Bibliography

1 DIETZ, P.; LEIGH, D.. **Diamondtouch: a multi-user touch technology**. In: UIST '01: PROCEEDINGS OF THE 14TH ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, p. 219–226, New York, NY, USA, 2001. ACM.

2 KRUEGER, W.; FROEHLICH, B.. **The responsive workbench [virtual work environment]**. Computer Graphics and Applications, IEEE, 14(3):12–15, May 1994.

3 SHNEIDERMAN, B.. **Direct manipulation: A step beyond programming languages**. IEEE Computer, 16(8):57–69, 1983.

4 STURMAN, D. J.. **Whole-hand input**. PhD thesis, MIT, 1992. Supervisor: David Zeltzer.

5 REHG, J. M.; KANADE, T.. **Visual tracking of high DOF articulated structures: an application to human hand tracking**. In: ECCV (2), p. 35–46, 1994.

6 RIJPKEMA, H.; GIRARD, M.. **Computer animation of knowledge-based human grasping**. In: SIGGRAPH '91: PROCEEDINGS OF THE 18TH ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, p. 339–348, New York, NY, USA, 1991. ACM.

7 YING WU; HUANG, T.. **Hand modeling, analysis and recognition**. Signal Processing Magazine, IEEE, 18(3):51–60, May 2001.

8 NIREI, K.; SAITO, H.; MOCHIMARU, M. ; OZAWA, S.. **Human hand tracking from binocular image sequences**, 1996.

9 PAVLOVIC, V.; SHARMA, R. ; HUANG, T.. **Visual interpretation of hand gestures for human-computer interaction: A review**, 1997.

10 QUEK, F. K. H.. **Toward a vision-based hand gesture interface**. In: VRST '94: PROCEEDINGS OF THE CONFERENCE ON VIRTUAL REALITY SOFTWARE AND TECHNOLOGY, p. 17–31, River Edge, NJ, USA, 1994. World Scientific Publishing Co., Inc.

11 QUEK, F.. **Eyes in the interface**. IVC, 13(6):511–525, August 1995.

12 GUIARD, Y.. **Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model**, 1987.

13 KENDON, A.. **Current issues in the study of gesture**. The Biological Foundations of Gesture, p. 23–47, 1986.

14 BOWMAN, D. A.; KRUIJFF, E.; LAVIOLA JR., J. J. ; POUPYREV, I.. **3D User Interfaces: Theory and Practice**. Addison-Wesley/Pearson, 2005.

15 POUPYREV, I.; BILLINGHURST, M.; WEGHORST, S. ; ICHIKAWA, T.. **The go-go interaction technique: non-linear mapping for direct manipulation in vr**. In: UIST '96: PROCEEDINGS OF THE 9TH ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, p. 79–80, New York, NY, USA, 1996. ACM.

16 STOAKLEY, R.; CONWAY, M. J. ; PAUSCH, R.. **Virtual reality on a WIM: Interactive worlds in miniature**. In: PROCEEDINGS CHI'95, 1995.

17 BOWMAN, D. A.; HODGES, L. F.. **An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments**. In: SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, p. 35–38, 182, 1997.

18 MINE, M.. **Virtual environment interaction techniques**. Technical Report TR93-005, UNC Chapel Hill, Dept of Computer Science, North Carolina, USA, 1995.

19 ZHANG, Z.. **A flexible new technique for camera calibration**. IEEE Trans. Pattern Anal. Mach. Intell., 22(11):1330–1334, 2000.

20 HARTLEY, R.; STURM, P.. **Triangulation**. 68(2):146–157, November 1997.

21 KAKUMANU, P.; MAKROGIANNIS, S. ; BOURBAKIS, N.. **A survey of skin-color modeling and detection methods**. Pattern Recogn., 40(3):1106–1122, 2007.

22 HARRIS, C.; STEPHENS, M.. **A combined corner and edge detector**. Alvey Vision Conference Proceedings, p. 147–152, 1988.

23 LUCAS, B.; KANADE, T.. **An iterative image registration technique with an application to stereo vision**, 1981.

24 TOMASI, C.; KANADE, T.. **Detection and tracking of point features**. Technical Report CMU-CS-90-166, Carnegie Mellon University, USA, Apr. 1991.

25 SHI, J.; TOMASI, C.. **Good features to track**. In: IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR'94), Seattle, June 1994.

26 LOWE, D.. **Distinctive image features from scale-invariant keypoints**. In: INTERNATIONAL JOURNAL OF COMPUTER VISION, volumen 20, p. 91–110, 2003.

27 BAY, H.; TUYTELAARS, T. ; GOOL, L.. **Surf: Speed-up robust features**. In: EUROPEAN CONFERENCE ON COMPUTER VISION, p. 404–417, 2006.

28 MIKOLAJCZYK, K.; SCHMID, C.. **A performance evaluation of local descriptors**. Transactions on Pattern Analysis and Machine Intelligence, 27(10):1615–1630, Oct. 2005.

29 VIOLA, P.; JONES, M.. **Robust real-time object detection**. International Journal of Computer Vision, 2002.

30 FREUND, Y.; SCHAPIRE, R. E.. **A decision-theoretic generalization of on-line learning and an application to boosting**. In: EUROPEAN CONFERENCE ON COMPUTATIONAL LEARNING THEORY, p. 23–37, 1995.

31 KÖLSCH, M.; TURK, M.. **Robust hand detection**. In: FGR, p. 614–619, 2004.

32 BRADSKI, G. R.. **Computer vision face tracking for use in a perceptual user interface**. Intel Technology Journal, (Q2):15, 1998.

33 ISARD, M.; BLAKE, A.. **Condensation – conditional density propagation for visual tracking**. International Journal of Computer Vision, 29(1):5–28, 1998.

34 ISARD, M.; BLAKE, A.. **ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework**. Lecture Notes in Computer Science, 1406:893–908, 1998.

35 KOLSCH, M.; TURK, M.. **Fast 2d hand tracking with flocks of features and multi-cue integration**. In: CVPRW '04: PROCEEDINGS OF THE 2004 CONFERENCE ON COMPUTER VISION AND PATTERN

RECOGNITION WORKSHOP (CVPRW'04) VOLUME 10, p. 158, Washington, DC, USA, 2004. IEEE Computer Society.

36 SHIMADA, N.; KIMURA, K. ; SHIRAI, Y.. **Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera**. ratfg-rts, 00:0023, 2001.

37 ATHITSOS, V.; SCLAROFF, S.. **3d hand pose estimation by finding appearance-based matches in a large database of training views**. Technical Report BU-CS-TR-2001-021, Computer Science Department, Boston University, Boston, USA, 2001.

38 SUTHERLAND, I.. **Sketchpad: A man-machine graphical communication system**. PhD thesis, Massachusetts Institute of Technology, January 1963.

39 NEWMAN, W. M.. **An experimental program for architectural design**. The Computer Journal, 9(1):21–26, May 1966.

40 NEWMAN, W. M.. **A system for interactive graphical programming**. In: AFIPS SPRING JOINT COMPUTER CONFERENCE, p. 47–43, 1968.

41 CLARK, J. H.. **Designing surfaces in 3-d**. Commun. ACM, 19(8):454–460, 1976.

42 KAY, A.; GOLDBERG, A.. **Personal dynamic media**. Computer, 10(3):31–41, 1977.

43 BOLT, R. A.. **"put-that-there": Voice and gesture at the graphics interface**. In: SIGGRAPH '80: PROCEEDINGS OF THE 7TH ANNUAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, p. 262–270, New York, NY, USA, 1980. ACM.

44 NEGROPONTE, N.. **The media room**. Technical Report Report for ONR and DARPA, Cambridge, MA, USA, 12, 1978.

45 SACHS, E.; STOOPS, D. ; ROBERTS, A.. **3-draw: a three dimensional computer aided design tool**. Systems, Man and Cybernetics, 1989. Conference Proceedings., IEEE International Conference on, p. 1194–1196 vol.3, 14-17 Nov 1989.

46 SACHS, E.; ROBERTS, A. ; STOOPS, D.. **3-draw: A tool for designing 3d shapes**. IEEE Computer Graphics and Applications, 11(6):18–26, 1991.

47 KRUEGER, M.. **Artificial Reality II**. Addison-Wesley: Reading, MA, 1991., second edition, 1991.

48 KRUEGER, M. W.; GIONFRIDDO, T. ; HINRICHSEN, K.. **Videoplace — an artificial reality**. SIGCHI Bull., 16(4):35–40, 1985.

49 ROBINETT, W.; HOLLOWAY, R.. **Implementation of flying, scaling and grabbing in virtual worlds**. In: SI3D '92: PROCEEDINGS OF THE 1992 SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, p. 189–192, New York, NY, USA, 1992. ACM.

50 SU, S. A.; FURUTA, R.. **A specification of 3d manipulation in virtual environments**, 1994.

51 STRAUSS, P. S.; CAREY, R.. **An object-oriented 3d graphics toolkit**. SIGGRAPH Comput. Graph., 26(2):341–349, 1992.

52 CONNER, D. B.; SNIBBE, S. S.; HERNDON, K. P.; ROBBINS, D. C.; ZELEZNIK, R. C. ; VAN DAM, A.. **Three-dimensional widgets**. In: PROCEEDINGS OF THE 1992 SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, SPECIAL ISSUE OF COMPUTER GRAPHICS, VOL. 26, p. 183–188, 1992.

53 BUTTERWORTH, J.; DAVIDSON, A.; HENCH, S. ; OLANO, M. T.. **3dm: a three dimensional modeler using a head-mounted display**. In: SI3D '92: PROCEEDINGS OF THE 1992 SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, p. 135–138, New York, NY, USA, 1992. ACM.

54 HINCKLEY, K.; PAUSCH, R.; GOBLE, J. C. ; KASSELL, N. F.. **A survey of design issues in spatial input**. In: UIST '94: PROCEEDINGS OF THE 7TH ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, p. 213–222, New York, NY, USA, 1994. ACM.

55 SHAW, C.; GREEN, M.. **THRED: A two-handed design system**. Multimedia Systems, 5(2):126–139, 1997.

56 MURAKAMI, T.; NAKAJIMA, N.. **Direct and intuitive input device for 3-d shape deformation**. In: CHI '94: PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, p. 465–470, New York, NY, USA, 1994. ACM.

57 LIANG, J.; GREEN, M.. **Jdcad: a highly interactive 3d modeling system**. Computers & Graphics, 18(4):499–506, 1994.

58 DEERING, M. F.. **Holosketch: a virtual reality sketching/animation tool**. ACM Trans. Comput.-Hum. Interact., 2(3):220–238, 1995.

59 GRIMM, C.; PUGMIRE, D.. **Visual interfaces for solids modeling**. In: ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, p. 51–60, 1995.

60 MAPES, D. P.; MOSHELL, J. M.. **A two-handed interface for object manipulation in virtual environments**. Presence, p. 403–416, 1995.

61 MINE, M. R.. **Working in a virtual world: Interaction techniques used in the chapel hill immersive modeling program**. Technical Report TR96-029, 12, 1996.

62 ZELEZNIK, R. C.; HERNDON, K. P. ; HUGHES, J. F.. **SKETCH: An interface for sketching 3D scenes**. In: Rushmeier, H., editor, SIGGRAPH 96 CONFERENCE PROCEEDINGS, p. 163–170. Addison Wesley, 1996.

63 MINE, M. R.; BROOKS, JR., F. P. ; SEQUIN, C. H.. **Moving objects in space: Exploiting proprioception in virtual-environment interaction**. Computer Graphics, 31(Annual Conference Series):19–26, 1997.

64 CUTLER, L. D.; FROEHLICH, B. ; HANRAHAN, P.. **Two-handed direct manipulation on the responsive workbench**. In: SYMPOSIUM ON INTERACTIVE 3D GRAPHICS, p. 107–114, 191, 1997.

65 ZHAI, S.. **User performance in relation to 3d input device design**. SIGGRAPH Comput. Graph., 32(4):50–54, 1998.

66 FJELD, M.; BICHSEL, M. ; RAUTERBERG, M.. **Build-it: An intuitive design tool based on direct object manipulation**. In: PROCEEDINGS OF THE INTERNATIONAL GESTURE WORKSHOP ON GESTURE AND SIGN LANGUAGE IN HUMAN-COMPUTER INTERACTION, p. 297–308, London, UK, 1998. Springer-Verlag.

67 FORSBERG, A.; LAVIOLA, J. ; ZELEZNIK, R.. **Ergodesk: A framework for two and three dimensional interaction at the activedesk**, 1998.

68 NISHINO, H.; UTSUMIYA, K. ; KORIDA, K.. **3d object modeling using spatial and pictographic gestures**. In: VRST, p. 51–58, 1998.

69 SCHKOLNE, S.; SCHRODER, P.. **Surface drawing**. Technical Report CS-TR-99-03, Pasadena, CA, USA, 1999.

70 SCHKOLNE, S.; PRUETT, M. ; SCHRÖDER, P.. **Surface drawing: creating organic 3d shapes with the hand and tangible tools**. In: CHI '01: PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN

FACTORS IN COMPUTING SYSTEMS, p. 261–268, New York, NY, USA, 2001. ACM.

71 LEIBE, B.; STARNER, T.; RIBARSKY, W.; WARTELL, Z.; KRUM, D.; SINGLETARY, B. ; HODGES, L. F.. **The perceptive workbench: Toward spontaneous and natural interaction in semi-immersive virtual environments**. In: VR, p. 13–20, 2000.

72 WU, M.; BALAKRISHNAN, R.. **Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays**. In: UIST '03: PROCEEDINGS OF THE 16TH ANNUAL ACM SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, p. 193–202, New York, NY, USA, 2003. ACM.

73 BUCHMANN, V.; VIOLICH, S.; BILLINGHURST, M. ; COCKBURN, A.. **Fingartips: gesture based direct manipulation in augmented reality**. In: GRAPHITE, p. 212–221, 2004.

74 PRATINI, E.. **Modeling with gestures: Sketching 3d virtual surfaces and objects using hands formation and movements**. In: ASCAAD INTERNATIONAL CONFERENCE, p. 35–41, 2004.

75 KIM, H.; ALBUQUERQUE, G.; HAVEMANN, S. ; W. FELLNER, D.. **Tangible 3d: Immersive 3d modeling through hand gesture interaction**. Technical Report TUBS-CG-2004-07, Institute of Computer Graphics, University of Technology, Braunschweig, Germany, 2004.

76 LINGRAND, D.; RENEVIER, P.; PINNA-DERY, A.-M.; CREMASCHI, X.; LION, S.; ROUEL, J.-G.; JEANNE, D.; CUISINAUD, P. ; SOULA, J.. **Gestaction3d: a platform for studying displacements and deformation of 3d objects using hands**. In: INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN OF USER INTERFACES (CADUI), p. 105–114, 2006.

77 DACHSELT, R.; HINZ, M.. **Three-dimensional widgets revisited - towards future standardization**. In: PROCEEDINGS OF THE WORKSHOP 'NEW DIRECTIONS IN 3D USER INTERFACES', 2005.

78 DACHSELT, R.; HUEBNER, A.. **Virtual environments: Three dimensional menus: A survey and taxonomy**. Comput. Graph., 31(1):53–65, 2007.

79 TRUYENQUE, M. A. Q.. **A computer vision application that uses hand gestures for human-computer interaction**. Master's thesis, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil, March 2005.

80 BETTIO, F.; GIACHETTI, A.; GOBBETTI, E.; MARTON, F. ; PINTORE, G.. **A practical vision based approach to unencumbered direct spatial manipulation in virtual worlds**. In: EUROGRAPHICS ITALIAN CHAPTER CONFERENCE, Conference held in Trento, Italy, February 2007. Eurographics Association.

81 AMIT, Y.; GEMAN, D. ; WILDER, K.. **Joint induction of shape features and tree classifiers**, 1997.

82 Solla, S. A.; Leen, T. K. ; Müller, K.-R., editors. **Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]**. The MIT Press, 2000.

83 ROWLEY, H. A.; BALUJA, S. ; KANADE, T.. **Neural network-based face detection**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):23–38, 1998.

84 SCHNEIDERMAN, H.; KANADE, T.. **A statistical method for 3d object detection applied to faces and cars**. Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on, 1:746–751 vol.1, 2000.

85 SUNG, K.-K.; POGGIO, T.. **Example-based learning for view-based human face detection**. IEEE Trans. Pattern Anal. Mach. Intell., 20(1):39–51, 1998.

86 LIENHART, R.; MAYDT, J.. **An extended set of haar-like features for rapid object detection**. In: PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, p. 900–903, Rochester, USA, September 2002. IEEE.

87 HARTLEY, R. I.; ZISSERMAN, A.. **Multiple View Geometry in Computer Vision**. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

A
Timeline of manipulation-related research

The following chronologically sorted and annotated list contains publications that are related to direct spatial manipulation of virtual geometric objects. The criteria to include a publication in this list are:

1. The publications must deal with manipulation of virtual geometrical three-dimensional objects, and/or their 2D sections or projections.

2. The spatial 3D manipulation must be initiated and executed through an interface involving human hands, fingers and/or arms, using any input device.

The second criterion above practically means that both input done vision-based tracking of human hands, as well as input done using devices like mice, pens, tables, datagloves and all sorts of many-d.o.f. devices will be included. The list is not meant to be exhaustive, but *representative* of prior work related to manipulation of 3D geometry, using hands.

A.1
Pre-1980s

1963

– PhD thesis [38] by Ivan Sutherland that influenced almost everything we know and think about computing, see Figures A.1 and A.2. The application (Sketchpad, also known as "Robot Draftsman") that Sutherland developed as a part of his thesis used a light-pen to draw and manipulate (grab, copy and move) 2D shapes on the screen, changing their sizes and using constraints. It can be considered to be the precursor to modern Computer-Aided Design (CAD) applications. This thesis influenced the development of Xerox Star workstation which later on influenced the development of Mac OS, Windows and X-Windows operating systems.

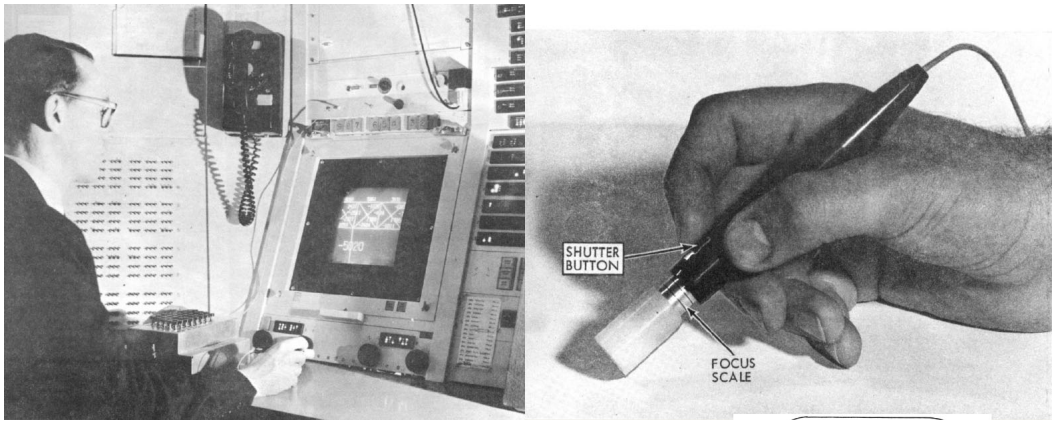


Figure A.1: Sutherland's Sketchpad in use (Lincoln TX-2 console, lightpen)

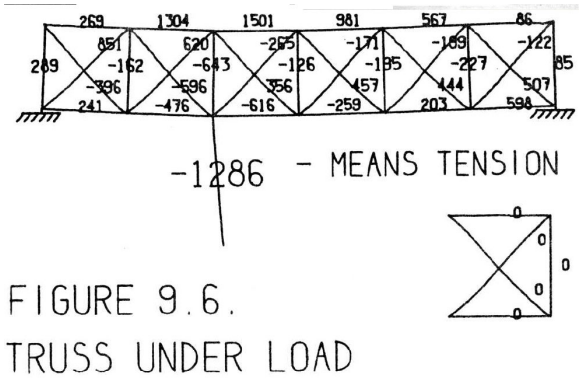


Figure A.2: Example of a drawing and calculation made in Sutherland's Sketchpad: truss load

1966

– [39] William Newman's early system for architectural design. Based on a PDP-7 computer, Type 340 Display and a type of light pen. The system uses modular building blocks. The paper describes the approaches used to organizing the display list for efficient manipulation and an algorithm for computing areas of enclosed spaces.

1968

– [40] William Newman's "reaction handler", based on tablet and stylus. Provided direct manipulation of graphical shapes. Introduced "light handles", a type of graphical potentiometer, which could be considered the first "manipulation widget".

1976

– [41] "Designing surfaces in 3-D" system [41] by another pioneer in the field of computer graphics (and co-founder of Silicon Graphics, Netscape Communications and some other companies) James H. Clark. One of the earliest interactive design systems for drawing free form 3D (parametric) surfaces, utilizing HMD. The mechanically-tracked HMD utilized in the Clark's system was designed by Ivan Sutherland. Surfaces can be controlled by manipulating control points on the associated wireframe grid (see Figure A.4). The system used a 3-D wand that computed its position my measuring its distance to the ceiling.

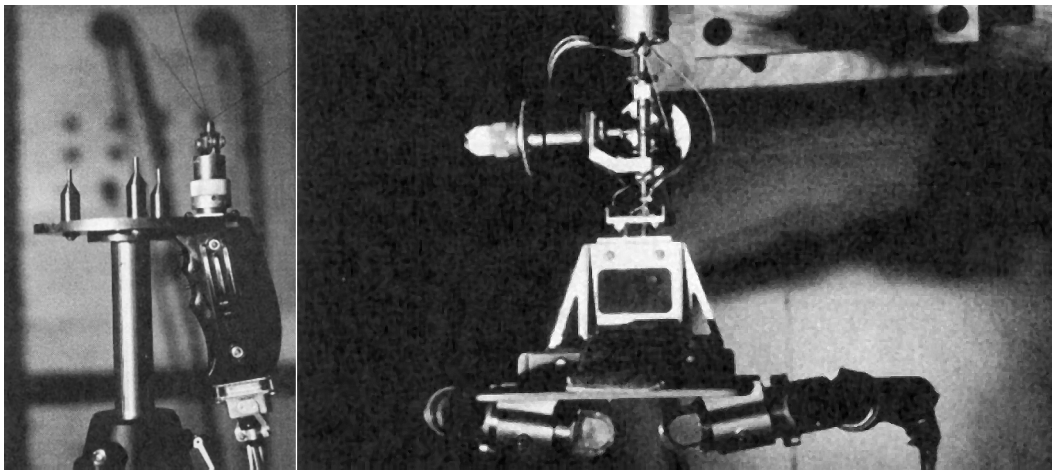


Figure A.3: James H. Clark's system: 3D-wand (left) and HMD armature (right)

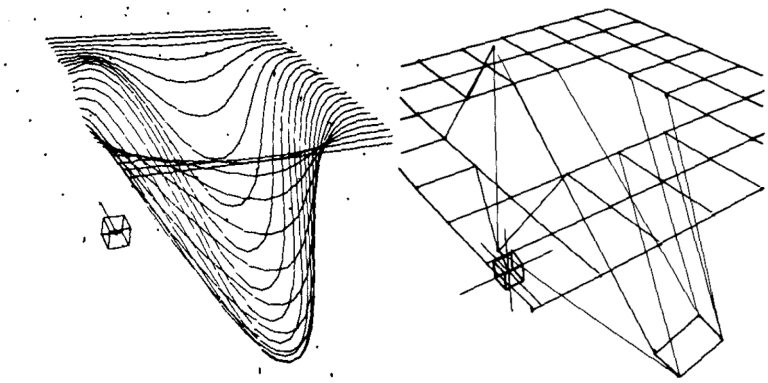


Figure A.4: James H. Clark's system: 3D surface being edited (left) and its grid of control points (right)

1977

– [42] Alan Kay's (Xerox PARC) gives a vision of direct manipulation interfaces for everyone, using the Smalltalk language and Dynabook.

A.2
1980s

1980

– Bolt's "Put-That-There" system [43]. Uses hand gestures together with speech input. Manipulates simple shapes on a large, wall-sized screen. The user can, for example, point to a shape, and then utter a command to modify the shape. Uses a "Media Room" by Negroponte [44], an enclosure which supplants the CRT display and turns the whole room into a sort of input-output space. The user sits in a modified chair: each arm of the chair has a small joystick sensitive to direction and pressure. Besides them there are two small touch-sensitive pads. On either sides of the chair are located TV monitors, whose cathode tube's surface has been coated with a touch-sensitive surface. Commands: CREATE ("create a blue square there"), MOVE ("move the blue triangle to the right of the green square"), MAKE THAT ("make that blue triangle smaller"), DELETE ("delete the large blue circle"), CALL THAT ("call that blue square the calendar"). Six-d.o.f. Polheus tracker epoxied in a cube, attached to user's wrist via a watchband.

1983

– [3] Ben Shneiderman coins the expression "direct manipulation" and defines its constituent components as well as psychological foundations. Describes graphically-based interaction, visibility of objects, incremental action and rapid feedback.

1987

– [12] Guiard gives a theoretical framework for the study of asymmetry in the context of human bimanual action. Most skilled manual activities involve two hands playing different roles. The two hands represent two motors, which cooperate with one another as if they were assembled in series, thereby forming a *kinematic chain*. In right-handed people, motion produced by the right hand tends to be articulated with motion produced by the left.

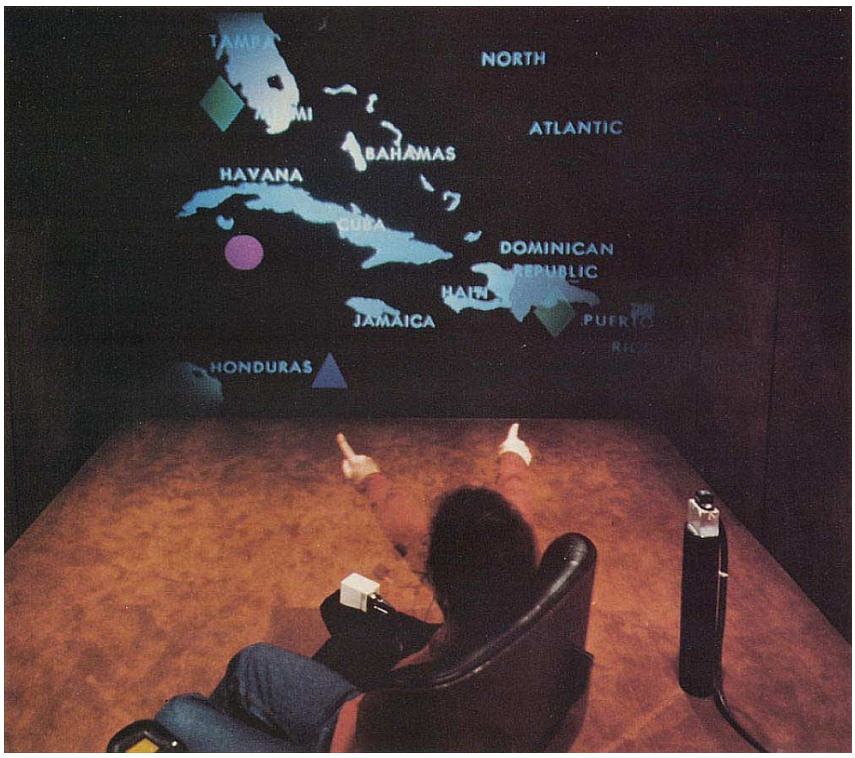


Figure A.5: "Put-That-There" system by Bolt: manipulating shapes on the wall-sized screen. The user currently points at the circular shape

1989

– 3-Draw, a 3D computer-aided design tool [45], [46], see Figure A.6. Output image is shown on a conventional non-stereo display. Capable of drawing complex free-form shapes. Uses two 6-d.o.f. sensors - one sensor is a configurable 3D drawing and editing tool, and the other sensor controls an object's position and orientation. One hand holds a tracked palette that acts as a movable reference frame in modeling space. The other hand holds a stylus and draws 2D curves on the palette. This combination thus resulted in curves in 3D space. Users found the interface natural and quick. Simultaneous use of two hands provided kinesthetic feedback that enabled users to feel as though they were holding the objects displayed on the screen.

A.3
1990s

1991

– VIDEODESK [47] — it consists of a large surface over which the user moves his fingers, hands and arms, see Figure A.7. The system uses over-

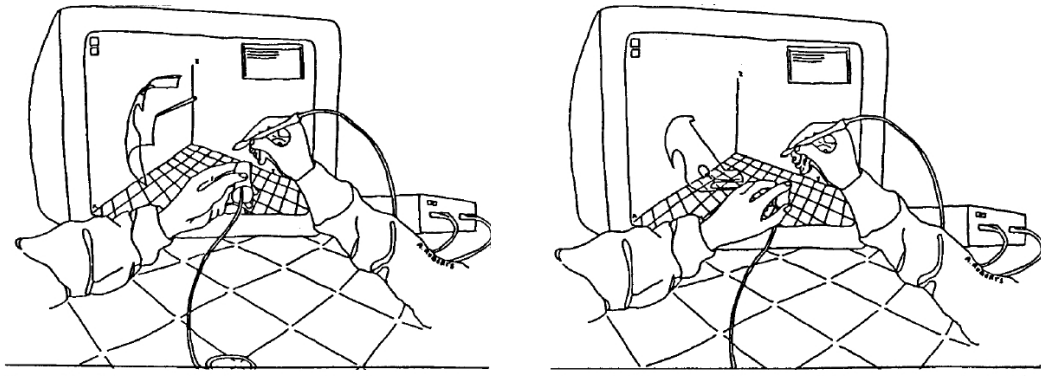


Figure 1. Designer sketches automobile fender in three dimensions using 3-Draw tools.

Figure 2. New orientation of model interactively obtained by moving left hand.

Figure A.6: 3-Draw by Sachs *et al* — based on two 6-d.o.f. sensors and a conventional non-stereo display.

head video cameras to track the appearance and 2D hand position and to detect image features such as the hand, fingers, and their orientation. The system uses a large horizontal table with a bright background (for easier detection and segmentation of hands). The geometry being manipulated is being modeled using splines. Control points on these splines can then be controlled using index fingers and thumbs of both hands. Similarly, using both index fingers it is possible to draw a circle on the screen.

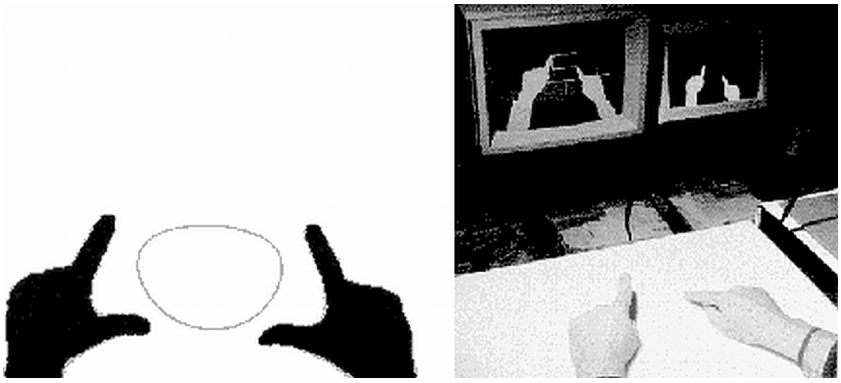
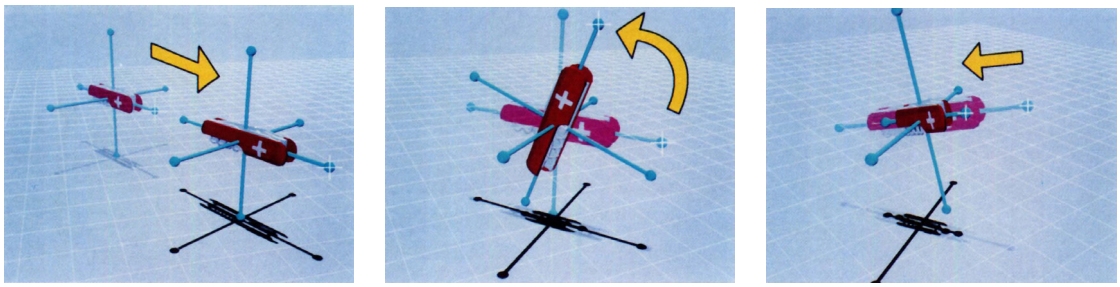


Figure A.7: Krueger's VIDEODESK. Splines are controlled by fingertip positions.

A related, older system by the same author is VIDEOPLACE [48] which tracks the whole body, not just hands. That is, the user uses his entire body as input to the system. It was installed as a part of a large installation on the Computers and Art exhibit at IBM building in New York, US. The author has linked this environment with VIDEODESK.

1992

- [49] Hand gestures initiate and terminate fundamental actions (translate, scale, rotate) that change the state of virtual world. Manual input device may be an instrumented glove or or a hand-held device with buttons. Specifies transforms for grabbing an object, flying, scaling the world.
- [50] A specification of 3D manipulation operations, based on hand gestures. Three basic gestures (touching, pointing and gripping) are defined. *Touching* is a simple gesture with no extra information; by tracking the logical hand in virtual 3D space, and using collision detection, it is said that the hand "touched" an object if the hand collided with the object. *Pointing* requires an extended index finger; its fingertip position then defines the starting point of the pointing, and the orientation of the index finger is the pointing direction. Using this information it is possible to determine the 3D object pointed at. Finally, *gripping* is defined as an analog of the click-and-drag operation in classical 2D WIMP interfaces.
- [51] Describes what is to become Open Inventor. Describes *manipulators* (trackball, one-axis scale, jack, handle box, spot light, directional light, one-axis translate).
- Conner et al [52] describe three-dimensional *widgets* (Figure [52]). Gives precise state diagrams. Virtual sphere, handles, snapping, color picker, rack, cone tree.



(a) (b) (c)

Figure A.8: Widgets by Conner et al. Translating a knife along its x axis (a), rotating a knife along an axis (b), and scaling a knife along an axis (c)

- [53] 3DM (Three-Dimensional Modeler). An interactive surface-modeling system. It uses a stereo HDM, and one single bat with 6 d.o.f. User can create 3D objects. Walking, flying, grabbing the world, scaling the user.

1994

- [54] A survey of design issues for developing effective free-space 3D user interfaces. People do not innately understand 3D reality, but rather they experience it. Concepts that facilitate 3D space perception: spatial references, relative vs. absolute gestures, two-handed interaction, multisensory feedback, physical constraints, head tracking techniques. Coarse vs. precise positioning tasks: gridding and snapping. Dynamics and size of the working volume of user's hands. Use of mice and keyboards in combination with free-space input devices; voice input, touch screen; hybrid interfaces. Clutching mechanisms. Importance of ergonomic details in spatial interfaces.
- [55] THRED (Two-Handed Refining EDitor). Output images displayed on a conventional, non-stereo monitor. The user manipulates two 3D position and orientation trackers with three buttons (i.e. button-enhanced Bats, that is, standard Polhemus sensors with three buttons attached), for each hand. Four postures for holding the Bat. Dominant hand for picking and manipulation, less-dominant hand for context setting. System intended for free-form sketching of polygonal surfaces (terrains, natural objects). Surfaces are hierarchically-refined polygonal surfaces based on quadrilaterals.
- A real elastic object, made from electrically conductive polyurethane, is being used as an input device for 3D shape deformation [56], see Figure A.9. The operation of twisting while bending is possible. Any combination of pressing, bending, twisting possible. A tactile input device, thus giving haptic feedback.

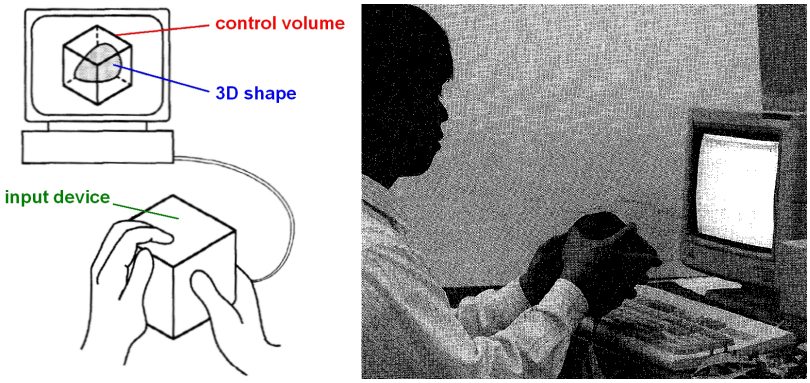


Figure A.9: Murakami's elastic cube for 3D deformation: schematic (left) and usage (right)

- JDCAD [57] — input is a 6-d.o.f. bat, and output is a kinetic head-tracked non-stereo display, see Figure A.10. Object selection using the

spotlight metaphor; innovative menus (daisy, ring); object creation, manipulation, viewing.

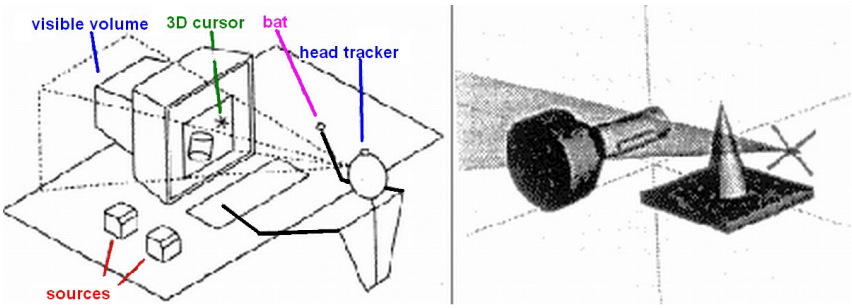


Figure A.10: JDCAD: schematic (left) and cone selection technique (right)

1995

– Mine discusses [18] virtual environment interaction techniques, and gives an introduction to fundamental forms of interaction (Figure A.11): movement (specifiying direction and speed), selection (local and at-a-distance), manipulation (change in position, orientation and center of rotation) and scaling (center of scaling and scaling center; uniform and non-uniform scaling). Lists hand tracking, gesture recognition, pointing, gaze direction. Lists physical and virtual controls. Gives coordinate system transforms in an appendix.

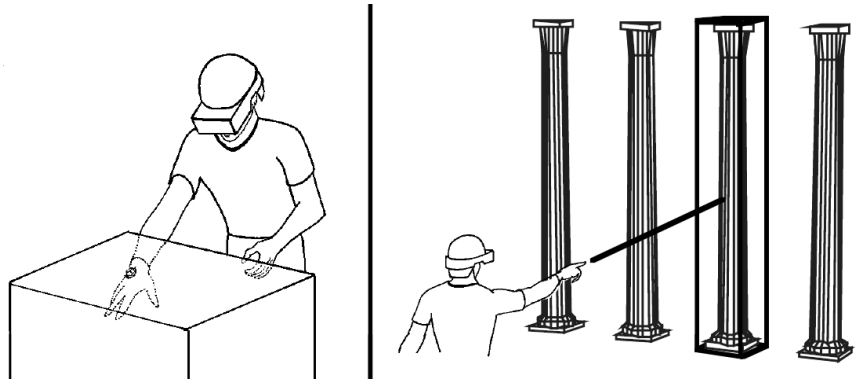


Figure A.11: Mine's local selection (left) and at-a-distance selection (right)

– HoloSketch [58], a VR system for 3D geometry creation and manipulation (Figure A.12). Fishtank stereo CRT, head-tracked stereo glasses, 3D mouse/wand ("one-fingered data glove") augmented by an offset digitizer rod, effectively making from it a six-axis wand. The wand tip feels like an extension of index finger. This 3D mouse has three top buttons and one side button. Multi-level 3D fade-up menu system, invoked by holding

down right wand button. Modal editor (a single current drawing or editing mode in force at any given moment). Draws rectangular solids, spheres, ellipsoids, cylinders, cones, rings, free-form tubes, 3D text, isolated line segments, free-form and polyline wires. Editing operations, Significant gains in productivity over 2D interface technology reported.

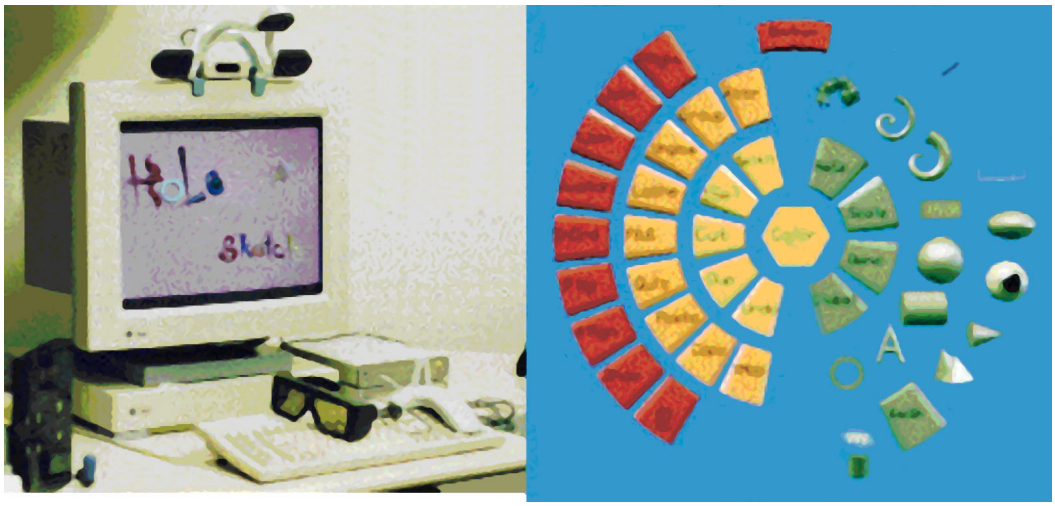


Figure A.12: Deering's Holosketch: head-tracked stereo glasses and 3D mouse/wand (left) and 3D fade-up menu (right)

- [59] Visual interfaces (manipulators) for solids modeling. Similar to [51]. Free-form operators such as blends, sweeps, and deformations. Sweep tool. Warp tool. Rail-tie tool. Rail-curve manipulation tool. Operator space. Visual tool should provide visual clues on its function and use. The design of the visual tool should be based on the user's intuition on how the operator should behave, not on the parameters to the operator.
- [60] PolyShop system. Two ChordGloves (datagloves which have electric contacts on fingertips and on the palm) used for bimanual input. Two hands used for translating, rotating and scaling of virtual objects. Hand gestures, read from different combinations of contacts between fingers and the palm.
- [16] Stoakley, Conway and Pausch present a two-handed architectural design system named WIM (Worlds In Miniature). The user is fully immersed into the virtual environment, and has a concurrent view to a miniature hand-held copy of the entire world attached to a tracker manipulated by the left hand. A clipboard is attached to the left tracker, and the surface of the clipboard represents the floor of the WIM. The right hand holds a ball containing another tracker with two buttons (the first button for selecting objects, and the other for moving them). The

WIM provides both an aerial perspective of the entire scene, and allows the user to manipulate objects in the miniature version of the scene.

1996

– CHIMP (Chapel Hill Immersive Modeling Program) [61], see Figure A.13. The system uses two separate bats, one for each hand. User can perform a unimanual operation for translations and rotations, and a bimanual symmetric movement for scaling. Uses action-at-a-distance for remote selection and interaction with objects, look-at menus, constrained object manipulation, flying, worlds-in-miniature, interactive numbers.

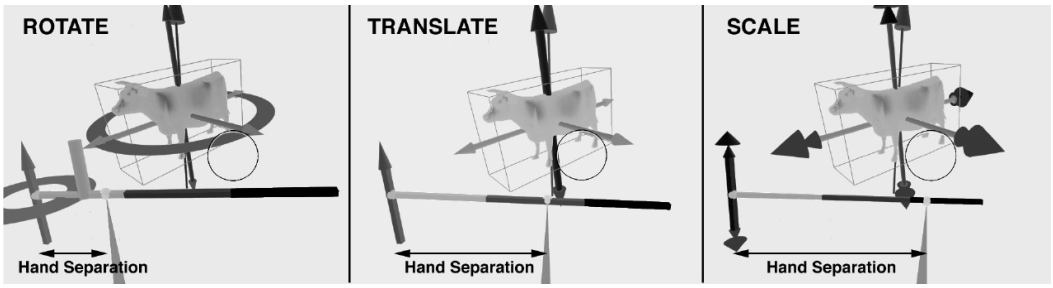


Figure A.13: CHIMP by Mine: Two-handed mode selection

– Go-go technique [15] for non-linear mapping for direct manipulation in VEs. The technique interactively "grows" the user's arm in order to reach remote objects which are to be manipulated.

– [62] Zeleznik's SKETCH system. Tries to bridge the gap between hand sketches and computer-based modeling programs. Uses stroke gestures to generate, move and rotate 3D solids.

1997

– [17] Evaluates techniques for grabbing and manipulating remote objects in virtual environments. Techniques: arm-extension, ray-casting, world-in-miniature (WIM), scaling the user, scaling the entire environment, go-go technique, hybrid techniques. Conclusions: grabbing and manipulation should be considered to be separate issues.

– Mine et al address [63] the lack of haptic feedback in VEs, see Figure A.14. To compensate, authors propose exploiting the only real object user has while in a VE — his own body. Thus the sense of proprioception. Three forms of body-relative interaction: direct manipulation, physical mnemonics, gestural actions. Select, grab, manipulate, release. Pull-down

menus, hand-held widgets, head-butt zoom, look-at menus, two-handed flying, throw-over-the-shoulder deletion, virtual object docking. Hand-held tablet as a real surface to work on.

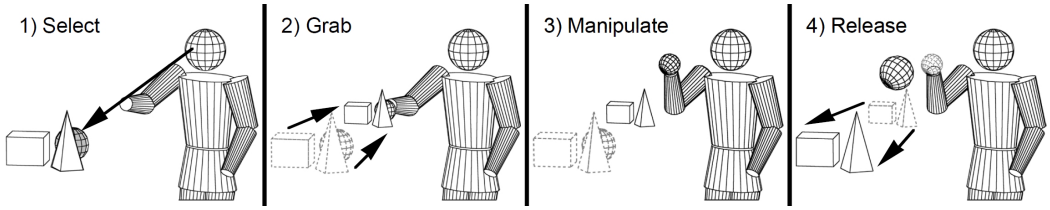


Figure A.14: Mine suggests proprioception as a way to address lack of haptic feedback

– [64] Two-handed direct manipulation on a "Responsive Workbench" [2] (Figures A.15 and A.16), a tabletop stereo display. Fakespace's PINCH gloves, stylus providing single dinstinguished point of action. Polhemus 6-d.o.f. tracker attached to stereo shutter glasses for head tracking. Four basic navigational tasks identified: 1) user slides, lifts up and pushes down the model; 2) user rotates the model around one of principal axes defined by tabletop or model; 3) user zooms in or out; 4) user changes his position relative to table thus relative to model, by walking around the table or by moving the head closer to/away from model. Devices, manipulators, tools. Tools can be *unimanual* (one-handed grab, panning, cutting plane, opacity, temperature, particle, streamline), *bimanual symmetric* (symmetric scale, slide-and-turn, turntable, grab-and-twirl, grab-and-carry), and *bimanual asymmetric* (grab-and-scale, trackball, zoom, free rotation, axis rotation, heuristic rotation, pinch rotation, constrained translation).

Figure A.15: Responsive Workbench: stereo video projected on mirrors below the desk (left), and persons observing a 3D house model displayed in stereo (right)

Figure A.16: Responsive Workbench: two-handed operation of zooming in

1998

– [65] Reviews the usability of various 6-d.o.f. input devices. Performance measures: speed, accuracy, ease of learning, fatigue, coordination, device persistence and acquisition. Mices modified for 6 d.o.f., the Bat, the Cricket, the MITS glove, Fingerball, Spaceball, SpaceMaster, Space Mouse, Elastic General-purpose Grip (EGG), Multi-d.o.f. armatures. Conclusion: none of the the existing devices fulfills all aspects of usability requirement for 3D manipulation; however, many insights into the characteristics and pros and cons of various designs; selection of various types of devices for different tasks (speed and short learning — free moving devices; fatigue, control trajectory quality and coordination — isometric or elastic rate control devices;)

– [66] "BUILD-IT" system. AR system, "Natural User Interface". Tangible, graspable control objects. To select an object, user puts a small "brick" (that is, interaction handler) at the object's position on the table. The object can then be rotated, translated and fixed by manipulating its associated brick. Multi-brick and multi-user interaction.

– "ErgoDesk" [67], see A.17. Interaction at ActiveDesk, a rear-projected drafting table-size display, similar to Responsive Workbench [2]. User creates 3D geometry using 2D lightpen-based input. Two-handed operation (user performs camera operations using a 3D tracker in his non-dominant hand). Speech input. Users had difficulty in creating 3D models. Weak modeling functionality. Many deficiencies in the hardware (display blurriness, tethered lightpen, noisy input from lightpen, difficult drawing in 3D).

Figure A.17: ErgoDesk by Forsberg et al

– In [68], one- and two-handed gestures (deform, grasp, point, scale, rotation) are used to model and manipulate 3D objects, using data gloves. See Figure A.18.

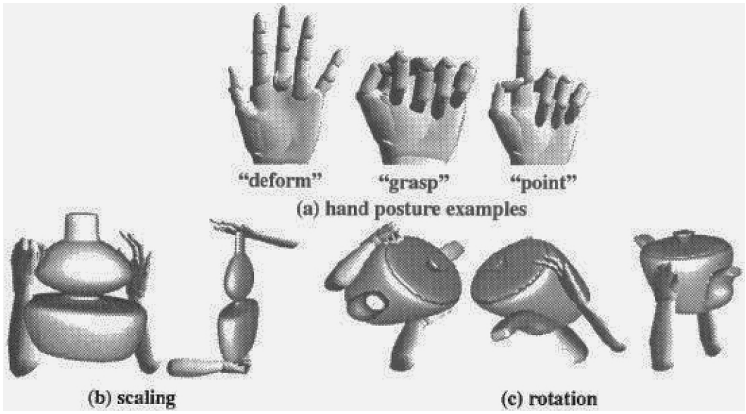


Figure A.18: Some manipulation gestures by Nishino et al

1999

– [69], [70] "Surface Drawing"[1] (Figures A.19 and A.20), a system for drawing organic 3D shapes, intended for artists. Wired glasses. Wired dataglove. Hand gestures. Users construct 3D shapes through "repeated marking". Hand marks 3D space in a semi-immersive environment (Responsive Workbench). Shapes created thus "float" in space above Re-

[1] schkolne.com/sdraw

sponsive Workbench. Tangible tools for edition and manipulation. Tongs[2] to move and scale 3D models. Magnet tool for free spatial hand motion. Magnet tool for drawing and editing (for example, bending) of 3D objects.

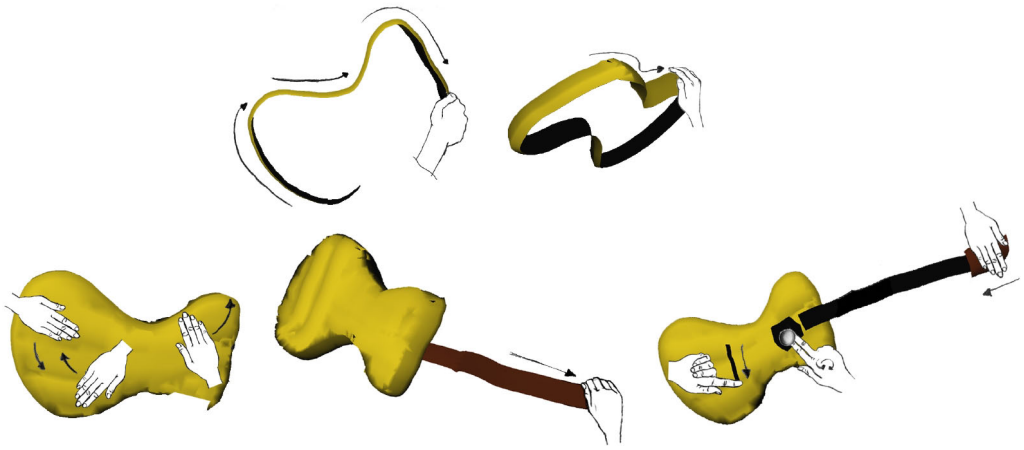


Figure A.19: "Surface drawing" by Schkolne et al: modeling a guitar in five steps

Figure A.20: "Surface drawing" by Schkolne et al: hand motions create 3D shapes which "float" over the Responsive Workbench

[2]A tong is a device for taking hold of objects; usually has two hinged legs with handles above and pointed hooks below.

A.4
2000-2008

2000

– [71] Perceptive Workbench by Leibe et al. Objects are recognized and tracked when placed on the display surface. Uses vision-based methods for interaction. Can identify 3D hand position, pointing direction, and arm gestures, which enhance selection, manipulation, and navigation tasks. Similar to Responsive Workbench however it uses infrared light instead.

2003

– [72] "RoomPlanner". Works on tabletop displays (MERL DiamondTouch used). Eight hand gestures defined. Tapping, dragging, flicking, catching, freeform rotation and scaling, tool palette manipulation and selection, parameter adjustment widget, flat hand, vertical hand, horizontal hand, tilted horizontal hand, two vertical hands, two corner-shaped hands.

2004

– FingARtips [73] by Buchmann et al. The technique tracks hand gestures by using image processing software and finger- and hand-based fiducial markers. The approach allows users to interact with virtual content using natural hand gestures.

– In [74], a sketching system prototype, utilizing gesture recognition and data gloves, was developed. Gestures grab, scale, and drop have been implemented (Figures A.21 and A.22).

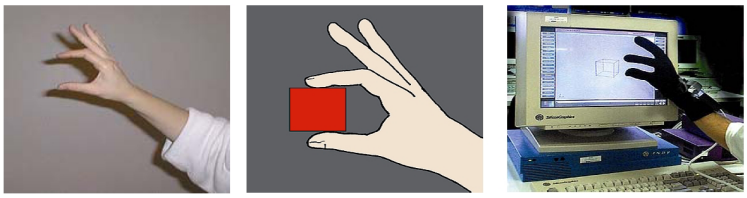


Figure A.21: Operation GRAB in Pratini's system

– In [75], vision-based gesture recognition utilizing white fingertip markers and so-called "black light" is used in order to manipulate 3D virtual objects in front of a large back-projection screen with two projectors for passive stereo (the user wears polarized glasses).

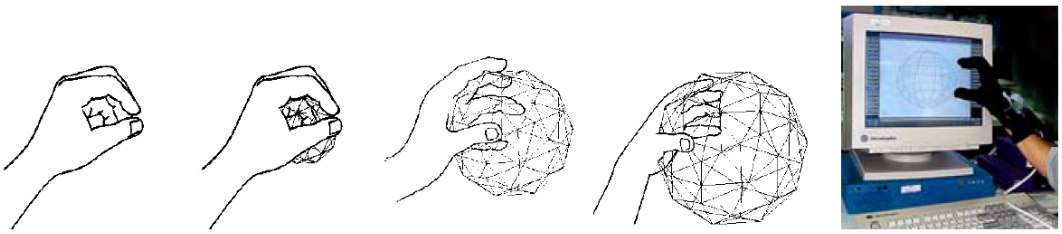


Figure A.22: Operation SCALE implemented as opening/closing the fist in Pratini's system

– In [76], a system for 3D translation and deformation using black gloves with five colors for each finger, stereo vision and passive stereo rendering is reported.

2005

– [77], [78] Gives a classification of 3D widgets, including 3D menus[3]. Especially suited for desktop 3D systems; classification is made according to interaction purpose. Four main classes of widgets:

1. widgets for direct 3D object interaction:
   - *object selection* (direct selection, occlusion selection, distance selection) and
   - *geometric manipulation* (linear transformation, non-linear transformation, high-level object manipulation).
2. widgets for 3D scene manipulation,
3. widgets for exploration and visualization, and
4. widgets for system/application control.

– In [79] unmarked hand gestures are being used for human-computer interaction. The prototype applications learns the background's characteristics in order to segment the hand, and detects and tracks fingertips for state switching.

[3]Online 3D-widget classification site: www.3d-components.org

2007

– [80] An approach for direct manipulation of 3D scenes (Figure A.23), based on visual, non-contact hand tracking and gesture recognition was presented. The system supports translation, rotation and scaling operations. The tracking cameras are located below the interaction volume. Six d.o.f. input is provided using both hands; the system does not require the user to wear any marker or any other kind of device.

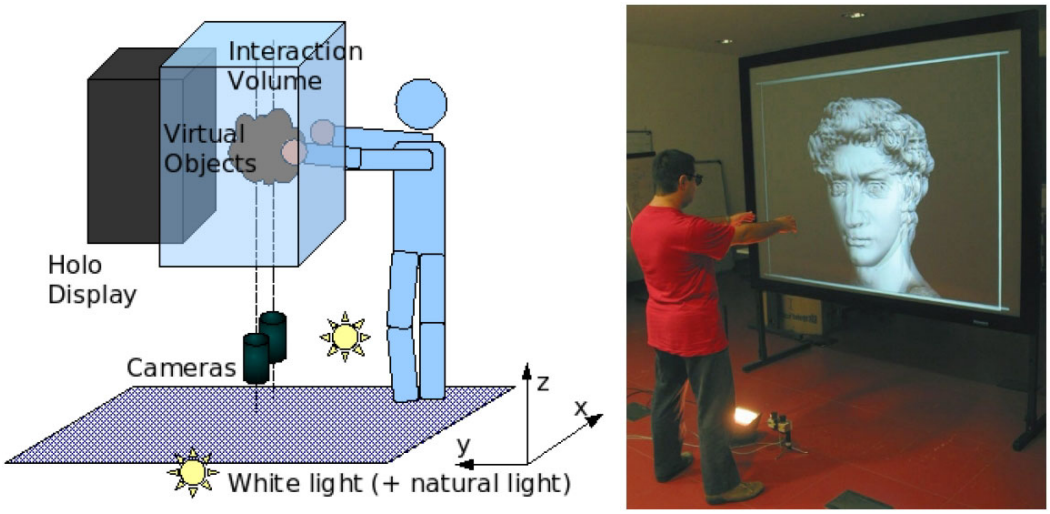


Figure A.23: The setup by Bettio et al. The user stands in front of a large stereo display, and manipulates the model using optically tracked hands.

B
Viola-Jones detection method

The Viola-Jones detection method [29] is a multi-stage detection method that has quickly found wide adoption in the computer vision community, due to its high speed of detection, and high detection rates. Compared to the best previously known detection methods [81] [82] [83] [84] [85], the Viola-Jones method is significantly faster (around fifteen times [29]) while achieving a comparable accuracy. There are four crucial features which distinguish this method:

- **Haar-like features** — Viola-Jones method classifies images based on the values of the so-called Haar-like features, which are simple features based on rectangles. (They are called "Haar-like" due to their similarity with the coefficients in the Haar wavelet transform.)
- **Integral image** — this is a novel data structure used in the pre-processing step of this algorithm, which allows the subsequent phases to run very quickly.
- **AdaBoost-based learning** — the learning part of the Viola-Jones method is based on AdaBoost [30], which combines a relatively small number of weak classifiers into a strong classifier.
- **Cascading strong classifiers** — this part of the Viola-Jones method combines strong classifiers into a "cascade" which discard regions of no interest quickly, thus leaving more processing times for regions that likely contain objects of interest.

B.1
Haar-like features

Haar-like features are prominent local aspects of an image which can be calculated very efficiently.

Let's take a look at Figure B.1, which depicts the extended Viola-Jones method [86]. Suppose we are dealing with a gray-level image I of $W \times H$ pixels. As we will see in Section B.2, there is a very fast way to compute the sum of all the pixels contained in either the upright rectangle, or rectangle inclined at 45°. A rectangle r, either the upright or inclined one, can be defined as:

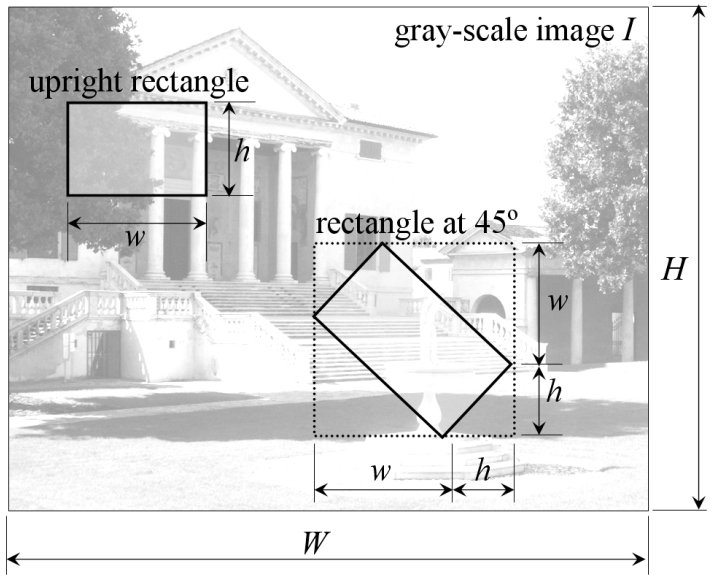


Figure B.1: Two types of rectangles used in the extended Viola-Jones method: 1) upright rectangle, and 2) rectangle inclined at 45°. We compute the sum of all gray-level intensities in rectangle r using function $\texttt{sum}(r)$.

$$r = (x, y, w, h, \alpha) \tag{B-1}$$

where

$$0 \leq x < x + w \leq W, \qquad 0 \leq y < y + h \leq H$$

$$x, y \geq 0, \qquad w, h > 0$$

$$\alpha = 0^\circ \text{ or } \ 45^\circ$$

The set Φ of all possible Haar-like features ϕ can then be defined as:

$$\Phi = \left\{ \phi \ \mid \ \phi = \sum_{i \in \{1, \ ..., \ N\}} \omega_i \cdot \texttt{sum}(r_i) \right\} \tag{B-2}$$

where N is an arbitrary number of rectangles chosen, r_i are parametrizations of those rectangles (see Equation (B-1)), $\omega_i \in \mathbb{R}$ are weights, and $\texttt{sum}(r_i)$ is the function that sums all the intensity values of all the pixels contained in rectangle r_i.

The problem with set (B-2) is that is infinitely large, therefore we reduce it to the following set:

$$\Phi = \left\{ \phi \ \mid \ \phi = \omega_1 \cdot \texttt{sum}(r_1) + \omega_2 \cdot \texttt{sum}(r_2), \quad \omega_1 = -1, \ \ \omega_2 = \frac{\texttt{area}(r_1)}{\texttt{area}(r_2)} \right\} \tag{B-3}$$

Thus in this newly defined set (B-3) of features we restrict N to 2, and constrain weights ω_1, ω_2 so that they have opposite signs and are used to compensate for the difference in area size between rectangles r_1, r_2.

We can now define the following set of 14 "template" or "prototype" features (Figure B.2), which will allow us to obtain real features (those that belong to set Φ in Equation (B-3)) by scaling and translating:

- Four edge features — two upright, two inclined
- Eight line features — four upright, four inclined
- Two center-surround features — one upright, one inclined

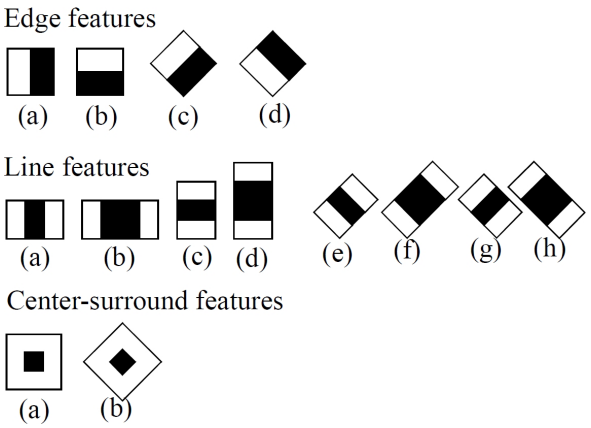


Figure B.2: Fourteen feature prototypes (templates) used in the extended Viola-Jones method

Let now $k = \lfloor W/w \rfloor$ and $l = \lfloor H/h \rfloor$. For seven upright features shown in Figure B.2, by scaling and translating we can generate a total of

$$kl\left(W+1-w\frac{k+1}{2}\right)\left(H+1-h\frac{l+1}{2}\right)$$

features, while for the remaining seven features inclined at 45° the total is

$$kl\left(W+1-z\frac{k+1}{2}\right)\left(H+1-z\frac{l+1}{2}\right), \qquad z=w+h$$

Note that line features can be calculated using two rectangles only, first rectangle r_1 encompassing the black *and* white rectangle, and second rectangle r_2 encompassing the black rectangle. For example (Figure B.3), line feature (a) with top left corner located at $(5,3)$ and dimensions 6×2 pixels can be written as:

$$\phi = -\texttt{sum}(5,3,6,2,0^\circ) + \frac{12}{4}\texttt{sum}(7,3,2,2,0^\circ)$$

which represents the combination of one big, encompassing 6×2 white rectangle r_1, and one smaller 2×2 black rectangle r_2 located in the middle of r_1.

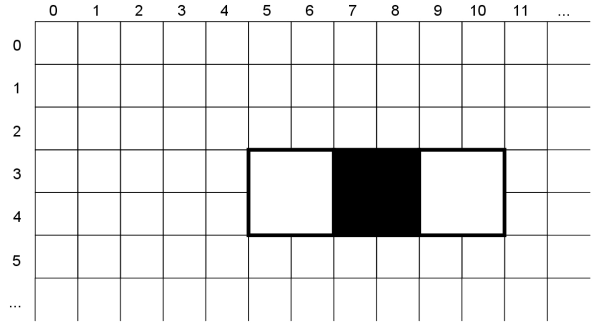


Figure B.3: Example: computing a 6×2-pixel "line feature" (see Figure B.2, feature (a) in the second row) whose top left corner is located at pixel $(5, 3)$

B.2 Integral images

Integral images are useful because, once computed, they enable the Viola-Jones method to subsequently compute features in constant time, i.e. in $O(1)$.

Let I be an $W \times H$ gray-level image. We define *integral image* $I_{\int}$ to be an image of same dimensions, whose value at pixel (x, y) is defined by:

$$I_{\int} = \sum_{u \leq x,\ v \leq y} I(u, v) \tag{B-4}$$

Intuitively, pixel $I_{\int}(x, y)$ contains the sum of all gray-level intensities for pixels that are to the left and up (relative to pixel (x, y)) in the original image I.

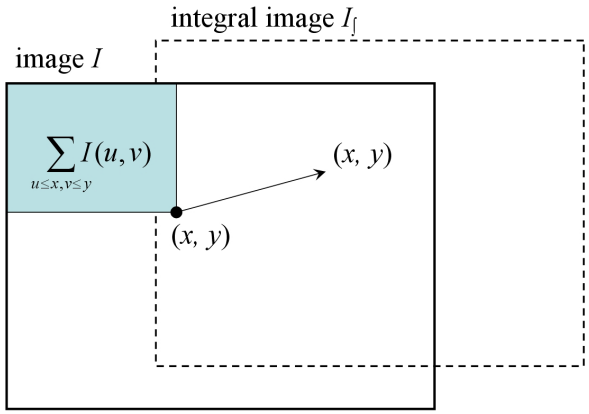


Figure B.4: The value of pixel (x, y) of the integral image $I_{\int}$ is equal to the sum of all pixels left and up from (x, y) in image I

We can use the following two recurrent relations to compute integral image $I_{\int}$ in just one pass over the original image I:

$$s(x, y) = x(x, y - 1) + I(x, y), \qquad s(x, -1) = 0$$

$$I_{\int}(x, y) = I_{\int}(x - 1, y) + s(x, y), \qquad I_{\int}(-1, y) = 0$$

Here $s(x, y)$ is the function that sums up row values in a cumulative fashion.

Integral images have the following beneficial properties:

- To compute the sum of any rectangle (sub-area) within the image I, just four array look-ups are needed.
- Therefore, the difference between two rectangles can be computed in just eight array lookups.
- Since two-rectangle features shown in Figure B.2 involve two adjacent rectangles, obviously just six array lookups are needed.
- Similarly, any three-rectangle features demands just eight array lookups.

B.3 AdaBoost-based learning

AdaBoost can be defined as "a general method for improving the accuracy of any given learning algorithm" [30]. As a special case, "any" learning algorithm could mean a learning algorithm that guesses the right answer just a little bit above 50%, i.e. just a little bit better than pure chance.

The AdaBoost algorithm:

- **GIVEN**: training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ where $x_i \in X$ ("instance space") and $y_i \in \{-1, +1\}$ ("set of labels"). In our context, instances $\{x_1, x_1, \ldots, x_m\}$ are $k \times k$-pixel images (for example, $k = 25$) containing ($y_i = +1$) or not containing ($y_i = -1$) human hand.
- **GOAL**: to output a final hypothesis $H(x)$ about the correct label for all $x \in X$
- **ALGORITHM**:
  - Initialize $D_1(i) = 1/m, \quad i \in \{1, \ldots, m\}$
  - For $t = 1, \ldots, T$:
    1. Train weak learner using current distribution D_t
    2. Get weak hypothesis $h_t: X \to \{-1, +1\}$ from the weak learner, so that error
    $$\epsilon_t = \sum_{i:\ h_t(x_i) \neq y_i} D_t(i)$$
    is low with respect to D_t
    3. Choose factor
    $$\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon}$$
    Factor α_t measures importance given to h_t.

4. Update D_{t+1}:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}$$

or equivalently

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t \, y_i \, h_t(x_i)}}{Z_t}$$

where Z_t is the normalization factor (chosen so that D_{t+1} is a distribution). This step increases (decreases) the weight of correctly (incorrectly) classified training example $i \equiv (x_i, y_i)$.

– Output the final hypothesis $H(x)$:

$$H(x) \colon X \to \{-1, +1\}$$

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right), \qquad x \in X \tag{B-5}$$

This final hypothesis ("strong classifier") $H(x)$ can be considered a weighted majority vote of T weak hypotheses, and factor α_t can be considered the weight attributed to the weak hypothesis h_t.

B.4 Cascading strong classifiers

In practice, a strong classifier (see Eq. B-5) can achieve any desired accuracy, however the speed is dissatisfying. Because of this strong classifiers are chained into the so-called "attentional cascade", or just "cascade", in order to achieve high frame rates. In such a chain, all strong classifiers are trained to detect approximately all objects *and* to reject a certain percentage of subwindows that do not contain the object.

For example:

– the first strong classifier in the cascade could be made of just two features, reject 50% of non-hand subwindows and detect hands correctly in 99.999% of all subwindows.

– the second strong classifier could consist of five features, reject 80% and detect correctly in 99.0% cases.

– the next six strong classifiers could consist of 25 features, reject 90% and detect correctly in 98.0% cases.

– ... and so on.

Taking now these classifiers, and chaining them into a series, we would obtain a cascade consisting of eight strong classifiers. The point in building a cascade is that a cascade significantly reduces processing times: the first strong classifiers reject many of subwindows that do not contain the object, while at the same time detecting correctly almost 100% of all the subwindows containing the object. All the subwindows that "passed through" the first strong classifier now have to be processed by the second classifier, which rejects even more subwindows, and so on. A subwindow must pass through all the classifiers in order to be classified as "positive".

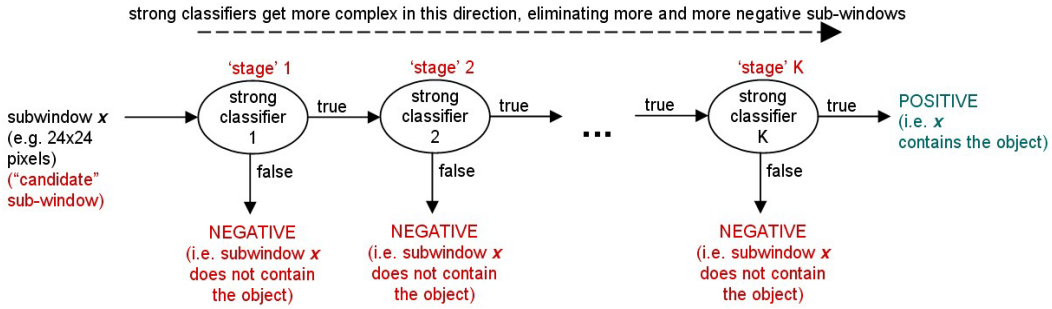


Figure B.5: Cascade of strong classifiers using Haar-like features

C
KLT features

As was already mentioned in Section 5.7 on page 50, *features* are properties of textured surfaces that allow us to "latch" onto them, see Figure 5.9 on page 50. By "latching" onto these features, we can thus effectively "latch" onto the object being tracked, therefore tracking the object.

The mathematical details behind KLT features [23] [24] [25] will now be given. Let $I(x, y, t)$ be "gray-level image sequence" functions defined on a sequence of $M \times N$ arrays at time moments $0, 1, 2, \ldots, L$, i.e.:

$$I : \{0, 1, 2, \ldots, N-1\} \times \{0, 1, 2, \ldots, M-1\} \times \{0, 1, 2, \ldots, L\} \longrightarrow [0, 1]$$

where N is the width of the image, M height, and L the time instant for the last image in the sequence. Let

$$I(x, y, t) = c, \ c \in [0, 1]$$

where c is a gray level between 0 (black) and 1 (white).

Let now W be a window in an image I, with dimensions $M' \times N'$, with the upper left corner located at (x', y'). Then we can restrict function I to the window W, thus obtaining function I_W:

$$I_W : W \longrightarrow [0, 1]$$

We are interested in tracking objects visible in the input image stream. Put differently, there exist certain patterns in the input image sequence which can be expressed formally like this:

$$I(x, y, t + \tau) = I(x - \xi, y - \eta, t) \tag{C-1}$$

Intuitively, Equation C-1 says that, having the current image $I(x-\xi, y-\eta, t)$, we can compute the next image (at time $t + \tau$) by moving all the pixels from the image $I(x - \xi, y - \eta, t)$ by a displacement vector $\vec{d} = (\xi, \eta)$.

Let now define $J(\vec{x}) = I(x, y, t+\tau)$ and $I(\vec{x}-\vec{d}) = I(x-\xi, y-\eta, t)$. Note that we omitted time parameter t for brevity (by definition, image J follows

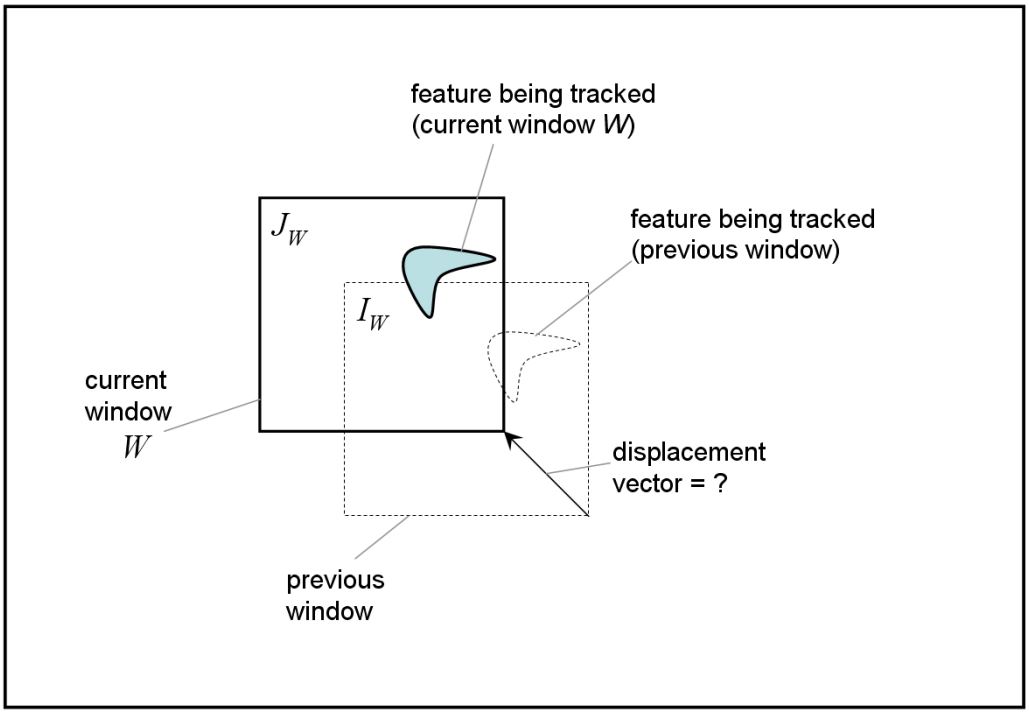


Figure C.1: Illustration of tracking based on KLT features. Window W is the current window, for example a rectangle of 10×10 pixels. J_W is the restriction of I on the current window W. I_W is the restriction of I on the previous window. What is being searched for, is the displacement vector $\vec{d}$, which enables us to position window W correctly in the current image.

I). The we can rewrite Equation C-1 as

$$J(\vec{x}) = I(\vec{x} - \vec{d}) + n(\vec{x}) \tag{C-2}$$

where $n(\vec{x})$ represents noise present in $J(\vec{x})$. The desired displacement vector $\vec{d}$ is then computed minimizing the following area integral over W:

$$\epsilon = \int_W \Big(I(\vec{x} - \vec{d}) - J(\vec{x})\Big)^2 w(\vec{x})\, d\vec{x} \tag{C-3}$$

Function $w(\vec{x})$ is the weighting function, which can be set to a desired function, for example to a constant function ($w(\vec{x}) = 1$) or to the Gaussian — depends on the application.

The question now is how to solve Equation C-3 for $\vec{d}$ so that:

$$\epsilon \longrightarrow \min$$

Note that when $\vec{d}$ is small, we can develop I into its Taylor series:

$$I(\vec{x} - \vec{d}) = I(\vec{x}) - \vec{g} \cdot \vec{d} + \ldots$$

... where $\vec{g}$ is a constant vector. We keep just the first two terms, so $I(\vec{x}-\vec{d}) = I(\vec{x}) - \vec{g}\cdot\vec{d}$, therefore equation C-3 becomes

$$\begin{aligned}\epsilon &= \int_W \Big(I(\vec{x}-\vec{d}) - J(\vec{x})\Big)^2 w(\vec{x})\,d\vec{x} = \\ &= \int_W \Big(I(\vec{x}) - \vec{g}\cdot\vec{d} - J(\vec{x})\Big)^2 w(\vec{x})\,d\vec{x} = \\ &= \int_W \Big(h(\vec{x}) - \vec{g}\cdot\vec{d}\Big)^2 w(\vec{x})\,d\vec{x} \end{aligned} \tag{C-4}$$

where $h(\vec{x}) = I(\vec{x}) - J(\vec{x})$. Equation C-4 can now be solved in the closed form, because ϵ is now a quadratic function. To find the minimum for ϵ, we now differentiate Equation C-4 relative to $\vec{d}$ and set the resulting expression to zero:

$$\int_W \Big(h(\vec{x}) - \vec{g}\cdot\vec{d}\Big)\,\vec{g}\;w(\vec{x})\,dA = 0$$

We can now replace $(\vec{g}\cdot\vec{d})\vec{g}$ by $(\vec{g}\cdot\vec{g}^{\tau})\,\vec{d}$. Since $\vec{d}$ can be considered constant for all pixels in W, we now obtain

$$\int_W h(\vec{x})\;\vec{g}\;w(\vec{x})\,dA = \left(\int_W (\vec{g}\cdot\vec{g}^{\tau})\;\;w(\vec{x})\,dA\right)\cdot\vec{d}$$

or simply switching the sides

$$\left(\int_W (\vec{g}\cdot\vec{g}^{\tau})\;\;w(\vec{x})\,dA\right)\cdot\vec{d} = \int_W h(\vec{x})\;\vec{g}\;w(\vec{x})\,dA$$

The previous equation can now be rewritten as

$$G\vec{d} = \vec{e} \tag{C-5}$$

where

$$G = \int_W (\vec{g}\cdot\vec{g}^{\tau})\;\;w(\vec{x})\,dA$$

and

$$\vec{e} = \int_W h(\vec{x})\;\vec{g}\;w(\vec{x})\,dA$$

Thus to find $\vec{d}$, we must, for each pair of consecutive frames, first compute G, then $\vec{e}$, and then using the linear system C-5 we can compute $\vec{d}$.

D
Hartley-Sturm triangulation method

The Hartley-Sturm triangulation method [20] is an algorithm that, under the assumption of Gaussian noise present in image point measurements, gives a *provably optimal* global solution to the triangulation problem.

In further text we assume that we know fundamental matrix F exactly, and that any error is due either to 1) the digitalization process on the CMOS/CCD chip of the camera, or 2) to the feature extraction process. It is assumed that these errors follow Gaussian distribution.

Let:

$\vec{u} \leftrightarrow \vec{u}'$ — an noisy, incorrect measured pair of correspondent features for the left and right camera respectively. This pair does not satisfy $\vec{u}'^{\tau} F \vec{u}$.

$\hat{\vec{u}} \leftrightarrow \hat{\vec{u}}'$ — a correct pair of correspondent features for the left and right camera respectively. Point $\hat{\vec{u}}$ should in general lie close to point $\vec{u}$, and $\hat{\vec{u}}'$ to $\vec{u}'$. Points $\hat{\vec{u}}, \hat{\vec{u}}'$ satisfy $\hat{\vec{u}}'^{\tau} F \hat{\vec{u}}$.

The goal therefore is to find points $\hat{\vec{u}}, \hat{\vec{u}}'$ that minimize the function

$$\left(d(\vec{u}, \hat{\vec{u}})\right)^2 + \left(d(\vec{u}', \hat{\vec{u}}')\right)^2 \tag{D-1}$$

where $d(\vec{u}, \vec{v})$ represents Euclidean distance between 2D points $\vec{u}, \vec{v}$. This minimization task is equivalent to finding real number t for which the following cost function attains minimum:

$$s(t) = \frac{t^2}{1 + f^2 t^2} + \frac{(ct + d)^2}{(at + b)^2 + f'^2 (ct + d)^2} \tag{D-2}$$

The algorithm (see [87], page 318):

- **GOAL** — compute 2D points $\hat{\vec{u}}, \hat{\vec{u}}'$ that minimize Eq. D-1. Given are measured 2D correspondent points $\vec{u}, \vec{u}'$, and fundamental matrix F.
- **ALGORITHM**:

  1. define transformation matrices

$$T = \begin{pmatrix} 1 & & -u \\ & 1 & -v \\ & & 1 \end{pmatrix} \text{ and } T' = \begin{pmatrix} 1 & & -u' \\ & 1 & -v' \\ & & 1 \end{pmatrix}$$

2. replace F by $T'^{-\tau}FT^{-1}$
3. compute epipoles $\vec{e} = (e_1, e_2, e_3)^\tau$ and $\vec{e}' = (e'_1, e'_2, e'_3)^\tau$ so that $\vec{e}'^\tau F = 0$ and $F\vec{e} = 0$. Normalize $\vec{e}, \vec{e}'$.
4. form matrices

$$R = \begin{pmatrix} e_1 & e_2 & \\ -e_2 & e_1 & \\ & & 1 \end{pmatrix} \text{ and } R' = \begin{pmatrix} e'_1 & e'_2 & \\ -e'_2 & e'_1 & \\ & & 1 \end{pmatrix}$$

5. replace F by $R'FR^\tau$
6. set $f = e_3, f' = e'_3, a = F_{22}, b = F_{23}, c = F_{32}, d = F_{33}$
7. form 6-degree polynomial

$$g(t) = t\left((at+b)^2 + f'^2(ct+d)\right)^2 - (ad-bc)(1+f^2t^2)^2(at+b)(ct+d)$$

8. solve $g(t)$ in order to obtain 6 roots
9. evaluate cost function $s(t)$ (see Eq. D-2) at the real part of each of the six roots. Also, find $\lim_{t\to\infty} s(t)$. Select t_{min} that gives the smallest value for $s(t)$.
10. evaluate two lines $\vec{l} = (tf, 1, -t)$ and $\vec{l}' = F(0, t, 1)^\tau = (-f'(ct + d), at + b, ct + d)^\tau$ at t_{min}, and find $\hat{\vec{u}}$, $\hat{\vec{u}}'$ as the closest points on these lines to the origin.
11. replace $\hat{\vec{u}}$ by $T^{-1}R^\tau\hat{\vec{u}}$ and $\hat{\vec{u}}'$ by $T'^{-1}R'^\tau\hat{\vec{u}}'$
12. compute the requested 3D point $\vec{X}$ by any other method, for example by mid-point method.