Image and Movement Understanding

J. Kennedy [1], A. Migliau [2], P. Morasso [3]
G. Sandini [3], H.L. Teulings [4], D. Vernon [5]

1. Introduction

This paper reports the preliminary work done in project P419 "Image and Movement Understanding", in the framework of the Esprit programme. Since Esprit is a collaborative programme, our emphasis was directed at establishing an effective cooperetive environment, which requires common goals, a common language, and a significant degree of concurrency among subgoals and subprojects. Naturally this is also the most difficult part but we think that, beyond predictable difficulties, the constraints imposed by transnational cooperetion are extremely stimulating and worth the effort. Since the paper reports only the initial part of our cooperation, it reflects dishomogeneities and disuniformities, that we are confident to overcome in the future.

2. Integrating Low-Level Features for 3-D Shape and Motion Analysis

General problems of early processing of visual information have been studied, with the purpose of integrating different aspects, and a particular case study has been considered, namely the "bin-of-parts" problem.

With regard to early processing the following aspects have been studied:

1) Integration of edge slope and orientation;

2) Integration of edge stereo information;

3) Extraction of regional stereo information.

Integration of edge slope and orientation

The detection of luminance discontinuities often produces edge contours that are perceptually unimportant for the analysis of the scene. To try to remove from the image those "irrelevant" edges is not a simple task due to the fact that this procedure usually involves the choice of rather arbitrary thresholds to be applied to certain measures of the local luminance discontinuity. This threshold is only rarely "image indipendent" and very often its optimal choice is linked to the particular image being processed. For this reason, the most

---

(1) CAPTEC, Dublin
(2) VDS, Florence
(3) Dept. Computer Science, U. Genoa
(4) Dept. Experimental Psychology, U. Nijmegen
(5) Dept. Computer Science, Trinity College Dublin

effective thresholding algorithms relay on some information about the image which is either known a-priori or measured from the image itself. In our implementation the edge detection procedure is based on the so-called Marr-Hildreth schema, i.e. on the detection of the zero-crossings (ZC) of the second derivative of Gaussian filtered images In this approach, the filtering operator is a rotationally symmetric Laplacian of Gaussian. Along with the location of the ZC, the slope and local orientation are measured. The integration of edge slope and orientatio has been attempted by developping different classes of thresholding techniques based on edge slope and orientation. In fact, from these measures of the luminance discontinuities two classes of thresholding algorithms have been developed:
- thresholding based on local orientation;
- thresholding based on local slope.

Very often the relevant discontinuities in an image are related to straight line segments. This is not only true in the so-called "blocks world scene" but it may play a crucial role, for example, in locating structured particulars from aerial photographs. A simple method to isolate straight segments from the ZC of an image can be based on the measurement of local orientation of luminance discontinuities. This thresholding method is based on the grouping of all the contiguous ZC differing in local orientation by less than a given threshold. It is worth noting that, besides the removal of irrelevant edges, this procedure also provides a grouping of the ZC points in straight segments.

The measure of local slope of ZC codes the "sharpness" of luminance discontinuities. This parameter is of obvious importance to descibe edges according to their physical nature (e.g. shadows vs. object boundaries). On the other hand, as its measure involve the computation of the first derivarive of the filtered image, the results is affected by computational noise. For this reason the selection of a threshold usually produces highly segmented ZC segments. Recently Canny suggested the use of two thresholds and a sort of "histeresys technique" to retain only those segments whose ZC are all above the lower threshold and have at least one slope value above the higher threshold. In Canny's algorithm the lower threshold is based on a local estimate of image noise (and therefore falls into the class of adaptive thresholding algorithms) while the higher threshold is set to "two to three times" the lower one. On the other hand, while the optimal choice of the lower threshold is mainly related to acquisition and processing noise, the higher threshold should be related to the particular image being processed ("foggy areas" in aerial photographs require thresholds that are quite different from those necessary for images acquired under controllable illumination conditions) even if the acquisition and processing algorithms are the same. In our implementation, the histogram of the ZC slopes is used to compute the value of the higher threshold, which is selected in such a way that only a given percentage of ZC has a higher slope value.

Integration of Edge Stereo Information

Many analogies exists between stereo and motion analysis, at least at the earlier stages of visual processing. The analogy stressed here is the feasibility of using matching techniques derived from image sequence analysis to compute the disparity in stereo pairs. For this purpose we tryed to solve the

problem without using an explicit correspondence between stereo pairs. The proposed algorithm belongs to the class of the so called "gradient schemata" and is based on the analysis of the difference between the two images.

The algorithm is based on the following steps:

a) Convolution between each image of the sequence and a Laplacian of Gaussain operator.

b) Extraction of the zero crossing from the convolved images

c) Computation of the difference between the convolutions of two successive frames of the sequence.

d) Computation of the disparity component in the direction perpendicular to the orientation of the contour.

e) Computattion of the disparity by making the assumtion that the "true" direction of "movement" is horizontal.

f) Search for the zero crossings of the second frame projected from the first frame in the direction of the velocity.

As to stereo analysis the size of the operator is particularly important. The computation of the disparity component perpendicular to the contour, is accurate only if the magnitude of the shift is less than about half the size of the positive lobe of the convolution mask. A problem arising from this fact is that it is necessary to be able to select between the different operators, the one that best suits the disparity of a given point of the contour. For this purpose we developped a technique that can be used to determine if a given operator is appropriate for the computation of the perpendicular component of the disparity.

Extraction of Regional Stereo Information

The regional stereo algorithm used is based on the correlation between corresponding image patches of the stereo pairs in a pyramidal-multiresolution schema. This algorithm does not use a correspondence procedure and is capable of providing rough but robust range information. The use of a pyramidal, multiple-resolution structure has the advantage of providing range information at a speed inversely proportional to the resolution necessary for a given task. The main advantage is the fact that the stereo matching is not based on point features, like the zero-crossings of a band-pass filtered image, but it is based on the signs between zero-crossings. By this technique the sensitivity to noise is greatly reduced at the expenses of a lower precision in the computation. The process is organized into the following steps:

a) A stereo pair of images is acquired using two cameras and stored into two image buffers of 256x256 pixels (8 bit per pixel).

b) Each image is filtered with a Laplacian of Gaussian operator at three different scales of resolution. The filtering is performed using a pyramidal convolution scheme providing images with a size inversely proportional to the size of the convolution mask. The output of the process is a set of 3 images from each image of the stereo pair. If the size of the original image is, say, 256x256, then, due to the pyramidal organization, the size of the filtered images are 64x64, 128x128 and 256x256.

c) Each of the convolved images is clipped to produce a binary sign representation which is used afterward to perform the matching.

d) The left and right sign representations, at the coarse, medium and fine scale, are matched pair-wise to produce coarse, medium and fine disparity measures, respectively. The matching is performed first at the coarse scale, to obtain a coarse 4x4 disparity map (disparity is defined as a measure of the translation required to "register" corresponding patches of the stereo pair). The disparity information obtained at the coarse scale is used to guide the computation at the medium scale. The same matching algorithm is applied at the medium scale to produce an 8x8 disparity map. The latter is used, again, to guide the matching algorithm at the fine scale and to produce the final disparity map composed by a 16x16 array. It is worth noting that the use of a pyramidal structure greatly reduces the computation at the coarse scale. The size of the image patches is in fact constant at all the scales whereas the number of patches decreases from fine to coarse (being 256, 64 and 16 respectively). As a consequence, the basic correlation procedure is exactly the same at all the resolution scales and the computation time decreases from fine to coarse proportionally to the number of patches. Moreover, the fact that the information at the low scale is used to guide the correlation at the high scale assures that the proportional dependence of the computation load represents the "worst case" situation.

e) Computation of depth measurement from the disparity map.

The resolution in depth near the fixation point is a function of the camera parameters which can be adjusted to achieve the desired precision. In the example presented in this paper, we used a focal length lens of 50 mm. with a distance between the cameras of 23 cm. and the fixation point at about 1.5 m from the cameras. This resulted in a depth resolution of less than 2 cm. with a 256x256 stereo pair.

Bin-of-Parts: The Initial Phase.

With regards to the Bin-of-Parts problem, in the initial phase the objective of this activity is to provide a comprehensive review of current approaches to the bin-of-parts (or bin-picking) problem, together with details of reported success in the field. Additionally, an implementation of an approach to the problem, based on boundary/edge cues exclusively, is to be described in order to make explicit the problems associated with using a single simple visual cue.

Progress in this activity has so far been confined to the review task and a literature survey has commenced and is continuing. No software has yet been implemented as the availability of an integral software development system, within which to develop and integrate image understanding algorithms, is dependent on the development of a compatible software base concerning early processing.

3. Iconic representation and matching techniques

A review of the works based on 2D template-matching for accomplishing object recognition invariant to rotations and/or scaling of the input patterns

has been made. It has been noted, from the analysis of the reviewed works, that similar results have been achieved from a variety of different researches being performed and methods being implemented in various fields.

Transformations of different types have been used to solve the scale and/or rotation invariance problems. These include:

- (x,y) to (log x,log y) mapping, to transform scale changes to simple shifts in the appropriate axes.
- (x,y) to (log r,Theta) mapping, to transform scale changes to shifts in the radial axis, and rotation changes to shifts in the angular axis.

Various methods have been proposed to achieve the above transformations, from systems based entirely on optical methods, to digital transformations (which may be limited by resolution and speed). Since a preprocessing step is often needed before the template matching, for purposes like enhancement of image features (such as edges) and noise reduction, some work has also been performed on the filtering of the images, taking into account the necessity to maintain the scale and rotation invariance in order for the overall processing (filtering and template matching) to be scale and rotation invariant. The filtering operations can be performed either on the Cartesian space, or after the image transformation.

The iconic matching methods present the advantage of being adaptable to real-time image aquisition and processing systems and can be integrated with top-down procedures based on symbolic approaches to scene analysis.

Computational spaces in iconic matching.

An investigation of the problem of mapping 2D signals from the Cartesian input space to the log-polar computational space has been performed. The reason for using a log-polar computational space, is that it transfers both scale (size) and rotation variances in the Cartesian image to simple shifts of related size in the log-polar image in the radial and angular axes, making the image suitable for iconic matching by just using a single object template. Moreover, whenever the input image has to be preprocessed by a scale and rotation invariant filter (e.g. for noise reduction or edge reinforcement), such a filtering is more easily implemented in the log-polar space. In fact, the space-variant filters needed to perform these operations in the Cartesian plane become classical shift-invariant filters in the log-polar domain. There is evidence to suggest that this mapping may occur in the human peripheral visual system.

Current methods of performing the transformation of a Cartesian input space to a log-polar output space follow two directions:

- Optical methods, which operate directly on the image during aquisition.
- Digital methods, which may provide a more flexible means of performing the transformation, since the original Cartesian image taken from the same viewpoint would also be available for processing.

The digital transformations have been performed on the original Cartesian image only after aquisition. Currently, the only digital implementations of the transformations are done in software, using random access to the data structures containing the images. These methods have proved unsuitable for real-time image applications on conventional equipment because of severe limitations in speed.

Analysis of currently used methods for performing the Cartesian to log-polar space has been simplified by the limited number of methods being employed. None of these methods based on digital techniques is suitable for

transformations based on real time image processing, and even less suitable for implementation on a hardware based, multiprocessor Cartesian to log-polar mapping machine. Work has commenced on the specifications of such a technique, and the limitations imposed by such a hardware configuration.

The main objective is to suggest the foundations for the implementation of a hardware multiprocessor based device for performing Cartesian to log-polar space mapping. It is already becoming clear that parallel processing is necessary due to constraints on the computational speed of even high speed circuits, and also on the speed and methods of accessing image memory, which must be shared with I/O devices and can be accessed optimally only in a sequential format.

4. Cursive script understanding

Some general concepts about early processing and normalization have been considered.

Handwriting data

These are integer x and y samples and generalized z samples. The sampling continues even when the pen is above the paper in order to obtain continuity. The generalized z samples are either pressure (with zero or nonzero offset), pen up/down or a series of run lengths of up/down.

Preprocessing: Spectrum

Determine amplitude spectrum of velocity. Find the top of the bell-shaped spectrum which is normally around 5 Hz.

Preprocessing: Filter

Determine the appropriate low-pass filter frequency transition band and filter the data.

Preprocessing: Time normalization

Resample the original signal such that the top of the bell-shaped spectrum appears at 5 Hz.

Geometric normalizations

This phase includes all further operations, preferrably global ones, but some may also use discrete procedues, required for the most simple segmentation phase.

Geometric normalizations: Orientation

First the baseline has to be found which may be position dependent. The baseline is not some unweighted average of the signal and therefore cannot be obtained by lowpass filtering. The baseline should approximate the envelope of the bottoms of the corpi of the letters. If we can extract from the direction of the progression what is down, then one can determine all down-up transitions (e.g., by the velocity minima). Some ordinal method might then determine

a smooth continuation of these corpus bottoms. The required transformation rotates (in a position dependent way) the writing movement to a horizontal and straight base line. (It may be noted that when the horizontal progression is not known the s ignal must be processed completely normal and up-set-down and the most sensefu l interpretation should be used).

Geometric normalizations: Slant

A constant local or slowly changing transformation should transform the slant to upright. Watch for any distortions in the latter case. Normally slant is derived from the long vertical downwards segments but one could also use the direction of the tangential at the segmentation point s.

Geometric normalization: Horizontal and vertical size

The average horizontal (i.e. only the rightward ones) and vertical corpu s stroke size can be normalized to a standard length, e.g. 2 mm.

Geometric normalization: Position

Two possible choices are the following ones: take the average position, i.e. the DC component, as the zero or the first sample as the zero.

Segmentation

The segmentation task can be decomposed into a number of different computational problems.

1. Finding segmentation points

This is a form if intelligent sampling (it may be considered as an extension of the theorem of Logan to the case of planar curves). It may be noted that also beginning and end of a writ ing trace are useful segmentation points.

2. Characterizing the segments whithout loosing information

Since cursive script is a continuous function of time, then the local behavior (between one segmentation point and the next one) can be reconstructed if a sufficient number of time derivatives are known at the segmentation points. For instance, each segmentation point could be characterized by zeroth, first and second derivative which allows three degrees of freedom per segment. The aim is a data reduction and a "natural" parametrization of the recorded trajectories.

Coding

We shall try to integrate two approaches:

1. a bottom-up approach In this phase the segments are characterized in terms of writing-trace symbols. Obvious letters can be indentified.

2. a top-down approach From bigram, trigram frequencies, word frequencies, recency, or context a sequence of the most obvious alternatives is checked.

The two procedures should proceed concurrently, in a message passing fashion. In other words, the two procedures can be considered as two "actors". We shall use actor programming concepts.

We started compiling a "lexicon" for handwriting research which is reported in the appendix.

5. Cognitive Modelling

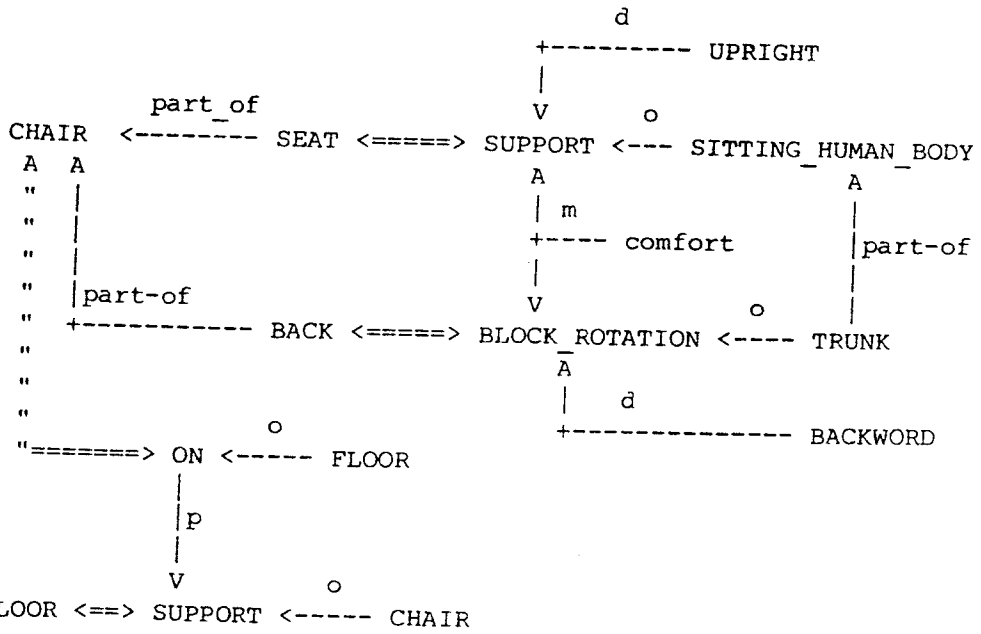
Qualitative theories for functional description of objects have been studied. In particular, three main topics were considered.

The first one was the development of a simplified prototype of a vision system, working on a block world where all the objects have plane surfaces and noiseless edges. This system makes inferences about local occlusions and groups surfaces according to its hypotheses about the 3-D structure of the scene. The resulting description is made in terms of interconnected basic volumes (typically parallelepipeds). The goal of this prototype is to serve as a working tool for a first investigation in the area of functional descriptions of physical objects. To this end a general framework has been defined, in which objects are described as conceptual graphs. The basic components of the conceptual graph are functional primitives and spatial relations. Each basic component, according to its semantics, may have one or more conceptual cases, whose fillers are identifiers of objects or parts of objects. Modifiers can also be introduced to put specific restrictions on the meaning of primitives. An example can be the functional description of a chair, shown below, where two functional primitives, SUPPORT and BLOCK_ROTATION, define the chair as that object which must support a sitting human body and block the backward rotation of his trunk; a spatial relation "ON FLOOR", which results in another functional requirement (FLOOR SUPPORT CHAIR) expresses the concept that a chair must be able to maintain a stable position on the floor, while performing its main functions. An ACTIVATION TREE encodes the graph crossing strategies. It establishes a partial order in the activation of primitives which match functional descriptions against visual patterns. The semantics of each functional primitive is embodied in a rule based expert which is able to judge if an arbitrary 3-D structure may perform the functions "it knows".

The development of these experts is the second main research topic. A first release of the support expert is currently under evaluation. It can check if an object A can be firmly supported by one or more surfaces. This checking is carried out by means of a set of hierarchically organized positioning strategies, which include special heuristics for common cases (as, for instance, a regular grating).

The third topic is an ongoing extensive analysis of those objects which are usually in a home environment, in order to find the best suited set of functional primitives and test the capabilities of the conceptual syntax. A tentative set has been defined including functions such as support, hooking, containment, rotation, sliding, block_rotation, block_sliding, closing, grasping, lifting and convey_flowing.

Furthermore, surface analysis through basic perceptual rules has been studied. A major problem in scene analysis is the recognition of partially occluded objects in a 3-D scene, what psychologists has referred to as amodal



completion. While partial occlusion is everyday experience for the human perceptual system, no proper computational theory has been implemented so far: computer vision programs make largely use of ad hoc criteria and simple heuristics which can work successfully only on restricted domains.

This report describe a first attempt to the implementation of a general model. It is based on a formal theory of visual perception, namely Structural Information Theory (henceforth SIT) which claims that the interpretation of a pattern depends strongly on the amount of its structural regularity, that's on its so-called structural information. This concept has been quantified with an encoding mechanism, providing a data structure for describing the image, and a formal language whose syntactical operators specify the regularity properties relevant for the perception process. The recovery of hidden parts of simple obscured objects is so performed choosing the solution which maximize the structural regularity.

The EXPOSE system makes use of SIT to perform a recovery of hidden parts in a 3-D scene, starting from the line-drawing of a simplified 3-D scene. The output consists of an interpretation of the occlusions in the scene based just on perceptual principles, which is meant to be the starting point for subsequent knowledge-based analysis (object recognition and so on).

The main features of the system are:

1) A domain-independent approach to 3-D partial occlusion, relying on a general theory of perception;

2) The use of regions as primitive input data. There are several reasons for this choice at this level of analysis. As Fisher points out: "the segmentation of a surface image into regions of uniform surface properties may be simpler... [while] the loss of information from using just points [or joints] makes it difficult to determine the image to model correspondences correctly [with stored models]".

3) An inference mechanism, accounting for possible modifications forced on neighbouring regions by the current completion. The solution of partial occlusion is so interpreted as a process of hypotheses setting and testing, where a backtrack is triggered in case of conflict and decision criteria are exploited to choose the right interpretation.

The system has been implemented in C_Prolog on a VAX 11/750 under UNIX operating system. Graphic output is also available on a VDS-701 system.

6. Compatible Software Base

An important goal of the whole project is to create a compatible software development base for all the partners. This involves a transfer of algorithms which were developed prior to the start of the project and also of algorithms developed during the past six months. Since hardware configurations currently differ among the partners, an essential and significant objective is the creation of a generally-useful software development environment which may be readily configured for all relevant hardware systems, facilitating rapid mutual interchange of software both now and in subsequent years. This necessitates not only a re-coding of processing software but also the specification of software standards and the specification, design, and implementation of an interactive re-configurable development system which will be used by all parties.

Work during this initial six-month period has been concerned with the objectives described above and, as such, the following sub-tasks has been completed.

A common system development environment has been identified and agreed upon: all low-level software will be written in the C programming language and software will run under the UNIX operating system.

Standard software documentation standards have been discussed by all partners and a distinct approach has been agreed upon; latitude is allowed for in-house software which might not be mutually useful.

All algorithms and software relevant to the generation of a compatible image understanding system have been identified and have been exchanged between partners. Translation of software to the agreed programming language, C, has commenced.

A provisional specification has been prepared for a virtual image understanding system. This system will allow rapid reconfiguration for all hardware systems to be used during this project and will facilitate complete software portability. A central objective of this system is that it also incorporates features to enable non-trivial integration of distinct low-level visual cues and to enable interactive investigation of the characterisitcs of such integration.

Finally, the usefulness of providing standard images for use in research by all parties has been discussed and such images are to be identified and utilised.

Presented at the Esprit Technical Week, Brussels, September 1985

APPENDIX:

A Lexicon for handwriting research in Esprit P419 IMU

It specifies the default meaning of relevant english, italian and dutch words.

Format:

English keyword.
Meaning (i=<italian>, nl=<dutch>, d=<german>).

"Cursive handwriting". Fluently connected handwriting.

"Cursive script". The result of cursive handwriting.

"Print". The result of performing block letters

Handwriting signal. The x, y and occasionally z (pen pressure or up/down information) as a function of time.

"Corpus". The central part of a letter (e.g., the a-shape contained in d, g, q; l has no corpus).

Corpus size. Vertical size of the corpus.

x-Height. (in typography) corpus size (nl=x-hoogte, d=Mittell"nge).

Writing size. See corpus size. (nl=schriftgrootte; bij voorkeur niet schrijfgrootte, i=dimensione del corpo)

"Body". Maximal vertical extension of a writing pattern (i.e., from top of ascender or corpus to bottom of descenderor corpus). (nl=corps, d=Kegel, i=corpo).

Ascender. Loop or leg (stem) above the corpus. (nl=stok, d=Oberl"nge, i=parte ascendente).

Descender. Loop and leg (stem) below the corpus. (nl=staart, d=Unterl"nge, i=parte discendente).

Aspect ratio. Ratio of width over heigth of a corpus.

Pitch. Number of letters per unit of width.

Roundness. Degree of uniformity of curvature or circle like writing trace.

Rotation. Undistorted turning of the writing trace. Counterclockwise is positive.

Orientation. Specific level of rotation.

Slant. Specific level of sheering. Forward is toward the right; backward is to the left.

Slope. See slant.

Tilt. See slant.

Context. Temporal or spatial environment of movements.

Micro context. The extent of the environment is one stroke.

Meso context. The extent of the environment is one handwriting pattern of several seconds.

Macro context. The extent of the environment is several spearate patterns.

Global. Without using segmentation information.

"Discrete". Using segmentation information.

"Local". Within segments.

Spatial. Refers to sizes and shapes of the writing trace.

Temporal. Refers to durations.

Dynamics. Refers to both spatial and temporal aspects.

"Stroke". Segment between two segmentation points. (nl=haal, i=battuta)

"Segment". See stroke.

Trace. Visible trace left behind by a moving pen on the paper.

Writing trace. See trace. (nl=schrijfspoor, i=traccia scritta).

Segmentation point. Crucial points in the writing movement used to split a movement into discrete segments (e.g., segmentation points can be defined as the minima of the absolute velocity, or the peaks of the curvature, or the moments where up and down movements join, or starting and end points of ballistic movements).

Horizontal stroke. Segment based on the horizontal component of movement.

Vertical stroke. Segment based on the vertical component of movement.

"Base line". Smoothly fitted curve through the segmentation points at the bottom of the letter corpi. (nl=basislijn, i=linea di base).

Connecting stroke. Stroke between two allographs. (nl=verbindingshaal, i=battuta di connessione).

Connection. See Connecting stroke.

Penlift. Event of the pen loosing paper contact.

Penup. State of the pen without paper contact.

Progression. Steady left-to-right movement during the writing.

Duration. Interval of time.

Time. Moment of time.

Writing time. Time needed to write a pattern.

Velocity. Vector pen speed.

Pen speed. Absolute velocity.

Tangential velocity. Absolute velocity.

Writing speed. Number of letters per unit of time. (nl=schrijfsnelheid, i=velocita` di scrittura).

Ballistic stroke. Movement without intermitting control.

Generation. Simulation of handwriting movements.

Regeneration. Resynthesis of movement information.

Fluency. Smoothness of a movement.

Reduction. Diminshing of movement information.

Digitizer. XY digitizing pad.

Pressure pen. Pen that anables also sampling of the axial pressure.

Pen pressure. Axial pressure.

Pen grip. Way of holding the pen.

X-axis. Horizontal digitizer axis.

Y-axis. Vertical digitizer axis.

Spectrum. Complex spectrum of x + i * y.

Fourier spectrum. See spectrum.

Frequency spectrum. See spectrum.

Power spectral density function. See power spectrum.

Power spectrum. Amplitude spectrum. Absolute spectrum.

Spectrum of the velocity. Complex spectrum of dx/dt + i * dy/dt.

Filter. Procedure that changes (or removes) specific characteristics of a signal

Filter frequency. The -3dB point of the lowpass frequency filter.

Sampling period. Intersample interval.

Sampling duration. Duration of the sampling.

Base-line density. Temporal frequency of samples projected on the perpendicular to the base line.

Curvature C. Inverse of the curve radius.

Curve radius R. Radius of the trace as a function of time.

Angular frequency W. Absolute velocity/curve radius as a function of time.

Running angle PHI. Cumulative of W as a function of time.

Polar distribution. Density distribution as a function of direction plotted in a polar diagram.

Phase. Time-onset difference of horizontal component relative to vertical component.