software [8] was developed in-house in order to simplify the programming and testing of applications that involve video-based interaction. It contains a number of utilities for acquiring, processing and storing images. For example, one of the library routines allows access to camera images from within an application program. Another routine detects the infrared-coated interaction elements and returns their position, orientation and size parameters. A third routine performs the mapping between

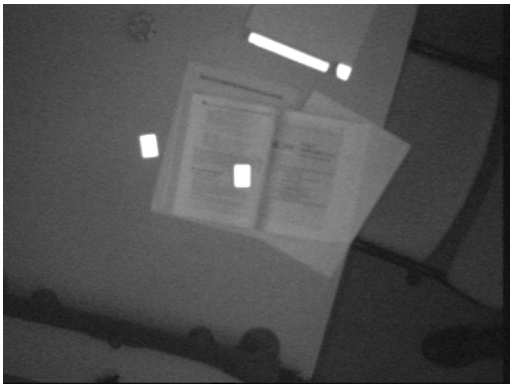
camera coordinates and projection (screen) coordinates that is required for accurate visual feedback. A fully automated calibration program can project a test pattern that is subsequently captured by the camera and analyzed to establish the transformation between both device coordinate systems. This calibration is required in order to guarantee that the visual feedback provided by the projector occurs at the actual brick positions.

Image analysis in the VIP

The image analysis in the VIP is fairly simple but nevertheless well-suited to illustrate some typical components of an image-analysis system.

The robustness and real-time implementation of the image analysis in the VIP is made possible by the image acquisition that has been optimized to simplify successive processing. In Figure 3, we show the captured image with and without the infrared filter in front of the camera. The interaction elements have a low contrast with the surrounding objects in case that no filter is used, while their detection becomes trivial in case of the filtered image. No image processing and local feature extraction need to be applied to this filtered image, since the grey value of the pixels carries sufficient information to distinguish the interaction objects from the other objects in the scene. The image segmentation reduces to a simple thresholding, followed by a labeling of connected white regions in the thresholded image. A **seed-fill algorithm** [2] is used to create connected regions. This algorithm starts by attributing a label to a white pixel that has not yet received one, and recursively assigns the same label to all the white pixels in a four-connected neighbourhood (i.e., the pixels to the north, south, east and west of the current pixel).





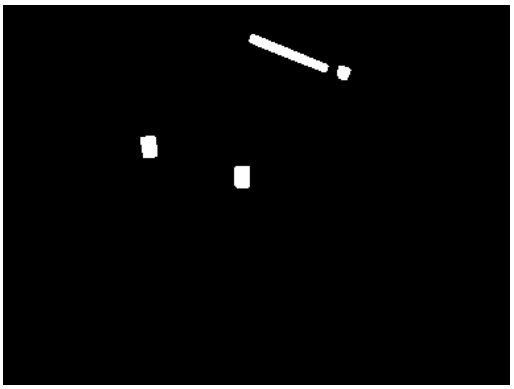


Figure 3: **Image acquisition** Image acquisition without (upper) and with (middle) infrared filter. Detection of the interaction elements can be done by simply thresholding the filtered image (bottom).

The object parameters derived for the labeled regions are: the position of the center of gravity, i.e.,

$$c_x = \frac{1}{n}\sum_{i=1}^{n} x_i,\ c_y = \frac{1}{n}\sum_{i=1}^{n} y_i, \qquad (1)$$

and the central moments

$$m_{pq} = \sum_{i=1}^{n} (x_i - c_x)^p\,(y_i - c_y)^q \qquad (2)$$

up to order $p + q = 2$, where (x_i, y_i), for $i = 1, \ldots, n$, denote the coordinates of the points that belong to one labeled region. Since the interaction elements are rectangular, the central moments can be used to derive the orientation

$$\phi = \frac{1}{2}\arctan\frac{2 \cdot m_{11}}{m_{20} - m_{02}}. \qquad (3)$$

and the width w and height h of the labeled regions [3], i.e.,

$$w \cdot h = m_{00},\ w^2 + h^2 = \frac{12 \cdot (m_{20} + m_{02})}{m_{00}}. \qquad (4)$$

The position (c_x, c_y) and orientation ϕ are available for the application program, and can for instance be used in a similar way as coordinates supplied by a mouse device[1]. The aspect ratio w/h and area $m_{00} = w \cdot h$ can be used to distinguish different interaction elements. For instance, in the example image of Figure 3, the two upper regions, which are attached to a sheet of paper, can be distinguished from the two bottom regions, which are separate (and identical) interactions elements used for selection.

Application of the VIP

One application that has been developed on the VIP is an interface for a medical image browser. Many hospitals have regular meetings to discuss patient records. During such meetings, a large light box, which is preloaded with lots of images, is currently used to view images. First, this way of working poses logistic problems (photos have to be developed or printed, collected and mounted before the meeting, usually within a limited time frame). Second, the approach only affords limited flexibility. For instance, processing images on-line and using image sequences is prohibited. Third, the viewing situation is typically such that only few participants can observe an image at the same time, which obviously is not beneficial to the discussion.

The VIP was used to demonstrate an alternative working environment for this application. The communication space of the VIP was used to project one or more images that are (currently) considered for closer examination. The action-perception space allowed to select between records of different patients, and to pick one or more images (presented by small thumb nails) for close-up viewing. A user-centered approach was used to create an interface with a minimal number of system functions. Medical experts were confronted with the system. The most important observations can be summarized as follows: 1) all users were able to use the system straight away and 2) users enjoyed the new style of interaction.

References

[1] A. Bovik. *Handbook of Image & Video Processing*. Academic Press, San Diego, 2000.

[2] P. Heckbert. *Graphic Gems I*. Academic Press, Boston, 1990.

[3] B. Jahne, H. Haußecker, and P. Geißler. *Handbook of Computer Vision and Applications*. Academic Press, San Diego, 1999.

[4] Pennebaker W.B. and Mitchell J.L. *JPEG still image compression standard*. Van Nostrand Reinhold, New York, 1993.

[5] Rauterberg, M. and Fjeld, M. and Krueger, H. and Bichsel, M. and Leonhard, U. and Meier, M. Build-it: A computer vision-based interaction technique for a planning tool. In *Proceedings of the HCI'97*, pages 303–314, Berlin, 1997. Springer Verlag.

[6] Smets, G.J.F. and Stappers, P.J. and Overbeeke, K.J. and van der Mast, C. Designing in virtual reality: perception-action coupling and affordances. In *Simulated and Virtual Realities: Elements of Perception*, pages 189–208, London, 1995. Taylor & Francis.

[7] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, San Diego, 1999.

[8] W. Wesselink. Vip++. Technical report, IPO - Center for User-System Interaction, Eindhoven University of Technology, 1999.

[1] In this case, the VIP behaves as a multi-mouse device.