as an ellipse, the key position is determined as the directions of the ellipse's main axes. Thus the key shape can be adjusted by rotating the ellipse that defines it. Depending on the physical dimensions of the device **100**, the processing unit **600** is configured to define limits for the keyboard **104** appearance that it cannot exceed.

[**0026**] **FIG. 8** illustrates a feasible structure of the processing unit **600**. The blocks belonging to the processing unit **600** are structural entities that can be implemented e.g. as program modules, i.e. by a programming language, such as the C programming language, C++ programming language, computer language, or by an assembler, which are stored as runnable versions in a memory provided in the processor and run on the processor. Instead of translatable programming languages, other interpretable programming languages may naturally also be used, provided that they satisfy the required processing rate and capacity. When the processing unit **600** is implemented as an ASIC, the structural entities are ASIC blocks. Information is transmitted from the touch pad **106** to the processing unit **600**. If the information is not in the form of x and y coordinates, it can be converted into this form by block **800**. Block **800** may naturally also be provided in connection with the touch pad **106**, e.g. in connection with the touch screen **102** when the touch screen technique is used. The processed information, which is presented e.g. as x and y coordinates, is then supplied to block **802**, where the pressed key is identified. The information on the identified key is supplied to the application **804** that needs this information. The information on the identified key is also supplied to block **806**, which collects information and carries out an analysis on the basis of which the tactile appearance of the keyboard **104** is changed. The definitions of the new tactile keyboard **104** appearance are then supplied from block **806** to block **808**, which controls the determination of the keyboard **104** appearance. The whole block **808** or part of it may be located in connection with the touch pad **106**, e.g. in connection with the touch screen **102** when the touch screen technology is employed. As can be seen from **FIG. 8**, there is also a connection from block **802** to block **808**: tactile feedback on the keyboard use is given from the feedback unit **612**, which is provided with a connection to block **808**.

[**0027**] In an embodiment, the processing unit **600** is configured to determine the visual appearance of the keyboard **104** in addition to its tactile appearance. In that case, the visual appearance of the keyboard **104** is re-determined on the basis of the collected information and the analysis carried out so as to make the keyboard **104** more ergonomic for the user, which makes the use of the keyboard **104** easier and/or the pressing of a wrong key less likely. The determination of visual appearance is described in Finnish application 20021162, which is incorporated herein by reference. In the structure shown in **FIG. 8**, the visual appearance can be determined in block **808**, which is provided with a connection to the display **102**. In an embodiment, the processing unit **600** is configured to determine the tactile appearance of the keyboard **104** and the visual appearance of the keyboard **104** so that they correspond to each other.

[**0028**] There are several alternatives for changing the shape of a single key; for example, the key shape is changed in the main directions, i.e. in the x and y directions, or the key shape is changed arbitrarily, i.e. the key is shaped to correspond best to the manner of pressing the key, or the predetermined appearance of the key (e.g. preliminary shape and location of the key) is changed adaptively within the set limits. The processing unit **600** may be configured to move the centre point of a key according to the mean of the coordinates of key presses. The processing unit may be configured to change the key shape according to the variance of the coordinates of key presses. Some ways of changing the key shape will be discussed in greater detail below but at least the following clustering techniques can be used for changing the key shape: vector quantization VQ and expectation maximization EM. Also, other suitable adaptive and/or optimizing methods can be applied to changing the key shape.

[**0029**] In the following, tests carried out by the applicant on the new adaptive keyboard **104** will be described with reference to **FIGS. 2, 3, 4** and **5**. A portable computer provided with a touch screen was used in the test. A keyboard **104** illustrated in **FIG. 2** was created both on the left and on the right side of the touch screen. Thus the keyboard appearance consisted of adjacent rectangular keys: "1", "2", "3", "4", "5", "6", "7", "8", and "9". The keyboard **104** on the left side of the touch screen was used to simulate a situation where the user of the subscriber terminal **100** presses the keys with his left-hand thumb, and the keyboard **104** on the right side was correspondingly used to simulate a situation where the user of the subscriber terminal **100** presses the keys with his right-hand thumb.

[**0030**] Each key "1" to "9" was parameterized as shown in **FIG. 3** by using its centre point **320, 322, 324, 326, 328, 330, 334, 336**. The borderlines **300, 302, 304, 306, 308, 310, 312, 314** between the keys were defined implicitly using the Voronoi regions of the centre points known from vector quantization. For example, the Voronoi region of key "1" is the rectangular area which is defined by borderlines **300, 302, 308** and **319** and is closer to the centre point **320** of key "1" than to any other centre point. Principles of vector quantization and computing of Voronoi regions are described in Allen Gersho & Robert M. Gray: Vector Quantization and Signal Compression, The Kluwer International Series in Engineering and Computer Science, 1992, which is incorporated herein by reference. In addition to the nine adaptive centre points **320, 322, 324, 326, 328, 330, 332, 334, 336** of the keys, sixteen fixed centre points **340, 342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 363, 364, 366, 368, 370** were determined. These were associated with the value 'nil' or with the keyboard **104** border. The Voronoi regions of these fixed centre points were defined as regions outside the keyboard **104** border.

[**0031**] Then the actual test started. The test was carried out first on the keyboard **104** on the left side and then on the keyboard on the right side. A sequence consisting of four random numbers R1, R2, R3, R4 was shown to the user (on the touch screen). The random numbers were selected from 1 to 9. The user tried to key in these numbers using the keyboard **104** to be tested with his respective thumb. The feeding thus took place by pressing the keyboard **104** key in question, i.e. the Voronoi region of the code book vector associated with the key concerned.

[**0032**] The real coordinates P1=(x1, y1), P2=(x2, y2), P3=(x3, y3) and P4=(x4, y4) of each key press were saved. Each press Pi was associated with the index (i.e. key "1" to