# EXHIBIT 4.13

injuries. These concerns are addressed by capacitive touchpads which encourage neutral postures and extremely light tapping [24]. The splits in various ergonomic keyboards have successfully neutralized the awkward wrist postures of conventional keyboards [44, 65, 66, 114, 164], but less progress has been made in minimizing typing force.

Human factors studies have uncovered interesting relationships between typing force and keyswitch construction. Though make force, the force which must be applied before the key makes electrical contact, ranges between .3 and 1.0 N on most keyboards, users apply 2.5 to 7.9 times more force than necessary as a safety factor to ensure keys are always activated [43, 45, 126, 139]. The safety factor decreases with increasing make force, but overall applied force is still minimized with lower make forces [45, 126]. Other keyswitch parameters such as make point travel, the distance the key depresses before making electrical contact, and overtravel, the distance the key depresses after making electrical contact but before bottoming out, can also affect applied force. Radwin [123] found that peak applied force dropped 24% as overtravel increased from 0.0 to 3.0 mm, while pretravel had no significant effect. Gerard [45] found that typists applied roughly the same force to keyswitches with .72 N buckling springs, which produced an abrupt change in force feedback and an audible click, as they did to cheaper .28 N rubber dome keyswitches. Peak applied forces were 53% higher for rubber dome switches with .83 N make force. Finger flexor EMG's were independent of pace as long as typists worked at or below their comfortable pace, possibly because faster typists make more efficient movements [45].

Though these studies indicate which keyswitch designs will lessen typing force, the need to support resting hands and avoid accidental activation of keys ultimately limits the reduction of typing force [123] with mechanical keyswitches. Rose [127] suggests make forces less than .5 N may lead to accidental activation and an increase in static extensor contraction for postural support, but .3 N seems tolerable on ergonomic keyboards such as the Kinesis which has firm wrist rests.

## A.4 Relevance to the MTS

The approach taken in this dissertation has been to eliminate pressure switches altogether in favor of touchpad-like surface tapping and algorithmic discrimination between resting fingers and keypresses. There is no doubt subjectively that finger impact and finger flexion forces are greatly reduced when typing on the MTS. Future research should verify that this translates to reduction in pressure inside the carpal tunnel and avoidance of the critical cumulative trauma threshold. Also, future research should ensure that the MTS does not impose excessive postural demands since the hands are sometimes suspended above its surface.

# Appendix B

## VERTICAL INTERPOLATION BIASES ON PARALLELOGRAM ELECTRODE ARRAYS

Though the interspersed wedges of parallelogram electrode arrays introduced in Section 2.1.8 generally aid vertical interpolation when row spacings are larger than fingertip contact heights, they can cause misleading vertical measurements for small, light contacts not centered exactly on or between electrodes. The interpolated vertical position is always unbiased if contacts are centered on a row, but as Figure B.1 illustrates, under certain conditions it is biased when a contact is between rows. The actual positions of all contacts in Figure B.1 are halfway between the top (light gray) and bottom (dark gray) rows, and the interpolated positions of contacts a), c), d), and f) accurately reflect this because the contacts overlap equal portions of top and bottom electrodes. However, contact b) appears to overlap two to three times as much light gray as dark gray, which will cause a stronger signal on the top middle electrode and the interpolated position to be about 3/4 of the way to the top row. For an electrode row spacing of 1.2 cm this corresponds to a position error of 3-5 mm.

For the wider contact in Figure B.1e the contact overlaps of the middle and left columns are roughly equal, and only the right column is severely biased because the contact only touches the top electrode there. When the right column interpolation is averaged with the unbiased middle and left column interpolations the final reported position will be only 1-2 mm too high. For even larger contacts the

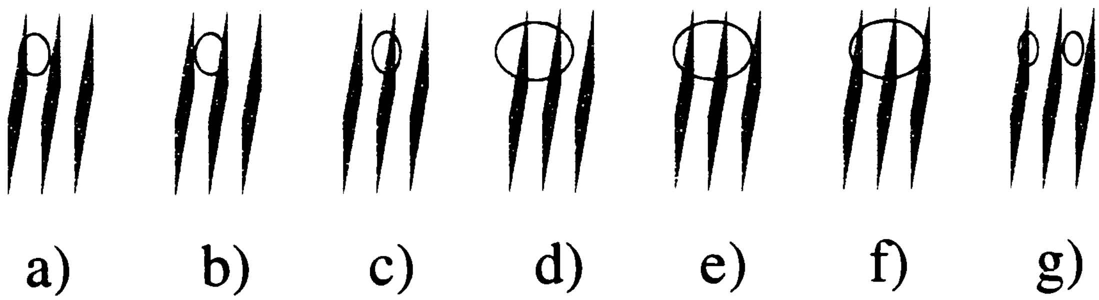



**Figure B.1:** Diagram illustrating the vertical interpolation biases which can arise when small-to-medium-sized contacts are halfway between parallelogram electrode rows but not centered on or between columns. In a) and d) small and medium contacts (ovals) are centered halfway between the left and middle columns and touch or overlap equal portions of top row electrodes (light gray) and bottom row electrodes (dark gray). In c) and f) the contacts are centered on the middle columns but still overlap equal portions of top and bottom rows, causing their vertical position to be correctly interpolated as halfway between the rows. But in b) and e) the contacts are just to the left of middle column center and overlap more of the top row electrodes than the bottom, causing their interpolated vertical position to be erroneously high. If they were just to the right of column center their interpolated vertical position would be erroneously low. The bias worsens for smaller contacts so that even though the tiny contacts in g) are centered between rows, the one on the left will be interpolated to be centered on the bottom row while the one on the right will be interpolated to be centered on the top row. Note that the parallelogram shadings in this figure are only meant to visually differentiate the top row electrodes from bottom row electrodes where the contacts overlap them, not to indicate the relative proximity measured in each row. The gaps between columns are necessary to route sensor wiring, but eliminating them would not eliminate the biases.

errors caused by biased overlap on peripheral columns become minuscule because the correct interpolations by columns toward the center of the contact dominate the centroid average.

Figure B.1g shows the worst case scenario, in which the position error can be as much as plus or minus half a row spacing. Here the contacts are so small that the left contact only overlaps the bottom left electrode, so its position will be reported as centered on the bottom row, a full 6 mm lower than the actual position, and the right contact will be reported a full 6 mm too high.

Note that the parallelograms cause no significant biases in horizontal centroid computation. This is easily proved by summing the proximity signals in each column and then interpolating horizontally using the column sums. As long as the vertical gaps between parallelograms are minuscule and the proximity sensors are linear, these computed column sums are equivalent to the proximity signal which would be obtained from a physically continuous rectangular electrode spanning the whole column, so they should produce exactly the same horizontal interpolation.

## B.1 Nonlinear Vertical Centroid for Parallelogram Interpolation

The MTS employs a number of methods to ameliorate the effects of these vertical interpolation biases. First, sensed electrode capacitance is inversely proportional to flesh proximity, so using a substantially thick dielectric cover between the fingers and electrodes weakens the signal at the center of a contact where it is firmly touching the dielectric, preventing the center from totally dominating the signals from the contact periphery. For very small, light contacts such as in Figure B.1g, this causes the contact to appear wider or more diffuse to the sensors and exhibit more moderate biases between those of Figure B.1b and Figure B.1e. When a finger is moving horizontally across the surface, the biases manifest themselves as vertical oscillations in the finger trajectory. Therefore during medium-to-high speed motion they can be removed with a low-pass motion filter.

For medium-sized contacts as in Figure B.1e, a nonlinear centroid computation which de-emphasizes the potentially biased columns at the periphery of the contact in favor of those columns at the center which are overlapped evenly can further diminish the errors by at least a factor of two. Note that in the biased right column the contact only overlaps the left side of the column at the tip of the top electrode, so the total signal from the right column is much less than that from the left or middle columns. This is the most straightforward indication that the right column is on the contact periphery and may contribute a biased vertical interpolation.

Instead of linearly weighting the contribution of each column's vertical interpolation by each column's total proximity as in a standard centroid (Equation 3.11), the following nonlinear centroid formulas square each column's total proximity to emphasize the stronger, unbiased interpolations of the central columns. Let $G_{E_j}$ be the electrodes in the intersection of group $G$ and electrode column $j$, $G_{z_j}$ be the total proximity of $G_{E_j}$, and $G_{y_j}$ be the vertical centroid as measured within the electrodes of column $j$:

$$G_{E_j} = G_E \cap \{e \in j\} \tag{B.1}$$

$$G_{z_j} = \sum_{e \in G_{E_j}} e_z \tag{B.2}$$

$$G_{y_j} = \sum_{e \in G_{E_j}} \frac{e_z e_y}{G_{z_j}} \tag{B.3}$$

Then the improved vertical centroid $G_{ypar}$ for parallelogram arrays is given by:

$$G_{zpar} = \sum_j G_{z_j}^2 \tag{B.4}$$

$$G_{ypar} = \sum_j \frac{G_{z_j}^2 G_{y_j}}{G_{zpar}} \tag{B.5}$$

Though these steps ameliorate the interpolation biases to less than a millimeter, ensuring that they are not a nuisance during evaluation of the MTS, they still cause noticeable oscillations in on-screen cursor motion given high cursor motion sensitivity and light fingertip contacts. The only known way to eliminate the biases entirely is to decrease electrode row spacing so much that parallelograms are no longer needed.

# Appendix C

## CONVERGENCE TRAPS FOR LOCALIZED COMBINATORIAL SEARCH ON AN ATTRACTOR RING.

Figures C.1 and C.2 demonstrate unfortunate cases in which a pairwise exchange sequence like bubble sort can get trapped in assignments whose cost is only a local minimum. In Figure C.1, the attractors do not lie on a perfect circle, and a single contact (circle) is poorly initialized to an attractor $x$ on the opposite side of the ring from its closest attractor $c$. For the contact's assignment to propagate around the ring to the closest attractor $c$, there must be a subsequence of the pairwise exchange sequence such that swapping in each pair of the subsequence decreases the contact's assignment distance. Moreover, this subsequence must connect the far attractor $x$ to the contact's closest attractor $c$ through overlapping attractor pairs.

In the case of Figure C.1a, where the attractors are arranged around a perfect circle, this convergence condition is met because the attractors $a$ and $d$ adjacent to the far attractor $x$ are closer to the contact than $x$. This can be verified visually by noticing that they lie within the circle of points (heavy arc) the same distance from the contact as attractor $x$. It can also be verified by noticing that these attractors are on the same side of the relevant perpendicular bisector as the contact $o$, which will cause the contact to be swapped out of the far attractor and propagate toward the near attractor $c$.

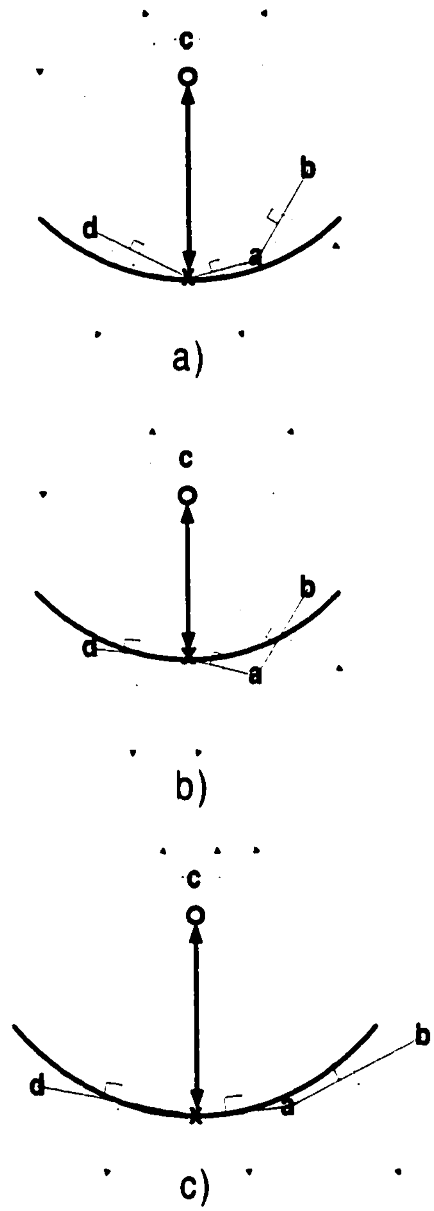



**Figure C.1:** Diagrams showing convergence failures for attractor rings which are not perfectly circular. In a) the attractor ring is perfectly circular so pairwise swapping of adjacent attractors can propagate a contact's (o) assignment from an attractor (x) on the opposite side of the ring to the closest attractor (c) on the near side. But the concavity in the ring of b) and the warping of the ring in c) trap the contact (o) in the far attractor (x) since attractors (a) and (d) adjacent to the far attractor are farther from the contact than (x).

The case of Figure C.1b contains a concavity in the attractor ring at $x$. This causes attractor $x$ to be closer to the contact $o$ than its adjacent attractors $a$ and $d$. Thus pairwise swaps with these adjacent attractors will not be accepted and the contact cannot escape from attractor $x$. This is verified visually by noting that the adjacent attractors $a$ and $d$ lie outside the equidistant arc which passes through $x$ or noting that attractor $x$ is on the same side of the adjacent perpendicular bisectors as the contact $o$.

The elliptical attractor ring of Figure C.1c shows that a concavity is not necessary to prevent convergence. Again the attractors adjacent to the far attractor are outside of the equidistant arc and perpendicular bisectors, trapping the contact's assignment on the far attractor.

Since the finger attractor ring is not perfectly circular, convergence failures similar to those of Figure C.1b and Figure C.1c have actually been observed in the identification system. Typically a fingertip contact will get trapped in a palm attractor and misidentified as a palm. This has been addressed successfully by increasing the size of the exchange neighborhood for palm attractors to include all attractors instead of just the next attractor in the ring. Thus when the exchange sequence reaches a palm attractor, pairwise swaps are considered between the palm attractor and all other attractors, and the swap which produces the minimum assignment cost over all such pairs is taken. This increases the worst case number of comparisons to $O(M^3)$, the same as algorithms for general assignment problems.

Figure C.2 shows that for attractor rings which are perfectly circular, symmetric and full of contacts, pairwise exchange can get stuck in rotational local minima. The assignment of Figure C.2a clearly has the globally lowest total assignment cost, but the rotated assignments of Figure C.2b are also a local minimum. Under the distance-squared metric, the assignments of Figure C.2b all meet the local pairwise ordering relation. All pairwise swaps such as the swap on the bottom attractors

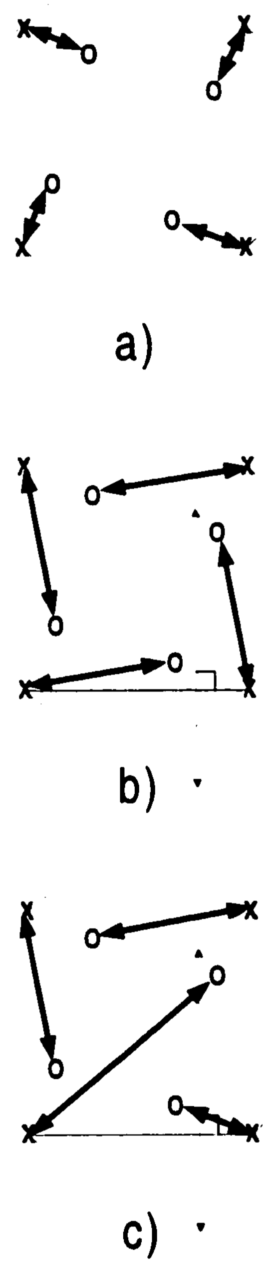



**Figure C.2:** Rotational local minimum for a perfectly symmetric attractor ring full of contacts. The global minimum assignment is shown in a). However, the rotated assignments in b) are also a stable local minimum because no pairwise swap such as that in c) will be accepted under the distance-squared metric.

in Figure C.2c would increase the assignment costs and will not be taken, trapping the assignments in this rotated state. However, this type of convergence failure has not been noticed in practice, presumably because the asymmetries in attractor spacings around the actual finger attractor ring discourage this type of convergence failure.

# BIBLIOGRAPHY


[1] Emile Aarts and Jan Karel Lenstra. *Local Search In Combinational Optimization.* John Wiley and Sons, Chichester, 1997.

[2] Johnny Accot and Shumin Zhai. Beyond Fitts' Law: Models for trajectory-based hci tasks. In *CHI '97*, pages 295–302. ACM, March 1997.

[3] Subutai Ahmad. A usable real-time 3D hand tracker. In *Proceedings of the 28th Asilomar Conference on Signals, Systems, and Computers - Part 2 (of 2)*, volume 2, Pacific Grove, CA, October 1994. IEEE.

[4] Samuel Audet. Hot Scroll 1.0 for OS/2. http://www.cam.org/~guardia, 1998.

[5] Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. The Rockin' Mouse: Integral 3D manipulation on a plane. In *Proceedings of CHI '97*, pages 311–318, Atlanta, GA USA, March 1997.

[6] Ravin Balakrishnan and I. Scott MacKenzie. Performance differences in the fingers, wrist, and forearm in computer output control. In *Proceedings of the CHI '97 Conference on Human Factors in Computing Systems*, pages 303–310, New York, 1997. ACM.

[7] Egon Balas, Donald Miller, Joseph Pekny, and Paolo Toth. A parallel shortest path algorithm for the assignment problem. Technical Report EDRC 05-38-89, Carnegie Mellon University, April 1989.

[8] M. Balinski. A competitive (dual) simplex method for the assignment problem. *Mathematical Programming*, 34:125–141, 1986.

[9] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[10] R. S. Barr, F. Glover, and D. Klingman. The alternating basis algorithm for assignment problems. *Mathematical Programming*, 13:1–13, 1977.

[11] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21:152–171, 1981.

[12] D. P. Bertsekas. A distributed asynchronous relaxation algorithm for the assignment problem. In *Proceedings of the 24th IEEE Conference on Decision and Control*, 1987.

[13] Paul J. Besl and Ramesh C. Jain. Range image segmentation. In Herbert Freeman, editor, *Machine Vision: Algorithms, Architectures, and Systems*, pages 221–256. Academic Press, Inc., New York, 1988.

[14] Eric A. Bier, Maureen C. Stone, Ken Fishkin, William Buxton, and Thomas Baudel. A taxonomy of see-through tools. In *Proceedings of CHI '94*, pages 358–364, New York, 1994.

[15] Sephen J. Bisset and Bernard Kasser. *Multiple Fingers Contact Sensing Method for Emulating Mouse Buttons and Mouse Operations on a Touch Sensor Pad.* U.S. Patent 5825352, October 20 1998.

[16] Klaus Boehm, Wolfgang Broll, and Michael Sokolewicz. Dynamic gesture recognition using neural networks; a fundament for advanced interaction construction. In *SPIE Conference on Electronic Imaging Science and Technology*, San Jose, California, February 1994.

[17] Robert A. Boie, Laurence W. Ruedisueli, and Eric R. Wagner. *Computer Mouse or Keyboard Input Device Utilizing Capacitive Sensors.* U.S. Patent 5463388, October 31 1995.

[18] J. E. Bolton and R. C. Wilkerson. Responsiveness of pain scales: a comparison of three pain intensity measures in chiropractic patients. *Journal of Manipulative and Physiological Therapeutics*, 21(1):1–7, January 1998.

[19] Grigore Burdea and Philippe Coiffet. *Virtual Reality Technology.* John Wiley and Sons, Inc., New York, 1994.

[20] Bill Buxton. Smoke and mirrors. *BYTE*, pages 205–210, July 1990.

[21] William Buxton. Chunking and phrasing and the design of human-computer dialogues. In *Information Processing 86*, pages 475–480, North Holland, 1986. Elsevier Science Publishers B.V.

[22] William Buxton, Eugene Flume, Ralph Hill, Alison Lee, and Carson Woo. Continuous hand-gesture driven input. In *Graphics Interface*, pages 191–195, Toronto, Ontario, 1983. Computer Systems Research Group, Department of Computer Science, University of Toronto.

[23] William Buxton and Brad A. Myers. A study in two-handed input. In *Proceedings of CHI '86*, pages 321–326, Toronto, Ontario, 1986.

[24] Ahmet E. Çakir, Gisela Çakir, Thomas Müller, and Pieter Unema. The Track-pad - a study on user comfort and performance. In *Proceedings of CHI '95*, http://www.acm.org/sigchi/chi95/Electronic/documnts/shortppr/aec1bdy.htm, 1995. ACM.

[25] S. K. Card, T. P. Moran, and A. Newell. *The Psychology of Human-Computer Interaction*. Erlbaum, Hillsdale, NJ, 1983.

[26] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. The design space of input devices. In M.M. Blattner and R.B. Dannenberg, editors, *Multimedia Interface Design*, pages 217–232. ACM Press, New York, 1992.

[27] G. Carpaneto and P. Toth. Primal-dual algorithms for the assignment problem. *Discrete Applied Mathematics*, 18:137–153, 1987.

[28] K. C. Chung, M. S. Pillsbury, M. R. Walters, and R. A. Hayward. Reliability and validity testing of the Michigan Hand Outcomes Questionnaire. *Journal of Hand Surgery*, 23(4):575–587, July 1998.

[29] Kinesis Corporation. Revolutionary ergonomic keyboards & high performance input accessories for PC, Macintosh, and Sun. http://www.kinesis-ergo.com/, 1999.

[30] J. L. Crowley and J. Coutaz. Vision for man machine interaction. In Leonard J. Bass and Claus Unger, editors, *Engineering for Human-Computer Interaction*, pages 28–43. Chapman and Hall, New York, 1995.

[31] U. Derigs. The shortest augmenting path method for solving assignment problems – motivation and computational experience. *Annals of Operations Research*, 4:57–102, 1985.

[32] Sarah A. Douglas and Anant Kartik Mithal. *The Ergonomics of Computer Pointing Devices*. Springer, London, 1997.

[33] J. Edmonds and R. M. Karp. Theoretical improvements on algorithmic efficiency of network flow problems. *Journal of the ACM*, 19:248–261, 1972.

[34] J. Egerváry. Matrixok kombinatorius tulajdonsagairol. *Mathematikai es Fizikai Lapok*, 38:16–26, 1931.

[35] Horst A. Eiselt, Giorgio Pederzoli, and Carl-Louis Sandblom. *Continuous Optimization Models*. Walter de Gruyter, New York, 1987.

[36] FakeSpace, Inc. Pinch Gloves. http://www.fakespace.com/prod-pnch.html.

[37] Donald L. Fisher, Robert O. Andres, David Airth, and Stephen S. Smith. Repetitive motion disorders: The design of optimal rate-rest profiles. *Human Factors*, 35(2):283–304, June 1993.

[38] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[39] George W. Fitzmaurice and William Buxton. An empirical evaluation of graspable user interfaces: Towards specialized, space-multiplexed input. In *Proceedings of CHI '97*, pages 43–50, Toronto, Ontario, March 1997.

[40] Patrick J. Franz and David H. Straayer. *Integrated Keyboard and Pointing Device System with Automatic Mode Change*. U.S. Patent 5189403, February 1993.

[41] Masaaki Fukumoto and Yoshinobu Tonomura. Body coupled fingering: Wireless wearable keyboard. In *CHI 97*, pages 147–154, Atlanta, GA, March 1997. ACM.

[42] John M. Gauch. Image segmentation and analysis via multiscale gradient watershed hierarchies. *IEEE Transactions on Image Processing*, 8(1):69–79, January 1999.

[43] Michael J. Gerard, T. J. Armstrong, J.A. Foulke, and B. J. Martin. Effects of key stiffness on force and the development of fatigue while typing. *American Industrial Hygiene Association Journal*, 57:849–854, 1996.

[44] Michael J. Gerard, Stephen K. Jones, Leo A. Smith, Robert E. Thomas, and Tai Wang. An ergonomic evaluation of the kinesis ergonomic computer keyboard. *Ergonomics*, 37(10):1661–1668, 1994.

[45] Michael James Gerard. *Effects of Keyswitch Stiffness, Typing Pace, and Auditory Feedback on Typing Force, Muscle Activity, and Subjective Discomfort*. Ph.D. dissertation, University of Michigan, 1997.

[46] George E. Gerpheide. *Electrical Charge Transfer Apparatus*. U.S. Patent 5349303, September 20 1994.

[47] George E. Gerpheide. *Methods and Apparatus for Data Input*. U.S. Patent 5305017, April 19 1994.

[48] George E. Gerpheide and Michael D. Layton. *Capacitance-Based Proximity with Interference Rejection Apparatus and Methods*. U.S. Patent 5565658, October 15 1994.

[49] Douglas J. Gillan, Kritina Holden, Susan Adam, Marianne Rudisill, and Laura Magee. How does Fitts' Law fit pointing and dragging? In *CHI '90 Proceedings*, pages 227–234. ACM, April 1990.

[50] David Gillespie, Timothy P. Allen, Robert J. Miller, and Federico Faggin. *Object Position Detector With Edge Motion Feature.* U.S. Patent 5543590. August 6 1996.

[51] David Gillespie, Timothy P. Allen, and Ralph Wolf. *Object Position Detector With Edge Motion Feature and Gesture Recognition.* U.S. Patent 5543591, August 6 1996.

[52] William G. Gillick and Clement C. Lam. *Roller Mouse for Implementing Scrolling in Windows Applications.* U.S. Patent 5530455, June 25 1996.

[53] S. A. Goldstein. *Biomechanical aspects of cumulative trauma to tendons and tendon sheaths.* Ph.D. dissertation, University of Michigan. 1981.

[54] Etienne Grandjean. *Ergonomic Design of VDT Workstations.* Taylor and Francis, Philadelphia, 1987.

[55] Yves Guiard. Asymmetric division of labor in hamn skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior*, 19(4):486–517, 1987.

[56] Handkey Corporation. Twiddler User's Manual. http://www.handykey.com/. 1997.

[57] Kostas Haris, Serafim N. Efstratiadis, Nicos Maglaveras, and Aggelos K. Katsaggelos. Hybrid image segmentation using watersheds and fast region merging. *IEEE Transactions on Image Processing*, 7(12):1684–1699, December 1998.

[58] Frederick S. Hillier and Gerald J. Lieberman. *Introduction to Operations Research.* Holden-Day, Inc., Oakland, California, 1986.

[59] W. Daniel Hillis. A high-resolution imaging touch sensor. *The International Journal of Robotics*, 1(2):33–44, 1982.

[60] Ken Hinckley. *Haptic Issues for Virtual Manipulation.* Ph.D. dissertation, University of Virginia, 1996.

[61] Ken Hinckley, Mary Czerwinski, and Mike Sinclair. Interaction and modeling techniques for desktop two-handed input. In *UIST '98 Symposium on User Interface Software and Technology*, pages 49–58, 1998.

[62] Ken Hinckley, Randy Pausch, Dennis Proffitt, James Patten, and Neal Kassell. Cooperative bimanual action. In *Proceedings of CHI '97*, pages 27–34, Atlanta, GA, March 1997.

[63] Ken Hinckley, Joe Tullio, Randy Pausch, Dennis Proffitt, and Neal Kassell. Usability analysis of 3D rotation techniques. In *Proceedings of UIST '97*, pages 1–10, Banff, Alberta, Canada, 1997.

[64] E. R. Hoffman. Fitts' Law with transmission delay. *Ergonomics*, 35:37–48, 1992.

[65] M. Honan, M. Jacobson, R. Tal, and D. Rempel. Changes in wrist postures during a prolonged typing task. In *Proceedings of the Human Factors and Ergonomics Society - 40th Annual Meeting*, pages 629–631, 1996.

[66] M. Honan, E. Serina, R. Tal, and D. Rempel. Wrist postures while typing on a standard and split keyboard. In *Proceedings of the Human Factors and Ergonomics Society - 39th Annual Meeting*, volume 1, pages 366–368, 1995.

[67] Don Hopkins. The design and implementation of pie menus. *Dr. Dobb's Journal*, December 1991.

[68] S. L. Horowitz and T. Pavlidis. A graph-theoretic approach to picture processing. *Computer Graphics and Image Processing*, 7:282–291, 1978.

[69] Infogrip, Inc. The BAT Personal Keyboard. http://www.infogrip.com/bat.htm, 1993.

[70] S. Inokuchi and R. Nevatia. Boundary detection in range pictures. In *Proceedings of the 5th International Conference Pattern Recognition*, pages 1301–1303, Miami, Florida, November 1980.

[71] Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen Jr. Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction*, 1:3–26, March 1994.

[72] R. A. Jarvis and Edward A. Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, C-22(11):1025–1034, November 1973.

[73] Herbert D. Jellinek and Stuart K. Card. Powermice and user performance. In *CHI '90 Proceedings*, pages 213–220. ACM, April 1990.

[74] Peter W. Johnson and Steven Lehman David M. Rempel. Measuring muscle fatigue during computer mouse use. In *Proceedings of the 1996 18th Annual Conference of the IEEE Engineering in Medicine Biology Society*, volume 4, pages 1454–1455, 1996.

[75] R. Jonkers and T. Volgenant. Shortest augmenting path method for dense and sparse linear assignment problems. *Computing*, 38:325–390, 1987.

[76] Paul Kabbash and William Buxton. The 'Prince' Technique: Fitts' Law and selection using area cursors. In *CHI '95 Mosaic of Creativity*, pages 273–279. ACM, May 1995.

[77] Paul Kabbash, I. Scott MacKenzie, and William Buxton. Human performance using computer input devices in the preferred and non-preferred hands. In *Proceedings of the InterCHI '93*, pages 474–481, 1993.

[78] Bernard Kasser, Bernhard Joss, and Sephen J. Bisset. *Touch Sensing Method and Apparatus*. U.S. Patent 5790107, August 4 1998.

[79] Donald E. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1973.

[80] D. König. *Theorie der endlichen und unendlichen Graphen*. Akademie-Verlag, Leipzig, 1936.

[81] Myron W. Krueger. *Artificial Reality II*. Addison-Wesley, New York, 1991.

[82] Valerie Kucharewski. Technology evaluation: One-handed typing devices. http://www.stanford.edu/~valya/hci.html, 1997.

[83] H. W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Quarterly*, 2:83–97, 1955.

[84] Shrawan Kumar. A conceptual model of overexertion, safety and risk of injury in occupational settings. *Human Factors*, 36(2):197–209, June 1994.

[85] Gordon Kurtenbach and William Buxton. The limits of expert performance using hierarchic marking menus. In *Proceedings of INTERCHI '93*, pages 482–487, April 1993.

[86] Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. The design of a gui paradigm based on tablets, two-hands and transparency. In *Proceedings of CHI '97*, pages 35–42, Atlanta, GA, March 1997.

[87] Wendi Ann Latko. *Development and Evaluation of an Observational Method for Quantifying Exposure to Hand Activity and Other Physical Stressors in Manual Work.* Ph.D. dissertation, University of Michigan, 1997.

[88] Seonkyoo Lee. A Fast Multiple-Touch-Sensitive Input Device. Master's thesis, University of Toronto, 1984.

[89] SK. Lee, W. Buxton, and K.C. Smith. A multi-touch three dimensional touch-sensitive tablet. In *Proceedings of CHI '85*, Toronto, Ontario, April 1985. ACM.

[90] Andrea Leganchuk, Shumin Zhai, and William Buxton. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Transactions on Computer Human Interaction*, http://www.dgp.utoronto.ca/people/andrea/bimanual.html. 1999. in press.

[91] Neil A. Levine. *Finger worn graphic interface device.* U.S. Patent 4954817, September 4 1990.

[92] James D. Logan. *System for Using A Touchpad Input Device for Cursor Control and Keyboard Emulation.* U.S. Patent 5666113, September 9 1997.

[93] I. Scott Mackenzie. A note on the information-theoretical basis for Fitts' Law. *Journal of Motor Behavior*, 21:323–330, 1989.

[94] I. Scott MacKenzie. Input devices and interaction techniques for advanced computing. In W. Barfield and T.A. Furness III, editors, *Virtual Environments and Advanced Interface Design*, pages 437–470. Oxford University Press, Oxford, UK, 1995.

[95] I. Scott MacKenzie and William Buxton. Extending Fitts' Law to two-dimensional tasks. In *Proceedings of the CHI '92 Conference on Human Factors in Computing Systems*, pages 219–226, New York, 1992.

[96] I. Scott MacKenzie and William Buxton. A tool for the rapid evaluation of input devices using Fitts' Law models. *SIGCHI Bulletin*, 25(3):58–63, 1993.

[97] I. Scott MacKenzie, Abigail Sellen, and William Buxton. A comparison of input devices in elemental pointing and dragging tasks. In *Proceedings of the CHI '91 Conference on Human Factors*, pages 161–166, New York, 1991. ACM.

[98] I. Scott MacKenzie, R. William Soukoreff, and Chris Pal. A two-ball mouse affords three degrees of freedom. In *Extended Abstracts of the CHI '97 Conference on Human Factors in Computing Systems*, pages 303–304, New York, 1997.

[99] I. Scott MacKenzie and Colin Ware. Lag as a determinant of human performance in interactive systems. In *INTERCHI '93*, pages 488–493. ACM, April 1993.

[100] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. A semantic analysis of the design space of input devices. *Human-Computer Interaction*, 5:145–190, 1990.

[101] Lillian G. Malt. Keyboard design in the electronic era. In *PIRA Eurotype Forum*, London Hilton Hotel, Park Lane, London http://www.teleprint.com/keyboard/history.html, September 14–15 1977. Malt Applied Systems, UK, Printing Industry Research Association.

[102] S. Martello and P. Toth. Linear assignment problems. *Annals of Discrete Mathematics*, 31:259–282, 1987.

[103] Edgar Matias, I. Scott MacKenzie, and William Buxton. Half-qwerty: A one-handed keyboard facilitating skill transfer from qwerty. In *Proceedings of the INTERCHI '93 Conference on Human Factors in Computing Systems*, pages 88–94, New York, 1994. ACM.

[104] Edgar Matias, I. Scott MacKenzie, and William Buxton. Half-QWERTY: Typing with one hand using your two-handed skills. In *Companion of the CHI '94 Conference on Human Factors in Computing Systems*, pages 51–52, New York, 1994. ACM.

[105] Edgar Matias, I. Scott MacKenzie, and William Buxton. One-handed touch-typing on a qwerty keyboard. *Human-Computer Interaction*, 11:1–27, 1996.

[106] Edgar Matias, I. Scott MacKenzie, and William Buxton. A wearable computer for use in microgravity space and other non-desktop environments. In *Companion of the CHI '96 Conference on Human Factors in Computing Systems*, pages 69–70, New York, 1996. ACM.

[107] Paul McAvinney. The Sensor Frame-a gesture-based device for the manipulation of graphic objects. Available from SensorFrame, Inc., Pittsburgh, Pa., December 1986.

[108] Paul McAvinney. Telltale gestures. *BYTE*, 15(7):237–240, July 1990.

[109] L. F. McGinnes. Implementation and testing of a primal-dual algorithm for the assignment problem. *Operations Research*, 31:277–291, 1983.

[110] Nimish Metha. A flexible human machine interface. Master's thesis, University of Toronto, 1982.

[111] Robert J. Miller, Stephen Bisset, Timothy P. Allen, and Gunter Steinbach. *Object Position and Proximity Detector.* U.S. Patent 5495077, February 27 1996.

[112] K.W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations.* Cambridge University Press, Great Britain, 1994.

[113] Gita Murthy, Norman J. Kahan, Alan R. Hargens, and David M. Rempel. Forearm muscle oxygenation decreases with low levels of voluntary contraction. *Journal of Orthopaedic Research*, 15(4):507–511, 1997.

[114] M. Nakaseko, E. Grandjean, W. Hunting, and R. Gierer. Studies on ergonomically designed alphanumeric keyboards. *Human Factors*, 27(2):175–187, 1985.

[115] Claudia Nölker and Helge Ritter. Detection of fingertips in human hand movement sequences. In Ipke Wachsmuth and Martin Fröhlich, editors, *Gesture and Sign Language in Human-Computer Interaction, Proceedings of the International Gesture Workshop*, pages 209–218. Springer, New York, 1997.

[116] A. Okabe, B. Boots, and K. Sugihara. *Spatial Tessellations - Concepts and Applications of Voronoi Diagrams.* John Wiley and Sons, Chichester, 1992.

[117] Emil Pascarelli and Deborah Quilter. *Repetitive Strain Injury: A Computer User's Guide.* John Wiley and Sons, Inc., New York, 1994.

[118] Mark Patrick and George Sachs. *X11 Input Extension Protocol Specification.* The X Consortium, ftp://ftp.x.org/pub/R6.4/xc/doc/hardcopy/Xi/proto_ps.gz, version 1.0 edition, 1992.

[119] P.C.D. Maltron Ltd. Keyboards for people with special needs. http://www.maltron.com, 1997.

[120] Marko M. Pecina, Jelena Krmpotic-Nemanic, and Andrew D. Markiewitz. *Tunnel Syndromes: Peripheral Nerve Compression Syndromes.* CRC Press, New York, 2nd edition, 1997.

[121] Francis K.H. Quek. Unencumbered gestural interaction. *IEEE Multimedia*, 3:36–47, Winter 1996.

[122] MD R. Werner, PhD T. J. Armstrong, MS C. Bir, and M.K. Aylard. Intracarpal canal pressures: The role of finger, hand, wrist and forearm position. *Clinical Biomechanics*, 12(1):44–51, 1997.

[123] Robert G. Radwin. Activation force and travel effects on overexertion in repetitive key tapping. *Human Factors*, 39(1):130–140, March 1997.

[124] Graham J. Reid, Cheryl A. Gilbert, and Patrick J. McGrath. The Pain Coping Questionnaire: preliminary validation. *Pain*, 76:83–96, 1998.

[125] David Rempel, Peter J. Keir, W. Paul Smutz, and Alan Hargens. Effects of static fingertip loading on carpal tunnel pressure. *Journal of Orthopaedic Research*, 15(3):422–426, 1997.

[126] David Rempel, Elaine Serina, and Edward Klinenberg. The effect of keyboard keyswitch make force on applied force and finger flexor muscle activity. *Ergonomics*, 40(8):800–808, 1997.

[127] M. J. Rose. Keyboard operating posture and activation force: Implications for muscle over-use. *Applied Ergonomics*, 2:198–203, 1991.

[128] David A. Rosenbaum. *Human Motor Control*. Academic Press, Inc., New York, 1991.

[129] Dean Rubine and Paul McAvinney. Programmable finger-tracking instrument controllers. *Computer Music Journal*, 14(1):26–41, 1990.

[130] Dean Harris Rubine. The automatic recognition of gestures. Master's thesis, Carnegie Mellon University, 1991.

[131] Joseph D. Rutledge and Ted Selker. Force-to-motion functions for pointing. In D. Diaper et al., editor, *Human-Computer Interaction - INTERACT '90*, pages 701–706, Yorktown N.Y., 1990. IBM T.J. Watson Research Center.

[132] Jacqueline L. Salmon. For students, a painful lesson on computers. *The Washington Post*, Sunday:A01, May 17 1998.

[133] Richard A. Schmidt. *Motor Control and Learning*. Human Kinetics Publishers, Inc., Champaign, Illinois, 1988.

[134] Elaine R. Serina, C. D. Mote Jr., and David Rempel. Force response of the fingertip pulp to repeated compression - effects of loading rate, loading angle and anthropometry. *J. Biomechanics*, 30(10):1035–1040, 1997.

[135] B. Shneiderman. An empirical comparison of pie vs. linear menus. In *Sparks of Innovation in Human-Computer Interaction*, pages 79–88. Ablex Publishers, Norwood, N.J., 1993.

[136] John L. Sibert and Mehmet Gokturk. A finger-mounted, direct pointing device for mobile computing. In *Proceedings of UIST '97 Banff*, pages 41–42, Alberta, Canada, 1997. ACM.

[137] B. A. Silverstein, L. J. Fine, and T. J. Armstrong. Hand, wrist cumulative trauma disorders in industry. *British Journal of Industrial Medicine*, 43:779–784, 1986.

[138] B. A. Silverstein, L. J. Fine, and T. J. Armstrong. Occupational factors and carpal tunnel syndrome. *American Journal of Industrial Medicine*, 11:343–358, 1987.

[139] Paula M. Sind. The effects of structual and overlay design parameters of membrane switches on the force exerted by users. In *Proceedings of the Human Factors Society - 34th Annual Meeting*, pages 380–384. Department of Industrial Engineering and Operations Research, 1990.

[140] R. William Soukoreff and I. Scott MacKenzie. Generalized Fitts' Law Model Builder. In *Companion Proceedings of the CHI '95 Conference on Human Factors in Computing Systems*, pages 113–114, New York, 1995.

[141] David Joel Sturman. *Whole-Hand Input*. Ph.D. dissertation, Massachusetts Institute of Technology, 1992.

[142] BJ Sumner. *User's Guide to Pen for OS/2*. IBM Corporation, version 1.03 edition, 1995.

[143] Synaptics, Inc. Synaptics touchpad software drivers and utilities. http://www.synaptics.com, 1998.

[144] Esa-Pekka Takala, Sirna Lammi, Hannu Nieminin, and Eira Viikari-Juntura. Electromyographic changes in the static holding test of the arm. *International Journal of Industrial Ergonomics*, 12(1-2):85–90, 1993.

[145] Mark A. Tapia and Gordon Kurtenbach. Some design refinements and principles on the appearance and behavior of marking menus. In *Proceedings of UIST '95*, pages 189–195, Pittsburgh, PA, USA, November 1995.

[146] Jochen Triesch and Christoph von der Malsburg. Robust classification of hand postures against complex backgrounds. *Proceedings of the 1996 2nd International Conference on Automatic Face and Gesture Recognition*, pages 170–175, October 14-16 1996.

[147] John Viega, Matthew Conway, George Williams, and Randy Pausch. 3D Magic Lenses. In *Proceedings of UIST '96*, pages 51–58, Seattle, Washington. 1996.

[148] Inc. VPL Research. Data Glove Model 2 User's Manual. Redwood City, CA. 1987.

[149] Wacom Technology Co. Intuos: the first intelligent graphics tablet system. http://www.wacom.com/productinfo/intuos.html, 1998.

[150] CH. Wagner. The pianist's hand: anthropometry and biomechanics. *Ergonomics*, 31(1):97–129, 1988.

[151] Demin Wang. A multiscale gradient algorithm for image segmentation using watersheds. *Pattern Recognition*, 30(12):2043–2052, 1997.

[152] Colin Ware. Using hand position for virtual object placement. *The Visual Computer*, 6:245–253, 1990.

[153] Colin Ware and Danny R. Jessome. Using the Bat: A six-dimensional mouse for object placement. *IEEE Computer Graphics and Applications*, pages 65–70, November 1988.

[154] Pierre Wellner. The DigitalDesk Calculator: Tactile manipulation on a desk top display. In *Proceedings of UIST '91*, pages 27–33. ACM, November 1991.

[155] Pierre Wellner. Interacting with paper on the digitaldesk. *Communications of the ACM*, pages 1–17, July 1993.

[156] Pierre D. Wellner. Adaptive thresholding for the digitaldesk. Technical Report EPC-93-110, Rank Xerox EuroPARC, 1993.

[157] Pierre D. Wellner. Self calibration for the digitaldesk. Technical Report EPC-93-109, Rank Xerox EuroPARC, 1993.

[158] Wayne Westerman. Design and evaluation of a touch typing recognizer for a multi-touch surface. *in preparation*, 1999.

[159] Wayne Westerman and John Elias. *A method and apparatus for capacitive imaging of multiple finger contacts.* U. S. Patent Provisional Application Serial #60/072,509, January 26 1998.

[160] Wayne Westerman and John Elias. Graphical manipulation via finger chords on a multi-touch surface. *in preparation*, 1999.

[161] Wayne Westerman and John Elias. Hand tracking and finger identification on a multi-touch surface. *in preparation*, 1999.

[162] Wayne Westerman and John Elias. *A method and apparatus for integrated manual input.* U. S. Patent Application Serial #09/236,513, January 25 1999.

[163] Alan Daniel Wexelblat. A feature-based approach to continuous-gesture analysis. Master's thesis, University of Pennsylvania, 1994.

[164] Ph.D. William Hargreaves, M.D. David Rempel, Nachman (Manny) Halpern, M.D. Robert Markison, Dr. Ing. Karl Kroemer, and Jack Litewka. Toward a more humane keyboard. In *CHI '92*, pages 365–368, May 1992.

[165] Shaojun Xiao and Samy CS Leung. Muscle fatigue monitoring using wavelet decomposition of surface EMG. *Biomedical Sciences Instrumentation*, 34:147–152, 1998.

[166] Wayne Yacco. The BAT: Infogrip's keyboard good pick for alternative input device. *MacWEEK*, 6(27):71, July 27 1993.

[167] Shumin Zhai. *Human performance in six degree of freedom input control.* Ph.D. dissertation, University of Toronto, 1995.

[168] Shumin Zhai, Eser Kandogan, Barton A. Smith, and Ted Selker. In search of the "Magic Carpet": Design and experimentation of a bimanual 3d navigation interface. *Journal of Visual Languages and Computing*, February 1999.

[169] Shumin Zhai, Paul Milgram, and Anu Rastogi. Anisotropic performance in six degree-of-freedom tracking: An evaluation of 3D display and control interfaces. *IEEE Transactions on Systems, Man, and Cybernetics - part A: Systems and Humans.*, 27(4):518–528, July 1997.

[170] Shumin Zhai, Barton A. Smith, and Ted Selker. Dual stream input for pointing and scrolling. In *Proceedings of CHI '97 Extended Abstracts*, 1997.

[171] Shumin Zhai, Barton A. Smith, and Ted Selker. Improving browsing performance: A study of four input devices for scrolling and pointing tasks. In *Proceedings of INTERACT '97: The Sixth IFIP Conference on Human-Computer Interaction*, pages 286–292, July 1997.

[172] S. W. Zucker. Region growing: childhood and adolescence. *Computer Graphics and Image Processing*, 5:382–399, 1976.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 817 000 A1

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
**07.01.1998 Bulletin 1998/02**

(51) Int Cl.$^{6}$: **G06F 3/033**

(21) Application number: **97480031.0**

(22) Date of filing: **13.06.1997**

(84) Designated Contracting States:
**AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**

(30) Priority: **25.06.1996 US 672518**

(71) Applicant: **International Business Machines Corporation**
**Armonk, N.Y. 10504 (US)**

(72) Inventor: **Shieh, Johnny Meng-Han**
**Austin, Texas 78727 (US)**

(74) Representative: **de Pena, Alain**
**Compagnie IBM France**
**Département de Propriété Intellectuelle**
**06610 La Gaude (FR)**

(54) **Reshapable pointing device for touchscreens**

(57) A method, apparatus, and article of manufacture direct a computer system, having at least a processor, memory, and touchscreen, to create a reshapable pointing device. The method includes the steps of displaying a pointing device on the touchscreen and, in response to detecting at least one finger placed on the pointing device, reshaping the pointing device in accordance with the size of the finger.

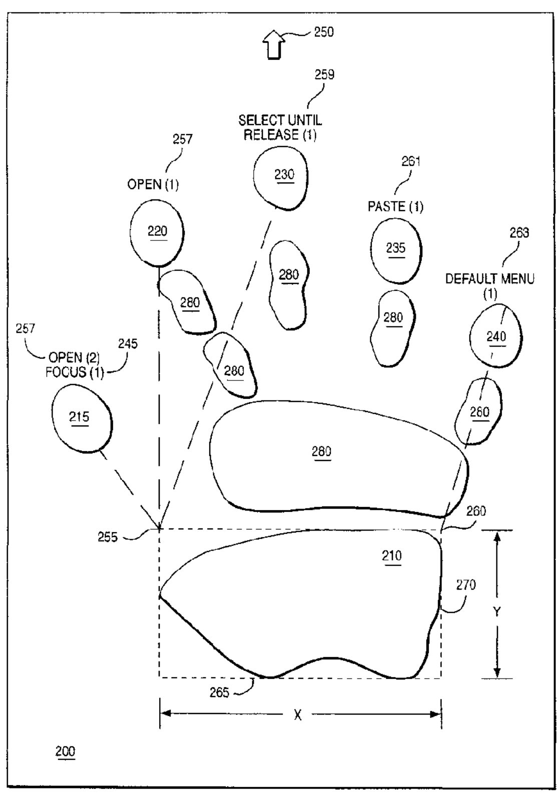



FIG. 2

EP 0 817 000 A1

## Description

### 1. Field of the Invention

The present invention relates to pointing devices and pointers and, more particularly, but without limitation, to pointing devices for use on touchscreen systems.

### 2. Background Information and Description of the Related Art

Conventionally, users interface with the desktop and operating system of their computer system using a "mouse". A mouse is a special hardware input device connected by a wire or infrared signal to the computer system. Typically, the mouse has one or more push buttons on its top and a roller on its bottom designed to roll along a surface next to the computer system. When the user moves the mouse's roller on the surface, a mouse pointer positioned on the computer system's display tracks the movement of the mouse's roller. When the user has positioned the pointer at a desirable location, such as over an object, the user clicks or multiple clicks, depending on how the mouse is programmed, one of the mouse push buttons to invoke or open the object.

The user may customize the operations of a mouse and mouse pointer. Through a customization menu maintained by some conventional operating systems, the user may customize, for example, the assignment of a single click of a first mouse button to invoke a first function and the assignment of a single click over a second mouse button to invoke a second function. Further, the user may customize the appearance of the mouse pointer on the display screen. For example, one user may prefer a small arrow to be the mouse pointer, while another user may prefer a large blinking arrow. Similarly, some users may prefer a fast mouse pointer (i.e., small movements of the mouse's roller cause large movement of the mouse pointer), while other users may prefer a slower mouse pointer. This feature is referred to as the "sensitivity" of the mouse pointer.

These types of mouse and mouse pointer behaviors may be customized for each individual user. However, most operating systems provide customization for only one user on one system. Therefore, for a multi-user system, the user must re-customize the mouse from the prior setting. This typically involves the user editing a mouse settings file or local database that maps button behavior to a specific function. Some systems, such as X11, have a special init file to do this.

Conventional mice suffer certain disadvantages and limitations. For example, the mouse is bulky, fixed in size so that very small hands or very large hands alike do not properly fit over the mouse, not permanently attached to the computer system, subject to corrosion, and requires the user to know the specific mouse behavior (e.g., which mouse button and how many clicks invoke a function). Accordingly, many customer oriented systems (e.g., ATM machines) and multi-user systems do not use mice. Rather, the trend for multi-user systems is to use touchscreens.

Conventional touchscreens allow the user's finger or a pointing device to replace the conventional mouse and mouse pointer. Conventional touchscreens utilize, for example, heat sensitive, sound sensitive, pressure sensitive, or motion sensitive grids/detectors to detect a hand, finger, or object placed on the touchscreen. However, conventional touchscreens suffer certain limitations and disadvantages. For example, unlike a mouse pointer, fingers vary in size and, therefore, the desktop must place contiguous object icons and text far apart to accommodate the largest fingers. Also, the user cannot select the customization features as found in conventional mice and mouse pointers.

Accordingly, there would be great demand for a new pointing device that uses touchscreen technology, but allows object icons and text to be placed close to one another and allows user customization of the pointing device.

### Summary

A method, apparatus, and article of manufacture direct a computer system, having at least a processor, memory, and touchscreen, to create a reshapable pointing device. The method includes the steps of displaying a pointing device on the touchscreen and, in response to detecting at least one finger placed on the pointing device, reshaping the pointing device in accordance with the size of the finger.

### Brief Description of the Drawings

Fig. 1 illustrates a conventional hardware configuration for use with the present invention.

Fig. 2 illustrates a virtual pointing device in accordance with the present invention.

Fig. 3 illustrates detailed logic in the form of a flowchart for performing the steps in accordance with the present invention.

Fig. 4 illustrates a variation of the virtual pointing device illustrated in Fig. 2.

Fig. 5 illustrates another view of the virtual pointing device shown in Fig. 2.

Fig. 6 illustrates a menu for defining the characteristics of the virtual pointing device in accordance with the present invention.

Fig. 7 illustrates a shape menu, define functionality menu, and define pointer menu in accordance with the present invention.

Fig. 8 illustrates detailed logic in the form of a flowchart for performing the steps in accordance with the present invention.

Fig. 9 illustrates detailed logic in the form of a flowchart for performing the steps in accordance with the present invention.

Detailed Description of the Preferred Embodiments

The preferred embodiments may be practiced in any suitable hardware configuration that uses a touchscreen, such as computing system 100 illustrated in Fiq. 1 or, alternatively, in a laptop or notepad computing system. Computing system 100 includes any suitable central processing unit 10, such as a standard microprocessor, and any number of other objects interconnected via system bus 12. For purposes of illustration, computing system 100 includes memory, such as read only memory (ROM) 16, random access memory (RAM) 14, and peripheral memory devices (e.g., disk or tape drives 20) connected to system bus 12 via I/O adapter 18. Computing system 100 further includes a touchscreen display adapter 36 for connecting system bus 12 to a conventional touchscreen display device 38. Also, user interface adapter 22 could connect system bus 12 to other user controls, such as keyboard 24, speaker 28, mouse 26, and a touchpad 32 (not shown).

One skilled in the art readily recognizes how conventional touchscreens operate, how conventional touchscreen device drivers communicate with an operating system, and how a user conventionally utilizes a touchscreen to initiate the manipulation of objects in a graphical user interface. For example, touchscreen technology includes electronic sensors positioned inside a flexible membrane covering a computer screen, a grid of infrared signals, or a method of detecting a touch by sensing a change in reflected sound waves through glass or plastic. Using current touchscreen technology, a user can initiate the display of a pull down menu by touching the touchscreen, and then selecting an object within that menu by dragging a finger down the pull down menu.

A graphical user interface (GUI) and operating system (OS) of the preferred embodiment reside within a computer-readable media and contain a touchscreen device driver that allows one or more users a user to initiate the manipulation of displayed object icons and text on a touchscreen display device. Any suitable computer-readable media may retain the GUI and operating system, such as ROM 16, RAM 14, disk and/or tape drive 20 (e.q., magnetic diskette, magnetic tape, CD-ROM, optical disk, or other suitable storage media).

In the preferred embodiments, the COSE™ (Common Operating System Environment) desktop GUI interfaces the user to the AIX™ operating system. The GUI may be viewed as being incorporated and embedded within the operating system. Alternatively, any suitable operating system or desktop environment could be utilized. Examples of other GUIs and/or operating systems include X11™ (X Windows) graphical user interface, Sun's Solaris™ operating system, and Microsoft's Windows 95™ operating system. While the GUI and operating system merely instruct and direct CPU 10, for ease in explanation, the GUI and operating system will be described as performing the following features and functions.

Referring to Fig. 2, touchscreen 200 includes any conventional, suitable touchscreen that is sensitive to, for example, heat, pressure, or the sound of palm and fingerprints. In this illustration, a user has placed his/her right hand (not shown) on touchscreen 200. While any suitable touchscreen technology may be used, for ease in explanation, the preferred embodiment will be described as using a touchscreen that detects sound patterns. In response to the user placing his/her hand on touchscreen 200, touchscreen 200 detects the sound pattern of the user's hand, including the sound from palmprint area 210, thumbprint area 215, fingerprint areas 220, 230, 235, and 240, and areas 280. Alternatively, only a portion of the hand (e.g., only fingers) and/or a unique object (e.g., stylus) could be substituted for the detection of a hand print. Moreover, more than one hand or object can be detected at a time.

When touchscreen 200 detects one or more hand/finger patterns similar to the one shown in Fig. 2, the OS attempts to identify the user(s). To do so, the OS measures the distance of each fingerprint area 215, 220, 230 and 240 from palmprint area 210, along with the X, Y coordinates of palmprint area 210 and the X, Y extremities of the palmprint area 210. The OS defines the cross point of the leftmost and uppermost point of the palmprint area 210 as the first reference point 255. The OS measures the longest distance from thumbprint 215 to the first reference point 255. Similarly, the OS measures the longest distance from fingerprint areas 220 and 230, respectively, to first reference point 255.

In the same manner, the OS defines the cross point of the rightmost and uppermost point of palmprint area 210 as the second reference point 260, whereby the longest distance from fingerprint area 240 to the second reference point 260 is determined. Finally, the OS measures the X and Y coordinates 265 and 270 of palmprint area 210. To add even more accuracy, the size of each fingerprint could be measured.

Next, the OS searches a user file database (not shown) stored in memory for a match of the newly determined measurements with any existing measurements to determine if a stored identity exists for the handprint. Specifically, the OS compares the four distance measurements and the X, Y coordinates of palmprint 210 with any existing measurements stored in the user file database. However, one skilled in the art realizes that numerous means exists for identifying the handprint (or object print) of a particular user (or user's object) without departing from the scope and spirit of this invention. Illustratively, only the width of the palmprint area 210 could be used to determine if a match existed.

If the OS finds a match within a user-defined (or default) acceptable tolerance (described herein), the OS reads the user file for pre-defined customization features, if any, and creates a virtual pointinq device under the hand (or a portion of the hand) positioned on touchscreen 200 using the pre-defined customization features. Additionally, one skilled in the art recognizes that a secondary confirmation of the user match could be made through, for example, a user id label displayed next to the virtual pointing device, or a specific color shading of the virtual pointing device. Therefore, the areas of touchscreen 200 under, for example, the user's thumb (i.e., thumbprint area 215), fingers (i.e., fingerprint areas 220, 230, 235, and 240), and palm (i.e., palmprint area 210) become "activated", such that certain defined movements of the user's fingers, thumb, and/or palm on those "activated" areas cause certain functions to be invoked. However, if the OS does not recognize the handprint, the OS can build a default virtual pointing device under the hand or a portion of the hand using a default set of functions or the user can create a customized virtual pointing device (described herein).

Fig. 5 illustrates how the user(s) move and operate the virtual pointing device(s). As the user slides his/her hand over touchscreen 200 such that the hand remains in substantial contact with touchscreen 200, the OS detects the position of the user's moving hand on touchscreen 200 and, in response, continuously re-defines the "activated" areas of the virtual pointing device to be the areas under the hand (or a portion of the hand). Therefore, the virtual pointing device moves with and according to the movement of the user's hand. For example, if an "activated" area is initially defined as the area contained within the touchscreen pixel coordinates [X1, Y1, X2, Y2, X3, Y3, and X4, Y4] (not shown) and the user moves a finger from that area to the touchscreen pixel coordinates [X5, Y5, X6, Y6, X7, Y7, and X8, Y8], the "activated" area moves to those new coordinates.

The OS positions pointer 250 near an activated area of the virtual pointing device (in this case, over fingerprint area 230) such that pointer 250 moves in lock step with the virtual pointing device. Therefore, the user could, for example, move the virtual pointing device and, therefore, pointer 250, such that pointer 250 is positioned over a desired object icon. Alternatively, the user could merely lift his hand and place it at a desired location, whereby the OS would re-create the virtual pointing device under the user's hand at the new location (described herein).

The user operates the virtual pointing device via movement of the user's fingers, thumb and/or palm. Illustratively, the user may invoke the "focus function" 245, whereby an object icon positioned under pointer 250 gains focus, by lifting his/her thumb and then placing the thumb back on thumbprint area 215 within a certain amount of time (e.g., two seconds) (referred to as "single clicking"). Similarly, the user may invoke the "paste" function by lifting and replacing his/her third finger on third fingerprint area 235 within a certain amount of time.

Each finger, palm, and thumb behavior and associated functionality/command can be specially defined, and later redefined, to invoke a specific function (described in more detail herein). The OS displays a dialog above each fingerprint/thumbprint area to indicate the finger behavior (a "(1)" representing a single click; a "(2)" representing a double click, etc.) and corresponding functionality/command (e.g., focus 245, open 257, select until release 259, paste 261 and default menu 262).

The default functionality/command, finger behavior and pointer are defined in the preferred embodiment as follows. A single click of the thumb on thumbprint area 215 causes the OS to invoke focus function 245 on any object icon or text positioned under pointer 250. A single click of a finger on fingerprint area 220 or a double click of thumbprint area 215 causes the OS to invoke an open function 230 on any object icon or text positioned under pointer 250. A single click on fingerprint area 230 invokes a select until release function 259 on any object icon or text positioned under pointer 250, while a single click of fingerprint area 235 invokes a paste function 261 on any object icon or text positioned under pointer 250. Finally, a single click of fingerprint area 240 invokes a default menu function 263. The default pointer 250 is in the shape of an arrow and is positioned near fingerprint area 230. However, one skilled in the art readily recognizes that any combination of default functions, pointer location, and/or finger behavior (e.g., multiple clicks) could have been used to define the default virtual pointing device. Moreover, a simultaneous single click (or multiple clicks) of two or more fingers could invoke a function/command.

Fig. 3 illustrates a flow chart containing detailed logic for implementing the preferred embodiments. At 302, touchscreen 200 detects sound/heat/pressure, etc., from a handprint (or object), or alternatively, a portion of a handprint. At 306, the OS reads the handprint and calculates the measurements previously described and illustrated in Fig. 2. At 310, the OS searches user files in a database for the handprint measurements. At 312, if the OS locates any existing handprint measurements within a default tolerance of 10% (which can later be changed by the user, described herein), at 320, the OS reads all information in that user file and, at 322, draws a virtual pointing device on the touchscreen under the user's hand (or portion of the hand) based on pre-defined characteristics found in the user file. Additionally, in the future, if any objects and/or text have been selected by the virtual pointing device, they will be drawn in a position relative to their previous location to the virtual pointing device (described herein). At 333, the OS requests a confirmation that the user match is correct. If the user continues to use the virtual pointing device, then the OS interprets this use as a confirmation. The OS displays a user id label next to the virtual pointing device or, alternatively, may display the virtual pointing device as a specific color shading. Otherwise, if the wrong user has been assumed, control returns to 310, where the OS searches the database for another possible match.

At 324, the OS determines if there is any consistent unusual behavior or undefined behavior for four or more seconds, such as, for example, failing to detect the fingerprint(s), the palmprint, or no handprint on the touchscreen. If the OS detects no unusual behavior, the OS performs a work event loop at 326 (see Fig. 9) and control returns to 324. Referring to Fig. 9, at 902, the OS determines if any movement of the hand across the touchscreen has occurred and, if so, at 904 the OS moves the virtual pointing device in accordance with the movement of the hand. At 906, the OS determines if movement of a finger or thumb has occurred to invoke a function/command and, if so, at 908 the OS invokes that function/command on any object/text positioned under the pointer. Control returns to 324.

Returning to 324 of Fig. 3, if the OS detects unusual behavior or undefined behavior for a certain amount of time (e.g., 4 seconds), at 328, the OS determines if all fingers have been lifted off the touchscreen while the palm remains on the touchscreen. Alternatively, one skilled in the art recognizes that many other indicators could replace the "all fingers lifted" indicator, such as determining if a combination of fingers have been lifted or determining if the palm has been lifted while the fingerprints remain in contact with the touchscreen. If the OS determines that all fingers have been lifted off the touchscreen, at 330, the OS displays a main menu 600 (see Fig. 6, described herein) prompting the user to provide any re-customization of the virtual pointing device. At 344, the OS displays the new virtual pointing device in accordance with any changes made at 330 and control returns to 324.

Returning to 328, if all fingers were not detected as being raised while the palm remained in contact with the touchscreen, at 342, control is directed to Fig. 8. Referring to Fig. 8, at 810, the OS determines if the entire hand (or object) has been lifted off the touchscreen. If the entire hand has not been lifted off the touchscreen, but unusual or undefined behavior has occurred, such as lifting a combination of fingers, thumb and/or palm (whose behavior does not have a corresponding defined functionality), control is directed to 814, where the OS re-draws the virtual pointing device under the hand based on the user file. This indicates to the user that the immediate past hand/finger behavior has no defined function. If the entire hand has been lifted off the touchscreen, at 811, the OS continues to display the virtual pointing device on the touchscreen in its current location for a period of time (e.g., 5 seconds), but in an obvious hibernated state, meaning the fingerprint and palmprint areas will be viewed as translucent areas on the touchscreen. When the virtual pointing device is in the obviously hibernated state, no functionality can be invoked until it is activated (i.e., brought out of hibernation, described herein). At 812, the OS determines if the hand has been re-positioned on the touchscreen within five seconds of detecting its removal. If the hand has not been re-positioned on the touchscreen within the five seconds, control is directed to 826 (described herein). However, if the OS detects the hand being re-positioned on the touchscreen within 5 seconds, at 816, the OS determines if more than one virtual pointing device is concurrently being used and, if so, if more than one user had lifted his/her hand off the touchscreen at the time the hand was re-positioned on the touchscreen. If not, at 814, control is directed to 322 of Fig. 3, whereby the OS activates and moves the virtual pointing identified by the user file under the re-positioned hand. Additionally, if any objects and/or text were selected by the virtual pointing device at the time the hand was lifted, they will be re-drawn in a position relative to their previous location to the virtual pointing device (described herein).

If more than one user had concurrently lifted his/her hand off the touchscreen, at 820, the OS reads the handprint of the re-positioned hand and calculates the measurements previously described and illustrated in Fig. 2. At 822, the OS searches the user files of the virtual pointing devices having a detected lifted hand for a hand measurement match. If a match is not found, at 823, the OS searches the user file database for the user identification of one of the virtual pointing devices having a detected lifted hand. The OS then displays a dialog (not shown) asking the user if he/she is the user identified by the user identification. If the user indicates that he/she is identified by the user identification at 825, at 826, control is directed to 322 of Fig. 3, whereby the OS moves the virtual pointing device identified by the user file under the re-positioned hand, and if any objects and/or text were selected by the virtual pointing device, they will be re-drawn in a position relative to their previous location to the virtual pointing device (described herein). However, if the user indicates that the identification does not identify the user at 825, the OS determines if that identification is the last user file of a virtual pointing device having a detected lifted hand. If not, control returns to 823 where the OS searches the next user file of a virtual pointing device having a detected lifted hand. This process repeats until a match is found between the user and the user identification and, therefore, the corresponding virtual pointing device having a detected lifted hand. If the OS has searched the last user file and no match has been found, at 839, control is directed to 310 of Fig. 3, where the OS search all the user files for the user's hand.

Returning to 812, if the hand has not been repositioned on the touchscreen within 5 seconds, at 826, the OS continues to display the virtual pointing device in the obvious hibernated state and, at 828, prompts the user in a dialog (not shown) if the user desires to quit. If the user desires to quit, control is directed to 830 where the OS removes the virtual pointing device from the display. If the user does not desire to quit, at 832, the OS places the mouse in a "hidden hibernation" state, which means that the mouse image displayed on the touchscreen in the obvious hibernated state (i.e., translucent) begins to fade with time, but can be instantly activated when the user next touches the touchscreen. Therefore, the OS transforms the virtual pointing device from obvious hibernation (e.g., displayed in an translucent form) to hidden hibernation. After a user specified time (e.g., 30 minutes), the OS interprets the time delay as meaning that the virtual pointing device is no longer needed. At 836, if the OS detects a hand placed on the touchscreen within 30 minutes, at 840, the OS brings the virtual pointing device out of hidden hibernation, redraws it under the hand, and control returns to 324 of Fig. 3. Otherwise, at 838, the OS removes the virtual pointing device currently in a hidden hibernation state from memory (e.g., RAM).

Returning to 312 of Fig. 3, the OS determines if a match has been found between a measured hand placed on the touchscreen and any existing user files. If the OS detects several user files having handprint measurements closely matching the handprint in question, at 316, the OS displays in a drop down menu (not shown) on the touchscreen showing those users having the closest match. At 318, the OS waits for the user to select (using his other hand) from the drop down menu a match in user identity, or a selection indicating that no match has occurred. If a match has occurred, control is directed to 320 (previously described). If no match has occurred, control is directed to 314, where the OS displays on the touchscreen a menu (see 510 in Fig. 5) asking the user to indicate if he/she desires to create a customized virtual pointing device. If the user does not desire to create a customized virtual pointing device, the OS prompts the user to place his/her hand on the touchscreen and, in response, the OS builds a generic virtual pointing device under the user's hand, as shown in Fig. 5, having the default finger/palm behavior and fingerprint functionality as previously described and control is directed to 324.

If the user does desire to create a customized virtual pointing device, at 332, the OS opens a user file. At 334, the OS stores the size of the fingerprints and palmprint in the user file. At 336, the OS calculates the distance between the first reference point (previously described and shown in Fig. 2) and the farthest point to each fingerprint of the first three fingers. Additionally, the OS could calculate the second reference point and distance therefrom to the fourth fingerprint. At 338, the OS prompts the user for a user identification and displays main menu 600, which prompts the user to enter virtual pointing device characteristics, such as the virtual pointing device shape, pointer location, behavior and sensitivity, and fingerprint functionality (described herein and shown in Fig. 6). At 340, the OS stores all information in the user file. Control is directed to 322, where the OS draws the virtual pointing device under the hand (or portion of the hand) based on the information stored in the user file.

At 324, the OS determines if any unusual behavior has occurred. If so, at 328, the OS determines if all fingers of the hand have been lifted off the touchscreen. If so, at 330, the OS displays a main menu 600 as illustrated in Fig. 6, prompting the user to provide any customization of the virtual pointing device.

Referring to Fig. 6, after the OS displays the main menu 600, the user may remove his/her hand from the touchscreen. If the user selects shape button 620, a "shape" menu appears (see 700 in Fig. 7) that allows the user to define/redefine the shape of the virtual pointing device. Referring to shape menu 700 of Fig. 7, the OS displays several options to the user. For example, the user could select a "fingers only" virtual pointing device (see Fig. 4, described herein) whereby only the fingers need to be in contact with the touchscreen to move the virtual pointing device, or a palm and thumb only virtual pointinq device, whereby only the thumb and palm need to be in contact with the touchscreen to move the virtual pointing device. In the latter case, movement of the fingers would not be assigned functionality. Additionally, "a thumb plus one finger" or "palm" virtual pointing device could be created. However, because the OS invokes the main menu 600 (see Fig. 6) by lifting all fingers while keeping the palm in contact with the touchscreen, if the user defines a new virtual pointing device that does not include the palm, the user could not later re-program the functionality of that special virtual pointing device. Rather, the user would have to start with a generic virtual pointing device to create a new device. Alternatively, a different technique could be used to activate the main menu 600 without departing from the scope of the invention.

The user may change the default accuracy tolerance amount from 10% to one of a number of pre-programmed values. To do so, the user presses accuracy button 702 and, in response, a drop-down list (not shown) of values (e.g., 4%, 8%, 20%) appears for the user's selection. The user enters/saves all selections by pressing button 704. In response, the main menu 600 shown in Fig. 6 reappears.

Returning to Fig. 6, if the user selects define function button 625, a "define function" menu appears that allows the user to define/redefine the functionality of the fingerprints/palmprint areas. Specifically, define functionality menu 730 in Fig. 7 allows the user to change the functionality of each fingerprint and thumbprint area by pressing the associated button next to the appropriate finger. For example, the user has pressed button 732, indicating that he/she desires to change the functionality of the second finger (i.e., fingerprint area 230). In response, the OS displays drop-down list 740 of predefined functions stored in memory. The user has selected open function 742 where, in response, the OS displays another drop-down list 746. The user selected a double click 744 of the second finger to invoke the open function. The user then presses save button 748 to save the entries in the user file. In response, the main menu 600 shown in Fig. 6 appears. However, one skilled in the art readily recognizes that other changes in finger behavior and fingerprint area functionality may be made without departing from the scope and spirit of this preferred embodiment.

Returning to Fig. 6, if the user selects define pointer button 630, a "define pointer" menu appears that allows the user to define/redefine the shape, sensitivity, and position of the pointer on the virtual pointing device. Referring to define pointer menu 760 in Fig. 7, the user has a number of choices regarding the pointer. For example, the user can select a small, medium or large arrow, and/or a blinking arrow. The user can also select small or large pointer sensitivity, and the position of the pointer with respect to the virtual pointing device. For example, the pointer may be positioned over the third finger (default position), over the first finger, or below the palm. However, one skilled in the art readily recognizes that numerous changes in pointer behavior may be made without departing from the scope and spirit of this preferred embodiment.
The user presses save button 762 to save the entries and, in response, the main menu 600 appears.

Finally, in Fig. 6, the user has the option of saving and exiting by pressing save/exit button 635, or cancelling all changes and returning to the default virtual pointing device by pressing cancel button 615.

Referring to Fig. 4, in a second embodiment, the OS displays pre-determined, standard size fingerprint areas 420, 430, 435 and 440 and pointer 450 as a non-activated (also referred to as "obviously hibernated") virtual pointing device. The fingerprint areas of the virtual pointing device are translucent such that object icons can be seen through them. To activate the virtual pointing device, the user places one or more fingers over a fingerprint area 420, 430, 435 or 440 on touchscreen 400. Once activated, the OS assigns a default function (e.g., default function displayed above each fingerprint area) to each fingerprint area.

In response to detecting a finger placed over one or more of the fingerprint areas, the OS resizes the fingerprint area(s) to the size of the finger placed on the fingerprint area. Therefore, for example, if a finger is smaller than a fingerprint area, that fingerprint area will be reduced to the size of the finger. Conversely, if the finger is larger than the fingerprint area, the fingerprint area will be enlarged to the size of the finger.

Alternatively, when the OS detects a sound pattern (or heat, pressure, etc.) over one or more of the translucent fingerprints areas 420, 430, 435 and 440, the OS activates only those areas of virtual pointing device having a finger placed thereon. In this case, the OS assigns a default function (e.g., default function displayed above each fingerprint area) to each fingerprint area having a finger placed over it. However, the fingerprint areas not having a finger placed over them will not be activated and, as such, will not have the default function assigned to them until they are activated. Each fingerprint area may be activated at any time.

As the user slides his/her fingers over touchscreen 400, the OS detects the touchscreen pixel coordinates under the user's moving fingers and, in response, continuously re-defines the "activated" areas of the virtual pointing device to be the touchscreen areas under the fingers. Therefore, the virtual pointing device moves with and according to the movement of the user's fingers. However, while not all of the fingerprint areas may be activated at once, all fingerprint areas move together as one object.

The OS positions pointer 450 near the fingerprint area 420 such that pointer 450 moves in accordance with movement of the virtual pointing device. Therefore, the user could, for example, move the virtual pointing device such that pointer 450 is positioned over a desired object icon. Alternatively, the user could merely lift his hand and place it at a desired location, whereby the OS would re-create the virtual pointing device under the user's fingers at the new location. Additionally, any objects or text selected by the virtual pointing device at the time the hand was lifted would also be re-drawn at the new location.

In this example, the user has placed his/her first finger over fingerprint area 420 to activate that area of the virtual pointing device. If the user desires to re-size the distance between the fingerprint areas of the virtual pointing device, the user merely places a separate finger, one by one, over each displayed fingerprint area (thereby activating them) and then slides each finger outward/inward or upward/downward, as appropriate, to customize the distance between the fingerprint areas of the virtual pointing device. In this manner, the user customizes the shape/size of the virtual pointing device to the shape/size of his/her hand. However, the user must actively customize the shape/size of the virtual pointing device each time he/she uses it.

Once the user positions pointer 450 over a desired object icon 422, the user could, for example, single click his first finger over fingerprint area 420 to transfer focus to object icon 422. However, only generic functions (or previously established functions) can be used for this embodiment.

## Claims

1. A method for directing a computer system, having at least a processor, memory, and touchscreen, to create a reshapable pointing device, comprising the steps of:

   (a) displaying a pointing device on the touchscreen; and

   (b) in response to detecting at least one finger placed on the pointing device, reshaping the pointing device in accordance with the size of the finger.

**2.** The method according to claim 1 wherein step (a) comprises the step of:
displaying at least one translucent area on the touchscreen of a first size.

**3.** The method according to claim 2 wherein step (b) further comprises the step of:
in response to detecting a finger placed on the translucent area, wherein the finger is a second size, reshaping the translucent area to substantially the first size.

**4.** The method according to claim 1 further comprising the steps of:

displaying at least two translucent areas on the touchscreen, wherein the translucent areas are a first distance apart;

in response to detecting at least one finger placed on each translucent area, activating each translucent area; and

in response to moving one of the fingers with respect to the other finger such that the fingers are a second distance apart, adjusting the translucent areas to be substantially the second distance apart.

**5.** The method according to claim 2 further comprising the step of:
in response to detecting at least one finger placed on the translucent area, activating the pointing device.

**6.** The method according to claim 2 further comprising the step of:
in response to detecting at least one finger placed on the translucent area, activating the translucent area.

**7.** The method according to claim 3 wherein step (b) further comprises the step of:
detecting heat, pressure, or sound from the finger to determine the second size of the finger.

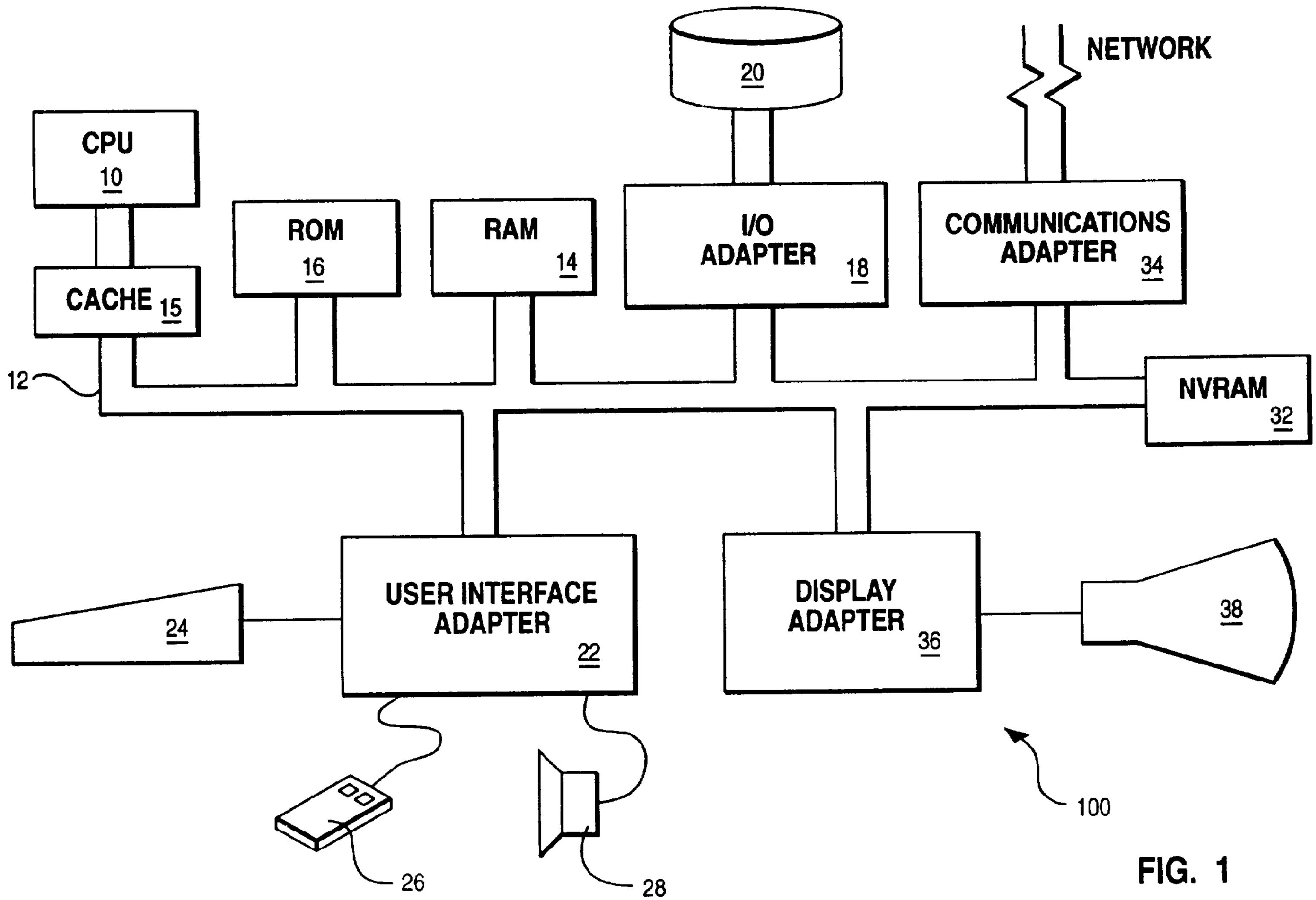
**8.** A system for carrying out the method for directing a computer having at least a processor, memory, and touchscreen, to create a reshapable pointing device according to anyone of the preceding claims.

**9.** An article of manufacture, comprising:

a computer usable medium having computer readable program code means embodied therein, the computer readable program code means in the article of manufacture comprising:

computer readable program code means for causing a computer system, having at least a touchscreen, to display a pointing device on the touchscreen; and

computer readable program code means for causing the computer system to, in response to detecting at least one finger placed on the pointing device, reshape the pointing device in accordance with the size of the finger.



NETWORK
20
CPU
10
ROM
16
RAM
14
I/O
ADAPTER
18
COMMUNICATIONS
ADAPTER
34
CACHE
15
12
NVRAM
32
USER INTERFACE
ADAPTER
22
DISPLAY
ADAPTER
36
24
38
26
28
100


FIG. 1

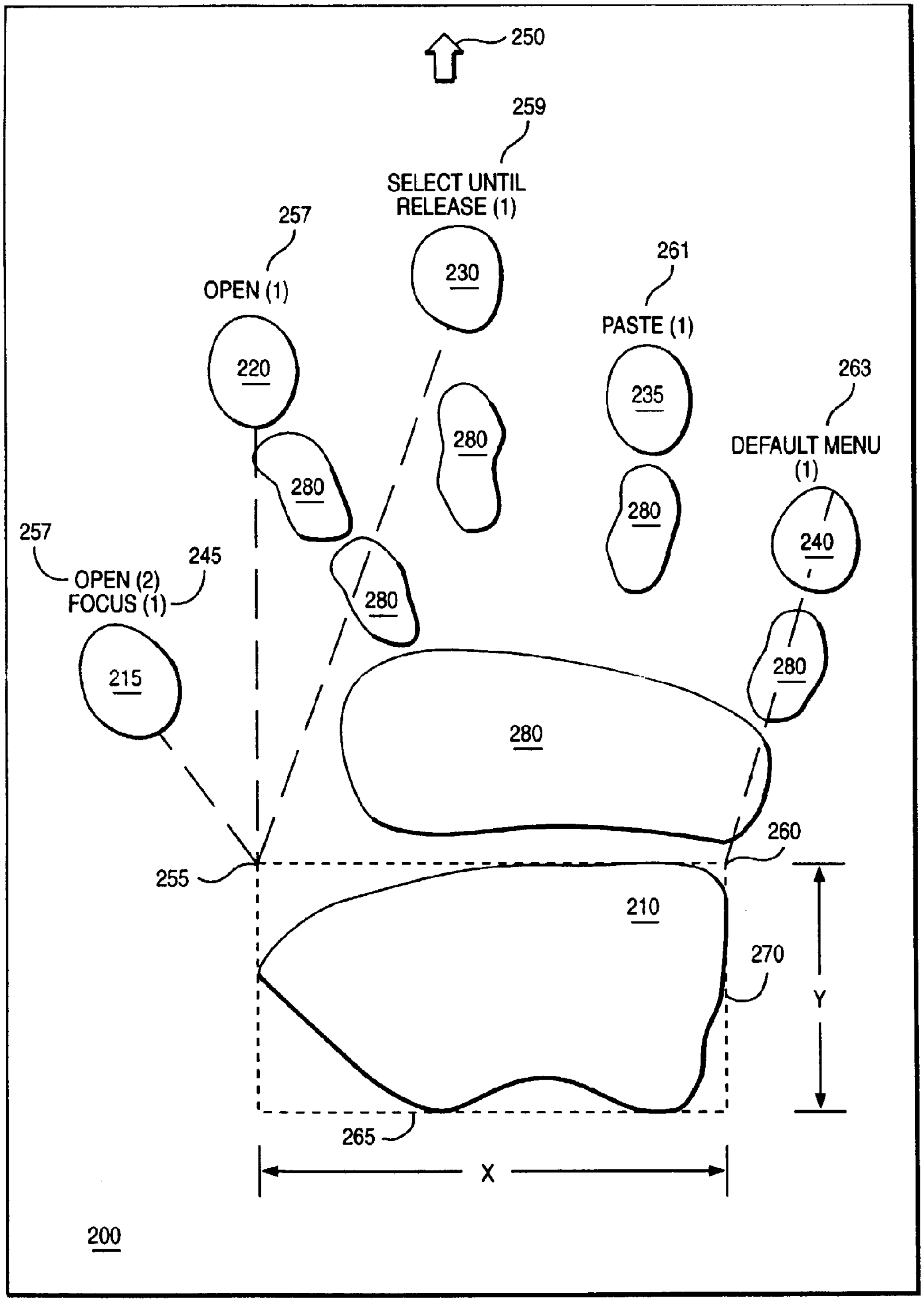

250
259
SELECT UNTIL RELEASE (1)
257
OPEN (1)
230
261
PASTE (1)
220
235
263
280
DEFAULT MENU (1)
280
280
257
245
240
OPEN (2)
FOCUS (1)
280
280
215
280
280
260
255
210
270
Y
265
X
200

FIG. 2

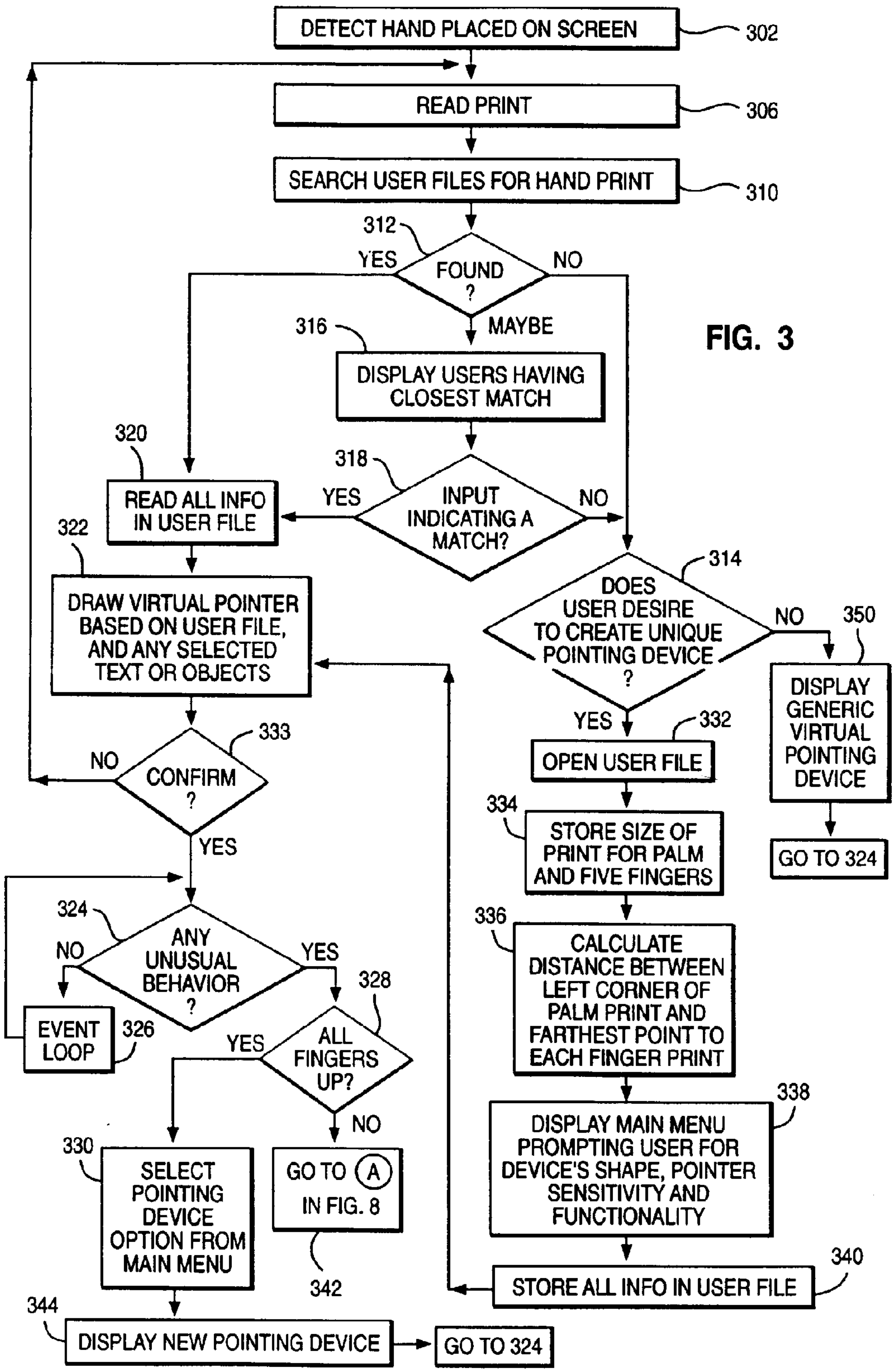

DETECT HAND PLACED ON SCREEN
302
READ PRINT
306
SEARCH USER FILES FOR HAND PRINT
310
312
FOUND ?
YES
NO
MAYBE
316
DISPLAY USERS HAVING CLOSEST MATCH
318
INPUT INDICATING A MATCH?
YES
NO
320
READ ALL INFO IN USER FILE
322
DRAW VIRTUAL POINTER BASED ON USER FILE, AND ANY SELECTED TEXT OR OBJECTS
314
DOES USER DESIRE TO CREATE UNIQUE POINTING DEVICE ?
NO
350
DISPLAY GENERIC VIRTUAL POINTING DEVICE
GO TO 324
YES
332
OPEN USER FILE
334
STORE SIZE OF PRINT FOR PALM AND FIVE FINGERS
336
CALCULATE DISTANCE BETWEEN LEFT CORNER OF PALM PRINT AND FARTHEST POINT TO EACH FINGER PRINT
338
DISPLAY MAIN MENU PROMPTING USER FOR DEVICE'S SHAPE, POINTER SENSITIVITY AND FUNCTIONALITY
340
STORE ALL INFO IN USER FILE
333
CONFIRM ?
NO
YES
324
ANY UNUSUAL BEHAVIOR ?
NO
YES
EVENT LOOP
326
328
ALL FINGERS UP?
YES
NO
330
SELECT POINTING DEVICE OPTION FROM MAIN MENU
GO TO (A) IN FIG. 8
342
344
DISPLAY NEW POINTING DEVICE
GO TO 324

FIG. 3

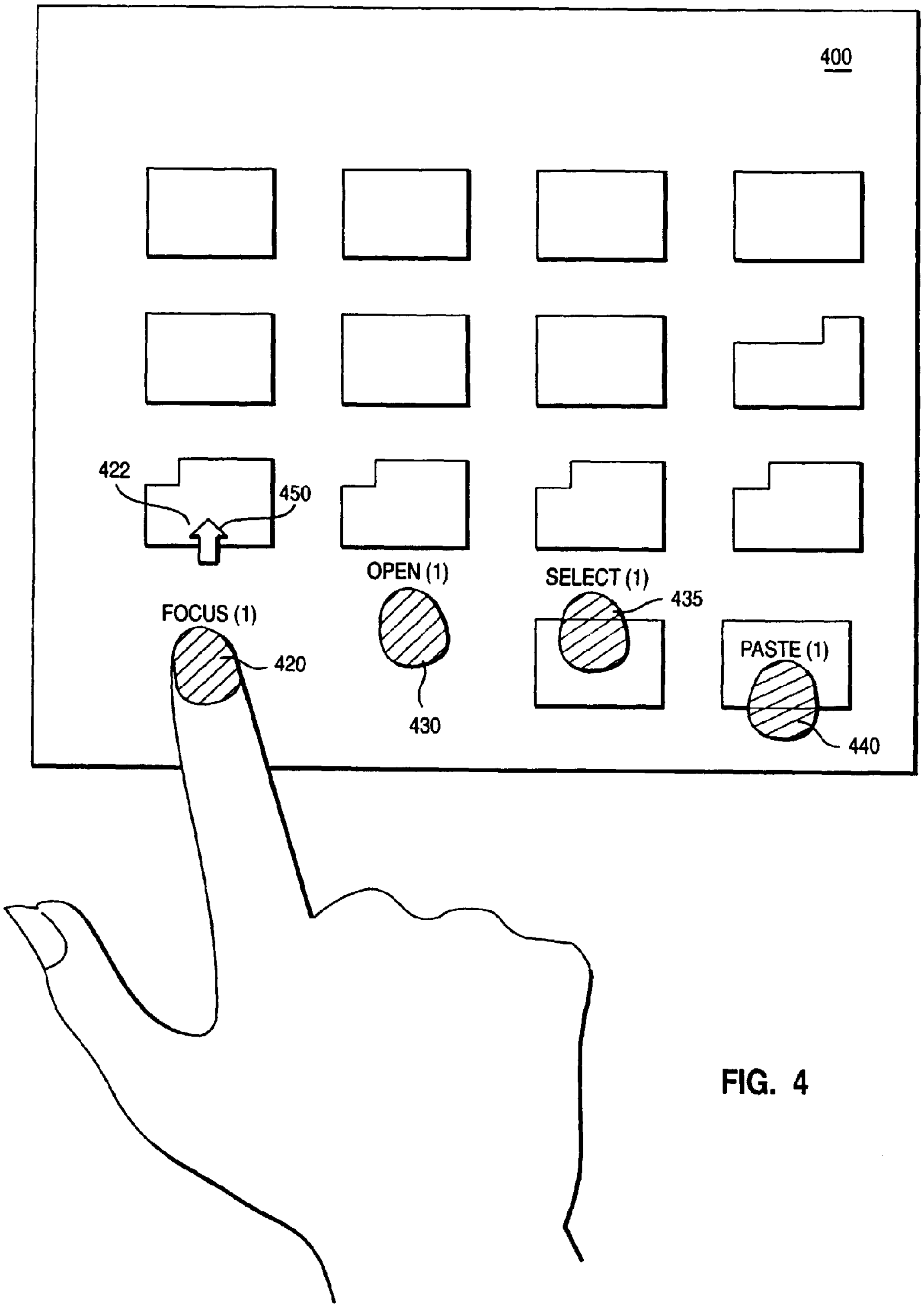



FIG. 4

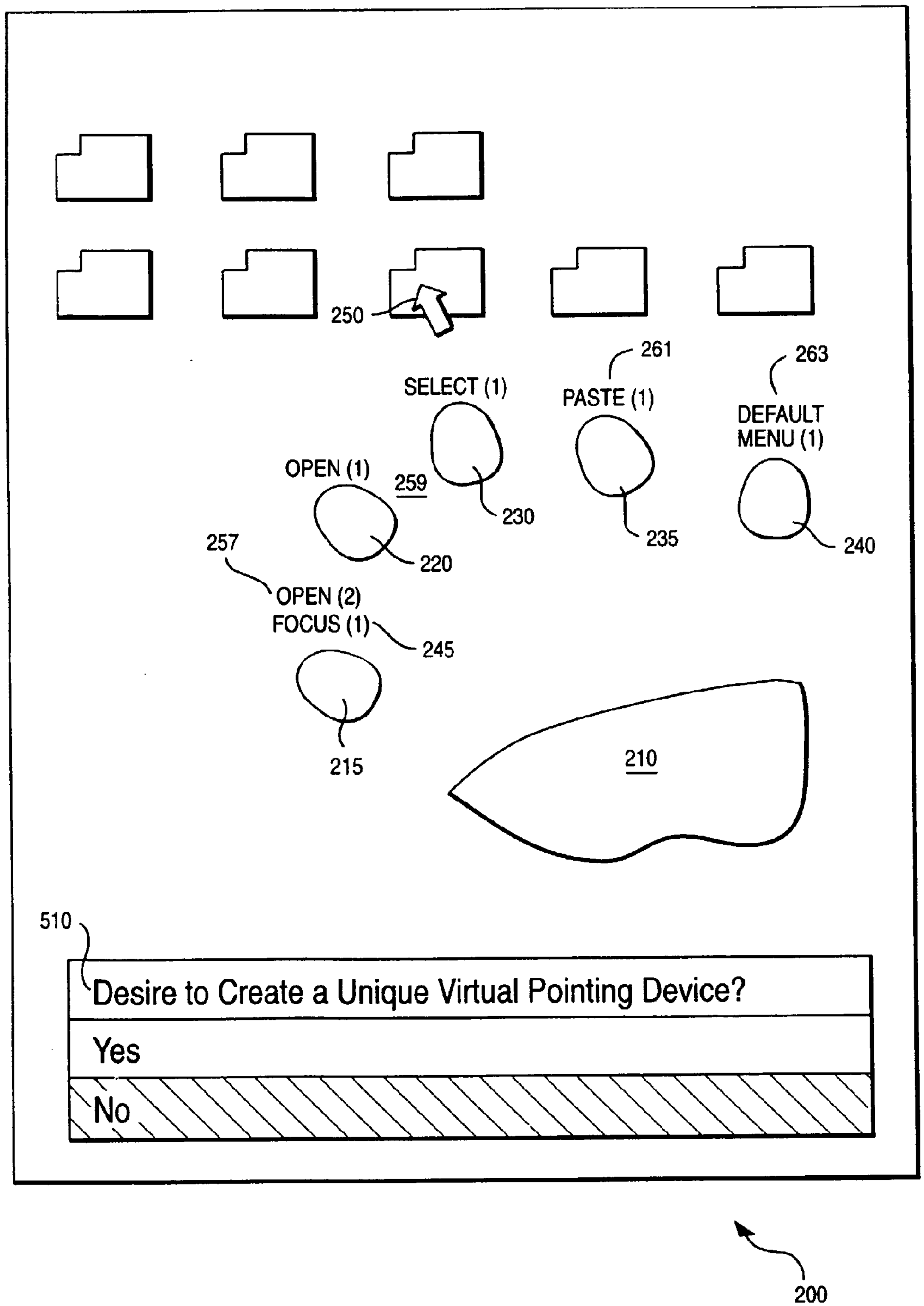

250
261
263
SELECT (1)
PASTE (1)
DEFAULT
MENU (1)
OPEN (1)
259
230
235
240
257
220
OPEN (2)
FOCUS (1)
245
210
215
510
Desire to Create a Unique Virtual Pointing Device?
Yes
No
200

FIG. 5

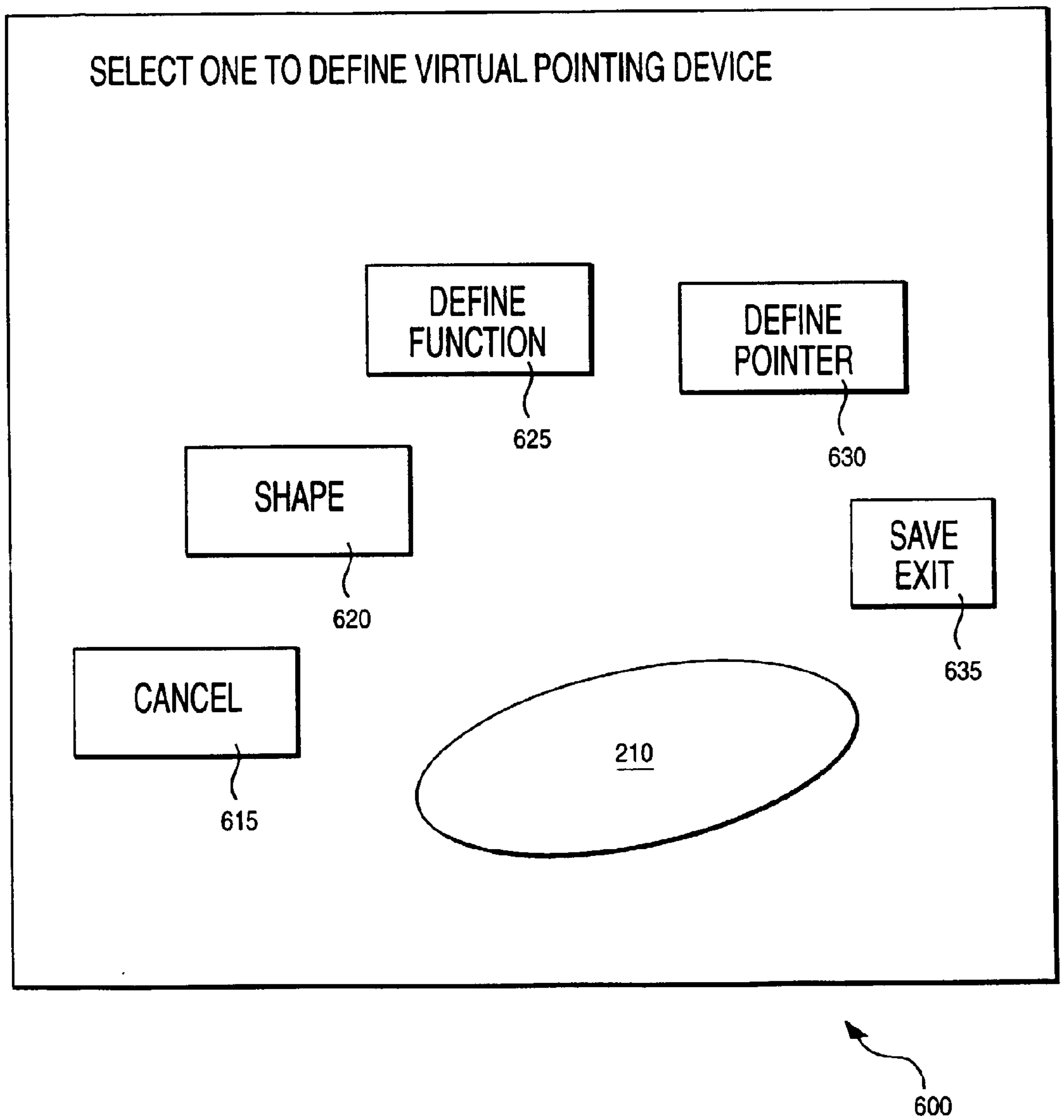

SELECT ONE TO DEFINE VIRTUAL POINTING DEVICE
DEFINE FUNCTION
625
DEFINE POINTER
630
SHAPE
620
SAVE EXIT
635
CANCEL
615
210
600

FIG. 6

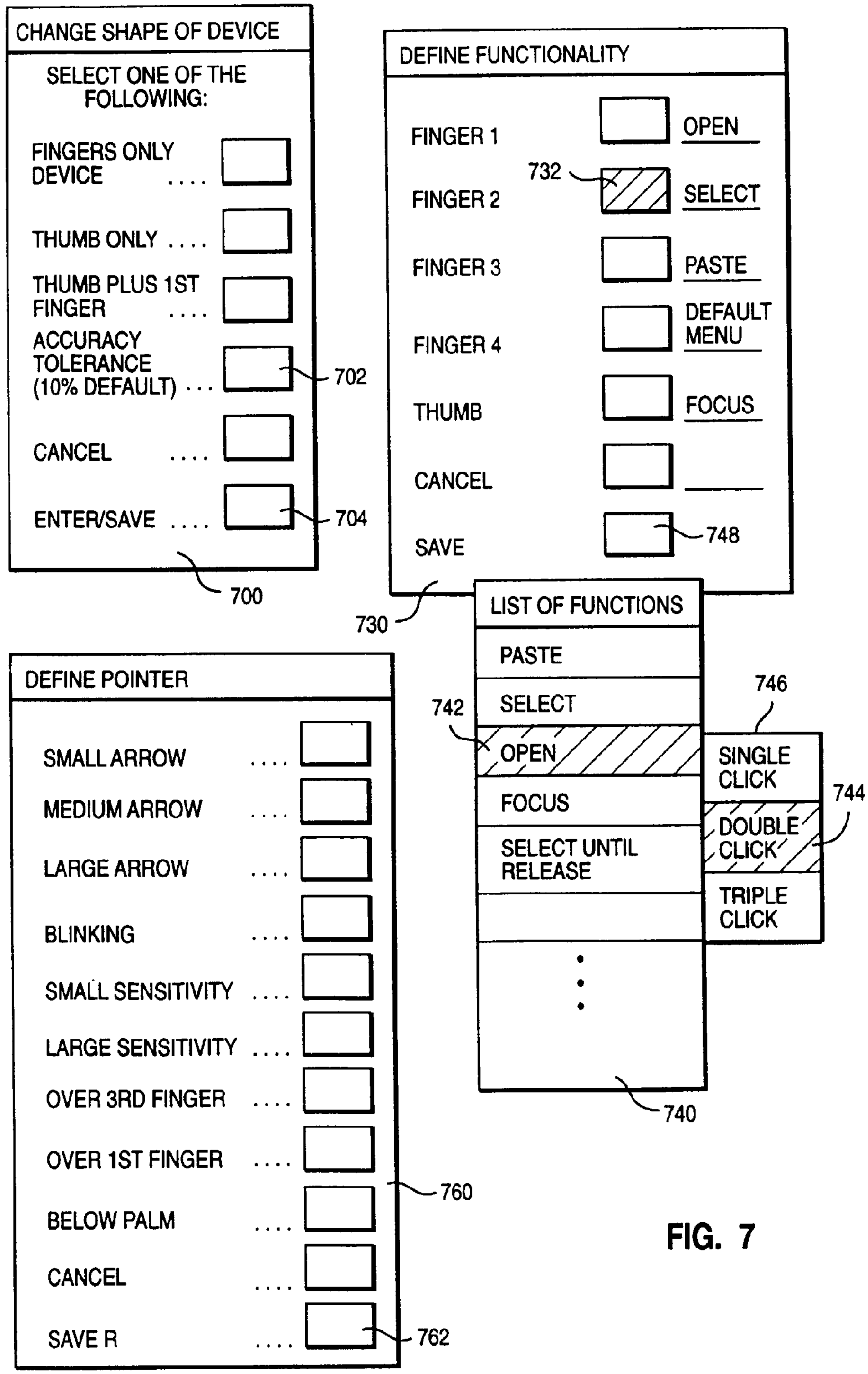
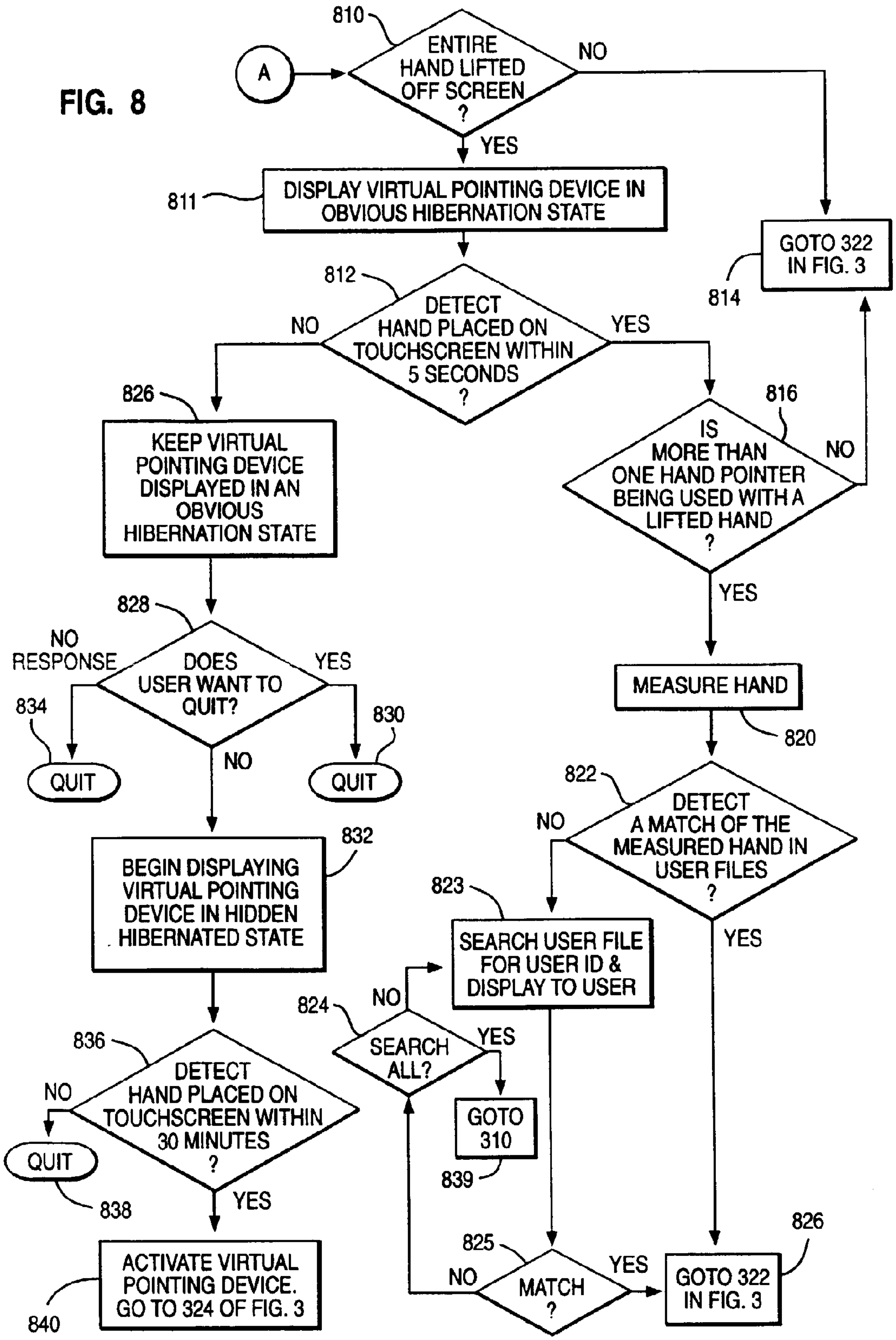

CHANGE SHAPE OF DEVICE
SELECT ONE OF THE FOLLOWING:
FINGERS ONLY DEVICE ....
THUMB ONLY ....
THUMB PLUS 1ST FINGER ....
ACCURACY TOLERANCE (10% DEFAULT) ...
702
CANCEL ....
ENTER/SAVE ....
704
700
DEFINE FUNCTIONALITY
FINGER 1
OPEN
732
FINGER 2
SELECT
FINGER 3
PASTE
FINGER 4
DEFAULT MENU
THUMB
FOCUS
CANCEL
SAVE
748
730
LIST OF FUNCTIONS
PASTE
SELECT
742
OPEN
FOCUS
SELECT UNTIL RELEASE
746
SINGLE CLICK
744
DOUBLE CLICK
TRIPLE CLICK
740
DEFINE POINTER
SMALL ARROW ....
MEDIUM ARROW ....
LARGE ARROW ....
BLINKING ....
SMALL SENSITIVITY ....
LARGE SENSITIVITY ....
OVER 3RD FINGER ....
OVER 1ST FINGER ....
760
BELOW PALM ....
CANCEL ....
SAVE R ....
762

FIG. 7

FIG. 8






APLNDC00021408

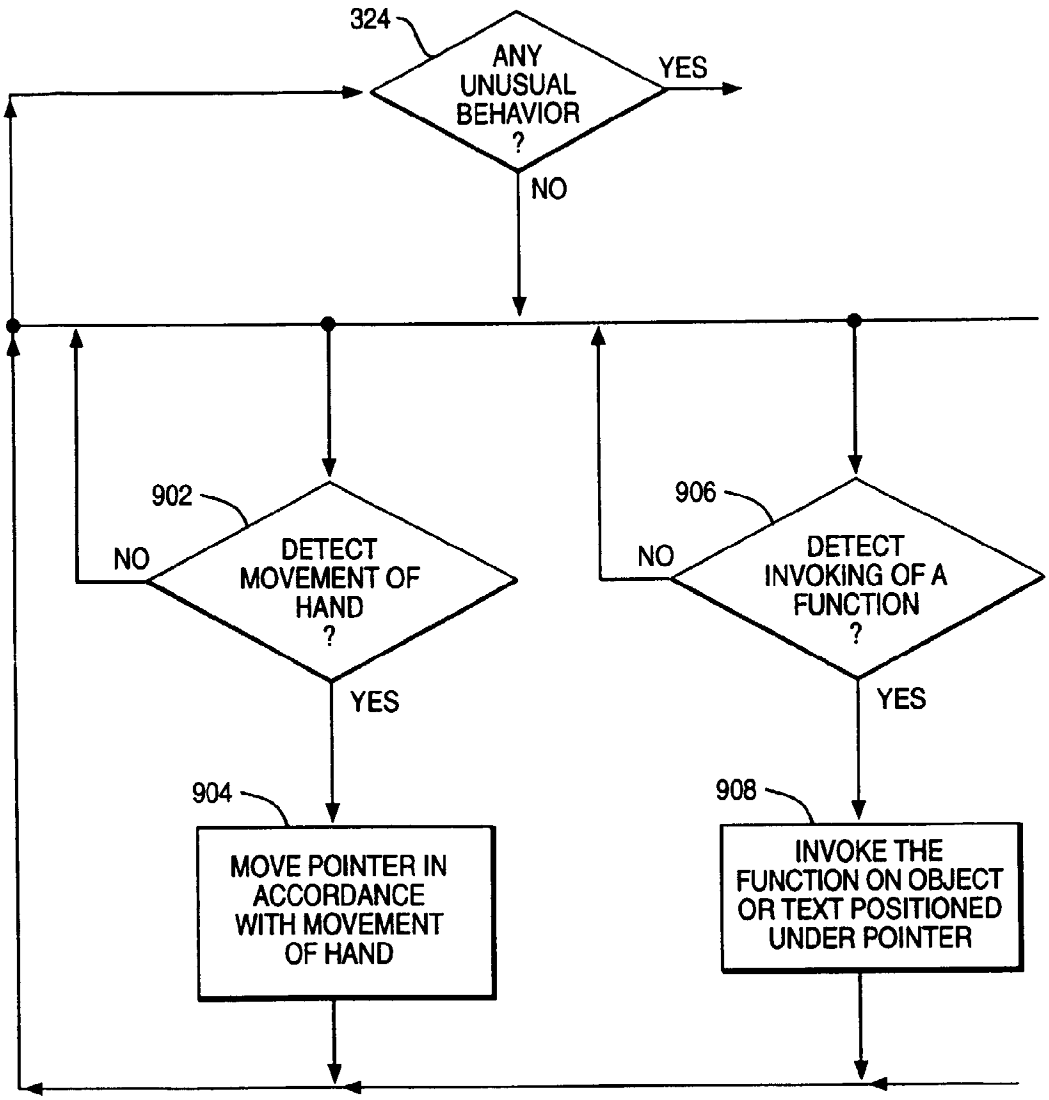

324
ANY UNUSUAL BEHAVIOR ?
YES
NO
902
DETECT MOVEMENT OF HAND ?
NO
YES
906
DETECT INVOKING OF A FUNCTION ?
NO
YES
904
MOVE POINTER IN ACCORDANCE WITH MOVEMENT OF HAND
908
INVOKE THE FUNCTION ON OBJECT OR TEXT POSITIONED UNDER POINTER

FIG. 9

European Patent Office

# EUROPEAN SEARCH REPORT

Application Number
EP 97 48 0031

## DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.6) |
|---|---|---|---|
| A | WO 92 09944 A (WANG LABORATORIES) 11 June 1992<br>* abstract *<br>* page 27, line 14 - line 19; figure 10 *<br>* page 44, line 23 - page 45, line 24 *<br>* page 47, line 1 - line 14 * | 1,8,9 | G06F3/033 |
| A | "SOFT ADAPTIVE FOLLOW-FINGER KEYBOARD FOR TOUCH-SCREEN PADS"<br>IBM TECHNICAL DISCLOSURE BULLETIN, vol. 36, no. 11, 1 November 1993, pages 5-7, XP000424761<br>* the whole document * | 1-4,8,9 | |
| A | "MOUSE EMULATION FOR DIGITIZER/TOUCH PANEL"<br>IBM TECHNICAL DISCLOSURE BULLETIN, vol. 33, no. 7, 1 December 1990, page 216/217 XP000108457<br>* the whole document * | 1,8,9 | |
| A | PATENT ABSTRACTS OF JAPAN<br>vol. 014, no. 352 (P-1085), 30 July 1990<br>& JP 02 127720 A (MITSUBISHI ELECTRIC CORP), 16 May 1990,<br>* abstract * | 1,8,9 | TECHNICAL FIELDS SEARCHED (Int.Cl.6)<br>G06F |
| P,A | EP 0 725 331 A (L G ELECTRONICS INC) 7 August 1996<br>* abstract *<br>* column 6, line 6 - column 7, line 42; figures 4A-5 * | 1,8,9 | |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| THE HAGUE | 15 September 1997 | Bravo, P |

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 817 000 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
**23.01.2002 Bulletin 2002/04**

(51) Int Cl.[7]: **G06F 3/033**

(21) Application number: **97480031.0**

(22) Date of filing: **13.06.1997**

(54) **Reshapable pointing device for touchscreens**

Umformbare Hinweisanordnung für Berührungsbildschirme

Dispositif de pointage déformable pour écrans tactiles

(84) Designated Contracting States:
**DE FR GB**

(30) Priority: **25.06.1996 US 672518**

(43) Date of publication of application:
**07.01.1998 Bulletin 1998/02**

(73) Proprietor: **International Business Machines Corporation**
**Armonk, N.Y. 10504 (US)**

(72) Inventor: **Shieh, Johnny Meng-Han**
**Austin, Texas 78727 (US)**

(74) Representative: **de Pena, Alain**
**Compagnie IBM France Département de Propriété Intellectuelle**
**06610 La Gaude (FR)**

(56) References cited:
**EP-A- 0 725 331** **WO-A-92/09944**

- **"SOFT ADAPTIVE FOLLOW-FINGER KEYBOARD FOR TOUCH-SCREEN PADS" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 36, no. 11, 1 November 1993, pages 5-7, XP000424761**
- **"MOUSE EMULATION FOR DIGITIZER/TOUCH PANEL" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 33, no. 7, 1 December 1990, page 216/217 XP000108457**
- **PATENT ABSTRACTS OF JAPAN vol. 014, no. 352 (P-1085), 30 July 1990 & JP 02 127720 A (MITSUBISHI ELECTRIC CORP), 16 May 1990,**



EP 0 817 000 B1

Printed by Jouve, 75001 PARIS (FR)

## Description

### 1. Field of the Invention

**[0001]** The present invention relates to pointing devices and pointers and, more particularly, but without limitation, to pointing devices for use on touchscreen systems.

### 2. Background Information and Description of the Related Art

**[0002]** Conventionally, users interface with the desktop and operating system of their computer system using a "mouse". A mouse is a special hardware input device connected by a wire or infrared signal to the computer system. Typically, the mouse has one or more push buttons on its top and a roller on its bottom designed to roll along a surface next to the computer system. When the user moves the mouse's roller on the surface, a mouse pointer positioned on the computer system's display tracks the movement of the mouse's roller. When the user has positioned the pointer at a desirable location, such as over an object, the user clicks or multiple clicks, depending on how the mouse is programmed, one of the mouse push buttons to invoke or open the object.

**[0003]** The user may customize the operations of a mouse and mouse pointer. Through a customization menu maintained by some conventional operating systems, the user may customize, for example, the assignment of a single click of a first mouse button to invoke a first function and the assignment of a single click over a second mouse button to invoke a second function. Further, the user may customize the appearance of the mouse pointer on the display screen. For example, one user may prefer a small arrow to be the mouse pointer, while another user may prefer a large blinking arrow. Similarly, some users may prefer a fast mouse pointer (i.e., small movements of the mouse's roller cause large movement of the mouse pointer), while other users may prefer a slower mouse pointer. This feature is referred to as the "sensitivity" of the mouse pointer.

**[0004]** These types of mouse and mouse pointer behaviors may be customized for each individual user. However, most operating systems provide customization for only one user on one system. Therefore, for a multi-user system, the user must re-customize the mouse from the prior setting. This typically involves the user editing a mouse settings file or local database that maps button behavior to a specific function. Some systems, such as X11, have a special init file to do this.

**[0005]** Conventional mice suffer certain disadvantages and limitations. For example, the mouse is bulky, fixed in size so that very small hands or very large hands alike do not properly fit over the mouse, not permanently attached to the computer system, subject to corrosion, and requires the user to know the specific mouse behavior (e.g., which mouse button and how many clicks invoke a function). Accordingly, many customer oriented systems (e.g., ATM machines) and multi-user systems do not use mice. Rather, the trend for multi-user systems is to use touchscreens.

**[0006]** Conventional touchscreens allow the user's finger or a pointing device to replace the conventional mouse and mouse pointer. Conventional touchscreens utilize, for example, heat sensitive, sound sensitive, pressure sensitive, or motion sensitive grids/detectors to detect a hand, finger, or object placed on the touchscreen. However, conventional touchscreens suffer certain limitations and disadvantages. For example, unlike a mouse pointer, fingers vary in size and, therefore, the desktop must place contiguous object icons and text far apart to accommodate the largest fingers. Also, the user cannot select the customization features as found in conventional mice and mouse pointers.

**[0007]** IBM Technical Disclosure Bulletin, vol. 36, no. 11, pages 5-7 discloses automatical adaption of the geometrical arrangement of a virtual keyboard on a touch screen according to the shape of a user's hand.

**[0008]** Accordingly, there would be great demand for a new pointing device that uses touchscreen technology, but allows object icons and text to be placed close to one another and allows user customization of the pointing device.

### Summary

**[0009]** The invention is set forth in independent claims 1, 8 and 9.

**[0010]** A method, apparatus, and computer program carrier direct a computer system, having at least a processor, memory, and touchscreen, to create a reshapable pointing device. The method includes the steps of displaying a pointing device on the touchscreen and, in response to detecting at least one finger placed on the pointing device, reshaping the pointing device in accordance with the size of the finger.

### Brief Description of the Drawings

**[0011]** Fig. 1 illustrates a conventional hardware configuration for use with the present invention.

**[0012]** Fig. 2 illustrates a virtual pointing device in accordance with the present invention.

**[0013]** Fig. 3 illustrates detailed logic in the form of a flowchart for performing the steps in accordance with the present invention.

**[0014]** Fig. 4 illustrates a variation of the virtual pointing device illustrated in Fig. 2.

**[0015]** Fig. 5 illustrates another view of the virtual pointing device shown in Fig. 2.

**[0016]** Fig. 6 illustrates a menu for defining the characteristics of the virtual pointing device in accordance with the present invention.

**[0017]** Fig. 7 illustrates a shape menu, define functionality menu, and define pointer menu in accordance with the present invention.

**[0018]** Fig. 8 illustrates detailed logic in the form of a flowchart for performing the steps in accordance with the present invention.

**[0019]** Fig. 9 illustrates detailed logic in the form of a flowchart for performing the steps in accordance with the present invention.

Detailed Description of the Preferred Embodiments

**[0020]** The preferred embodiments may be practiced in any suitable hardware configuration that uses a touchscreen, such as computing system 100 illustrated in Fig. 1 or, alternatively, in a laptop or notepad computing system. Computing system 100 includes any suitable central processing unit 10, such as a standard microprocessor, and any number of other objects interconnected via system bus 12. For purposes of illustration, computing system 100 includes memory, such as read only memory (ROM) 16, random access memory (RAM) 14, and peripheral memory devices (e.g., disk or tape drives 20) connected to system bus 12 via I/O adapter 18. Computing system 100 further includes a touchscreen display adapter 36 for connecting system bus 12 to a conventional touchscreen display device 38. Also, user interface adapter 22 could connect system bus 12 to other user controls, such as keyboard 24, speaker 28, mouse 26, and a touchpad 32 (not shown).

**[0021]** One skilled in the art readily recognizes how conventional touchscreens operate, how conventional touchscreen device drivers communicate with an operating system, and how a user conventionally utilizes a touchscreen to initiate the manipulation of objects in a graphical user interface. For example, touchscreen technology includes electronic sensors positioned inside a flexible membrane covering a computer screen, a grid of infrared signals, or a method of detecting a touch by sensing a change in reflected sound waves through glass or plastic. Using current touchscreen technology, a user can initiate the display of a pull down menu by touching the touchscreen, and then selecting an object within that menu by dragging a finger down the pull down menu.

**[0022]** A graphical user interface (GUI) and operating system (OS) of the preferred embodiment reside within a computer-readable media and contain a touchscreen device driver that allows one or more users a user to initiate the manipulation of displayed object icons and text on a touchscreen display device. Any suitable computer-readable media may retain the GUI and operating system, such as ROM 16, RAM 14, disk and/or tape drive 20 (e.g., magnetic diskette, magnetic tape, CD-ROM, optical disk, or other suitable storage media).

**[0023]** In the preferred embodiments, the COSE™ (Common Operating System Environment) desktop GUI interfaces the user to the AIX™ operating system. The GUI may be viewed as being incorporated and embedded within the operating system. Alternatively, any suitable operating system or desktop environment could be utilized. Examples of other GUIs and/or operating systems include X11™ (x windows) graphical user interface, Sun's Solaris™ operating system, and Microsoft's Windows 95™ operating system. While the GUI and operating system merely instruct and direct CPU 10, for ease in explanation, the GUI and operating system will be described as performing the following features and functions.

**[0024]** Referring to Fig. 2, touchscreen 200 includes any conventional, suitable touchscreen that is sensitive to, for example, heat, pressure, or the sound of palm and fingerprints. In this illustration, a user has placed his/her right hand (not shown) on touchscreen 200. While any suitable touchscreen technology may be used, for ease in explanation, the preferred embodiment will be described as using a touchscreen that detects sound patterns. In response to the user placing his/her hand on touchscreen 200, touchscreen 200 detects the sound pattern of the user's hand, including the sound from palmprint area 210, thumbprint area 215, fingerprint areas 220, 230, 235, and 240, and areas 280. Alternatively, only a portion of the hand (e.g., only fingers) and/or a unique object (e.g., stylus) could be substituted for the detection of a hand print. Moreover, more than one hand or object can be detected at a time.

**[0025]** When touchscreen 200 detects one or more hand/finger patterns similar to the one shown in Fig. 2, the OS attempts to identify the user(s). To do so, the OS measures the distance of each fingerprint area 215, 220, 230 and 240 from palmprint area 210, along with the X, Y coordinates of palmprint area 210 and the X, Y extremities of the palmprint area 210. The OS defines the cross point of the leftmost and uppermost point of the palmprint area 210 as the first reference point 255. The OS measures the longest distance from thumbprint 215 to the first reference point 255. Similarly, the OS measures the longest distance from fingerprint areas 220 and 230, respectively, to first reference point 255.

**[0026]** In the same manner, the OS defines the cross point of the rightmost and uppermost point of palmprint area 210 as the second reference point 260, whereby the longest distance from fingerprint area 240 to the second reference point 260 is determined. Finally, the OS measures the X and Y coordinates 265 and 270 of palmprint area 210. To add even more accuracy, the size of each fingerprint could be measured.

**[0027]** Next, the OS searches a user file database (not shown) stored in memory for a match of the newly determined measurements with any existing measurements to determine if a stored identity exists for the handprint. Specifically, the OS compares the four distance measurements and the X, Y coordinates of palmprint 210 with any existing measurements stored in the user file database. However, one skilled in the art realizes that numerous means exists for identifying the handprint (or object print) of a particular user (or user's object) without departing from the scope and spirit of this invention. Illustratively, only the width of the palmprint area 210 could be used to determine if a match existed.

**[0028]** If the OS finds a match within a user-defined (or default) acceptable tolerance (described herein), the OS reads the user file for pre-defined customization features, if any, and creates a virtual pointing device under the hand (or a portion of the hand) positioned on touchscreen 200 using the pre-defined customization features. Additionally, one skilled in the art recognizes that a secondary confirmation of the user match could be made through, for example, a user id label displayed next to the virtual pointing device, or a specific color shading of the virtual pointing device. Therefore, the areas of touchscreen 200 under, for example, the user's thumb (i.e., thumbprint area 215), fingers (i.e., fingerprint areas 220, 230, 235, and 240), and palm (i.e., palmprint area 210) become "activated", such that certain defined movements of the user's fingers, thumb, and/or palm on those "activated" areas cause certain functions to be invoked. However, if the OS does not recognize the handprint, the OS can build a default virtual pointing device under the hand or a portion of the hand using a default set of functions or the user can create a customized virtual pointing device (described herein).

**[0029]** Fig. 5 illustrates how the user(s) move and operate the virtual pointing device(s). As the user slides his/her hand over touchscreen 200 such that the hand remains in substantial contact with touchscreen 200, the OS detects the position of the user's moving hand on touchscreen 200 and, in response, continuously re-defines the "activated" areas of the virtual pointing device to be the areas under the hand (or a portion of the hand). Therefore, the virtual pointing device moves with and according to the movement of the user's hand. For example, if an "activated" area is initially defined as the area contained within the touchscreen pixel coordinates [X1, Y1, X2, Y2, X3, Y3, and X4, Y4] (not shown) and the user moves a finger from that area to the touchscreen pixel coordinates [X5, Y5, X6, Y6, X7, Y7, and X8, Y8], the "activated" area moves to those new coordinates.

**[0030]** The OS positions pointer 250 near an activated area of the virtual pointing device (in this case, over fingerprint area 230) such that pointer 250 moves in lock step with the virtual pointing device. Therefore, the user could, for example, move the virtual pointing device and, therefore, pointer 250, such that pointer 250 is positioned over a desired object icon. Alternatively, the user could merely lift his hand and place it at a desired location, whereby the OS would re-create the virtual pointing device under the user's hand at the new location (described herein).

**[0031]** The user operates the virtual pointing device via movement of the user's fingers, thumb and/or palm. Illustratively, the user may invoke the "focus function" 245, whereby an object icon positioned under pointer 250 gains focus, by lifting his/her thumb and then placing the thumb back on thumbprint area 215 within a certain amount of time (e.g., two seconds) (referred to as "single clicking"). Similarly, the user may invoke the "paste" function by lifting and replacing his/her third finger on third fingerprint area 235 within a certain amount of time.

**[0032]** Each finger, palm, and thumb behavior and associated functionality/command /command can be specially defined, and later redefined, to invoke a specific function (described in more detail herein). The OS displays a dialog above each fingerprint/thumbprint area to indicate the finger behavior (a "(1)" representing a single click; a "(2)" representing a double click, etc.) and corresponding functionality/command (e.g., focus 245, open 257, select until release 259, paste 261 and default menu 262).

**[0033]** The default functionality/command, finger behavior and pointer are defined in the preferred embodiment as follows. A single click of the thumb on thumbprint area 215 causes the OS to invoke focus function 245 on any object icon or text positioned under pointer 250. A single click of a finger on fingerprint area 220 or a double click of thumbprint area 215 causes the OS to invoke an open function 230 on any object icon or text positioned under pointer 250. A single click on fingerprint area 230 invokes a select until release function 259 on any object icon or text positioned under pointer 250, while a single click of fingerprint area 235 invokes a paste function 261 on any object icon or text positioned under pointer 250. Finally, a single click of fingerprint area 240 invokes a default menu function 263. The default pointer 250 is in the shape of an arrow and is positioned near fingerprint area 230. However, one skilled in the art readily recognizes that any combination of default functions, pointer location, and/or finger behavior (e.g., multiple clicks) could have been used to define the default virtual pointing device. Moreover, a simultaneous single click (or multiple clicks) of two or more fingers could invoke a function/command.

**[0034]** Fig. 3 illustrates a flow chart containing detailed logic for implementing the preferred embodiments. At 302, touchscreen 200 detects sound/heat/pressure, etc., from a handprint (or object), 'or alternatively, a portion of a handprint. At 306, the OS reads the handprint and calculates the measurements previously described and illustrated in Fig. 2. At 310, the OS searches user files in a database for the handprint measurements. At 312, if the OS locates any existing handprint measurements within a default tolerance of 10% (which can later be changed by the user, described herein), at 320, the OS reads all information in that user file and, at 322, draws a virtual pointing device on the touchscreen under the user's hand (or portion of the hand) based on pre-defined characteristics found in the user file. Additionally, in the future, if any objects and/or text have been selected by the virtual pointing device, they will be drawn in a position relative to their previous location to the virtual pointing device (described herein). At 333, the OS requests a confirmation that the user match is correct. If the user continues to use the virtual pointing device, then the OS interprets this use as a confirmation. The OS displays a user id label next to the virtual pointing device or, alternatively, may display the virtual pointing device as a specific color shading. Otherwise, if the wrong user has been assumed, control returns to 310, where the OS searches the database for another possible match.

**[0035]** At 324, the OS determines if there is any consistent unusual behavior or undefined behavior for four or more seconds, such as, for example, failing to detect the fingerprint(s), the palmprint, or no handprint on the touchscreen. If the OS detects no unusual behavior, the OS performs a work event loop at 326 (see Fig. 9) and control returns to 324. Referring to Fig. 9, at 902, the OS determines if any movement of the hand across the touchscreen has occurred and, if so, at 904 the OS moves the virtual pointing device in accordance with the movement of the hand. At 906, the OS determines if movement of a finger or thumb has occurred to invoke a function/command and, if so, at 908 the OS invokes that function/command on any object/text positioned under the pointer. Control returns to 324.

**[0036]** Returning to 324 of Fig. 3, if the OS detects unusual behavior or undefined behavior for a certain amount of time (e.g., 4 seconds), at 328, the OS determines if all fingers have been lifted off the touchscreen while the palm remains on the touchscreen. Alternatively, one skilled in the art recognizes that many other indicators could replace the "all fingers lifted" indicator, such as determining if a combination of fingers have been lifted or determining if the palm has been lifted while the fingerprints remain in contact with the touchscreen. If the OS determines that all fingers have been lifted off the touchscreen, at 330, the OS displays a main menu 600 (see Fig. 6, described herein) prompting the user to provide any re-customization of the virtual pointing device. At 344, the OS displays the new virtual pointing device in accordance with any changes made at 330 and control returns to 324.

**[0037]** Returning to 328, if all fingers were not detected as being raised while the palm remained in contact with the touchscreen, at 342, control is directed to Fig. 8. Referring to Fig. 8, at 810, the OS determines if the entire hand (or object) has been lifted off the touchscreen. If the entire hand has not been lifted off the touchscreen, but unusual or undefined behavior has occurred, such as lifting a combination of fingers, thumb and/or palm (whose behavior does not have a corresponding defined functionality), control is directed to 814, where the OS re-draws the virtual pointing device under the hand based on the user file. This indicates to the user that the immediate past hand/finger behavior has no defined function. If the entire hand has been lifted off the touchscreen, at 811, the OS continues to display the virtual pointing device on the touchscreen in its current location for a period of time (e.g., 5 seconds), but in an obvious hibernated state, meaning the fingerprint and palmprint areas will be viewed as translucent areas on the touchscreen. When the virtual pointing device is in the obviously hibernated state, no functionality can be invoked until it is activated (i.e., brought out of hibernation, described herein). At 812, the OS determines if the hand has been re-positioned on the touchscreen within five seconds of detecting its removal. If the hand has not been re-positioned on the touchscreen within the five seconds, control is directed to 826 (described herein). However, if the OS detects the hand being re-positioned on the touchscreen within 5 seconds, at 816, the OS determines if more than one virtual pointing device is concurrently being used and, if so, if more than one user had lifted his/her hand off the touchscreen at the time the hand was re-positioned on the touchscreen. If not, at 814, control is directed to 322 of Fig. 3, whereby the OS activates and moves the virtual pointing identified by the user file under the re-positioned hand. Additionally, if any objects and/or text were selected by the virtual pointing device at the time the hand was lifted, they will be re-drawn in a position relative to their previous location to the virtual pointing device (described herein).

**[0038]** If more than one user had concurrently lifted his/her hand off the touchscreen, at 820, the OS reads the handprint of the re-positioned hand and calculates the measurements previously described and illustrated in Fig. 2. At 822, the OS searches the user files of the virtual pointing devices having a detected lifted hand for a hand measurement match. If a match is not found, at 823, the OS searches the user file database for the user identification of one of the virtual pointing devices having a detected lifted hand. The OS then displays a dialog (not shown) asking the user if he/she is the user identified by the user identification. If the user indicates that he/she is identified by the user identification at 825, at 826, control is directed to 322 of Fig. 3, whereby the OS moves the virtual pointing device identified by the user file under the re-positioned hand, and if any objects and/or text were selected by the virtual pointing device, they will be re-drawn in a position relative to their previous location to the virtual pointing device (described herein). However, if the user indicates that the identification does not identify the user at 825, the OS determines if that identification is the last user file of a virtual pointing device having a detected lifted hand. If not, control returns to 823 where the OS searches the next user file of a virtual pointing device having a detected lifted hand. This process repeats until a match is found between the user and the user identification and, therefore, the corresponding virtual pointing device having a detected lifted hand. If the OS has searched the last user file and no match has been found, at 839, control is directed to 310 of Fig. 3, where the OS search all the user files for the user's hand.

**[0039]** Returning to 812, if the hand has not been re-positioned on the touchscreen within 5 seconds, at 826, the OS continues to display the virtual pointing device in the obvious hibernated state and, at 828, prompts the user in a dialog (not shown) if the user desires to quit. If the user desires to quit, control is directed to 830 where the OS removes the virtual pointing device from the display. If the user does not desire to quit, at 832, the OS places the mouse in a "hidden hibernation" state, which means that the mouse image displayed on the touchscreen in the obvious hibernated state (i.e., translucent) begins to fade with time, but can be instantly activated when the user next touches the touchscreen. Therefore, the OS transforms the virtual pointing device from obvious hibernation (e.g., displayed in an translucent form) to hidden hibernation. After a user specified time (e.g., 30 minutes), the OS interprets the time delay as meaning that the virtual pointing device is no longer needed. At 836, if the OS detects a hand placed on the touchscreen within 30 minutes, at 840, the OS brings the virtual pointing device out of hidden hibernation, redraws it under the hand, and control returns to 324 of Fig. 3. Otherwise, at 838, the OS removes the virtual pointing device currently in a hidden hibernation state from memory (e.g., RAM).

**[0040]** Returning to 312 of Fig. 3, the OS determines if a match has been found between a measured hand placed on the touchscreen and any existing user files. If the OS detects several user files having handprint measurements closely matching the handprint in question, at 316, the OS displays in a drop down menu (not shown) on the touchscreen showing those users having the closest match. At 318, the OS waits for the user to select (using his other hand) from the drop down menu a match in user identity, or a selection indicating that no match has occurred. If a match has occurred, control is directed to 320 (previously described). If no match has occurred, control is directed to 314, where the OS displays on the touchscreen a menu (see 510 in Fig. 5) asking the user to indicate if he/she desires to create a customized virtual pointing device. If the user does not desire to create a customized virtual pointing device, the OS prompts the user to place his/her hand on the touchscreen and, in response, the OS builds a generic virtual pointing device under the user's hand, as shown in Fig. 5, having the default finger/palm behavior and fingerprint functionality as previously described and control is directed to 324.

**[0041]** If the user does desire to create a customized virtual pointing device, at 332, the OS opens a user file. At 334, the OS stores the size of the fingerprints and palmprint in the user file. At 336, the OS calculates the distance between the first reference point (previously described and shown in Fig. 2) and the farthest point to each fingerprint of the first three fingers. Additionally, the OS could calculate the second reference point and distance therefrom to the fourth fingerprint. At 338, the OS prompts the user for a user identification and displays main menu 600, which prompts the user to enter virtual pointing device characteristics, such as the virtual pointing device shape, pointer location, behavior and sensitivity, and fingerprint functionality (described herein and shown in Fig. 6). At 340, the OS stores all information in the user file. Control is directed to 322, where the OS draws the virtual pointing device under the hand (or portion of the hand) based on the information stored in the user file.

**[0042]** At 324, the OS determines if any unusual behavior has occurred. If so, at 328, the OS determines if all fingers of the hand have been lifted off the touchscreen. If so, at 330, the OS displays a main menu 600 as illustrated in Fig. 6, prompting the user to provide any customization of the virtual pointing device.

**[0043]** Referring to Fig. 6, after the OS displays the main menu 600, the user may remove his/her hand from the touchscreen. If the user selects shape button 620, a "shape" menu appears (see 700 in Fig. 7) that allows the user to define/redefine the shape of the virtual pointing device. Referring to shape menu 700 of Fig. 7, the OS displays several options to the user. For example, the user could select a "fingers only" virtual pointing device (see Fig. 4, described herein) whereby only the fingers need to be in contact with the touchscreen to move the virtual pointing device, or a palm and thumb only virtual pointing device, whereby only the thumb and palm need to be in contact with the touchscreen to move the virtual pointing device. In the latter case, movement of the fingers would not be assigned functionality. Additionally, "a thumb plus one finger" or "palm" virtual pointing device could be created. However, because the OS invokes the main menu 600 (see Fig. 6) by lifting all fingers while keeping the palm in contact with the touchscreen, if the user defines a new virtual pointing device that does not include the palm, the user could not later re-program the functionality of that special virtual pointing device. Rather, the user would have to start with a generic virtual pointing device to create a new device. Alternatively, a different technique could be used to activate the main menu 600 without departing from the scope of the invention.

**[0044]** The user may change the default accuracy tolerance amount from 10% to one of a number of preprogrammed values. To do so, the user presses accuracy button 702 and, in response, a drop-down list (not shown) of values (e.g., 4%, 8%, 20%) appears for the user's selection. The user enters/saves all selections by pressing button 704. In response, the main menu 600 shown in Fig. 6 reappears.

**[0045]** Returning to Fig. 6, if the user selects define function button 625, a "define function" menu appears that allows the user to define/redefine the functionality of the fingerprints/palmprint areas. Specifically, define functionality menu 730 in Fig. 7 allows the user to change the functionality of each fingerprint and thumbprint area by pressing the associated button next to the appropriate finger. For example, the user has pressed button 732, indicating that he/she desires to change the functionality of the second finger (i.e., fingerprint area 230). In response, the OS displays drop-down list 740 of pre-defined functions stored in memory. The user has selected open function 742 where, in response, the OS displays another drop-down list 746. The user selected a double click 744 of the second finger to invoke the open function. The user then presses save button 748 to save the entries in the user file. In response, the main menu 600 shown in Fig. 6 appears. However, one skilled in the art readily recognizes that other changes in finger behavior and fingerprint area functionality may be made without departing from the scope and spirit of this preferred embodiment.

**[0046]** Returning to Fig. 6, if-the user selects define pointer button 630, a "define pointer" menu appears that allows the user to define/redefine the shape, sensitivity, and position of the pointer on the virtual pointing device. Referring to define pointer menu 760 in Fig. 7, the user has a number of choices regarding the pointer. For example, the user can select a small, medium or large arrow, and/or a blinking arrow. The user can also select small or large pointer sensitivity, and the position of the pointer with respect to the virtual pointing device. For example, the pointer may be positioned over the third finger (default position), over the first finger, or below the palm. However, one skilled in the art readily recognizes that numerous changes in pointer behavior may be made without departing from the scope and spirit of this preferred embodiment. The user presses save button 762 to save the entries and, in response, the main menu 600 appears.

**[0047]** Finally, in Fig. 6, the user has the option of saving and exiting by pressing save/exit button 635, or cancelling all changes and returning to the default virtual pointing device by pressing cancel button 615.

**[0048]** Referring to Fig. 4, in a second embodiment, the OS displays pre-determined, standard size fingerprint areas 420, 430, 435 and 440 and pointer 450 as a non-activated (also referred to as "obviously hibernated") virtual pointing device. The fingerprint areas of the virtual pointing device are translucent such that object icons can be seen through them. To activate the virtual pointing device, the user places one or more fingers over a fingerprint area 420, 430, 435 or 440 on touchscreen 400. Once activated, the OS assigns a default function (e.g., default function displayed above each fingerprint area) to each fingerprint area.

**[0049]** In response to detecting a finger placed over one or more of the fingerprint areas, the OS resizes the fingerprint area(s) to the size of the finger placed on the fingerprint area. Therefore, for example, if a finger is smaller than a fingerprint area, that fingerprint area will be reduced to the size of the finger. Conversely, if the finger is larger than the fingerprint area, the fingerprint area will be enlarged to the size of the finger.

**[0050]** Alternatively, when the OS detects a sound pattern (or heat, pressure, etc.) over one or more of the translucent fingerprints areas 420, 430, 435 and 440, the OS activates only those areas of virtual pointing device having a finger placed thereon. In this case, the OS assigns a default function (e.g., default function displayed above each fingerprint area) to each fingerprint area having a finger placed over it. However, the fingerprint areas not having a finger placed over them will not be activated and, as such, will not have the default function assigned to them until they are activated. Each fingerprint area may be activated at any time.

**[0051]** As the user slides his/her fingers over touchscreen 400, the OS detects the touchscreen pixel coordinates under the user's moving fingers and, in response, continuously re-defines the "activated" areas of the virtual pointing device to be the touchscreen areas under the fingers. Therefore, the virtual pointing device moves with and according to the movement of the user's fingers. However, while not all of the fingerprint areas may be activated at once, all fingerprint areas move together as one object.

**[0052]** The OS positions pointer 450 near the fingerprint area 420 such that pointer 450 moves in accordance with movement of the virtual pointing device. Therefore, the user could, for example, move the virtual pointing device such that pointer 450 is positioned over a desired object icon. Alternatively, the user could merely lift his hand and place it at a desired location, whereby the OS would re-create the virtual pointing device under the user's fingers at the new location. Additionally, any objects or text selected by the virtual pointing device at the time the hand was lifted would also be re-drawn at the new location.

**[0053]** In this example, the user has placed his/her first finger over fingerprint area 420 to activate that area of the virtual pointing device. If the user desires to resize the distance between the fingerprint areas of the virtual pointing device, the user merely places a separate finger, one by one, over each displayed fingerprint area (thereby activating them) and then slides each finger outward/inward or upward/downward, as appropriate, to customize the distance between the fingerprint areas of the virtual pointing device. In this manner, the user customizes the shape/size of the virtual pointing device to the shape/size of his/her hand. However, the user must actively customize the shape/size of the virtual pointing device each time he/she uses it.

**[0054]** Once the user positions pointer 450 over a desired object icon 422, the user could, for example, single click his first finger over fingerprint area 420 to transfer focus to object icon 422. However, only generic functions (or previously established functions) can be used for this embodiment.

## Claims

1. A method for directing a computer system, having at least a processor, memory, and touchscreen, to create a reshapable pointing device, comprising the

steps of:

(a) displaying a pointing device on the touchscreen; and

(b) in response to detecting at least one finger placed on the pointing device, reshaping the pointing device in accordance with the size of the finger.

**2.** The method according to claim 1 wherein step (a) comprises the step of: displaying at least one translucent area on the touchscreen of a first size.

**3.** The method according to claim 2 wherein step (b) further comprises the step of:

in response to detecting a finger placed on the translucent area, wherein the finger is a second size, reshaping the translucent area to substantially the second size.

**4.** The method according to claim 1 further comprising the steps of:

displaying at least two translucent areas on the touchscreen, wherein the translucent areas are a first distance apart;

in response to detecting at least one finger placed on each translucent area, activating each translucent area; and

in response to moving one of the fingers with respect to the other finger such that the fingers are a second distance apart, adjusting the translucent areas to be substantially the second distance apart.

**5.** The method according to claim 2 further comprising the step of:

in response to detecting at least one finger placed on the translucent area, activating the pointing device.

**6.** The method according to claim 2 further comprising the step of:

in response to detecting at least one finger placed on the translucent area, activating the translucent area.

**7.** The method according to claim 3 wherein step (b) further comprises the step of:

detecting heat, pressure, or sound from the finger to determine the second size of the finger.

**8.** A computer system adapted for carrying out the method for directing a computer having at least a processor, memory, and touchscreen, to create a reshapable pointing device according to anyone of the preceding claims.

**9.** A computer program carrier comprising:
a computer usable medium having computer readable program code means embodied therein, the computer readable program code means in the article of manufacture comprising:

computer readable program code means for causing a computer system, having at least a touchscreen, to display a pointing device on the touchscreen; and

computer readable program code means for causing the computer system to, in response to detecting at least one finger placed on the pointing device, reshape the pointing device in accordance with the size of the finger.

## Patentansprüche

**1.** Verfahren, das dazu dient, ein Computersystem, welches mindestens aus einem Prozessor, einem Speicher und einem Sensorbildschirm besteht, anzuweisen, am Sensorbildschirm eine in ihrer Form veränderliche Zeigevorrichtung zu erstellen, und das folgende Schritte umfaßt:

(a) Anzeigen einer Zeigevorrichtung am Sensorbildschirm;

(b) In Reaktion auf das Erkennen von mindestens einem Finger, der auf der Zeigevorrichtung plaziert wurde: Neugestaltung der Form der Zeigevorrichtung entsprechend der Größe des Fingers.

**2.** Verfahren nach Anspruch 1, wobei der Schritt (a) folgenden Schritt umfaßt:

Anzeigen von mindestens einer durchsichtigen Fläche einer ersten Größe am Sensorbildschirm.

**3.** Verfahren nach Anspruch 2, wobei der Schritt (b) weiterhin folgenden Schritt umfaßt:

In Reaktion auf des Erkennen eines auf der durchsichtigen Fläche plazierten Fingers einer zweiten Größe: Neugestaltung der Form der durchsichtigen Fläche, so daß sie im wesentlichen der zweiten Größe entspricht.

**4.** Verfahren nach Anspruch 1, das darüber hinaus folgende Schritte umfaßt:

Anzeigen von mindestens zwei durchsichtigen Bereichen am Sensorbildschirm, die sich in einem ersten Abstand zueinander befinden;

In Reaktion auf das Erkennen von zumindest einem Finger, der auf einer der durchsichtigen Flächen plaziert wurde: Aktivieren aller durchsichtigen Flächen; und

In Reaktion auf das Bewegen der Finger in bezug auf den anderen Finger, so daß sich die Finger in einem zweiten Abstand zueinander befinden: Einstellen der durchsichtigen Flächen, so daß sie sich im wesentlichen in einem zweiten Abstand zueinander befinden.

**5.** Verfahren nach Anspruch 2, das darüber hinaus folgenden Schritt umfaßt:

In Reaktion auf das Erkennen von mindestens einem Finger, der auf der durchsichtigen Fläche plaziert wurde: Aktivieren der Zeigevorrichtung.

**6.** Verfahren nach Anspruch 2, das darüber hinaus folgenden Schritt umfaßt:

In Reaktion auf das Erkennen von mindestens einem Finger, der auf der durchsichtigen Fläche plaziert wurde: Aktivieren der durchsichtigen Fläche.

**7.** Verfahren nach Anspruch 3, wobei Schritt (b) außerdem folgenden Schritt umfaßt:

Erkennen von Wärme-, Druck- oder Schallwirkungen, die vom Finger ausgehen, um so die zweite Größe des Fingers bestimmen zu können.

**8.** Computersystem, das für die Ausführung des Verfahrens zur Steuerung eines Computers mit mindestens einem Prozessor, Speicher und Sensorbildschirm so angepaßt wurde, daß es eine in ihrer Form veränderliche Zeigevorrichtung nach einem der vorstehenden Ansprüche erstellen kann.

**9.** Computerprogrammträger, der folgendes umfaßt: ein von einem Computer verwertbares Medium mit computerlesbaren Programmcode-Mitteln, wobei die computerlesbaren Programmcode-Mittel werkseitig folgendes umfassen:

Computerlesbare Programmcode-Mittel, die ein mit mindestens einem Sensorbildschirm ausgestattetes Computersystem zum Aufbau einer virtuellen Zeigevorrichtung am Sensorbildschirm veranlassen; und

Computerlesbare Programmcode-Mittel, die das Computersystem veranlassen, in Reaktion auf das Erkennen von mindestens einem Finger, der auf der Zeigevorrichtung plaziert wurde, die Form der Zeigevorrichtung entsprechend der Größe des Fingers neu zu gestalten.

**Revendications**

**1.** Procédé pour commander un système informatique, ayant au moins un processeur, une mémoire et un écran tactile, pour créer un dispositif de pointage déformable, le précédé comprenant les phases qui consistent à :

a. Afficher un dispositif de pointage sur l'écran tactile ; et

b. En réponse à la détection d'au moins un doigt sur le dispositif de pointage, déformer le dispositif de pointage en se conformant à la taille du doigt.

**2.** Procédé selon la revendication 1 où la phase (a) comprend la phase qui consiste à :

afficher, sur l'écran tactile, au moins une zone translucide ayant une première taille.

**3.** Procédé selon la revendication 2où la phase (b) comprend en outre la phase qui consiste à :

en réponse à la détection d'un doigt sur la zone translucide, où le doigt a une seconde taille, déformer la zone translucide pour qu'elle ait sensiblement la seconde taille.

**4.** Procédé selon la revendication 1 comprenant le outre les phases qui consistent à :

afficher au moins deux zones translucides sur l'écran tactile, où une première distance sépare les zones translucides ;

en réponse à la détection d'au moins un doigt sur chaque zone translucide, activer chaque zone translucide ; et

en réponse au déplacement de l'un des doigts par rapport à l'autre, de telle sorte qu'une deuxième distance sépare les deux doigts, ajuster les zones translucides pour qu'elles soient espacées sensiblement de la deuxième distance.

**5.** Procédé selon la revendication 2 comprenant en outre la phase qui consiste à :

en réponse à la détection d'au moins un doigt sur la zone translucide, activer le dispositif de pointage.

**6.** Procédé selon la revendication 2 comprenant en outre la phase qui consiste à :

en réponse à la détection d'au moins un doigt sur la zone translucide, activer la zone translucide.

**7.** Procédé selon la revendication 3 où la phase (b) comprend en outre la phase qui consiste à :

détecter la chaleur, la pression ou le bruit provoqués par le doigt pour déterminer la seconde taille du doigt.

**8.** Système informatique adapté pour exécuter le procédé de commande d'un ordinateur ayant au moins un processeur, une mémoire et un écran tactile pour créer un dispositif de pointage déformable selon l'une quelconque des revendications précédentes.

**9.** Support de programme informatique comprenant : un support utilisable sur machine dans lequel est réalisé un moyen de code de programme lisible par machine, le moyen de code de programme lisible par machine du support de programme informatique comprenant :

un moyen de code de programme lisible par machine pour qu'un système informatique, qui possède au moins un écran tactile, affiche un dispositif de pointage sur l'écran tactile ; et
un code de programme lisible par machine pour que, en réponse à la détection d'au moins un doigt sur le dispositif de pointage, le système informatique déforme le dispositif de pointage en se conformant à la taille du doigt.

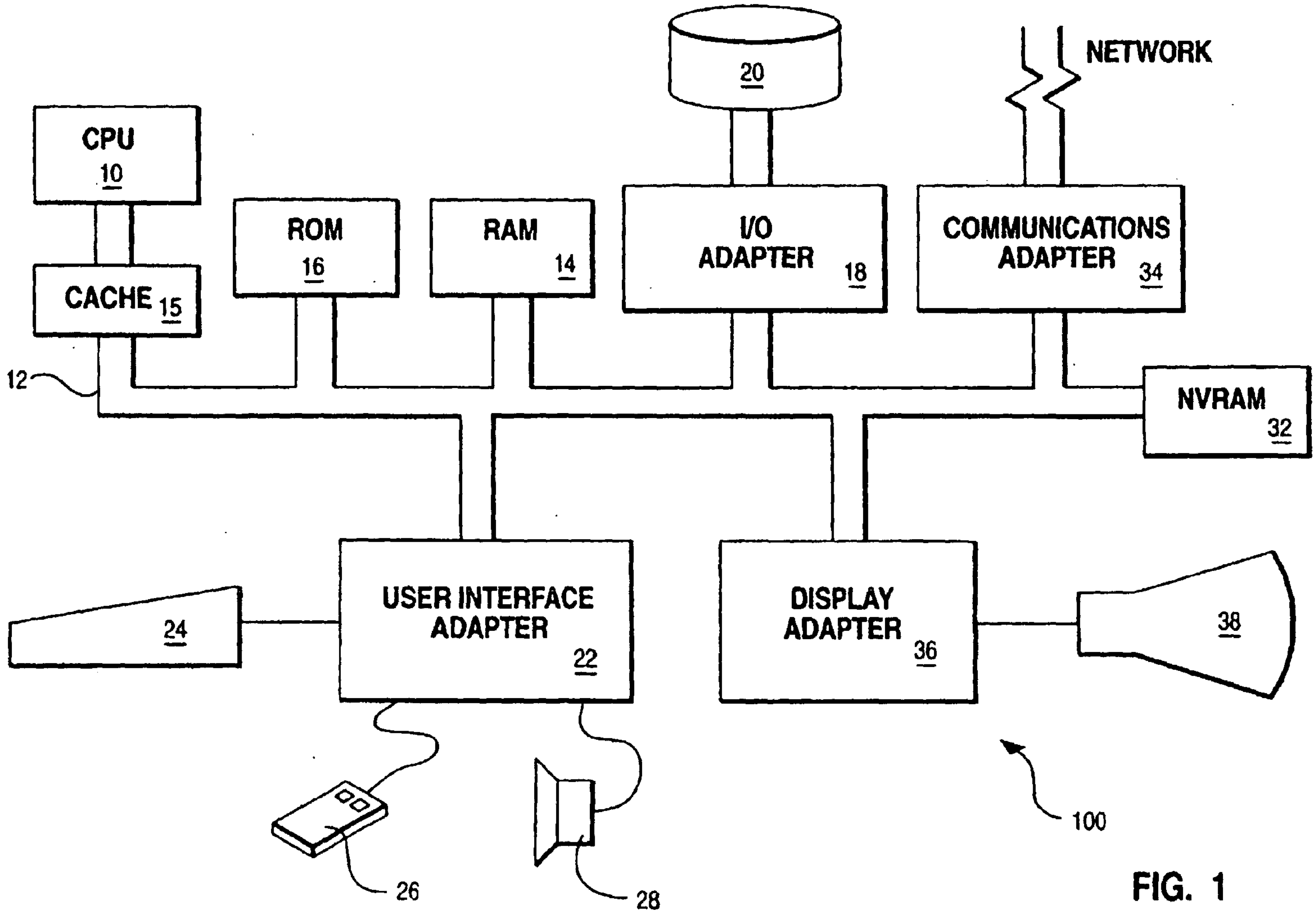

CPU
10
CACHE 15
ROM
16
RAM
14
20
I/O
ADAPTER
18
NETWORK
COMMUNICATIONS
ADAPTER
34
NVRAM
32
12
USER INTERFACE
ADAPTER
22
DISPLAY
ADAPTER
36
24
38
26
28
100

FIG. 1

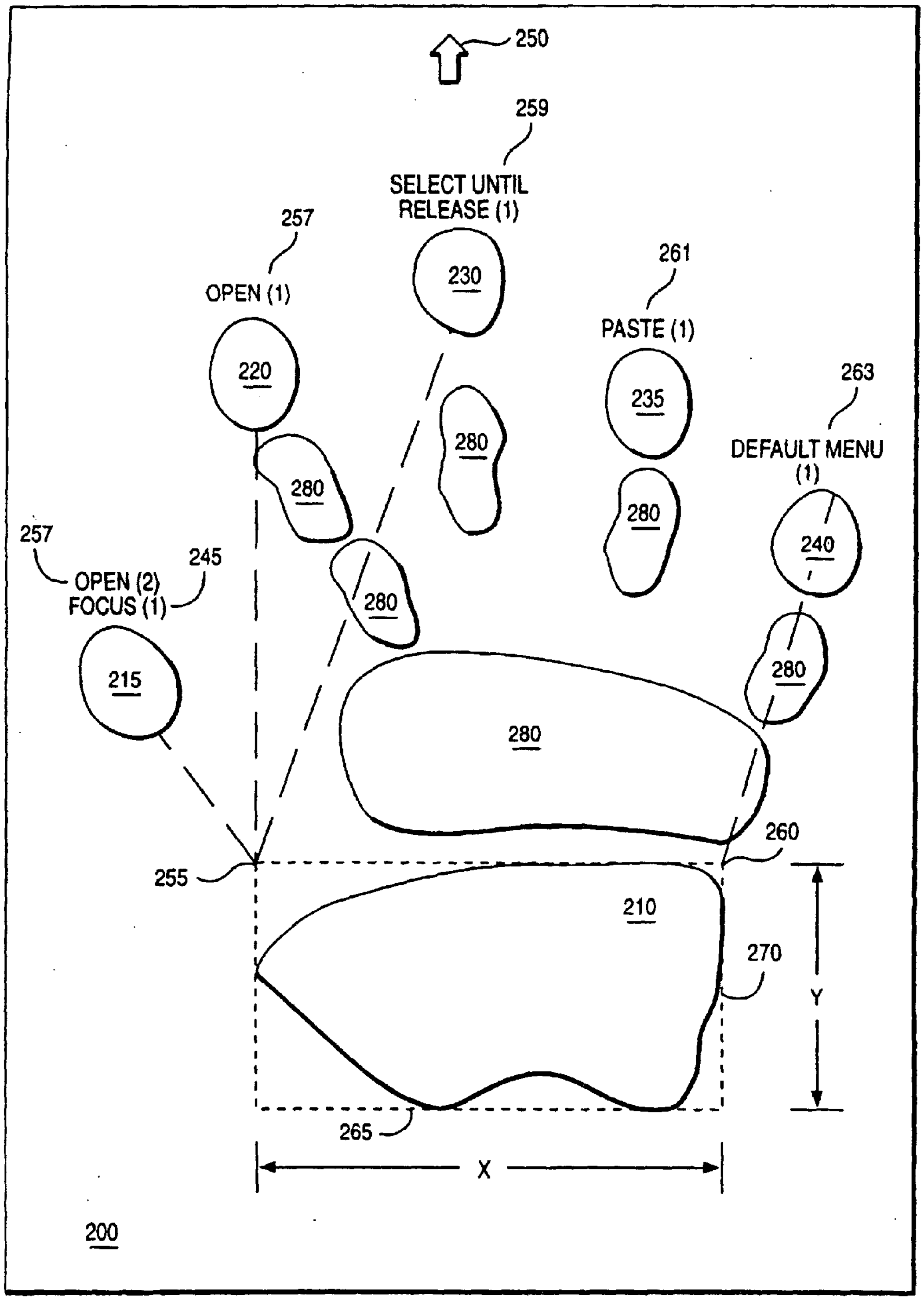

250
259
SELECT UNTIL RELEASE (1)
257
OPEN (1)
230
261
PASTE (1)
220
280
235
263
DEFAULT MENU (1)
280
280
257
OPEN (2)
FOCUS (1)
245
240
280
280
215
280
280
260
255
210
270
Y
265
X
200

FIG. 2

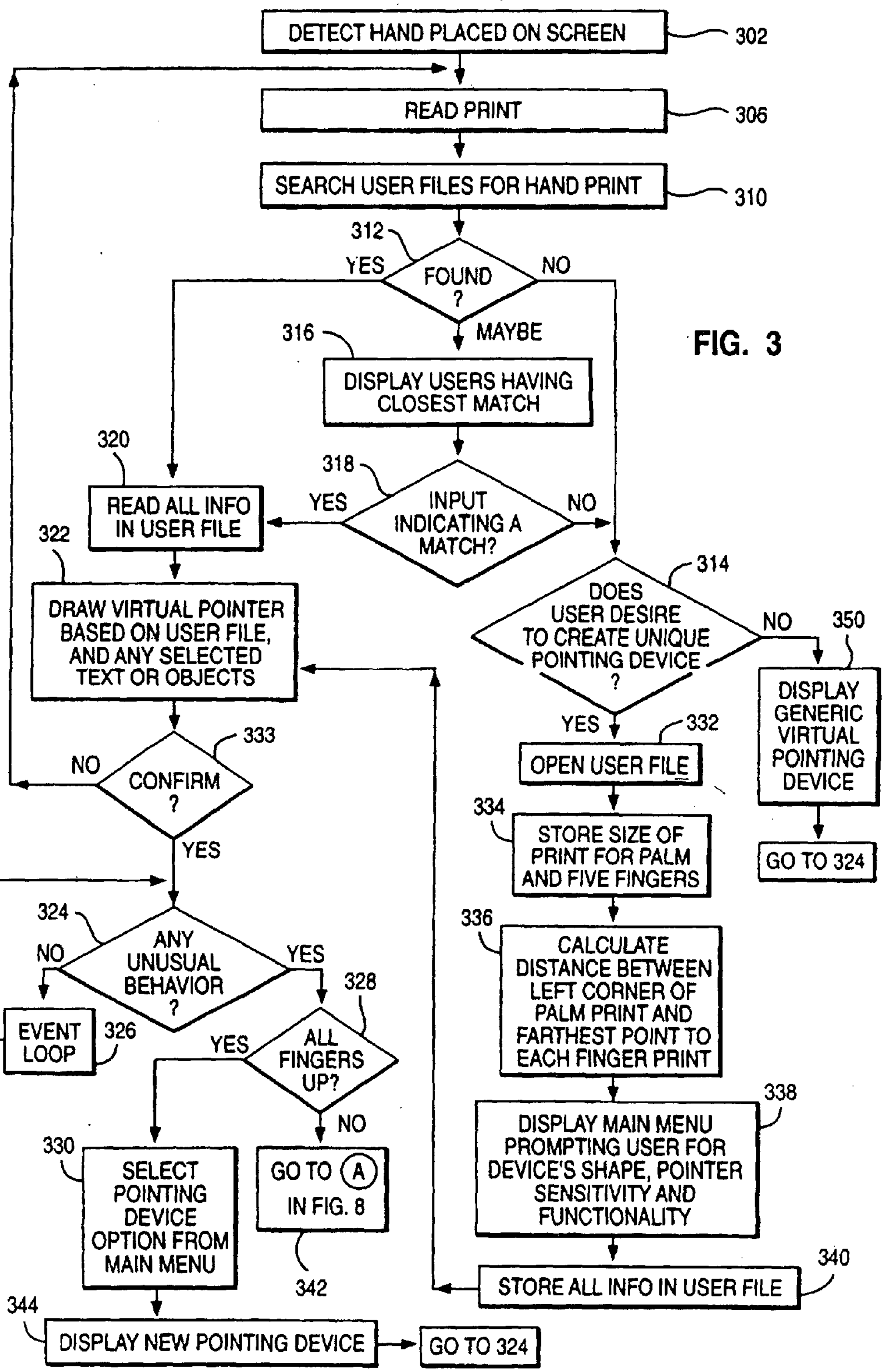

DETECT HAND PLACED ON SCREEN
302
READ PRINT
306
SEARCH USER FILES FOR HAND PRINT
310
312
FOUND ?
YES
NO
MAYBE
316
DISPLAY USERS HAVING CLOSEST MATCH
318
INPUT INDICATING A MATCH?
YES
NO
320
READ ALL INFO IN USER FILE
322
DRAW VIRTUAL POINTER BASED ON USER FILE, AND ANY SELECTED TEXT OR OBJECTS
314
DOES USER DESIRE TO CREATE UNIQUE POINTING DEVICE ?
NO
350
DISPLAY GENERIC VIRTUAL POINTING DEVICE
GO TO 324
YES
332
OPEN USER FILE
333
CONFIRM ?
NO
YES
334
STORE SIZE OF PRINT FOR PALM AND FIVE FINGERS
336
CALCULATE DISTANCE BETWEEN LEFT CORNER OF PALM PRINT AND FARTHEST POINT TO EACH FINGER PRINT
324
ANY UNUSUAL BEHAVIOR ?
NO
YES
328
EVENT LOOP
326
ALL FINGERS UP?
YES
NO
338
DISPLAY MAIN MENU PROMPTING USER FOR DEVICE'S SHAPE, POINTER SENSITIVITY AND FUNCTIONALITY
330
SELECT POINTING DEVICE OPTION FROM MAIN MENU
GO TO (A) IN FIG. 8
342
340
STORE ALL INFO IN USER FILE
344
DISPLAY NEW POINTING DEVICE
GO TO 324

FIG. 3

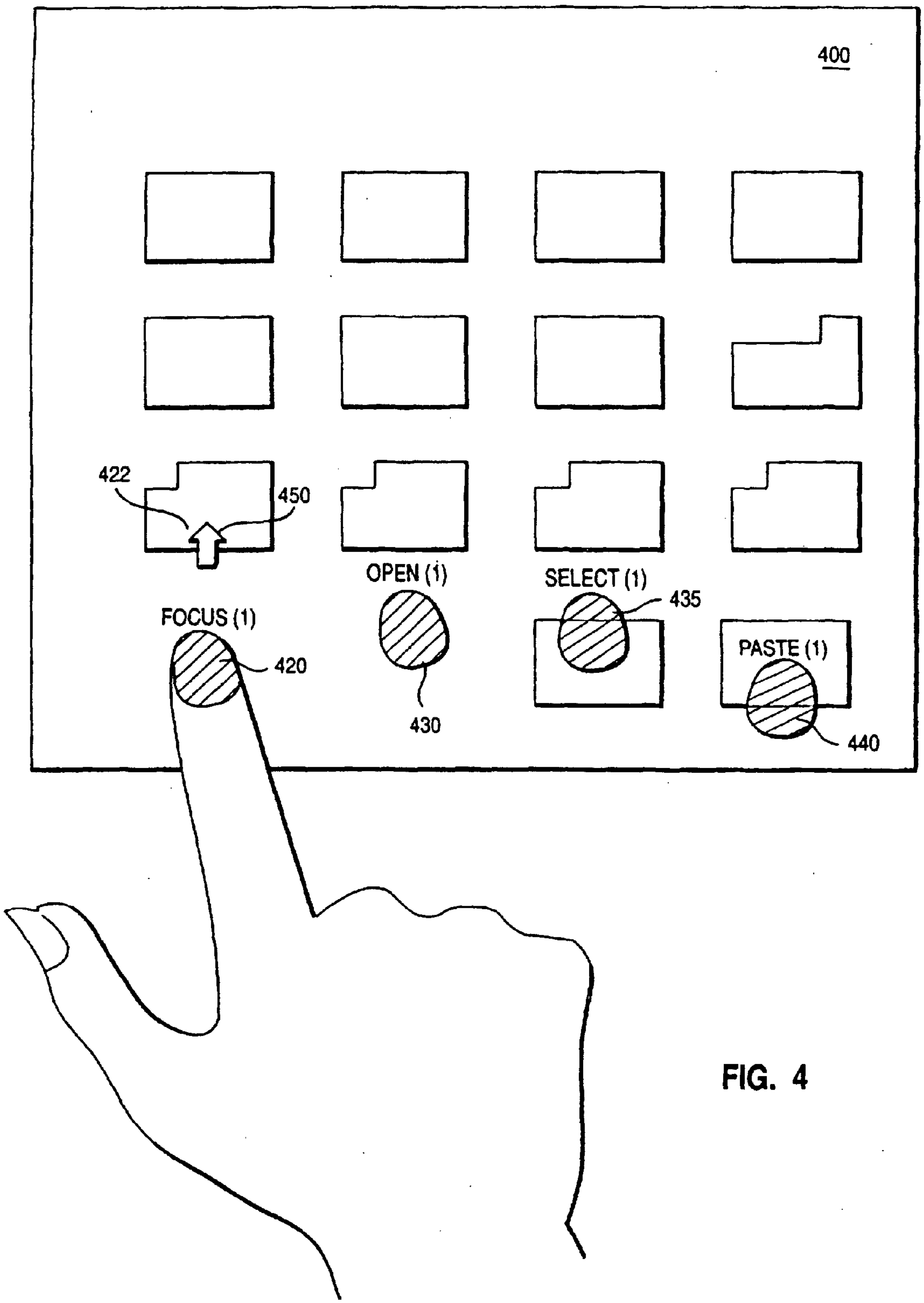



FIG. 4

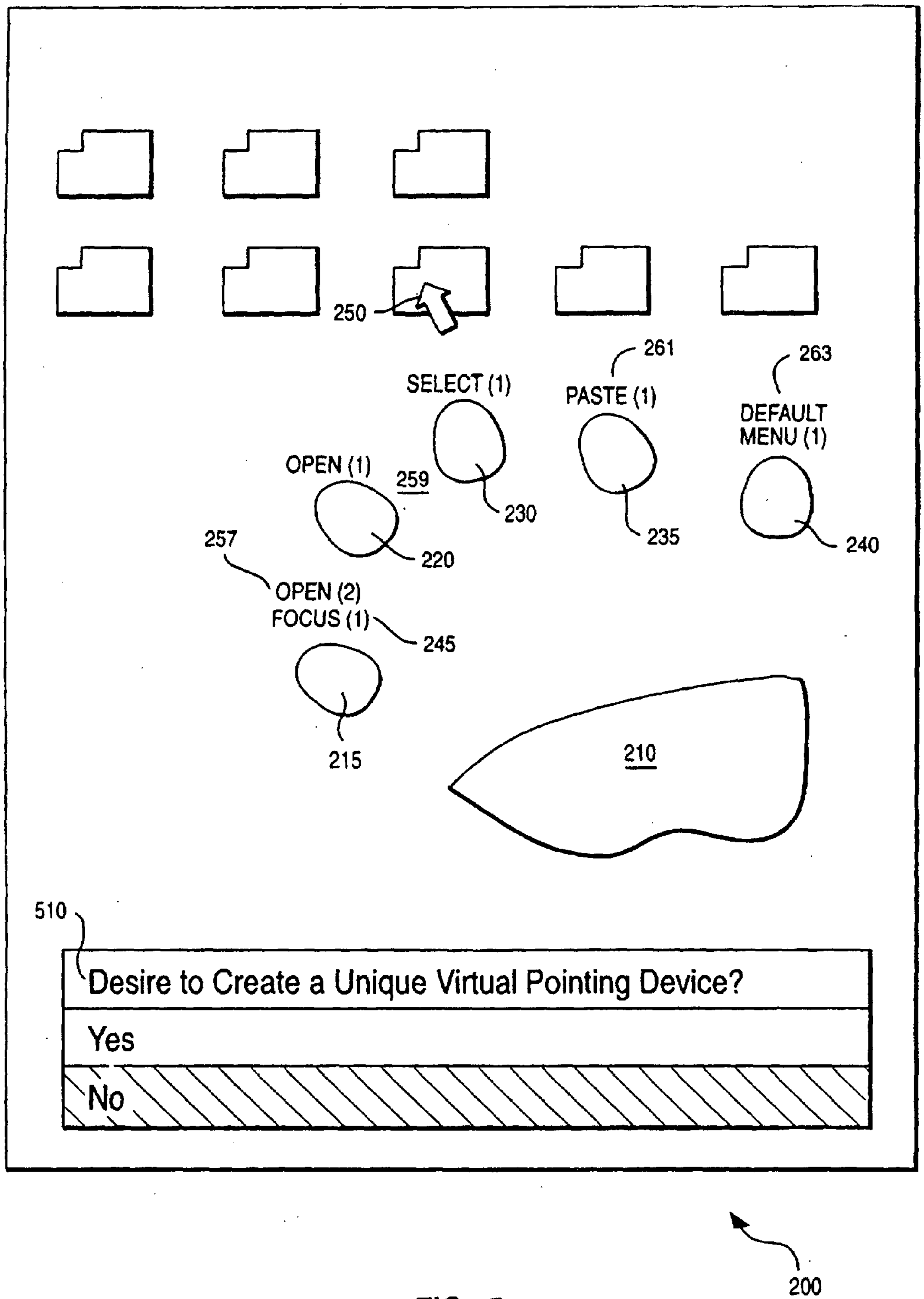

250
261
SELECT (1)
PASTE (1)
263
DEFAULT MENU (1)
OPEN (1)
259
230
235
240
257
220
OPEN (2)
FOCUS (1)
245
215
210
510
Desire to Create a Unique Virtual Pointing Device?
Yes
No
200

FIG. 5

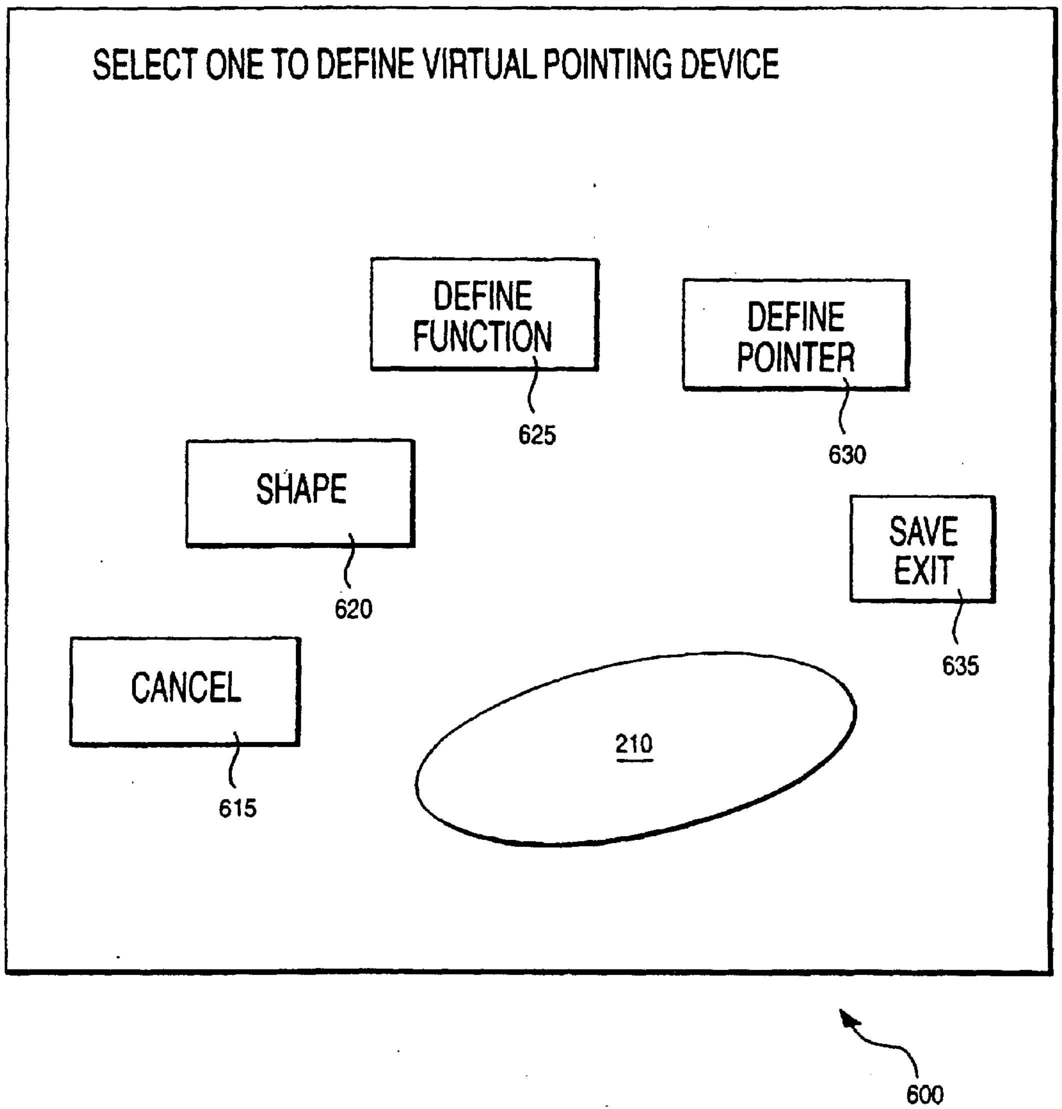

SELECT ONE TO DEFINE VIRTUAL POINTING DEVICE
DEFINE FUNCTION
625
DEFINE POINTER
630
SHAPE
620
SAVE EXIT
635
CANCEL
615
210
600

FIG. 6

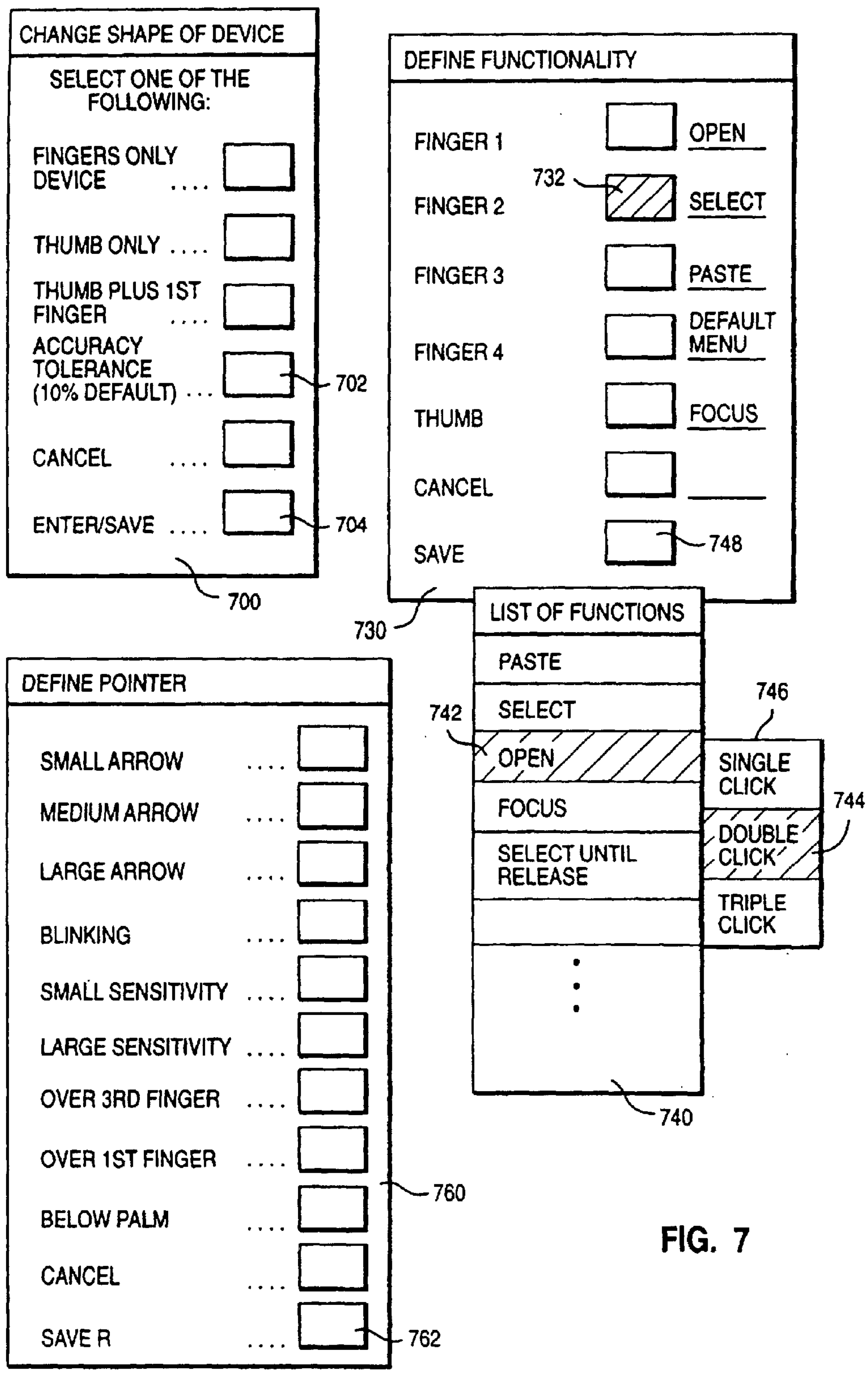
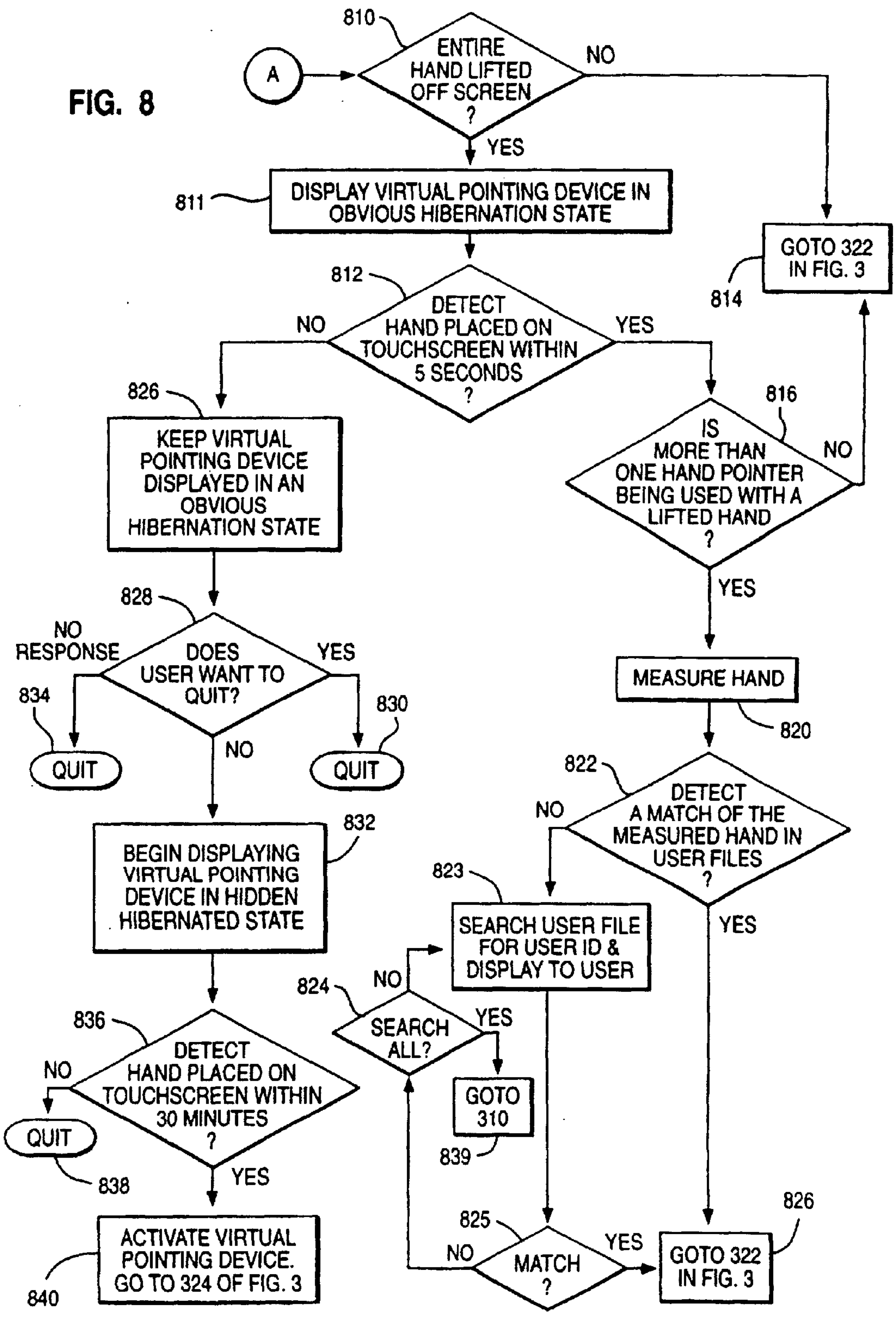

CHANGE SHAPE OF DEVICE
SELECT ONE OF THE FOLLOWING:
FINGERS ONLY DEVICE ....
THUMB ONLY ....
THUMB PLUS 1ST FINGER ....
ACCURACY TOLERANCE (10% DEFAULT) ...
CANCEL ....
ENTER/SAVE ....
702
704
700
DEFINE FUNCTIONALITY
FINGER 1
FINGER 2
FINGER 3
FINGER 4
THUMB
CANCEL
SAVE
OPEN
SELECT
PASTE
DEFAULT MENU
FOCUS
732
748
730
LIST OF FUNCTIONS
PASTE
SELECT
OPEN
FOCUS
SELECT UNTIL RELEASE
742
746
SINGLE CLICK
DOUBLE CLICK
TRIPLE CLICK
744
740
DEFINE POINTER
SMALL ARROW ....
MEDIUM ARROW ....
LARGE ARROW ....
BLINKING ....
SMALL SENSITIVITY ....
LARGE SENSITIVITY ....
OVER 3RD FINGER ....
OVER 1ST FINGER ....
BELOW PALM ....
CANCEL ....
SAVE R ....
760
762

FIG. 7

FIG. 8


A
810
ENTIRE HAND LIFTED OFF SCREEN ?
NO
YES
811
DISPLAY VIRTUAL POINTING DEVICE IN OBVIOUS HIBERNATION STATE
GOTO 322 IN FIG. 3
814
812
DETECT HAND PLACED ON TOUCHSCREEN WITHIN 5 SECONDS ?
NO
YES
826
KEEP VIRTUAL POINTING DEVICE DISPLAYED IN AN OBVIOUS HIBERNATION STATE
816
IS MORE THAN ONE HAND POINTER BEING USED WITH A LIFTED HAND ?
NO
YES
828
NO RESPONSE
DOES USER WANT TO QUIT?
YES
834
830
QUIT
NO
QUIT
MEASURE HAND
820
822
DETECT A MATCH OF THE MEASURED HAND IN USER FILES ?
NO
832
BEGIN DISPLAYING VIRTUAL POINTING DEVICE IN HIDDEN HIBERNATED STATE
823
SEARCH USER FILE FOR USER ID & DISPLAY TO USER
YES
824
NO
SEARCH ALL?
YES
836
DETECT HAND PLACED ON TOUCHSCREEN WITHIN 30 MINUTES ?
NO
QUIT
GOTO 310
839
838
YES
826
825
ACTIVATE VIRTUAL POINTING DEVICE. GO TO 324 OF FIG. 3
NO
MATCH ?
YES
GOTO 322 IN FIG. 3
840

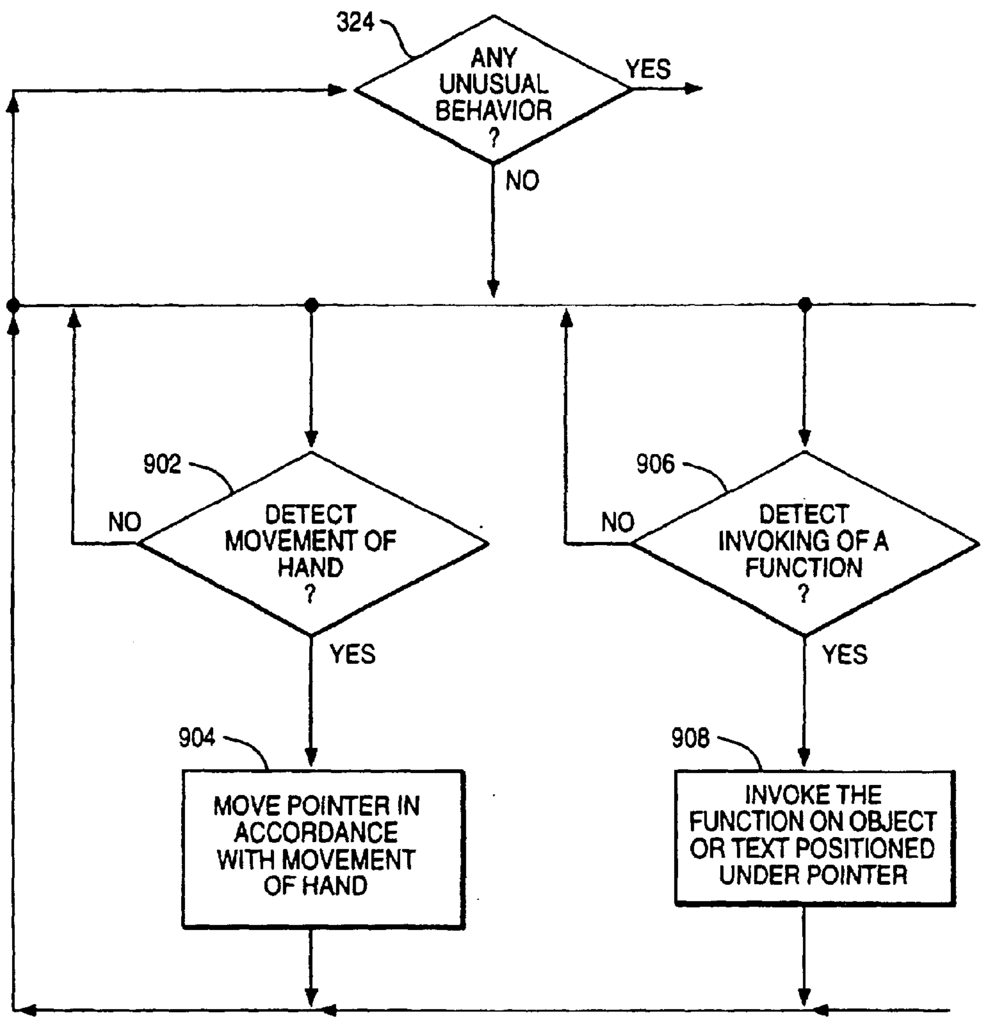

324
ANY UNUSUAL BEHAVIOR ?
YES
NO
902
DETECT MOVEMENT OF HAND ?
NO
YES
904
MOVE POINTER IN ACCORDANCE WITH MOVEMENT OF HAND
906
DETECT INVOKING OF A FUNCTION ?
NO
YES
908
INVOKE THE FUNCTION ON OBJECT OR TEXT POSITIONED UNDER POINTER

FIG. 9

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 622 722 A2**

# (12) EUROPEAN PATENT APPLICATION

(21) Application number : **94302912.4**

(22) Date of filing : **25.04.94**

(51) Int. Cl.$^5$ : **G06F 3/00,** G06F 3/033

(30) Priority : **30.04.93 GB 9308955**
**01.07.93 GB 9313637**

(43) Date of publication of application :
**02.11.94 Bulletin 94/44**

(84) Designated Contracting States :
**DE FR GB**

(71) Applicant : **RANK XEROX LIMITED**
**Parkway**
**Marlow, Buckinghamshire SLY 1YL (GB)**

(72) Inventor : **Wellner, Pierre**
**108 Allen Avenue**
**Allenhurst, N.J. 07711 (US)**

(74) Representative : **Reynolds, Julian David et al**
**Rank Xerox Ltd**
**Patent Department**
**Parkway**
**Marlow Buckinghamshire SL7 1YL (GB)**

(54) **Interactive copying system.**

(57) A system for generating new documents (20) from originals (4) containing text and/or images (22,28) employing e.g. a camera-projector system (6,8) focussed on a work surface (2), in conjunction with a copier or printer. In use, the camera 6 captures various manual operations carried out by the user, e.g. by pointing with fingers and tapping on the surface on the text or images (22,28) in an original paper document (4) on the surface (2) and representing manipulations of the text or images (22,28). Feedback to the user is provided by projection of an image (21,24) onto the surface or onto the original, or using some other visual display.

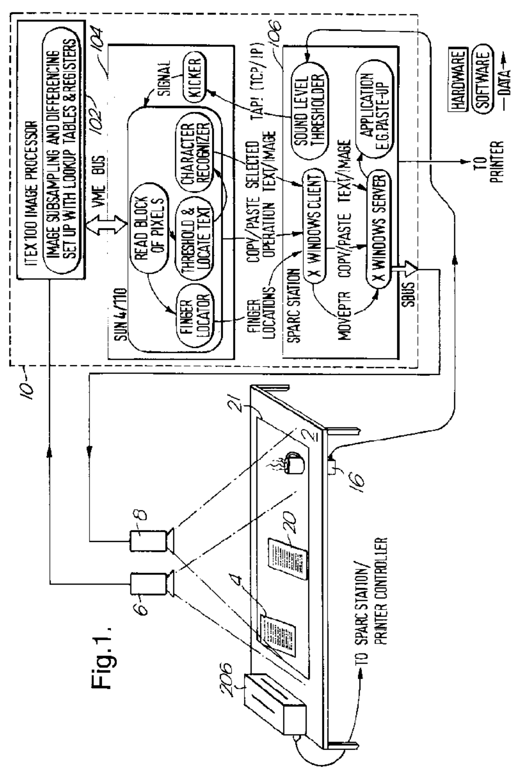



Fig.1.

EP 0 622 722 A2

Jouve, 18, rue Saint-Denis, 75001 PARIS

The present invention relates to interactive image reproduction machines, and more particularly reproduction machines for performing various operations on text or images during the creation of a new document.

It is common for office workers and others who work with documents on a regular basis to have effectively two desks-the "electronic desktop" provided by a workstation or personal computer by means of a graphical interface, and the physical desk on which paper documents are received and processed.

The electronic desktop, which has become more and more like the physical one, can perform numerous useful operations on documents stored in electronic form; but in dealing with tangible documents such devices have limitations. A paper document must either be converted into electronic form before the operations are performed on it in the electronic environment, or copying operations are carried out on the tangible document using an electrophotographic copier (such as a photocopier with an editing function) or a combined scanning and printing device, the available functions of which are restricted in nature.

US-A-5,191,440 discloses a photocopier system for combining plural image segments taken from a series of different documents and printing the series of image segments as a composite image on a common copy sheet. The documents are sequentially scanned in, and the results of page creation and edit functions may be previewed on a display screen.

It is known from EP-A-495 622 to use a camera-projector arrangement positioned above a desk, in order to select functions to be performed by selecting items located within the field of view of the camera. Such functions include calculating and translating operations carried out on data [e.g., in a paper document] located on the desk.

The present invention seeks to reduce the above-mentioned limitations on the operations which may be performed on data in a paper document, using interactive techniques in which the paper document effectively becomes part of the means for designating which operations (such as text/image selection, creation and manipulation) are carried out on the information contained in it, in order to create new documents using a processor-controlled copying or printing device.

It is an object of the present invention to provide an interactive copying system in which the available functions are expanded beyond both those provided by the conventional electronic environment alone, and those provided by existing copying equipment alone.

The present invention provides a copying system, comprising: a work surface; means for displaying images on the work surface; a camera, focussed on the work surface, for generating video signals representing in electronic form image information present within the field of view of the camera; processing means for recognising one or more manual operations relating to the image information which are executed by a user within the field of view of the camera, and for performing electronic operations, corresponding to said manual operation(s), on the electronic form to produce a modified electronic form; the displaying means being adapted to display, under the control of the processing means, simultaneously with or subsequent to said electronic operation performing step, images defined by said manual operations; and wherein said images defined by said manual operations include an image of the newly created document.

The copying system may further including a scanner, coupled to the processing means, for scanning a document containing said image information.

The system preferably further includes means, for sensing vibrational signals on the surface; the processing means being adapted to recognise a tap or strike by a user on the surface.

Preferably, the processing means includes a frame grabber, for storing video frames, and differencing means, for establishing the difference between pixel data values of corresponding pixels in successive video frames, and for displaying the resultant video frame data. Preferably, the processing means includes thresholding means, for converting multi-bit per pixel video frame data to 1 bit per pixel video frame data. Preferably, the thresholding means is adapted for carrying out said converting operation based on an estimate equal to the moving average of pixel intensities in a local area. Preferably, the local area comprises 1/nth of the width of a video frame, where n is preferably about 8.

Preferably, the processing means includes a frame grabber, for storing video frames, and means for calibrating positions in the frame grabber relative to positions within the display. Preferably, the calibrating means includes means for projecting a mark at four points in the display and carrying out said calibration by means of a four point mapping, given by

$$x' = c_1x + c_2y + c_3xy + c_4$$
$$y' = c_5x + c_6y + c_7xy + c_8,$$

where (x,y) is a point in the display (21) and (x',y') is a corresponding point in the video frame stored in the frame grabber.

The processing means may further include means for determining whether the user is right- or left-handed.

The present invention further provides a method of generating documents, comprising: providing a work surface, means for displaying images on the work surface, and a camera focussed on the work surface, said camera generating video signals representing in electronic form image information present within the field of view of the camera; recognising one or more manual operations relating to the image information which are executed by a user within the field of view of the camera; performing electronic operations, corresponding to said manual operation(s), on the electronic form to produce a modified electronic form; displaying, simultaneously with or subsequent to said electronic operation performing step, images defined by said manual operation(s); and wherein said images defined by said manual operations include an image of a newly created document.

The manual operation(s) may include designating a plurality of the extremities of a shape encompassing said selected portion of text or image information.

Preferably, the images defined by said manual operation(s) include an outline of, or a shaded area coincident with, said shape. Preferably, the shape is a rectangle.

The manual operation(s) may include pointing with a plurality of fingers at the corners of said shape.

Alternatively, said extremities are designated using a stylus in association with a position sensing tablet on the surface.

The manual operation(s) may include designating a text or image unit in the document by pointing a finger at it. The manual operation(s) may include designating a successively larger text or image unit in the document by tapping on the surface. The manual operation(s) include confirming a text or image selection by tapping on the surface. The manual operation(s) may include copying the selected text or image to a location in a new document displayed on the surface by pointing at the selected text or image using a finger or stylus and dragging the finger or stylus across the surface to said location in the new document, and dropping the selected text or image at said location by tapping on the surface. The manual operation(s) may include changing the dimensions of selected text or image by changing the separation of finger tips of the user defining extremities of the selected text or image. The manual operation(s) include placing paper signs within the field of view of the camera, the signs defining operations to be performed on selected text or image information.

The invention further provides a programmable copying apparatus when suitably programmed for carrying out the method of any of claims 4 to 7, or any of the above described particular embodiments.

The present invention further provides a copying system according to claim 9 of the appended claims.

The copying system may further including means for scanning the second document to generate an electronic version of the second document; wherein the processing means includes means for recognising the positions of the transferred image information in said electronic version; the system further including means for printing said transferred image information on said second document.

The present invention further provides an interactive image reproduction system, comprising: a plurality of workstations interconnected by a communications link, each workstation comprising a copying system according to any of claims 1 to 3, 8 or 9, each workstation being adapted for displaying the video output from the camera of the or each other workstation. The system may further include an audio or videoconferencing link between the workstations.

Embodiments of the present invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Fig. 1 is a schematic diagram of a copying system according to the invention;

Fig. 2 shows schematically a known imaging system into which the system of Fig. 1 may be incorporated;

Fig. 3 illustrates an image generated in the finger-tracking technique employed in the present invention;

Fig. 4 shows a four point mapping technique employed by the present invention;

Fig. 5 illustrates four ways to sweep out a selection rectangle when using the present invention;

Figs 6(a) to (f) show successive scenes of the desk surface in a copying operation according to one embodiment of the invention;

Fig. 7 is a flow chart of the procedure of Fig. 6;

Figs. 8(a) to (e) illustrate successive scenes of the desk surface in a copying operation according to a second embodiment of the invention;

Fig. 9 is a flow chart of the procedure of Fig. 8;

Figs. 10(a) to (h) show successive scenes of the desk surface in a copying operation according to a third embodiment of the invention;

Fig. 11 is a flow chart of the procedure of Fig. 10;

Fig. 12 shows a view from above of the desk surface during a copying operation according to a fourth embodiment of the invention;

Fig. 13 is a flow chart of the procedure of Fig. 12;

Fig. 14 shows a view from above of the desk surface during a copying operation according to a fifth embodiment of the invention; and

Fig. 15 is a flow chart of the procedure of Fig. 14.

Referring to Fig. 1, this illustrates schematically the copying system of the present invention. A flat desk surface 2 has placed on it a document 4 to be used as a source if textual or graphical information during manipulations which are described in detail below. The document 4 is located within the field of view of a video camera 6 mounted above the desk surface 2. A video projector 8 is mounted adjacent the camera 6 and projects onto the surface 2 a display 21 which is generally coincident with the field of view of the camera 6, and which, in the example shown, includes an image of a newly created document 20, as discussed below. The camera 6 and the projector 8 are both connected to a signal processing system, generally designated 10, which is in turn connected to a printing device 208 and, optionally, a document scanner 206 (see Fig. 2). A small snare-drum microphone 16 (preferably with built in amplifier) is attached to the bottom of the desk and picks up audible or vibrational signals. The system 10 monitors the (digitised) signal amplitude of the microphone 16 to determine (e.g. by comparison with a threshold value) when the user taps on the desk 2 (e.g. to designate an operation (see below)).

The architecture of the signal processing system 10 is schematically illustrated in Fig. 1. This implementation runs on standard X Window applications using the human finger as a pointing device. The system is implemented so that finger tip location and desk tapping information are sent through X in such a way that from the point of view of applications, these events are indistinguishable from those of a conventionally-used mouse. The system runs on two machines: a Sun 4/110 (104) and a SPARCstation (106). This is because the image processing board 102 plugs into a VME bus, while the projected video (LCD) display plugs into an Sbus. The images captured by the camera 6 are initially processed by an Itex100 image processing board 102. Any other suitable architecture could be used to achieve the image signal handling. Figure 1 illustrates how the software modules interface to each other and to the hardware. The system is implemented in C + + and C under SunOS and TCP/IP.

The desk-camera-projector arrangement (2,6,8) may be located remotely from the printing device 208 and any number of such arrangements may be linked up to a common printer. Alternatively, the surface 2 may itself constitute an upper surface of a copying or printing machine, or the surface of a desk next to such a machine, with the advantage that any documents created using the system may be immediately printed out and taken away by the user. The processor 10 may form an integral part of a copying or printing machine, or may be remotely located in a separate device and coupled to the printer by a conventional communications link.

In a preferred embodiment, the system of Fig. 1 forms an integral part of a printing system, for example as schematically illustrated in Fig. 2 and described in detail in EP-A-592108, with the exception that appropriate elements of the control section 207 are replaced by hardware from Fig. 1, such as the user interface 252 (which is implemented by the camera-projector arrangement (6,8)), the system control 254, etc. For additional control detail, reference is made to US-A-s 5,081,494, 5,091,971 and 4,686,542.

Interacting with objects with bare fingers is facilitated in the present invention through video-based finger-tracking. A bare finger is too thick, however, to indicate small objects such as a single letter, for which a pen or other thin object is used.

The current implementation uses simple image processing hardware to achieve the desired interactive response time (although suitable algorithms could be used to achieve the same result): it initially subsamples the image and processes it at very low resolution to get an approximate location for the finger. Only then does the system scale to its full resolution in order to get a precise location, so only small portions of the image need to be processed. If the user moves too quickly, the system loses track of where the finger is, so it immediately zooms back out to find it. The result is that large, quick movements are followed less precisely than fine movements, but for pointing applications this seems acceptable.

Interaction techniques using video-based finger tracking, are demonstrated by M. Krueger (*Artificial Reality II*, Addison-Wesley, 1991). The disadvantages of his system are discussed in UK patent application 9313637.2 (hereafter Ref. 1), a copy of which was filed with the present application.

In contrast, the present invention senses motion (since most objects on the desk 2 do not move except the user's hands and the objects they are holding): it captures sequential video frames and examines the image produced by subtracting the sequential values of each pixel in two successive frames. The result for, e.g., a moving hand looks like Figure 3. Further processing is then carried out to remove noise and to locate the precise position of the fingertips.

Motion detection uses an image loop-back feature of the board 102 that allows the most significant bits of two images to be sent through a look-up table. This table is set up to subtract the two images, allowing very fast differencing of successive frames. Current finger-tracking performance using the Sun 4/110 and Itex100 image processing board 102 is 6-7 frames/sec.

Determining when the user taps on the desk is preferably achieved using the microphone 16. Another way to detect tapping is to use a touch screen, which can provide dragging information as well as extra location data.

Projection from above provides similar capabilities to a large flat display screen, but it has the key advantage that computer-generated images 21 can be superimposed onto paper documents. This is necessary for creating merged paper and electronic documents, and for providing feedback when making selections 22, 28, 31 (see below) on paper. Overhead projection, however, does produce problems, such as shadows: these are hardly noticed when the projector is mounted above a horizontal desk 2, but special measures must be taken to avoid shadow problems on a nearly vertical surface, if this is used as the work surface (like a drawing board).

The brightness of the room may affect the clarity of the projected display. This is not a problem with normal fluorescent lights, but a bright desk lamp or direct sunlight should be avoided. The area onto which display 21 is projected should preferably be white.

In the implementations described herein the image output device may be any device that provides an image in the work surface: e.g. a CRT display conveyed to the surface 2 by means of mirror elements above or below the surface 2; or a flat panel LCD display integral with the desk and disposed either at the surface or below it. Any number of displays may be used.

Document images are captured through an overhead video camera 6, but a difficulty with standard video cameras is their low resolution (compared to scanners), which may not permit good quality copying from a document on the work surface. Several solutions are possible.

One technique is to use a very high resolution camera 6 for the system.

Another solution is to use multiple cameras. At least one camera 6 is set up with a wide field of view that encompasses the substantially the entire work surface 2. At least one other camera (hereafter-"subsidiary camera";not shown) is mounted adjacent the main camera 6 and zoomed in to cover a small part of the desk surface 2 (within or outside the display area 21) at high resolution (e.g. about 200 spots/inch; 8 spots/mm). Multiple fixed subsidiary cameras may be used to cover the whole area at high resolution, or fewer movable subsidiary cameras could be used. The video signal from each camera is processed via a respective channel of the image processing board 102, by means of suitable multiplexing techniques which are well known in the art. When such a subsidiary camera with a relatively small field of view is used, a light area (e.g. a white "window" or other visual indication such as a black rectangular outline) is projected onto the surface 2 so as to coincide with the field of view of the high resolution subsidiary camera(s) and indicate to the user exactly what part of the work surface is within that field of view (i.e., the active area). The user can therefore place source documents 4 within this high resolution "window" to enable text or image information to be scanned in by the system at high resolution. Since it has been found so far that only small parts of a document at a time need be used, and sliding a piece of paper into the camera's "window" is so easy, the use of multiple fixed cameras to cover the whole desk appears unnecessary.

A further possible technique solves this problem by storing information about the positions in the source document 4 of the part(s) to be copied, by means of image recognition techniques and the use of document descriptors. The document is then put through a (desktop) scanner 206 (or is pre-scanned before image manipulation takes place), preferably a high resolution (e.g. 24 dots/mm; 600 dots/inch) scanning machine, and this position information is used to determine what parts of the scanned image to use in the eventual copy. With this technique, the user interacts with the documents at the lower (camera) resolution, but the finished product is constructed from higher (scanner) resolution images. Prescanning is inconvenient for many interactive applications, however, so it is preferred to use one of the abovementioned alternative methods.

The image produced from a video camera 6 and frame grabber (not shown) on the board 102 is greyscale (typically eight bits/pixel). This grey-scale image must be thresholded, or converted to a one bit/pixel black and white image, before it can be used for character recognition or any of the other embodiments which are described herein.

Simple thresholding is not adequate for obtaining an image suitable for character recognition. Another problem can be automatic grey balancing on the camera. This can cause a change in brightness in one part of the image to affect the values in all other parts.

Consequently, the system uses an adaptive thresholding algorithm which varies the threshold value across the image according to its background value at each pixel. The present system produces results in a single pass which are nearly as good as systems requiring multiple passes through the image, by calculating the threshold value at each point from an estimate of the background illumination based on a moving average of local (within about 1/8th the width of the image) pixel intensities. This method is fast and can be combined with a scaling operation if necessary.

Finally, when dealing with text, the thresholded image is skew-corrected and recognised by an OCR server (in this case, Xerox Image System's ScanWorkX). If the resolution is high enough relative to the text size, then it returns the associated ASCII string. For accuracy, it is important to provide both quick feedback to the user (by displaying immediately the number or character which the system "thinks" it has recognised), and a simple way for the user to correct unrecognised characters.

To support interaction, projected feedback 24, 26 (see Fig.6) to the user, and selective grabbing of images through the camera 6, the system must map coordinates in the projected display 21 to coordinates in the frame grabber of image processor board 102. This calibration may be difficult because the projected display 21 is not a perfect rectangle (there are optical distortions such as "keystoning"), the camera 6 and/or tablet may be rotated relative to the projected display 21, and it may be necessary for the camera 6 to view the projected display from an angle. Also, vibrations caused by, e.g., air conditioners or slamming doors cause movements which disrupt the calibration, as do any adjustments to the equipment.

In the case where stylus input is used to indicate position on a tablet, the system first maps absolute positions on the digitising tablet to positions on the display 21 in order to provide feedback. Second, positions on the display 21 are mapped to corresponding positions in the frame grabber in order to support grabbing of selected areas 22, 28, 31 (see below) on the desk. Obtaining the data to calibrate the pointing device (stylus + tablet; touchscreen) to the display 21 is relatively straightforward: a series of points are displayed and the user is prompted to touch them with a pointer 32 (see Fig. 8).

In the case where position indication by finger tip location is used, obtaining data for calibrating the video camera 6 to the display 21 is not as simple.

Figure 4 shows an approach which improves on prior techniques: this is to project an object that can be located by the image processing system, allowing the system to self-calibrate without any assistance from the user. The present system projects a thick "plus" sign (+), and uses image morphology (see D. Bloomberg & P. Maragos, "Image Algebra and Morphological Image Processing", *SPIE Conference Procs,* San Diego, CA, July 1990) to pinpoint the centre of the mark in the frame grabber coordinate space.

For accurate mapping, preferably a four point calibration system (Fig. 4) is used, which compensates for rotation and keystoning. The mapping is given by the equations

$$x' = c_1x + c_2y + c_3xy + c_4 \quad (1)$$

$$y' = c_5x + c_6y + c_7xy + c_8 \quad (2)$$

where (x,y) are coordinates in the projected display, and (x',y') are coordinates in the frame grabber.

With four point pairs, the set of simultaneous linear equations can be quickly solved by Gaussian Elimination. Then, a fifth plus mark (+) is projected and its location is checked to make sure it is close enough to the position produced by the above mapping. The result is accurate to within one or two display pixels, allowing the user to select areas 22, 28, 31 on the desk 2 and rely on the displayed feedback 24 to precisely indicate what will appear in the grabbed image.

Unlike a traditional workstation, user interfaces on the present system must take account of handedness. If feedback 24 (see, e.g., Fig. 6) is projected to the lower left of the pointer 32 (finger, stylus), for example, then a right-handed person has no trouble seeing it, but a left-handed person does have trouble because it gets projected on the hand. Not only is feedback affected, but also the general layout of applications, and left-handed users are inconvenienced because it requires them to reach their arm farther than right-handed subjects, and their arms hide the paper 4 they are reading. The system's video camera 6 can see the user's hands, so it preferably recognises automatically which hand the user is pointing with, and then uses this information in implementing the interface during the following work session. A pop-up menu, for example, is preferably projected to the left of the pointer for a right-handed person, and to the right of the pointer for a left-handed person.

When pointing at paper with the present system, the camera 6 must be able to "see" the paper 4, and this means that fingers and other pointing devices 32 must be out of the way. However, new users do not seem to have much difficulty learning how to interact with the system in a way that keeps selections (22,28,31) visible. When sweeping out a rectangle 24 (for example in the Fig, 6 embodiment described below) there are four ways of doing this (see Fig. 5). If right-handed people use method ①, or if left-handed people use method ②, they obscure the selection. But users do not seem to repeat the mistake: in general, the system cannot see a selection (22, 28, 31) unless the user can see it too, and that seems easy for people to learn.

Selection feedback 24 can also play an important role in preventing obscuration. In the implementation discussed herein, the projected selection rectangle 24 floats slightly ahead of the pointer, so it is easy to avoid placing the pointer inside.

## Selecting parts of a document to be copied

In Figs 6(a)-(f) a basic user interface technique made possible by the copying system of the present invention - the selection of parts of a paper document 4 directly on the paper itself while the system reads the image selected - is illustrated in successive scenes, viewed from above the desk surface 2. The user 18 is creating a new document (generally designated 20) within the projected display 21, and here the source document 4 is a book page. The user selects a figure 22 on the book page 4 by first touching his two index fingers together at the top right hand corner of the figure 22: the system recognises this as a gesture for starting a selection. As the user then moves his left hand index finger to the bottom left hand corner of the figure 22 (motion ③ in Fig.5), the system processor recognises the movement and causes the projector to display, as feedback to the user, a selection block 24 (here a rectangular outline; alternatively a grey rectangle) which increases in size until this movement ceases (Fig.6(a)). The user can see exactly what is encompassed by the selection block 24, and when this is as desired, the user taps on the desk to confirm the selection (the tap being interpreted by the processor as such confirmation). The processor obtains via the camera 6 information indicating the positions of the boundaries of the selection block 24 relative to the original document 4 and therefore the extent of part of the document which has been selected.

Next, the user puts his pointed finger on the page 4 in the selection block 24 and "drags" a projected image of the selection block 24 by moving his finger across the display 21 (the selection block 24 is displayed by the projector 8 for feedback and moves to follow the position of the moving finger tip), positions it in the appropriate location in the document 20, and taps on the desk with a finger to confirm the positioning (FIg. 6(b)). The figure 22 is captured by the camera 6, thresholded by the processor and the stored form of the document 20 edited accordingly; and the result is that the projected display is modified so that the figure 22 is "pasted" into the document 20 at the desired location (Fig. 6(c)); here, the dimensions of the pasted-in figure 22 are adapted by the processor 10 to the available height and width of the text area of the new document 20.

It is then possible for the user to add a legend to the pasted-in figure 22 by typing out the text thereof on a conventional keyboard (not shown) linked to the processor 10. The stored electronic form of the document 20 is edited by the processor, and the projected display 21 is simultaneously modified to show the legend 26 (Fig. 6(d)) as it is typed in.

Next, the user selects a portion 28 of text from the book page 4 which is to be pasted in below the figure in document 20 (Fig. 6(e)). This is done in exactly the same way as selecting the figure 22 in Fig. 6(a), except that the user starts with both index fingers at the top left hand corner of the text portion 28 and moves his right hand index finger to the bottom right hand corner of the text portion 28 (motion ④ in Fig. 5). The text selection 28 is positioned in the document by tapping on the surface 2, as before. The difference in this case is that optical character recognition (OCR) is performed on the selected text portion 28 captured by the camera 6. The font of the text 28 is automatically converted into that of the rest of the electronic document 20, and is reformatted so as to fit into the text flow of the document 20 being made up. (Alternatively the text 28 could be treated in the same way as the figure 22 in Figs 6(a) to (c), by selecting from top right to bottom left: i.e. motion ③ in Fig.5.) The stored electronic form of the document 20 is updated accordingly by the processor 10 and the projector 8 automatically displays the modified document 20 (Fig.6(f)).

Once completed, the document 20 can be printed out by conveying to the processor 10 a command to send the electronic version of the document 20 to the printer 14. This command may be entered via the keyboard or conventional mouse operation, but is preferably designated by the user selecting an appropriate item from a pull-down menu (not shown) accessable, e.g. by finger pointing, in the display area 21 on the surface 2.

In an alternative implementation, the work surface 2 may incorporate a touch pad or other position sensing device; and the user can use an appropriate stylus to indicate corners of a rectangular selection block designating a part to be copied, such as by starting in one corner and moving the stylus to the opposite corner. (In this case, in order for the stylus and position-sensing tablet to operate, the document 4 must only be a single-sheet thick.) It is also possible to select non-rectangular regions by tracing a "lasso" around the part of the paper document to be copied. Another possibility is for the user to simply point at a region and the system can use image morphology techniques to determine the scope of the selection. One tap on the work surface could select only the smallest discernable element pointed to (e.g. a letter or word). Another tap in the same location would expand the selection to include the sentence containing that letter or word, a further tap causing selection of the paragraph containing that sentence, or larger visual unit, and so on. With all of these selection techniques, precise feedback is projected so the user can see exactly what is selected, and can therefore adjust the selection if it is not exactly what the user wants (e.g. selecting a "don't care" location - beyond the boundaries of the document 20 - whereupon the projected selection is cancelled; and then re-selecting from the source document).

Figure 7 illustrates, by means of a flow chart of appropriate software running in the signal processing system 10 of Fig. 1, the steps involved in carrying out the procedure sequentially illustrated in Fig. 6.

## Copying onto marked document

Another basic technique made possible by the present invention is the copying onto a previously marked document in novel ways. For example, a form can be filled in with data from parts of one or more other documents.

This technique is illustrated in Figs 8(a) to (f), which show successive scenes, viewed from above the surface 2. The technique is similar to that used in the embodiment illustrated in Fig. 6, except that the document 20 consists of the information to be added to a marked document 30 (in this case a form) placed on the work surface 2. Operations are performed to indicate how the form 30 should be completed, producing a projected image showing the additional marks that are to be made on the document.

As illustrated in Fig.8(a), the source document 4 comprises a receipt positioned within the camera's field of view. The user selects the numerical total 31 indicated on the receipt using the above-mentioned stylus and position-sensing tablet method (but any of the above-mentioned image-selection techniques could be used). An image of the selected number, captured by the camera 6, is projected back onto the display 21 at the position of the point of the stylus 32. As the projected image of selected number is dragged to the appropriate box 34 of the form 30, in a similar way to the moving selection block 24 in the Fig.6 embodiment, the motion of the number is shown in the display by the projector 8 (Fig.8 (b)). The number is recognised by the processor 10 using OCR and dropped in the box 34 by releasing a button on the stylus, or by the user tapping on the desk with his free hand.

Figure 8(c) illustrates an operation performed in the case where the appropriate data is not present in the source document 4: the user writes a date in the appropriate box 36 of the form by hand. The movement of the point of the stylus 32 is tracked as the user writes, and an image is simultaneously projected down onto the form 30 showing the ink which would have been left on the form if the stylus were a pen. The system recognises the user's characters as they are written, converts the projected "ink" characters into the same font as the other numbers on the form 30 and modifies the projected characters to make them appear in that font. Once one entry (e.g. a date in numerical form) has been made in this way, it can be copied to other boxes in in the same or neighbouring column using the above-described drag and drop process, or even copied by making ditto signs by hand in the appropriate places.

Once the relevant numbers have been "entered" in the form 30, an operation can be performed on a group of numbers. In Fig.8(d) a column 38 containing a set of "entered" numbers is selected using the stylus 32. Next, the user places on the form 30 a small piece of paper having a button 39 printed on it and designated "SUM", with its arrow pointing at the interior of a box 40 on the form 30 in which a total is to be entered. When the user "presses" the paper button by tapping a finger on the piece of paper as shown in Fig. 8(e), the sum of the numbers in the selected column 38 is projected into the box 40. In doing this, the system (1) recognises the function of the button (e.g. by means of morph glyphs present in the drop shadow 42 of the button 39), (2) recognises the tapping of the button so as to be aware of where and when to perform the summing operation, (3) carries out the summing operation, and (4) projects the resulting numerical sum into the box 40.

When all the necessary entries have been made in the form 30, the latter can be fed through a printer 14 in order to make the projected marks permanent. [For this purpose it may be convenient, in the case where the main printer 14 is remote from the desk 2, to have an additional compact ink jet printer (not shown) on the desk surface 2, enabling the printing of the additional marks on the form and, if necessary, the signing of the form by a user, to be carried out immediately.] The processor 10, which stores the relative positions of all the projected characters and numbers with respect to the features of the form 30, causes the corresponding ink marks to be made in the appropriate locations (row/column/box) in the form during the printing operation.

Figure 9 illustrates, by means of a flow chart of appropriate software running in the signal processing system 10 of Fig. 1, the steps involved in carrying out the procedure sequentially illustrated in Fig. 8.

This technique is preferably extended by performing OCR on a selection from a source document 4 and then projecting the recognised numbers or characters in image to fill in a form or other pre-marked document. In general, text selections are discriminated from graphics selections and OCR performed, where necessary. Optionally, a user may select a location on a form 30 using one of the above-described techniques, and then type characters into the projected image with a conventional keyboard (not shown) linked to the processor 10.

The above-mentioned paper buttons may also be used to extend this technique: various buttons displaying appropriate recognisable codes (as mentioned above) are used to perform commonly executed operations, such as currency conversions, averaging etc.

This technique further includes recognising other tools and modifying the projected image appropriately. When a stylus is used as a marking tool, and moved across the surface (e.g. for handwriting), a mark is produced in the image. If the resulting marks meet some criterion, recognition is performed and the marks replaced by the appropriate characters or numbers. Also, an eraser is recognised similarly and, in addition to its physical erasure of marks, the projected image is modified appropriately.

If a paper form is recognised by the system, then it can assist the user with prompts as to how to fill it out, and it can perform calculations (e.g. adding a column of numbers) that are specified on the form. In general, the system can augment any recognisable paper form with features now available only with electronic forms.

## Scaling and positioning document parts in projected image

Another user interface technique made possible by the present invention is scaling or positioning parts of a document before copying. In this disclosure, the term "arrange" is used generally to include an operation that scales (re-sizes) a document part, an operation that positions a document part, or an operation that both scales and positions a document part. Position of a document part also includes orientation. The basic technique for arranging a document part is to perform arranging operations in the projected image. In other words, a user can provide signals through the camera requesting operations so that the document part appears at a different scale or different position in the projected image. In effect, the user changes the projected image until it shows a desired final, output document with the indicated document part scaled and positioned as desired. The output document can then be printed.

To perform scaling, the user can indicate a different spacing between the opposite corners of a document part bounded by a selection rectangle, such as by moving the fingertips together or apart. Scale of the selected part in the projected image can then be changed in proportion to the change in spacing.

To perform selection and positioning, the user proceeds as in the embodiments of Figs 6 and 8, except as mentioned below.

Figure 10 illustrates this technique: successive scenes, viewed from above the surface 2, show how the invention is employed by a user in producing a sketch.

Initially, the user sketches out a scene using an ordinary pencil and which includes a tree 46 (Fig. 10(a)). Next, the user, desiring to create a row of trees, selects the image of the tree 46 by sweeping out a selection block 24 in the same manner as described with reference to the Fig.6 embodiment (Fig. 10(b)). The user then moves a copy of the selection block 24 as in the Fig.6 embodiment, except that two fingers (or finger and thumb) are used which, when the copied block 24 is in the desired position, are used to reduces the size of the block to the desired scale (Fig. 10(c)). The user taps on the desk surface 2 to "drop" the reduced tree 48 in position, and the projector displays the new tree 48 spaced apart from the original 46. This process is repeated three more times with the user's fingers progressively closer together, to produce a row of trees (46-54) in perspective along the lane 56 (Fig. 10(d)).

Next, the user begins to draw some slates 58 on the roof 60 of the house 62 (Fig. 10(e)). In order to save time the user places a paper button 64 designated "FILL" with its arrow pointing at the roof 60. The code in the drop shadow 66 of the button 64 is captured by the camera 6 and the command is recognised by the processor 10; similarly, the slate pattern 58 is captured when the user taps on the button 64. The user then moves the button 64 to the empty region of the roof and taps on the button again; and the system executes the button command by replicating the slate pattern 58 to fill the area within the boundaries of the roof 60 (Fig.10(f)). The resulting pattern is displayed by the projector 8.

A further step is illustrated by Figs 10(g) and (h): the user decides to include a window 66 in the roof 60, so some of the slate pattern must be erased. An "eraser" 68 having on it a printed sticker 70 displaying a code, similar to that on the above-mentioned paper buttons (39, 64), by means of which the system recognises the implement as an eraser. As the user sweeps out an area with the "eraser" 68, the erasing motion is recognised by the system, and the displayed pattern 58 is modified so as to omit the slate pattern from that area Fig. 10(g)). The user then draws in the dormer window 66 by hand Fig. 10(h)).

The result of these operations is a merged physical and electronically projected sketch similar to the combined form described in the Fig. 8 embodiment. Again, in order to make the projected marks (e.g. trees 48-54 and the slate pattern 58) permanent, the sheet 30 containing the sketch would be passed through a printer connected to the processor 10. The processor, which stores the relative positions of all the projected images with respect to either the features of the sketch or the boundaries of the sheet 30, causes the corresponding ink marks to be made in the appropriate locations on the sheet 30 during the printing operation.

Figure 11 illustrates, by means of a flow chart of appropriate software running in the signal processing system 10 of Fig. 1, the steps involved in carrying out the procedure sequentially illustrated in Fig. 10.

Another possibility for rotating and positioning parts of a source document 4 in the projected image is to move a paper original, for example containing image elements to be included in the final sketch, into the desired position within the projected document 20, and to select the image element of interest (e.g. by tapping on the surface 2) to be "pasted down" in place. This natural interaction technique allows any printed or hand-drawn image to be used as a sort of rubber stamp and advantageously allows the user to try an image element in various positions without having to produce a new complete sketch each time.

Random copying of document parts

The above basic techniques are especially powerful when considered together with the possibility of randomly copying from a set of input documents 4 to produce output documents. This user interface technique is based on obtaining information indicating the relationship between the input documents 4 and the output documents.

One way to obtain this information is to operate on the documents in sequence. In other words, the output documents include parts from input documents 4 in sequence, so that the input documents can be copied in order into the output documents. This can be inconvenient, however, such as when one of the input documents 4 include different parts that are copied into several of the output documents.

Another way to obtain this information without operating on the documents 4 in sequence is to use document recognition techniques. This is similar to the previous way except that it is unnecessary to provide identifiers on the documents. Instead, a document characteristic that can be detected at low resolution, such as line length pattern, can be used to obtain identifiers that are very likely to be unique for each document. Document classification techniques are described briefly in EP-A-495 622.

"Select and Paste" (or "Copy and Paste")

Although selecting text or images from one document, and "pasting" the selection into a second document is now a standard feature when manipulating electronic documents, the same operation is awkward to perform with real paper, requiring a photocopier, scissors, and some glue or tape. The system of the present invention, however, makes it possible to select and paste paper documents 4 in the same way that we select and paste electronic documents. In this implementation, a sketch 80 on paper 4 can be electronically selected by sweeping out an area 24 of the paper (e.g. with a stylus 32) in a similar manner to that described above. When the stylus 32 is raised, the system snaps a picture of the selection 80, and the projected rectangle 24 is replaced by a thesholded electronic copy of the area. This copy can then be moved about and copied to other parts 82 of the paper 4 as described in the aforementioned application. Sliding this electronically projected copy over the drawing to place it somewhere else is very similar to sliding a paper copy (see Fig. 12).

In Fig. 12, the user has selected the sketch 80 of the window on the sheet 4, and has made two copies of it (82). Now he has moved and is about to "paste down" a copy 86 of the flower 84 that he drew.

Figure 13 illustrates, by means of a flow chart of appropriate software running in the signal processing system 10 of Fig. 1, the steps involved in carrying out the procedure illustrated in Fig. 12.

User testing revealed another way of using this tool which is also very powerful. Instead of constructing a mixed paper and projected drawing, it has been found that a user can construct a purely projected drawing from selected portions taken from any number of their paper sketches. The user can sketch a figure on paper, move it to the desired location in the projected drawing, then select it using the above-mentioned techniques so that it remains "pasted down" in that location even after moving the paper away. The effect is like that of dry-transfer lettering or "rubber stamping", but in this case from any piece of paper onto a projected drawing. This interaction technique is quite different from the standard "copy and paste" found on most workstations and takes advantage of unique qualities of the present invention: using both hands for manipulating and pointing as well as the superimposition of paper and electronic objects.

Multi-user systems

People often use documents when working together, and they often need simultaneously to see and modify these documents. Physical paper is normally constrained in that it cannot be written on, pointed to, or otherwise manipulated by two people simultaneously who are, for example, located on separate continents; but this constraint can also be addressed by the present invention.

Shared editing of documents has been disclosed in, e.g. J.S. Olsen et al., "Concurrent editing: the group's interface" in D. Daiper *et al.* (eds) *Human Computer Interaction - Interact '90,* pp. 835-840, Elsevier, Amsterdam). Most of this work has concentrated on screen-based documents, but the multi-user implementation of the present invention makes it possible to share real *paper* documents. It allows users in (at least) two separate locations to "share" their physical desks, by enabling both users to see and to edit each other's paper documents 4.

Referring to Fig. 14, in the case of a two-user system, the two processors 10 are connected by means of a conventional communications link. Each installation continuously grabs images 88 from its local desk 2 and projects thresholded images 90 from the remote desk 2'. The result is that both users see what is on both desks. When a paper document 4 is placed on a desk 2 of user A, it is projected onto desk 2' of user B and vice versa. The projections are digitally scaled and positioned to provide What You See Is What I See (WYSIWIS), and both users can draw (using a real pen 92, 92') on either paper documents 4 or on virtual documents. On both sides, the remote user B will see the new drawing projected in the corresponding place. Hand motions are also transmitted over the communications link and displayed, so if a user points to a certain place on a document 4 the other user can see this. (The partner's hands block the view of what is underneath them, just as with an ordinary desk, so this must be dealt with through social protocols and speech: not pictured in Fig. 12 is an audio link through telephones or speakerphones which is preferably provided to facilitate this. Another useful and even more preferable addition is a face-to-face audio-visual link.)

In Fig. 14, the local user A is drawing a "X" 88 on a paper sheet 4 in ink, while the remote user's (B) paper and hand can be seen having just finished drawing a "O" 90.

Figure 15 illustrates, by means of a flow chart of appropriate software running in the signal processing system 10 of Fig 1, the steps involved in carrying out the procedure illustrated in Fig. 14.

**Claims**

**1.** A copying system, comprising:

a work surface (2);

means (8) for displaying images on the work surface;

a camera (6), focussed on the work surface, for generating video signals representing in electronic form image information present within the field of view of the camera;

processing means (10) for recognising one or more manual operations relating to the image information which are executed by a user within the field of view of the camera, and for performing electronic operations, corresponding to said manual operation(s), on the electronic form to produce a modified electronic form;

the displaying means (8) being adapted to display, under the control of the processing means (10), simultaneously with or subsequent to said electronic operation performing step, images defined by said manual operations; and

wherein said images defined by said manual operations include an image of the newly created document.

**2.** The copying system according to claim 1, wherein the modified electronic form includes an electronic version of a newly created document.

**3.** The copying system according to claim 2, further including means (208) for printing a document corresponding to at least part of said modified electronic form.

**4.** A method of generating documents, comprising:

providing a work surface, means for displaying images on the work surface, and a camera focussed on the work surface, said camera generating video signals representing in electronic form image information present within the field of view of the camera;

recognising one or more manual operations relating to the image information which are executed by a user within the field of view of the camera;

performing electronic operations, corresponding to said manual operation(s), on the electronic form to produce a modified electronic form;

displaying, simultaneously with or subsequent to said electronic operation performing step, images defined by said manual operation(s); and

wherein said images defined by said manual operations includes an image of the newly created document.

**5.** The method according to claim 4, wherein the modified electronic form includes an electronic version of a newly created document.

**6.** The method according to claim 4 or 5, further including the step of supplying to a printing device said electronic version; and printing out said newly created document.

**7.** The method according to claim 6, wherein said manual operation(s) include selecting a portion of text or image information in a document (4) located within the field of view of the camera.

**8.** A programmable printing apparatus when suitably programmed for carrying out the method of any of claims 4 to 7.

**9.** A copying system, comprising:

a work surface (2);

means (8) for displaying images on the work surface;

a camera (6), focussed on the work surface, for generating video signals representing in electronic form image information present in at least first and second documents within the field of view of the camera;

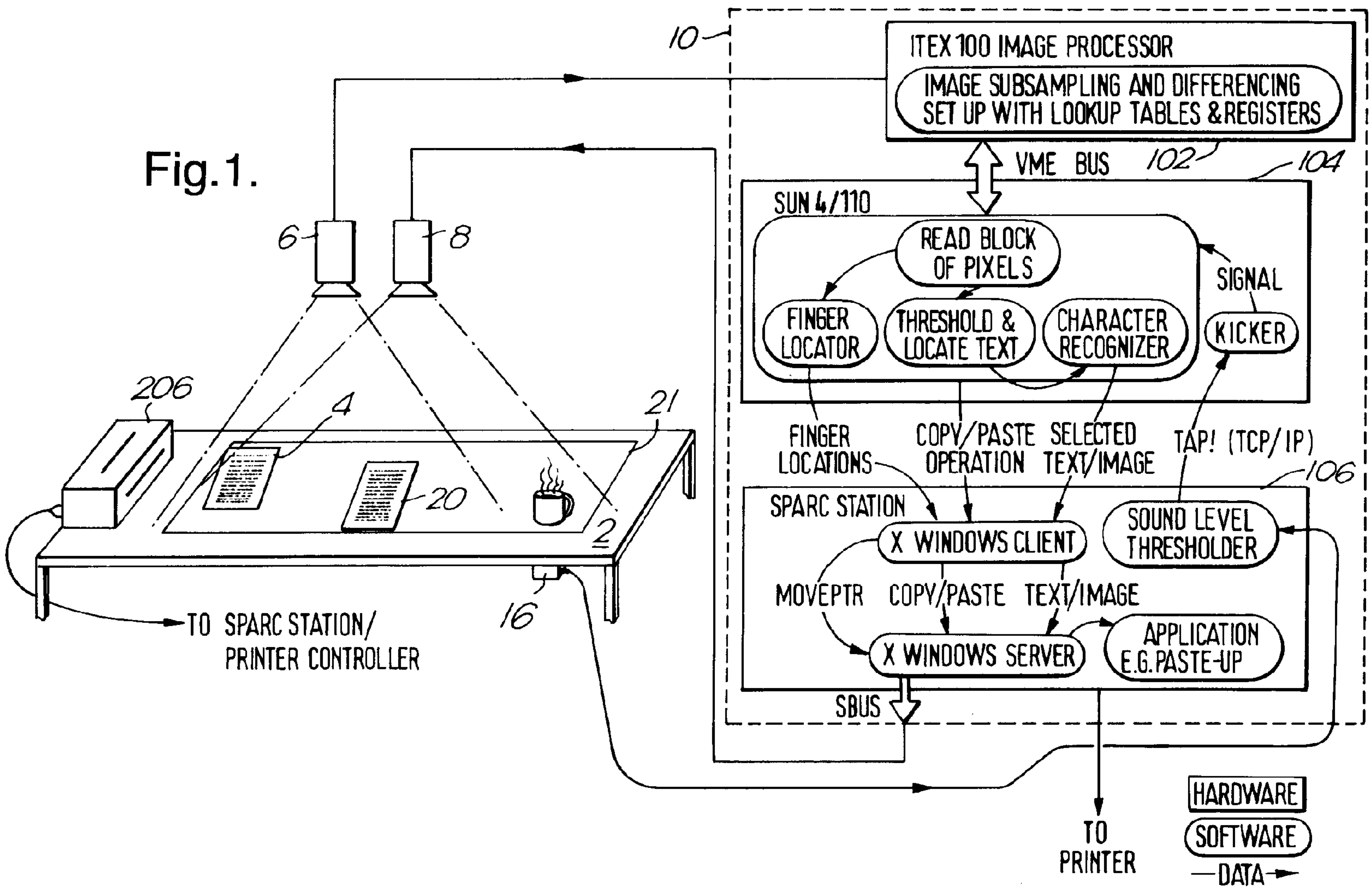
processing means (10) for recognising one or more manual operation(s) which are executed by a user within the field of view of the camera and represent the transfer of image information from the first document to the second document, and for performing electronic operations, corresponding to said manual operation(s), on the electronic form of said second document to produce a modified electronic form;

the displaying means (8) being adapted to display, under the control of the processing means (10), simultaneously with or subsequent to said electronic operation performing step, images defined by said manual operations.

**10.** An interactive image reproduction system, comprising:

a plurality of workstations interconnected by a communications link, each workstation comprising a system according to any of claims 1 to 3, 8 or 9, each workstation being adapted for displaying the video output from the camera (6) of the or each other workstation.

Fig.1.


10
ITEX 100 IMAGE PROCESSOR
IMAGE SUBSAMPLING AND DIFFERENCING SET UP WITH LOOKUP TABLES & REGISTERS
VME BUS
102
104
SUN 4/110
READ BLOCK OF PIXELS
FINGER LOCATOR
THRESHOLD & LOCATE TEXT
CHARACTER RECOGNIZER
SIGNAL
KICKER
FINGER LOCATIONS
COPY/PASTE OPERATION
SELECTED TEXT/IMAGE
TAP! (TCP/IP)
106
SPARC STATION
X WINDOWS CLIENT
SOUND LEVEL THRESHOLDER
MOVEPTR
COPY/PASTE
TEXT/IMAGE
X WINDOWS SERVER
APPLICATION E.G. PASTE-UP
SBUS
TO PRINTER
HARDWARE
SOFTWARE
DATA
6
8
206
4
21
20
2
16
TO SPARC STATION/ PRINTER CONTROLLER

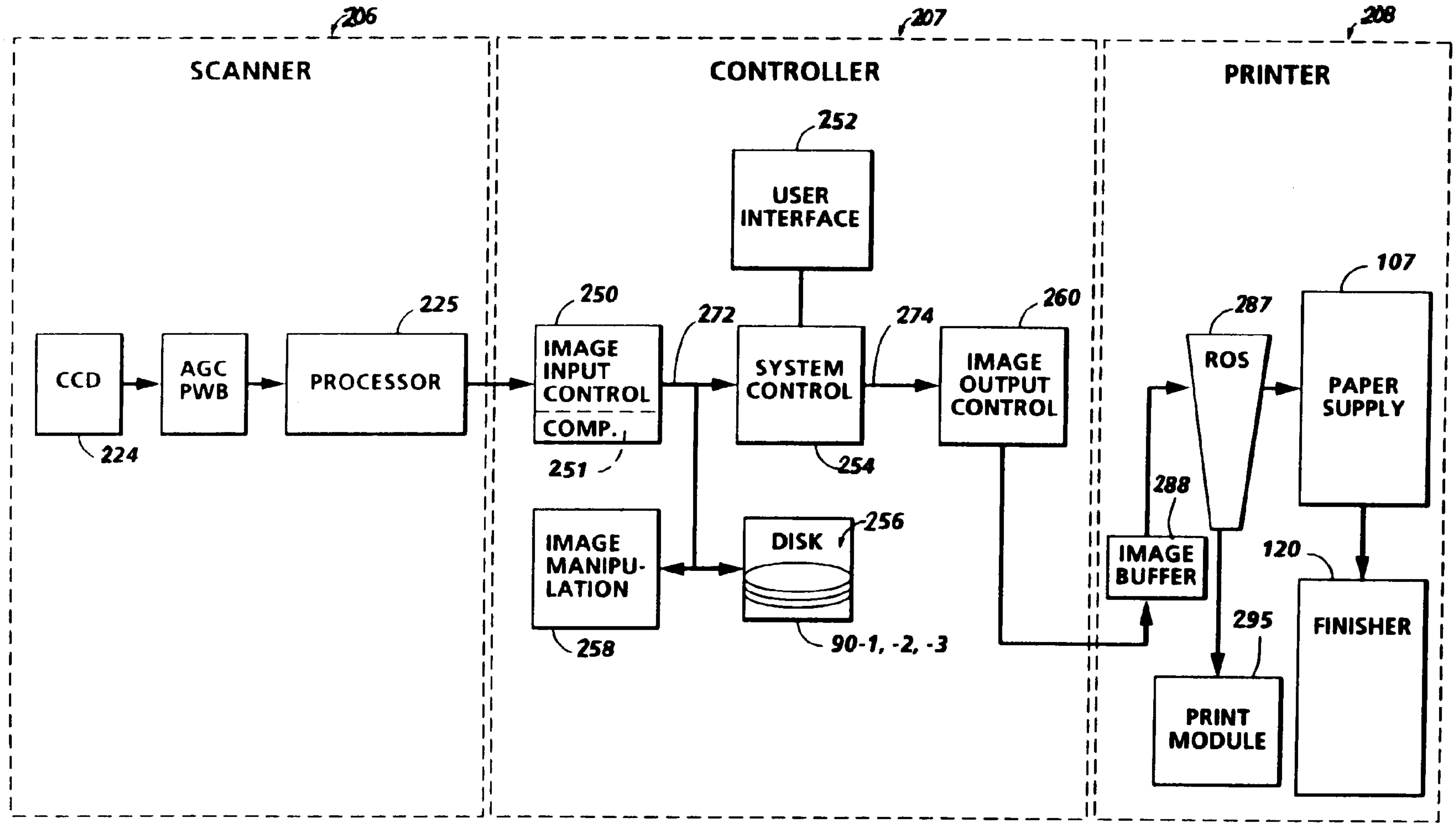

206
207
208
SCANNER
CONTROLLER
PRINTER
252
USER INTERFACE
107
250
272
274
260
287
225
ROS
IMAGE INPUT CONTROL
SYSTEM CONTROL
IMAGE OUTPUT CONTROL
PAPER SUPPLY
CCD
AGC PWB
PROCESSOR
COMP.
224
251
254
288
256
DISK
120
IMAGE MANIPU-LATION
IMAGE BUFFER
FINISHER
295
258
90-1, -2, -3
PRINT MODULE

Fig.2.

Fig.3.

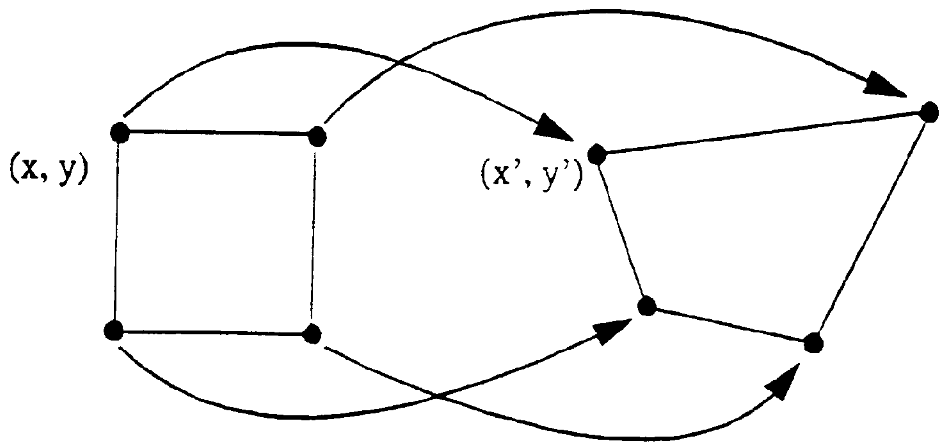



Fig.4.

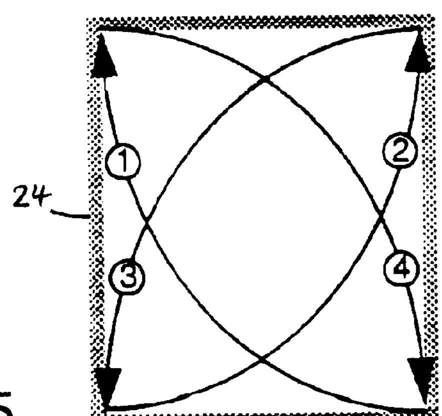



Fig.5.

Fig.6a

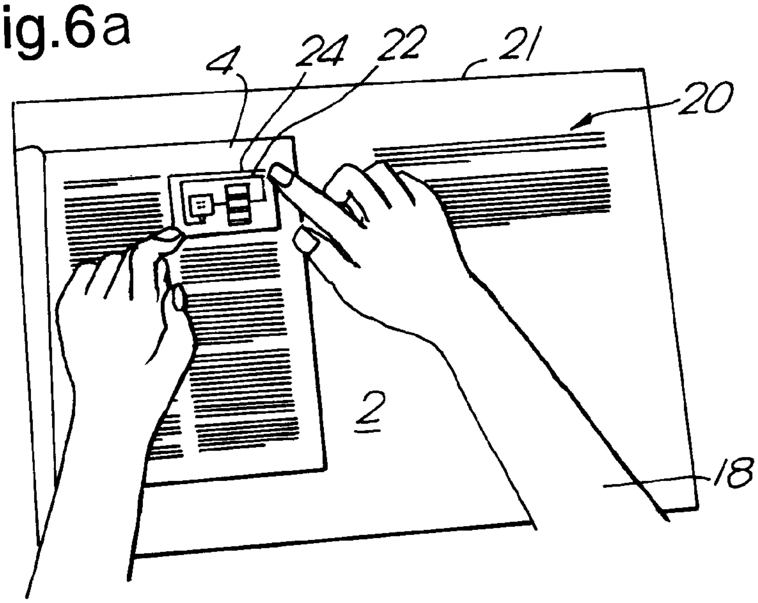
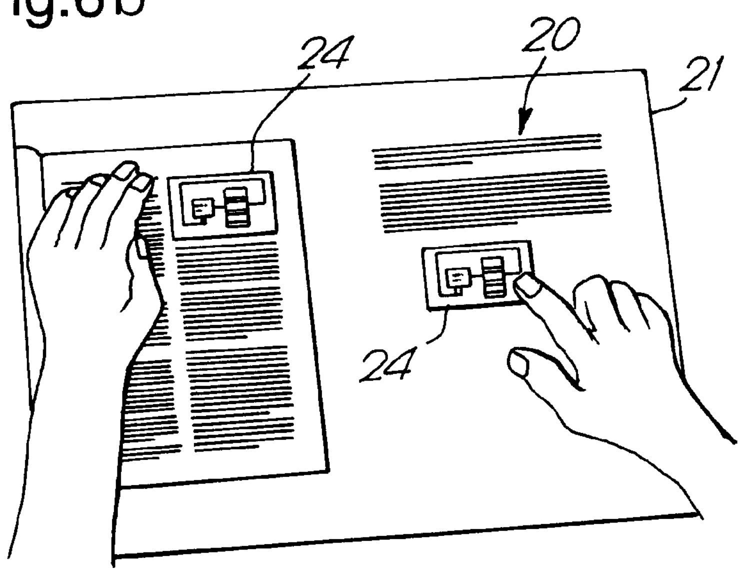



Fig.6b






APLNDC00021444

Fig.6c

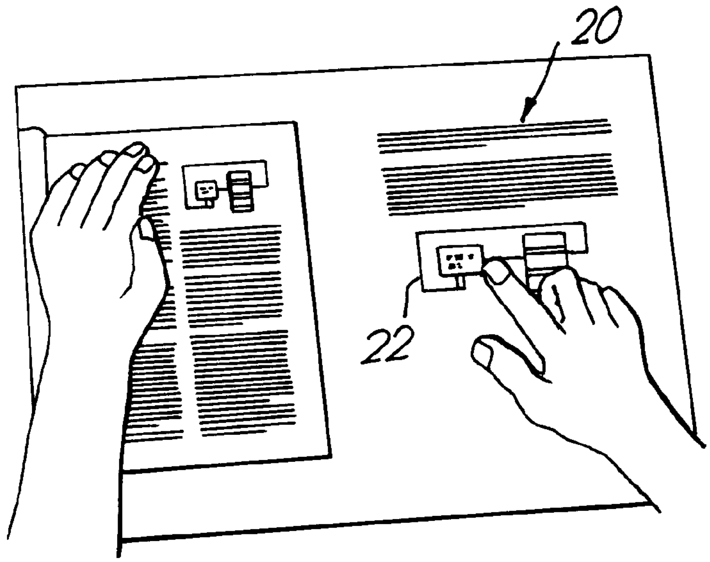



Fig.6d

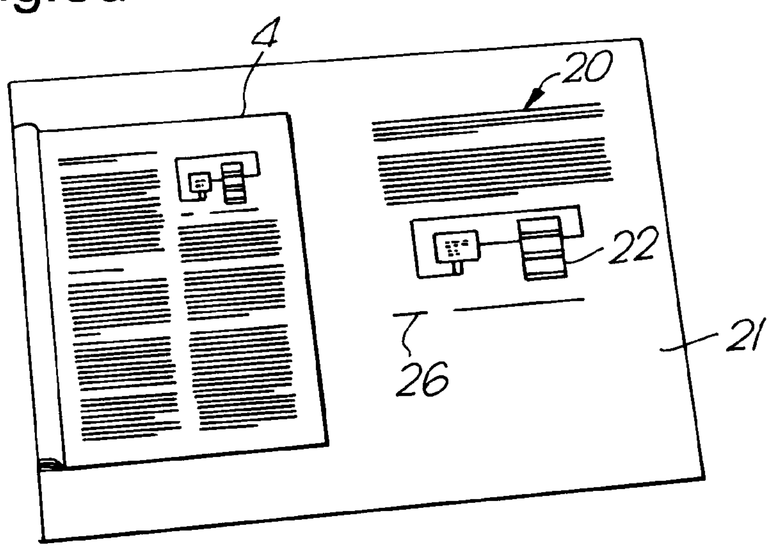


Fig.6e

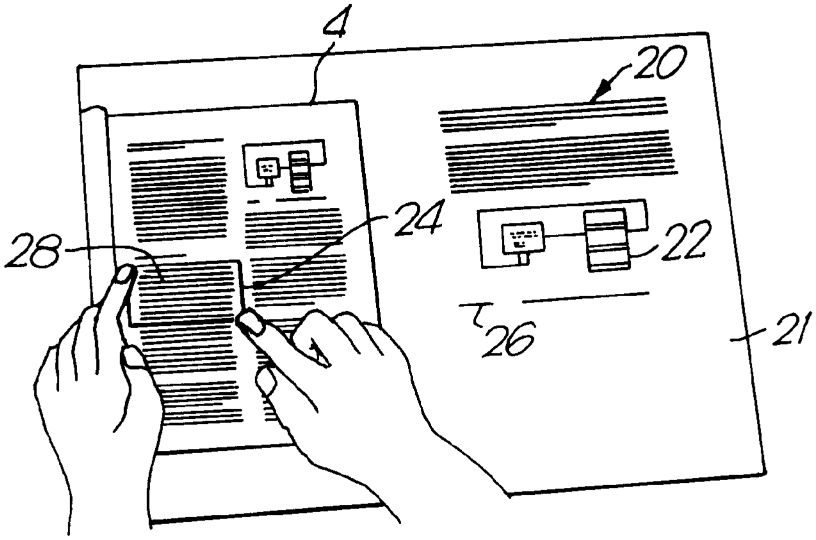
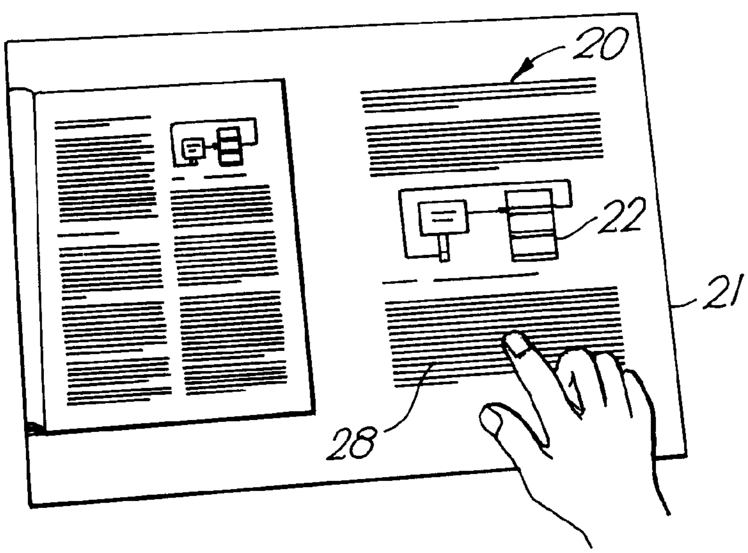



Fig.6f






APLNDC00021446

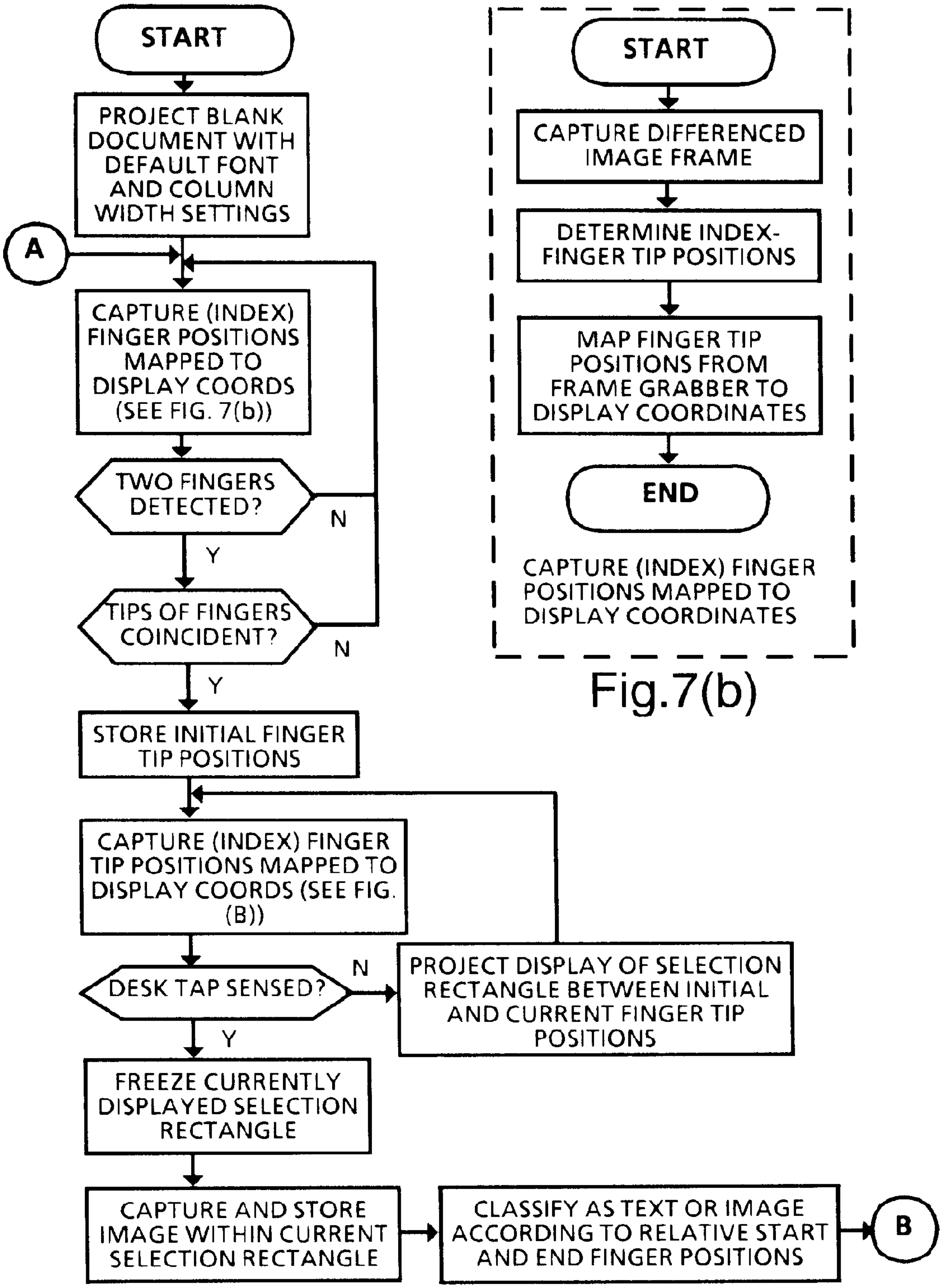



Fig.7(a)

Fig.7(b)



APLNDC00021447

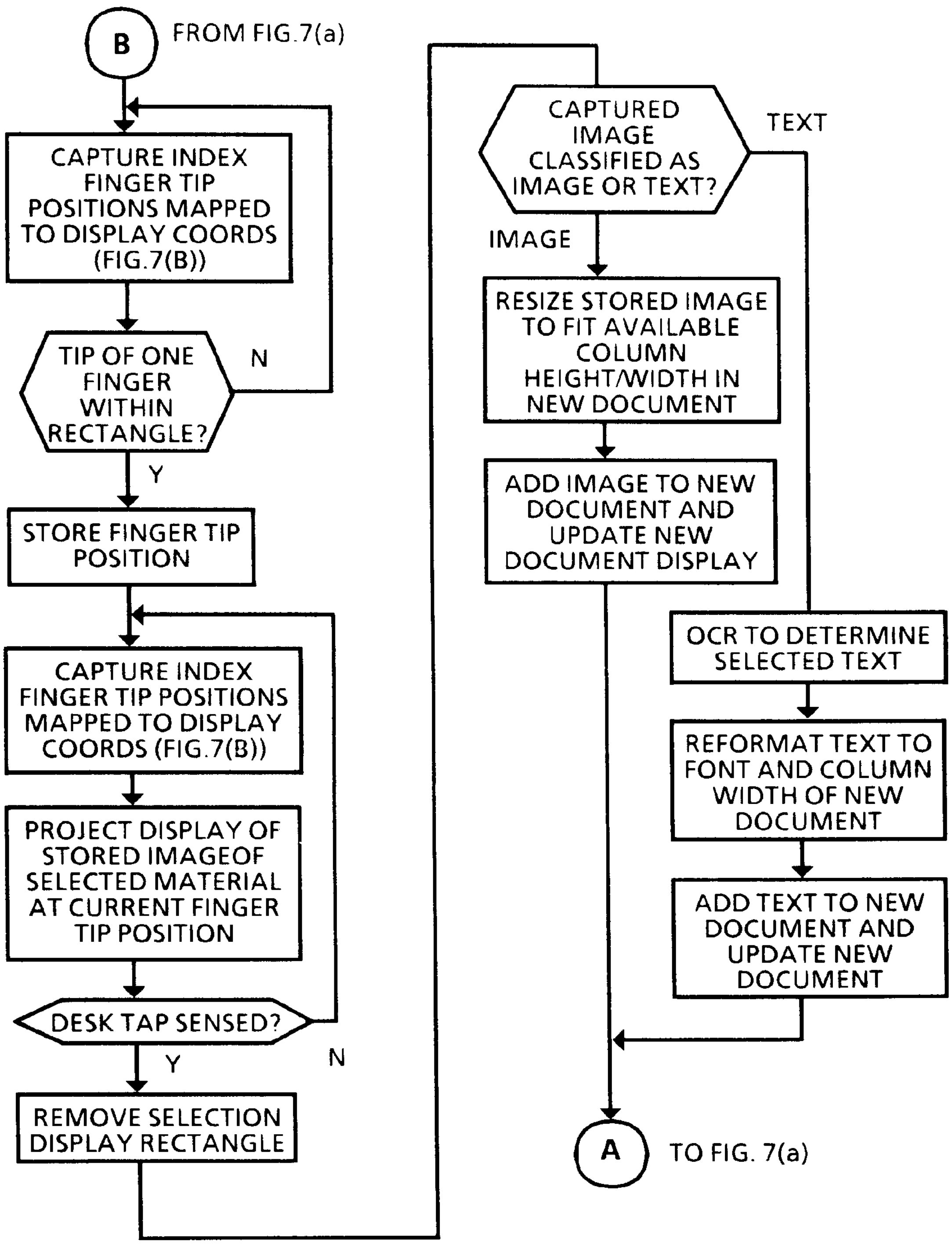

B
FROM FIG.7(a)
CAPTURE INDEX FINGER TIP POSITIONS MAPPED TO DISPLAY COORDS (FIG.7(B))
TIP OF ONE FINGER WITHIN RECTANGLE?
N
Y
STORE FINGER TIP POSITION
CAPTURE INDEX FINGER TIP POSITIONS MAPPED TO DISPLAY COORDS (FIG.7(B))
PROJECT DISPLAY OF STORED IMAGEOF SELECTED MATERIAL AT CURRENT FINGER TIP POSITION
DESK TAP SENSED?
Y
N
REMOVE SELECTION DISPLAY RECTANGLE
CAPTURED IMAGE CLASSIFIED AS IMAGE OR TEXT?
TEXT
IMAGE
RESIZE STORED IMAGE TO FIT AVAILABLE COLUMN HEIGHT/WIDTH IN NEW DOCUMENT
ADD IMAGE TO NEW DOCUMENT AND UPDATE NEW DOCUMENT DISPLAY
OCR TO DETERMINE SELECTED TEXT
REFORMAT TEXT TO FONT AND COLUMN WIDTH OF NEW DOCUMENT
ADD TEXT TO NEW DOCUMENT AND UPDATE NEW DOCUMENT
A
TO FIG. 7(a)

Fig.7(c).

Fig.8a

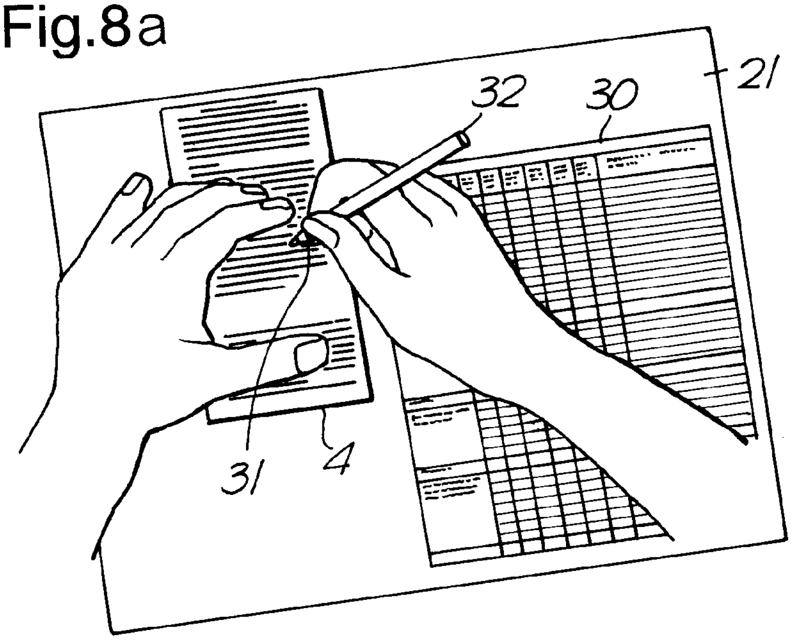



Fig.8b

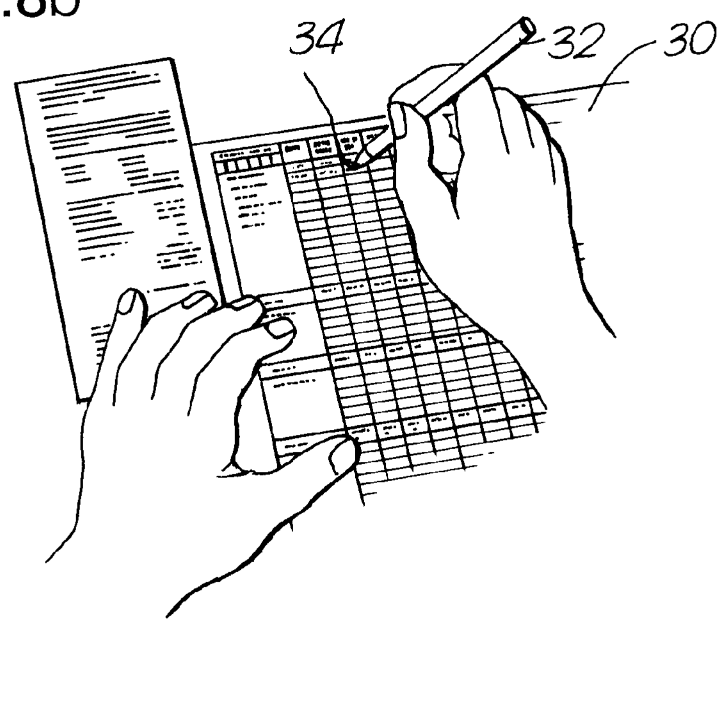


Fig.8c

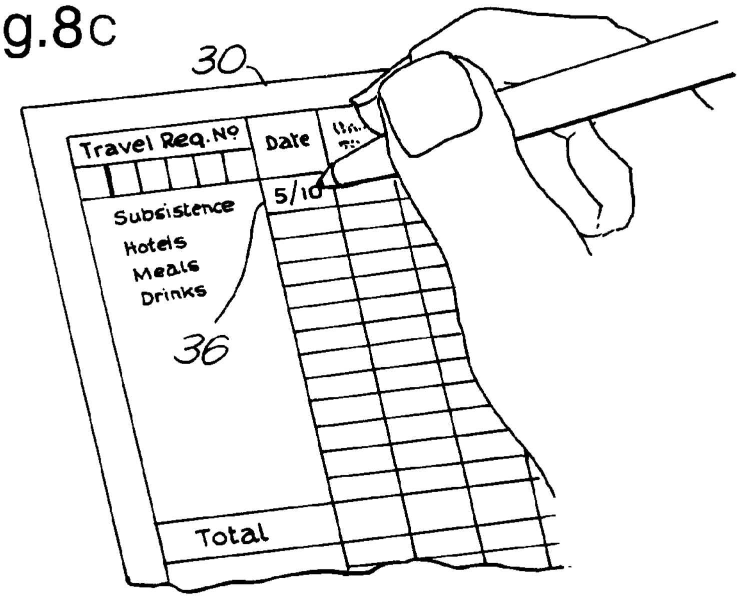
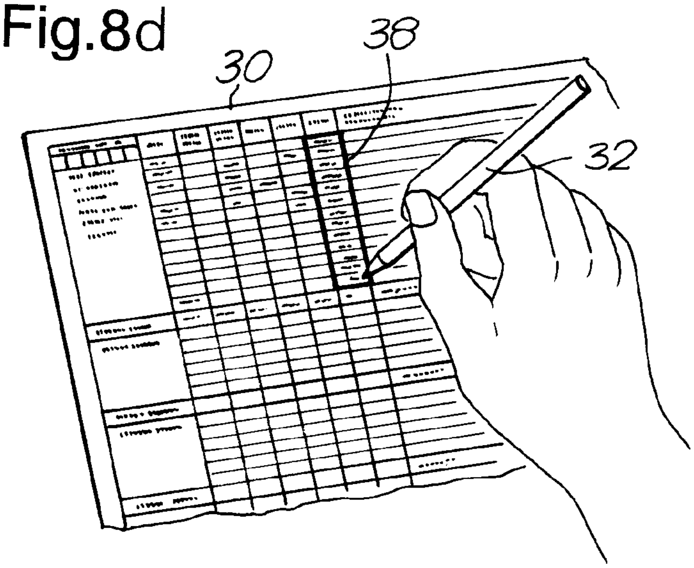


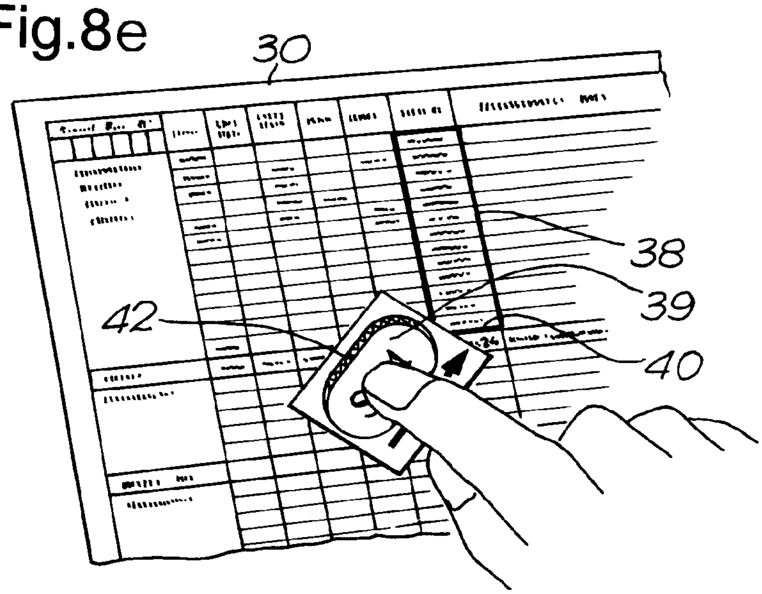

Fig.8d






APLNDC00021450

Fig.8e






APLNDC00021451

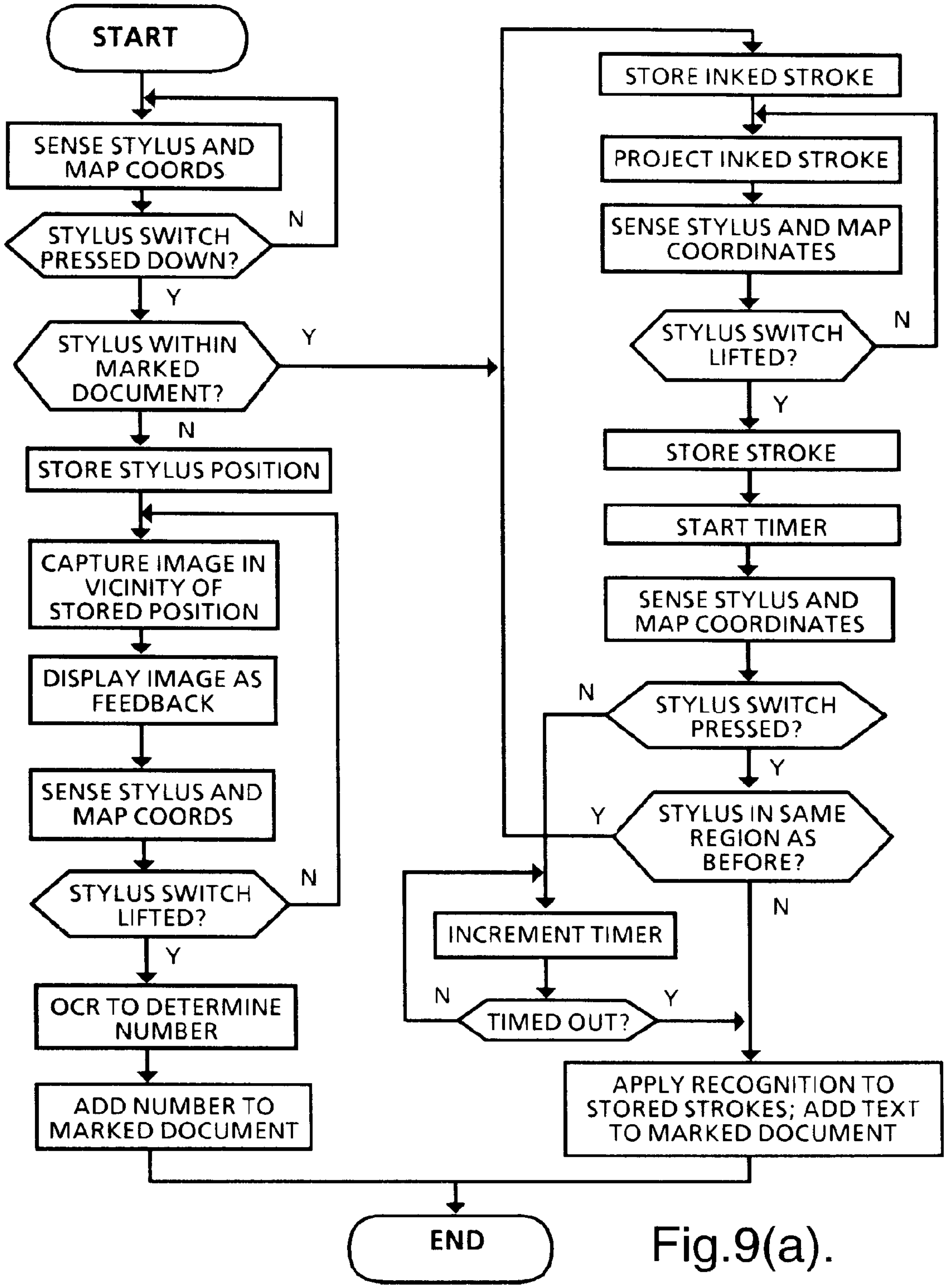

START
SENSE STYLUS AND MAP COORDS
STYLUS SWITCH PRESSED DOWN?
N
Y
STYLUS WITHIN MARKED DOCUMENT?
Y
N
STORE STYLUS POSITION
CAPTURE IMAGE IN VICINITY OF STORED POSITION
DISPLAY IMAGE AS FEEDBACK
SENSE STYLUS AND MAP COORDS
STYLUS SWITCH LIFTED?
N
Y
OCR TO DETERMINE NUMBER
ADD NUMBER TO MARKED DOCUMENT
STORE INKED STROKE
PROJECT INKED STROKE
SENSE STYLUS AND MAP COORDINATES
STYLUS SWITCH LIFTED?
N
Y
STORE STROKE
START TIMER
SENSE STYLUS AND MAP COORDINATES
STYLUS SWITCH PRESSED?
N
Y
STYLUS IN SAME REGION AS BEFORE?
Y
N
INCREMENT TIMER
TIMED OUT?
N
Y
APPLY RECOGNITION TO STORED STROKES; ADD TEXT TO MARKED DOCUMENT
END

Fig.9(a).

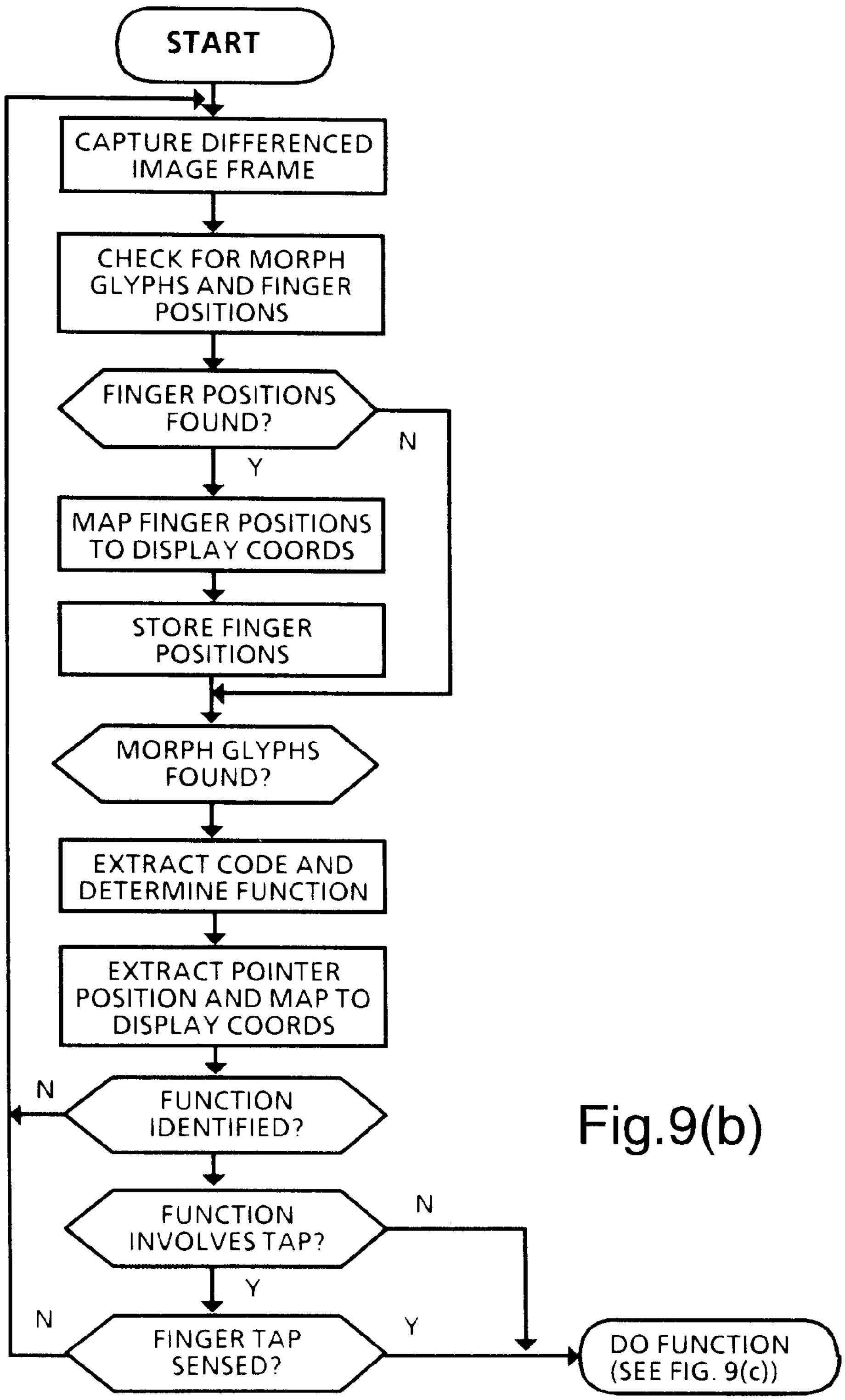

START
CAPTURE DIFFERENCED IMAGE FRAME
CHECK FOR MORPH GLYPHS AND FINGER POSITIONS
FINGER POSITIONS FOUND?
N
Y
MAP FINGER POSITIONS TO DISPLAY COORDS
STORE FINGER POSITIONS
MORPH GLYPHS FOUND?
EXTRACT CODE AND DETERMINE FUNCTION
EXTRACT POINTER POSITION AND MAP TO DISPLAY COORDS
N
FUNCTION IDENTIFIED?
N
FUNCTION INVOLVES TAP?
Y
N
FINGER TAP SENSED?
Y
DO FUNCTION (SEE FIG. 9(c))

Fig.9(b)

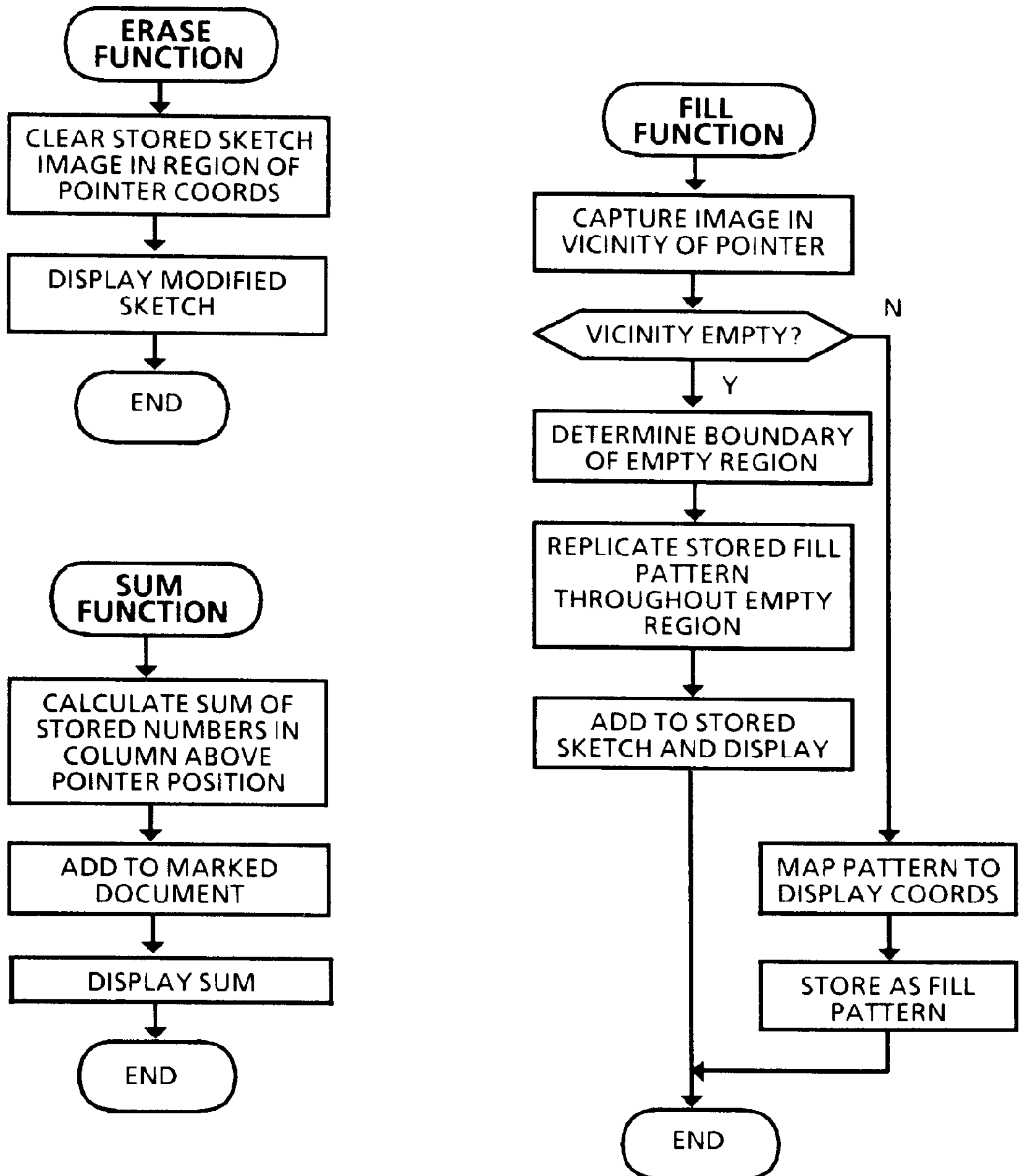

ERASE FUNCTION
CLEAR STORED SKETCH IMAGE IN REGION OF POINTER COORDS
DISPLAY MODIFIED SKETCH
END
SUM FUNCTION
CALCULATE SUM OF STORED NUMBERS IN COLUMN ABOVE POINTER POSITION
ADD TO MARKED DOCUMENT
DISPLAY SUM
END
FILL FUNCTION
CAPTURE IMAGE IN VICINITY OF POINTER
VICINITY EMPTY?
N
Y
DETERMINE BOUNDARY OF EMPTY REGION
REPLICATE STORED FILL PATTERN THROUGHOUT EMPTY REGION
ADD TO STORED SKETCH AND DISPLAY
MAP PATTERN TO DISPLAY COORDS
STORE AS FILL PATTERN
END

Fig.9(c).

Fig.10a

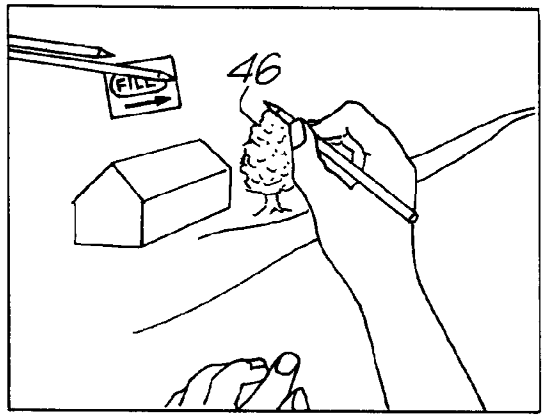



Fig.10b

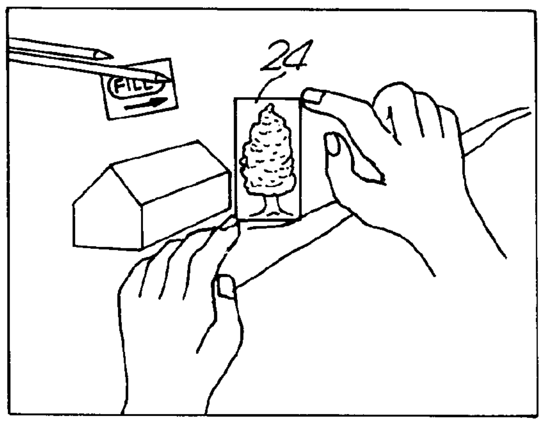



Fig.10c

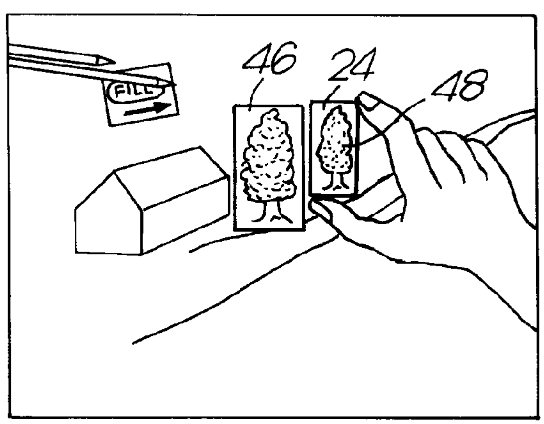


Fig.10d

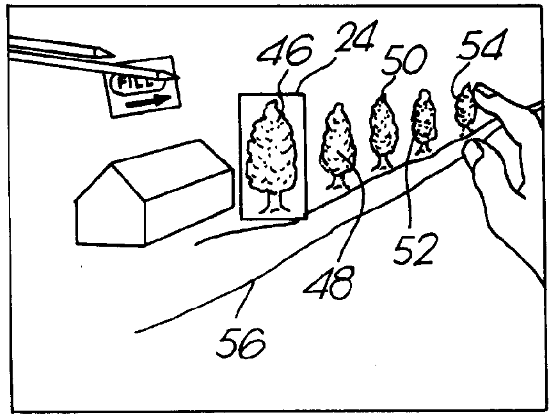



Fig.10e

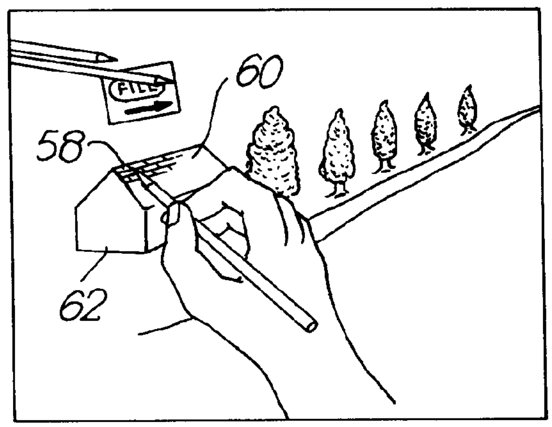



Fig.10f

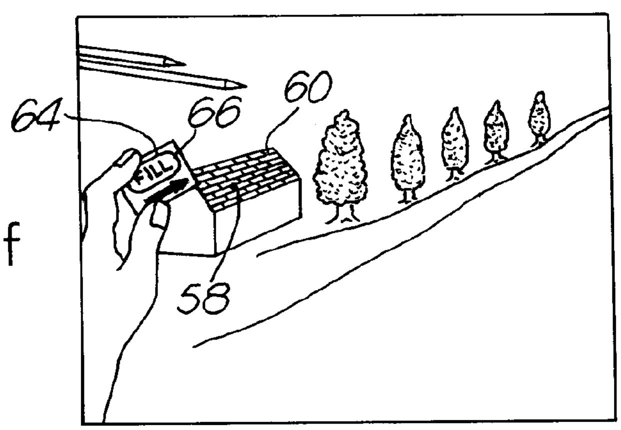


Fig.10g

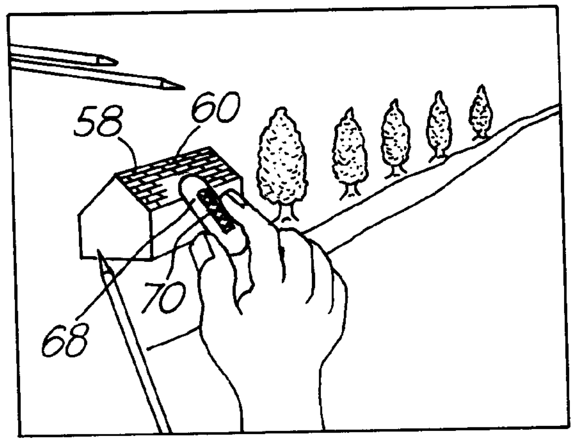
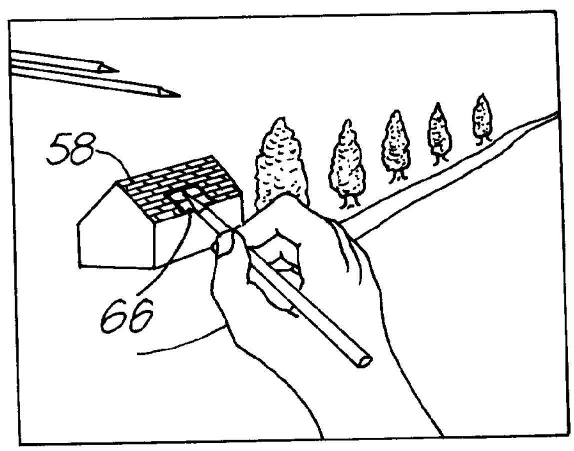



Fig.10h



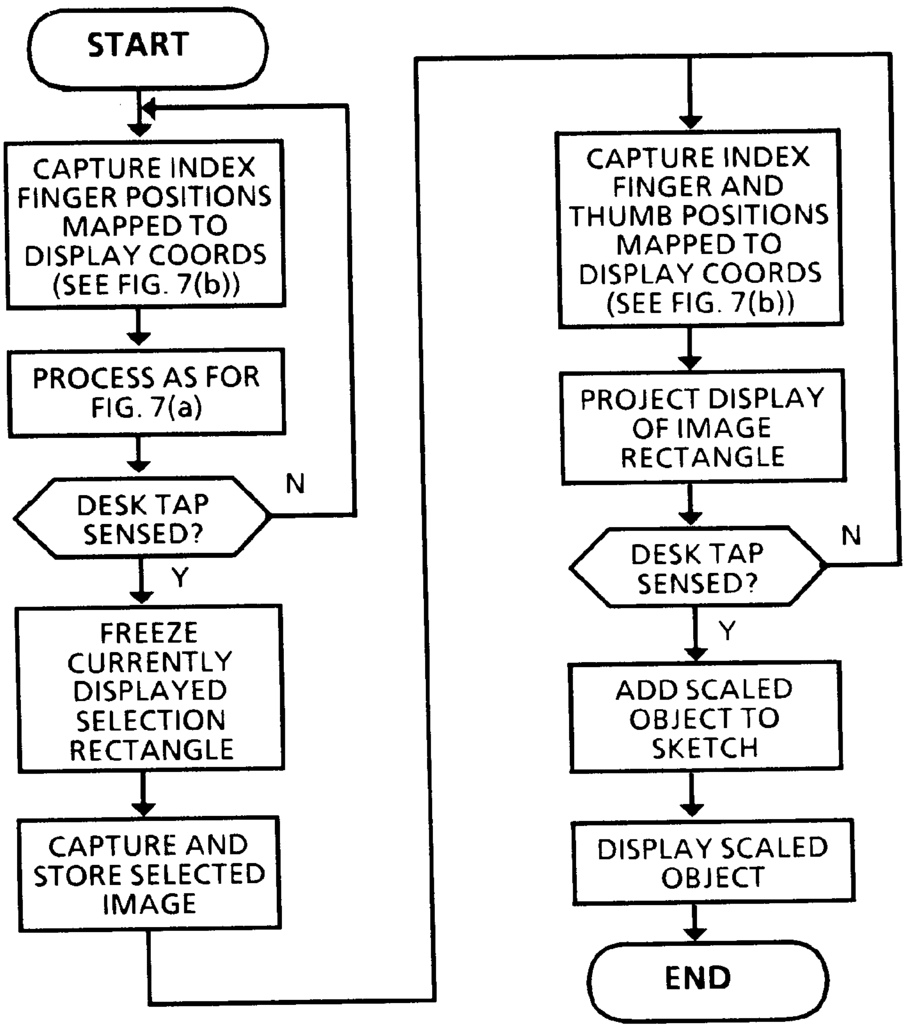

START
CAPTURE INDEX FINGER POSITIONS MAPPED TO DISPLAY COORDS (SEE FIG. 7(b))
PROCESS AS FOR FIG. 7(a)
DESK TAP SENSED?
N
Y
FREEZE CURRENTLY DISPLAYED SELECTION RECTANGLE
CAPTURE AND STORE SELECTED IMAGE
CAPTURE INDEX FINGER AND THUMB POSITIONS MAPPED TO DISPLAY COORDS (SEE FIG. 7(b))
PROJECT DISPLAY OF IMAGE RECTANGLE
DESK TAP SENSED?
N
Y
ADD SCALED OBJECT TO SKETCH
DISPLAY SCALED OBJECT
END

Fig.11

Fig.12.

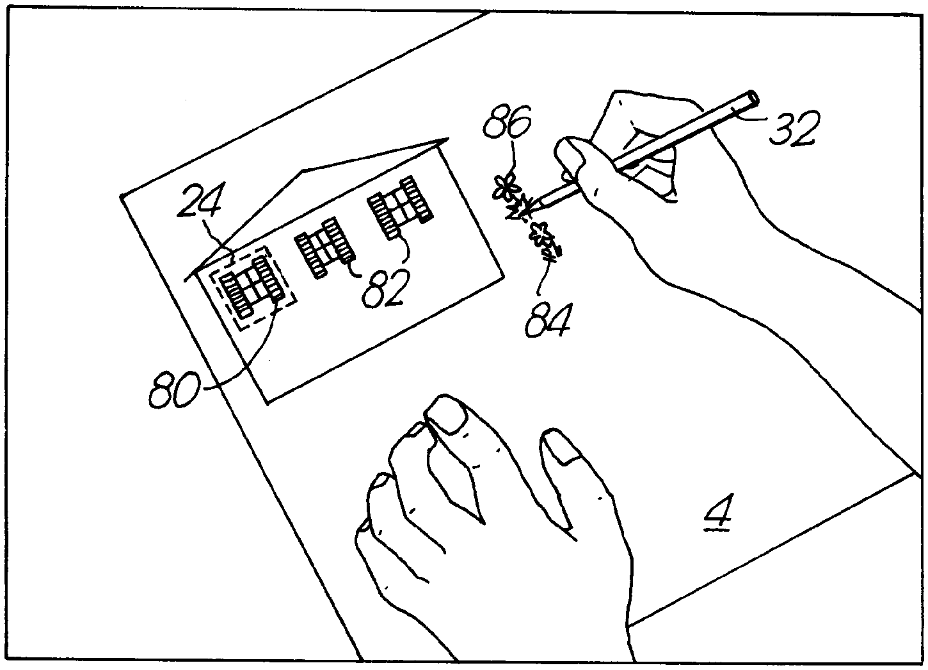
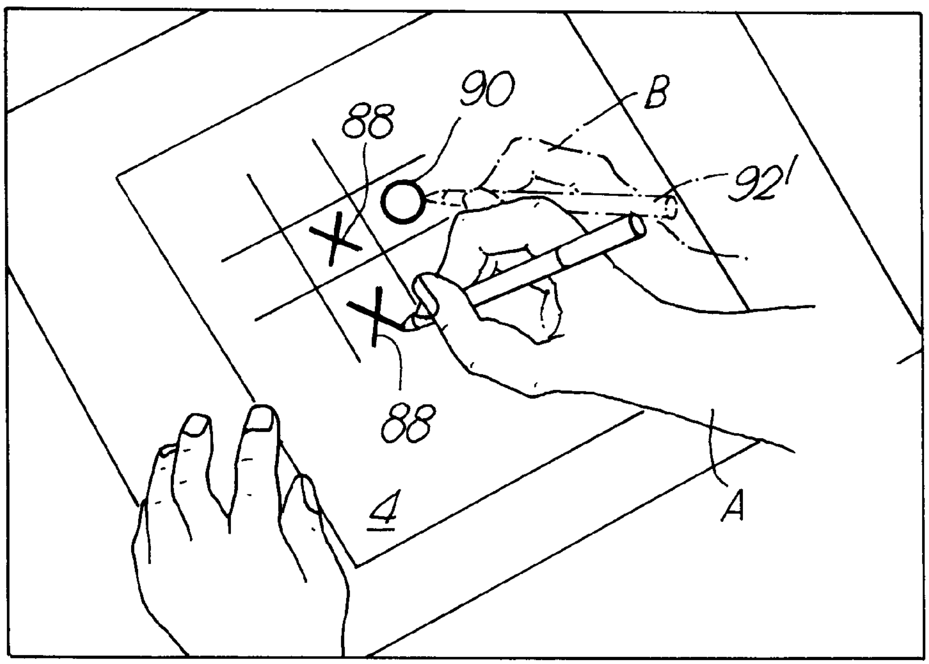



Fig.14.



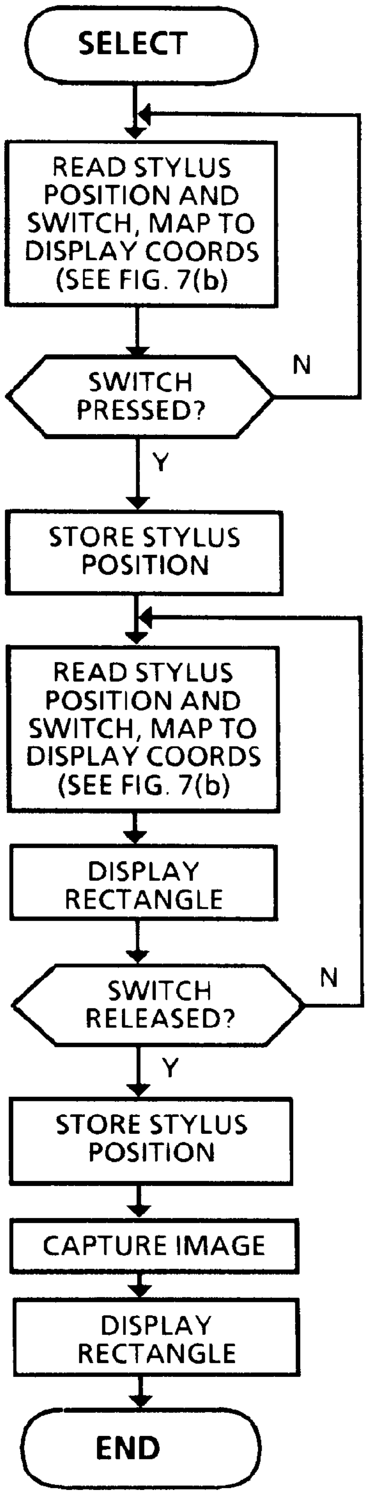


Fig.13(a).

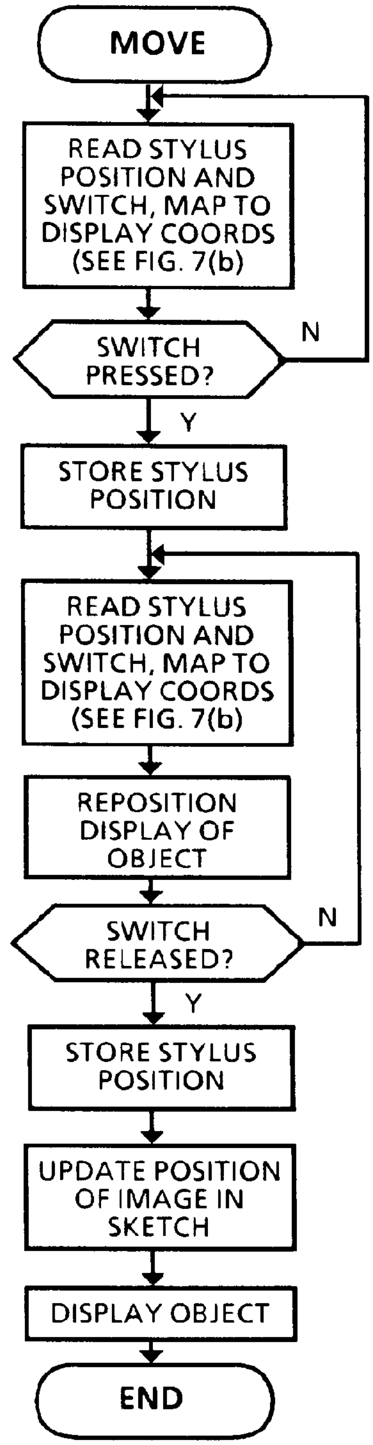


Fig.13(b).

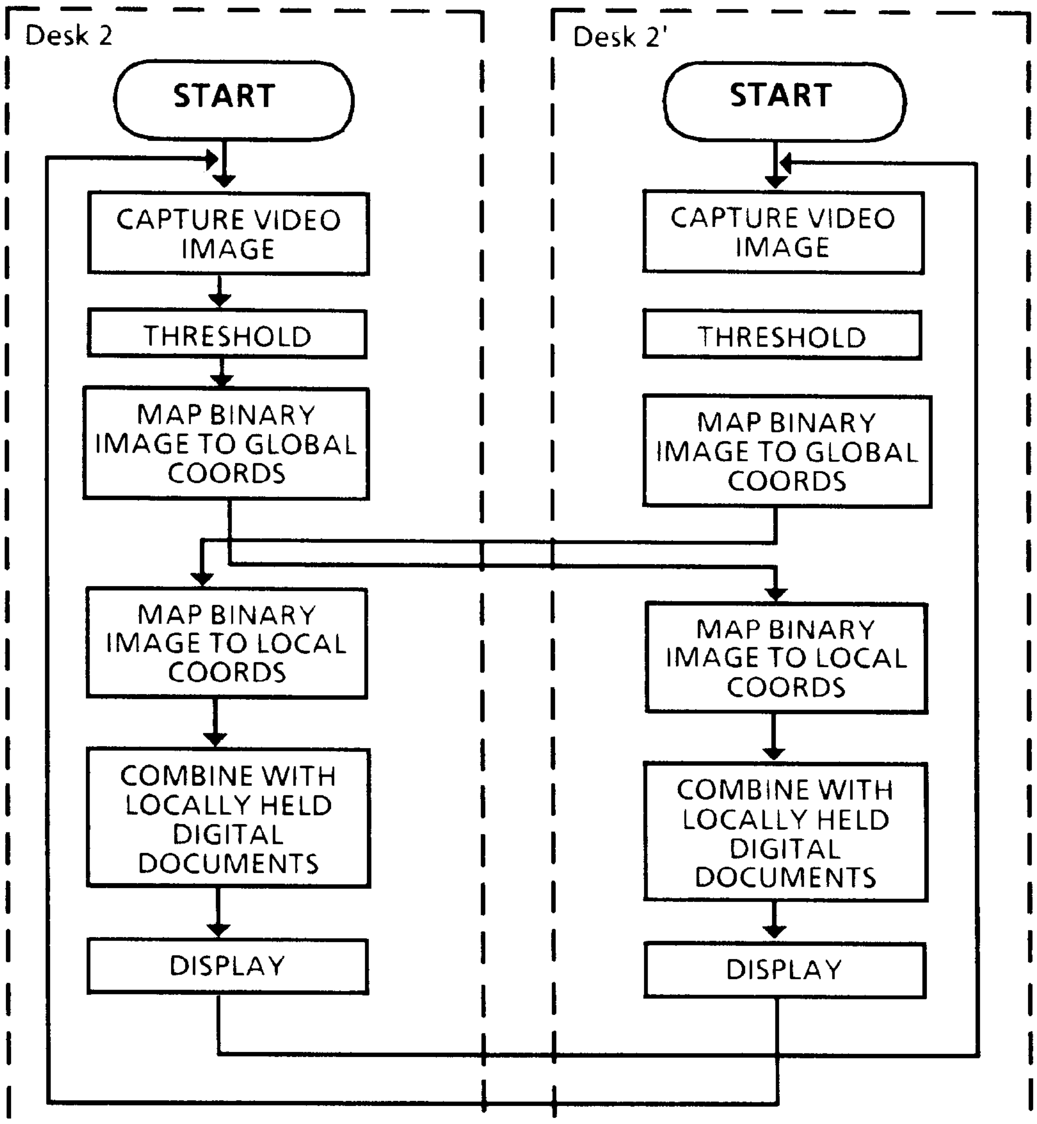

Desk 2
START
CAPTURE VIDEO IMAGE
THRESHOLD
MAP BINARY IMAGE TO GLOBAL COORDS
MAP BINARY IMAGE TO LOCAL COORDS
COMBINE WITH LOCALLY HELD DIGITAL DOCUMENTS
DISPLAY
Desk 2'
START
CAPTURE VIDEO IMAGE
THRESHOLD
MAP BINARY IMAGE TO GLOBAL COORDS
MAP BINARY IMAGE TO LOCAL COORDS
COMBINE WITH LOCALLY HELD DIGITAL DOCUMENTS
DISPLAY

Fig.15

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 622 722 A3

(12) # EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
**14.04.1999 Bulletin 1999/15**

(43) Date of publication A2:
**02.11.1994 Bulletin 1994/44**

(21) Application number: **94302912.4**

(22) Date of filing: **25.04.1994**

(51) Int Cl.6: **G06F 3/00**, G06F 3/033, G06K 11/08

(84) Designated Contracting States:
**DE FR GB**

(30) Priority: **30.04.1993 GB 9308955**
**01.07.1993 GB 9313637**

(71) Applicant: **RANK XEROX LIMITED**
**Marlow, Buckinghamshire SL7 1YL (GB)**

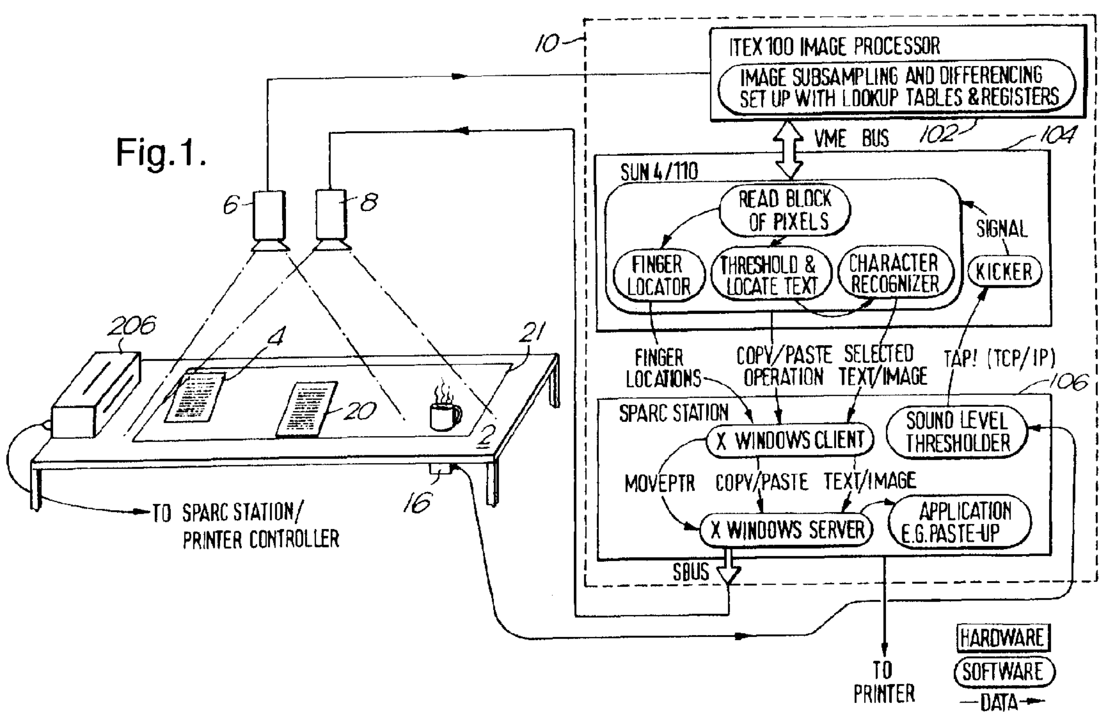
(72) Inventor: **Wellner, Pierre**
**Allenhurst, N.J. 07711 (US)**

(74) Representative: **Reynolds, Julian David et al**
**Rank Xerox Ltd**
**Patent Department**
**Parkway**
**Marlow Buckinghamshire SL7 1YL (GB)**

(54) **Interactive copying system**

(57) A system for generating new documents (20) from originals (4) containing text and/or images (22,28) employing e.g. a camera-projector system (6,8) focussed on a work surface (2), in conjunction with a copier or printer. In use, the camera 6 captures various manual operations carried out by the user, e.g. by pointing with fingers and tapping on the surface on the text or images (22,28) in an original paper document (4) on the surface (2) and representing manipulations of the text or images (22,28). Feedback to the user is provided by projection of an image (21,24) onto the surface or onto the original, or using some other visual display.

Fig.1.




EP 0 622 722 A3

European Patent Office

# EUROPEAN SEARCH REPORT

Application Number
EP 94 30 2912

## DOCUMENTS CONSIDERED TO BE RELEVANT

| Category | Citation of document with indication, where appropriate, of relevant passages | Relevant to claim | CLASSIFICATION OF THE APPLICATION (Int.Cl.5) |
|---|---|---|---|
| X | WELLNER P: "THE DIGITALDESK CALCULATOR: TANGIBLE MANIPULATION ON A DESK TOP DISPLAY" PROCEEDINGS OF THE SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOL ( UIST ), HILTON HEAD, S. CAROLINA, NOV. 11 - 13, 1991, no. SYMP. 4, 11 November 1991, pages 27-33, XP000315063 ASSOCIATION FOR COMPUTING MACHINERY * the whole document * | 1-10 | G06F3/00 G06F3/033 G06K11/08 |
| A | KRUEGER M W: "SELECTED PRACTICAL APPLICATIONS" ARTIFICIAL REALITY,1 January 1983, pages 140-149, XP002009298 * the whole document * | 1-9 | |
| A | US 5 025 314 A (TANG JOHN C ET AL) 18 June 1991 * abstract * * column 3, line 30 - column 4, line 26; figure 1 * * column 10, line 22 - column 11, line 24; figure 5 * | 10 | TECHNICAL FIELDS SEARCHED (Int.Cl.5) G06F G06K |
| P,X | WELLNER P: "INTERACTING WITH PAPER ON THE DIGITALDESK" COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY, vol. 36, no. 7, 1 July 1993, pages 87-96, XP000384570 * the whole document * | 1-10 | |
| P,X | EP 0 568 161 A (XEROX CORP) 3 November 1993 * the whole document * | 1-9 | |

The present search report has been drawn up for all claims

| Place of search | Date of completion of the search | Examiner |
|---|---|---|
| THE HAGUE | 17 February 1999 | Bravo, P |

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

## ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 94 30 2912

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

17-02-1999

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 5025314 A | 18-06-1991 | NONE | |
| EP 0568161 A | 03-11-1993 | EP 0495622 A<br>JP 6110755 A | 22-07-1992<br>22-04-1994 |

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 622 722 B1

# (12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
**17.07.2002 Bulletin 2002/29**

(51) Int Cl.[7]: **G06F 3/00**, G06F 3/033, G06K 11/08

(21) Application number: **94302912.4**

(22) Date of filing: **25.04.1994**

## (54) Interactive copying system

Interaktives Kopiersystem

Système de copie interactive

(84) Designated Contracting States:
**DE FR GB**

(30) Priority: **30.04.1993 GB 9308955**
**01.07.1993 GB 9313637**

(43) Date of publication of application:
**02.11.1994 Bulletin 1994/44**

(73) Proprietor: **Xerox Corporation**
**Rochester, New York 14644 (US)**

(72) Inventor: **Wellner, Pierre**
**Allenhurst, N.J. 07711 (US)**

(74) Representative: **Grünecker, Kinkeldey, Stockmair & Schwanhäusser Anwaltssozietät**
**Maximilianstrasse 58**
**80538 München (DE)**

(56) References cited:
**EP-A- 0 568 161** **US-A- 5 025 314**

- **WELLNER P: "THE DIGITALDESK CALCULATOR: TANGIBLE MANIPULATION ON A DESK TOP DISPLAY" PROCEEDINGS OF THE SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOL ( UIST ), HILTON HEAD, S. CAROLINA, NOV. 11 - 13, 1991, no. SYMP. 4, 11 November 1991, pages 27-33, XP000315063 ASSOCIATION FOR COMPUTING MACHINERY**
- **KRUEGER M W: "SELECTED PRACTICAL APPLICATIONS" ARTIFICIAL REALITY,1 January 1983, pages 140-149, XP002009298**
- **WELLNER P: "INTERACTING WITH PAPER ON THE DIGITALDESK" COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY, vol. 36, no. 7, 1 July 1993, pages 87-96, XP000384570**

## Description

**[0001]** The present invention relates to interactive image reproduction machines, and more particularly reproduction machines for performing various operations on text or images during the creation of a new document.

**[0002]** it is common for office workers and others who work with documents on a regular basis to have effectively two desks—the "electronic desktop" provided by a workstation or personal computer by means of a graphical interface, and the physical desk on which paper documents are received and processed.

**[0003]** The electronic desktop, which has become more and more like the physical one, can perform numerous useful operations on documents stored in electronic form; but in dealing with tangible documents such devices have limitations. A paper document must either be converted into electronic form before the operations are performed on it in the electronic environment, or copying operations are carried out on the tangible document using an electrophotographic copier (such as a photocopier with an editing function) or a combined scanning and printing device, the available functions of which are restricted in nature.

**[0004]** Wellner P.: "The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display", Proceedings of the Symposium on User Interface Software and Technology (UIST), Hilton Head, S. Carolina, November 11 to 13, 1991, No. Symp. 4, 11 November 1991, pages 27 to 33, XP000315063, Association for Computing Machinery describes a digital desk with a computer-controlled camera and projector above it. The camera sees where the user is pointing and it reads portions of documents that are placed on the desk. The projector displays feedback and electronic objects onto the desk surface. In more detail, the video camera is mounted above the digital desk pointing down at the work surface. The camera's output is fed through a computer and image processing system. The computer and image processing system is further connected with a computer-driven projector that is also mounted above the desk, allowing to superimpose electronic objects onto paper documents on the users work surface. The digital desk permits to project feedback onto the desk instead of on a separate screen. The key features of this digital desk are: using a camera to allow pointing with the fingers, reading paper documents on the desk, and projecting images onto the desk. Further to fingers, other things that can be used on the digital desk are described: an ordinary eraser may be made to erase electronic documents in addition to physical documents, a stapler could be used to attach electronic documents together, and a staple remover would detach them. Furthermore, a digital desk calculator is described as an example for an application of the digital desk.

**[0005]** The present invention seeks to reduce the above-mentioned limitations on the operations which may be performed on data in a paper document, using interactive techniques in which the paper document effectively becomes part of the means for designating which operations (such as text/image selection, creation and manipulation) are carried out on the information contained in it, in order to create new documents using a processor-controlled copying or printing device.

**[0006]** It is an object of the present invention to provide an interactive document creation system in which the available functions are expanded beyond both those provided by the conventional electronic environment alone, and those provided by existing copying equipment alone. This object is solved by the subject-matter of the independent claims. Preferred embodiments are the subject-matter of the dependent claims.

**[0007]** Embodiments of the present invention will now be described, by way of example, with reference to the accompanying drawings, in which:

Fig. 1 is a schematic diagram of a copying system according to the invention;
Fig. 2 shows schematically a known imaging system into which the system of Fig. 1 may be incorporated;
Fig. 3 illustrates an image generated in the finger-tracking technique employed in the present invention;
Fig. 4 shows a four point mapping technique employed by the present invention;
Fig. 5 illustrates four ways to sweep out a selection rectangle when using the present invention;
Figs 6(a) to (f) show successive scenes of the desk surface in a copying operation according.to one embodiment of the invention;
Fig. 7 is a flow chart of the procedure of Fig. 6;
Figs. 8(a) to (e) illustrate successive scenes of the desk surface in a copying operation according to a second embodiment of the invention;
Fig. 9 is a flow chart of the procedure of Fig. 8;
Figs. 10(a) to (h) show successive scenes of the desk surface in a copying operation according to a third embodiment of the invention;
Fig. 11 is a flow chart of the procedure of Fig. 10;
Fig. 12 shows a view from above of the desk surface during a copying operation according to a fourth embodiment of the invention;
Fig. 13 is a flow chart of the procedure of Fig. 12;
Fig. 14 shows a view from above of the desk surface during a copying operation according to a fifth embodiment of the invention; and
Fig. 15 is a flow chart of the procedure of Fig. 14.

**[0008]** Referring to Fig. 1, this illustrates schematically the copying system of the present invention. A flat desk surface 2 has.placed on it a document 4 to be used as a source if textual or graphical information during manipulations which are described in detail below. The document 4 is located within the field of view of a video camera 6 mounted above the desk surface 2. A video pro-