Chapter 44

Networked Robots: from Telerobotics to Cloud Robotics

Summary

As of 2013, almost all robots have access to computer networks that offer extensive computing, memory, and other resources that can dramatically improve performance. The underlying enabling framework is the focus of this chapter: networked robots. Networked robots trace their origin to telerobots, or remotely controlled robots. Telerobots are widely used to explore undersea terrains and outer space, to defuse bombs, and to clean up hazardous waste. Until 1994, telerobots were accessible only to trained and trusted experts through dedicated communication channels. This chapter will describe relevant network technology, the history of networked robots as it evolves from teleoperation to cloud robotics, properties of networked robots, how to build a networked robot, example systems. Later in the chapter we focus on the recent progress on cloud robotics, and topics for future research.

44.1 Overview and Background

As illustrated in Fig. 44.1, the field of networked robots locates at the intersection between two exciting fields: robotics and networking. Similarly, teleoperation (Chapter 43) and multiple mobile robot systems (Chapter 51) also find their overlaps in the intersection. The primary concerns of the teleoperation are stability and time delay. Multiple mobile robot systems concerns coordination and planning of autonomous robots and sensors communicating over local networks. *The subfield of Networked Robots focuses on the robot system architectures, interfaces, hardware, software, and applications that use networks (primarily the Internet / Cloud).*

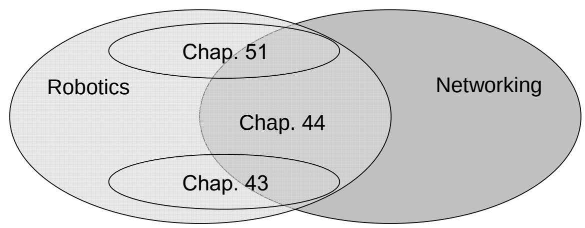

Figure 44.1: Relationship between the subjects of networked robots (Chapter 44, the present chapter), teleoperation (Chapter 43), and multiple mobile robot systems (Chapter 51)

By 2012, several hundred networked robots have been developed and put online for public use. Many papers have been published describing these systems and a book on this subject by *Goldberg* and *Siegwart* is available [1]. Updated information about new research and an archive/survey of networked robots is available on the website of the IEEE technical committee on networked robots, which fosters research in this area (IEEE Technical Committee on Networked Robots `http://tab.ieee-ras.org/`).

The rest of the chapter is organized as follows:

we first review the history and related work in Section 44.2. In Section 44.3, we review network and communication technology to provide necessary background for the following two main Sections 44.4 and 44.5. Section 44.4 focus on traditional networked robots while Section 44.5 summarize the new development in cloud robotics. Section 44.6, we conclude the chapter with recent applications and future directions.

44.2 A Brief History

44.2.1 Networked Teleoperation

Networked robots have their root in teleoperation systems, which started as remotely controlled devices. However, thanks to the recent evolution of the Internet and wireless networks, networked robots quickly expand their scope from the traditional master-slave teleopereation relationship to an integration of robots, human, agents, off-board sensors, databases, and clouds over the globe. To review the history of networked robots, we trace back to the root: remotely controlled devices.

Like many technologies, remotely controlled devices were first imagined in science fiction. In 1898, Nicola Tesla [2] demonstrated a radio-controlled boat in New York's Madison Square Garden. The first major experiments in teleoperation were motivated by the need to handle radioactive materials in the 1940s. Goertz demonstrated one of the first bilateral simulators in the 1950's at the Argonne National Laboratory [3]. Remotely operated mechanisms have been designed for use in inhospitable environments such as undersea [4] and space exploration [5]. At General Electric, Mosher [6] developed a two-arm teleoperator with video cameras. Prosthetic hands were also applied to teleoperation [7]. More recently, teleoperation is being considered for medical diagnosis [8], manufacturing [9] and micromanipulation [10]. See Chapter 43 and the book from Sheridan [11] for excellent reviews on teleoperation and telerobotics research.

The concept of hypertext (linked references) was proposed by Vannevar Bush in 1945 and was made possible by subsequent developments in computing and networking. In the early 1990's, Berners-Lee introduced the Hypertext Transmission Protocol (HTTP). A group of students led by Marc Andreessen developed an open source version of the first graphical user interface, the "Mosaic" browser, and put it online in 1993. The first networked camera, the predecessor of today's "webcam", went online in November 1993 [12]

Approximately nine months later, the first networked telerobot went online. The "Mercury Project" combined an IBM industrial robot arm with a digital camera and used the robot's air nozzle to allow remote users to excavate for buried artifacts in a sandbox [13, 14]. Working independently, a team led by K. Taylor and J. Trevelyan at the University of Western Australia demonstrated a remotely controlled six-axis telerobot in September 1994 [15, 16]. These early projects pioneered a new field of networked telerobots. See [17–25] for other examples.

Networked telerobots are a special case of "supervisory control" telerobots, as proposed by Sheridan and his colleagues [11]. Under supervisory control, a local computer plays an active role in closing the feedback loop. Most networked robotics are type (c) supervisory control systems (see Fig 44.2).

Although a majority of networked telerobotic systems consist of a single human operator and a single robot [26–33], Chong et al. [34] propose a useful taxonomy: Single Operator Single Robot (SOSR), Single Operator Multiple Robot (SOMR) [35, 36], Multiple Operator Single Robot (MOSR), and Multiple Operator Multiple Robot (MOMR) [37, 38]. These frameworks greatly extend system architecture of networked robots. In fact, human operators can often be replaced with autonomous agents, off-board sensors, expert systems, and programmed logics, as demonstrated by Xu et al. [39] and Sanders et al. [40]. The extended networked connectivity also allows us to employ techniques such as crowd sourcing and collaborative control for demanding applications such as nature observation and environment monitoring [41, 42]. Hence networked telerobots fully evolute into networked robots: an integration of robots, humans [43], computing power, off-board sensing, and databases over the Internet.

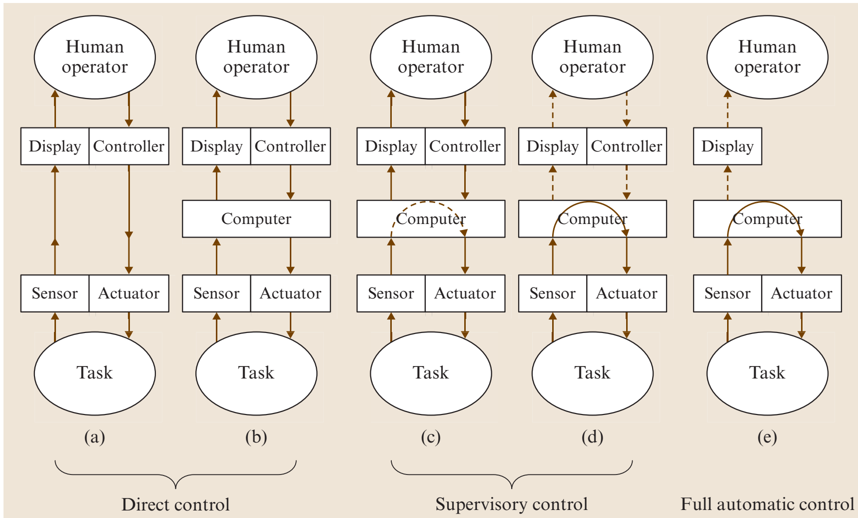

Figure 44.2: A spectrum of teleoperation control modes adapted from Sheridan's text [11]. We label them a-e, in order of increasing robot autonomy. At the far left would be a mechanical linkage where the human directly operates the robot from another room through sliding mechanical bars, and on far right would be a system where the human role is limited to observation/monitoring. In c-e, the dashed lines indicated that communication may be intermittent.

The last 18 years (1994-2012) witnessed the extensive development in networked robots. New systems, new experiments and new applications go well beyond traditional fields such as defense, space, and nuclear material handing [11] that motivated teleoperation in early 1950s. As the Internet introduces universal access to every corner of life, the impact of networked robots becomes broader and deeper in modern society. Recent applications range from education, industry, commercial, health care, geology, environmental monitoring, to entertainment and arts.

Networked robots provide a new medium for people to interact with remote environment. A networked robot can provide more interactivity beyond what a normal videoconferencing system. The physical robot not only represents the remote person but also transmits multi-modal feedback to the person, which is often referred as "telepresence" in literature [29]. Paulos and Canny's Personal ROving Presence (PRoP) robot [44], Jouppi and Thomas' Surrogate robot [29], Takayama et al.'s Texai [45], and Lazewatsky and Smart's inexpensive platform [46] are representative work.

Networked robots have great potential for education and training. In fact, one of the earliest networked telerobot systems [47] originates from the idea of a remote laboratory. Networked telerobots provide universal access to the general public, who may have little to no knowledge of robots, with opportunities to understand, learn, and operate robots, which were expensive scientific equipment limited to universities and large corporate laboratories before. Built on networked telerobots, online remote laboratories [48, 49] greatly improves distance learning by providing an interactive experience. For example, teleoperated telescopes help students to understand astronomy [50]. Teleoperated microscope [51] helps student to observe micro-organisms. The Tele-Actor project [52] allows a group of students to remotely control a human teleactor to visit environments that are normally not accessible to them such as clean-room environments for semi-conductor manufactory facility and DNA analysis laboratories.

44.2.2 Cloud Robotics and Automation

Recent development of cloud computing provide new means and platform for networked robots. In 2010, James Kuffner at Google introduced the term "Cloud Robotics" [53] to describe a new approach to robotics that takes advantage of the Internet as a resource for massively parallel computation and real-time sharing of vast data resources. The Google autonomous driving project exemplifies this approach: the system indexes maps and images that are collected and updated by satellite, Streetview, and crowdsourcing from the network to facilitate accurate localization. Another example is Kiva Systems new approach to warehouse automation and logistics using large numbers of mobile platforms to move pallets using a local network to coordinate planforms and update tracking data. These are just two new projects that build on resources from the Cloud. Steve Cousins of Willow Garage aptly summarized the idea: "No robot is an island." Cloud Robotics recognizes the wide availability of networking, incorporates elements of open-source, open-access, and crowdsourcing to greatly extend earlier concepts of "Online Robots" [54] and "Networked Robots" [55, 56].

The Cloud has been used as a metaphor for the Internet since the inception of the World Wide Web in the early 1990s. As of 2012, researchers are pursuing a number of cloud robotics and automation projects [57] [58] . New resources range from software architectures [59] [60] [61] [62] to computing resources [63]. The RoboEarth project [64] aims to develop "a World Wide Web for robots: a giant network and database repository where robots can share information and learn from each other about their behavior and their environment" [65]. Cloud Robotics and Automation is related to concepts of the "Internet of Things" [66] and the "Industrial Internet," which envision how RFID and inexpensive processors can be incorporated into a vast array of objects from inventory items to household appliances to allow them to communicate and share information.

44.3 Communications and Networking

Below is a short review of relevant terminologies and technologies on networking. For details, see the texts by [67].

A communication network includes three elements: *links*, *routers/switchers*, and *hosts*. Links refer to the physical medium that carry bits from one place to another. Examples of links include copper or fiber-optic cables and wireless (radio frequency or infrared) channels. Switches and routers are hubs that direct digital information between links. Hosts are communication end points such as browsers, computers, and robots.

Networks can be based in one physical area (local-area network, or LAN), or distributed over wide distances (wide-area network, or WAN). Access control is a fundamental problem in networking. Among a variety of methods, the *ethernet* protocol is the most popular. Ethernet provides a broadcast-capable multiaccess LAN. It adopts a carrier-sense multiple-access (CSMA) strategy to address the multiple-access problem. Defined in the IEEE 802.x standard, CSMA allows each host to send information over the link at any time. Therefore, collisions may happen between two or more simultaneous transmission requests. Collisions can be detected either by directly sensing the voltage in the case of wired networks, which is referred to as collision detection (CSMA/CD), or by checking the time-out of an anticipated acknowledgement in wireless networks, which is referred to as collision avoidance (CSMA/CA). If a collision is detected, both/all senders randomly back off a short period of time before retransmitting. CSMA has a number of important properties: (1) it is a completely decentralized approach, (2) it does not need clock synchronization over the entire network, and (3) it is very easy to implement. However, the disadvantages of CSMA are: (1) the efficiency of the network is not very high and (2) the transmission delay can change drastically.

As mentioned previously, LANs are interconnected with each other via routers/switchers. The information transmitted is in packet format. A packet is a string of bits and usually contains the source address, the destination address, content bits, and a checksum. Routers/switchers distribute packets according to their routing table. Routers/switchers have no memory of packets, which ensures scalability of the network. Packets are usually routed according to a first-in first-out (FIFO) rule, which is independent of the application. The packet formats and addresses are independent of the host technology, which ensures extensibility. This routing mechanism is referred to as packet switching in the networking literature. It is quite different from a traditional telephone network, which is referred to as circuit switching. A telephone network is designed to guarantee a dedicated circuit between a sender and a receiver once a phone call is established. The dedicated circuitry ensures communication quality. However, it requires a large number of circuits to ensure the quality of service (QoS), which leads to poor utilization of the overall network. A packet-switching network cannot guarantee dedicated bandwidth for each individual pair of transmissions, but it improves overall resource utilization. The Internet, which is the most popular communication media and the infrastructure of networked telerobots, is a packet-switching network.

44.3.1 The Internet

The creation of the Internet can be traced back to US Department of Defense's (DoD) APRA NET network in the 1960s. There are two features of the APRA NET network that enabled the successful evolution of the Internet. One feature is the ability for information (packets) to be rerouted around failures. Originally this was designed to ensure communication in the event of a nuclear war. Interestingly, this dynamic routing capability also allows the topology of the Internet to grow easily. The second important feature is the ability for heterogeneous networks to interconnect with one another. Heterogeneous networks, such as X.25, G.701, ethernet, can all connect to the Internet as long as they can implement the Internet protocol (IP). The IP is media, operating system (OS), and data rate independent. This flexible design allows a variety of applications and hosts to connect to the Internet as long as they can generate and understand IP.

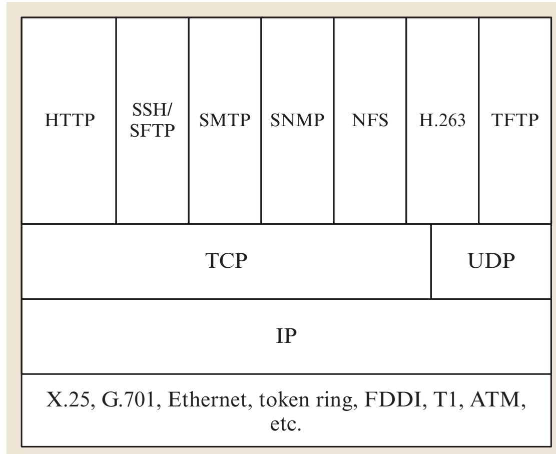

Figure 44.3: A four-layer model of internet protocols (after [67])

Figure 44.3 illustrates a four-layer model of the protocols used in the Internet. On the top of the IP, we have two primary transport layer protocols: the transmission control protocol (TCP) and the user data protocol (UDP). TCP is an end-to-end transmission control protocol. It manages packet ordering, error control, rate control, and flow control based on packet round-trip time. TCP guarantees the arrival of each packet. However, excessive retransmission of TCP in a congested network may introduce undesirable time delays in a networked telerobotic system. UDP behaves differently; it is a broadcast-capable protocol and does not have a retransmission mechanism. Users must take care of error control and rate control themselves. UDP has a lot less overhead compared to TCP. UDP packets are transmitted at the sender's preset rate and the rate is changed based on the congestion of a network. UDP has great potential, but it is often blocked by firewalls because of a lack of a rate control mechanism. It is also worth mentioning that the widely accepted term TCP/IP refers to the family of protocols that build on IP, TCP, and UDP.

In the application layer of the Internet protocols, the HTTP is one of the most important protocols. HTTP is the protocol for the World Wide Web (WWW). It allows the sharing of multimedia information among heterogeneous hosts and OSs including text, image, audio, and video. The protocol has significantly contributed to the boom of the Internet. It also changes the traditional client/server (C/S) communication architecture to a browser/server (B/S) architecture. A typical configuration of the B/S architecture consists of a web server and clients with web browsers. The web server projects the contents in hypertext markup language (HTML) format or its variants, which is transmitted over the Internet using HTTP. User inputs can be acquired using the common gateway interface (CGI) or other variants. The B/S architecture is the most accessible because no specialized software is needed at the client end.

44.3.2 Wired Communication Links

Even during peak usage, the network backbones of the Internet often run at less than 30% of their overall capacity. The average backbone utilization is around 15 – 20%. The primary speed limitation for the Internet is the *last mile*, the link between clients and their local Internet service providers (ISP).

Table 44.1 lists typical bit rates for different connection types. It is interesting to note the asymmetric speeds in many cases, where upstream bit rate (from the client to the Internet), are far slower than downstream bit rates (from the Internet to the client). These asymmetries introduce complexity into the network model for teleoperation. Since the speed difference between the slowest modem link and the fastest Internet II node is over 10,000, designers of a networked telerobotic system should anticipate a large variance of communication speeds.

44.3.3 Wireless Links

Table 44.2 compares the speed, band, and range of wireless standards as of 2012. Increasing bit rate and communication range requires increasing power. The amount of radio frequency (RF) transmission power required over a distance d is proportional to d^k, where $2 \leq k \leq 4$ depending on the antenna type. In Table 44.2, Bluetooth and Zigbee are typical low-power transmission standards that are good for short distances. HSPA+ and LTE are commercially marketed as the 4G cellphone network.

By providing high-speed connectivity at low cost, WiFi is the most popular wireless standard in 2012. Its range is approximate 100 m line of sight and the WiFi wireless network usually consists of small-scale interconnected access points. The coverage range usually limits these networks to an office building, home, and other indoor environments. WiFi is a good option for indoor mobile robots and human operators. If the robot needs to navigate in the outdoor environment, the 3G or 4G cellphone network can provide the best coverage available. Although obvious overlap exists among wireless standards in coverage and bandwidth, there are two import issues that have not been covered by Table 44.2. One is mobility. We know that, if an RF source or receiver is moving, the corresponding Doppler effect causes a frequency shift, which could cause problems in communication. WiFi is not designed for fast-moving hosts. 3G HSPA cellphone allows the host to move at a vehicle speed under 120 km/h. However, LTE allows the host to move at a speed of 350 km/h or 500 km/h, which even works for high-speed trains.

Long range wireless links often suffer from latency problem, which may drastically decreases system performer as discussed in Chapter 43. One may notice that we did not list satellite wireless in Table 44.2 because the long latency (0.5–1.7 secs) and high price makes it difficult to be useful for robots. The large antenna size and high power consumption rate also limits its usage in mobile robots. In fact, the best option for long range wireless is LTE. LTE is designed with a transmission latency of less that 4 ms whereas 3G HSPA cellphone networks have a variable latency of 10 – 500 ms.

44.3.4 Video and Audio Transmission Standards

In networked robots systems, the representation of the remote environment is often needed to be delivered to online users in video and audio format.

Types	Bits per second
Dialup Modem (V.92)	Up to 56 K
Integrated Services Digital Network (ISDN)	64 – 160 K for BRI, Up to 2048 K for PRI
High Data Rate Digital Subscriber Line (HDSL)	Up to 2.3 M duplex on two twisted-pair lines
Assymetric Digital Subscriber Line (ADSL)	1.544 – 24.0 M downstream, 0.5 – 3.3 M upstream
Cable modem	2 – 400 M downstream, 0.4 – 108 M upstream
Fiber to the home (FTTH)	0.005 – 1 G downstream, 0.002 – 1 G upstream
Direct Internet II node	1.0 – 10.0 G

Table 44.1: *Last-mile* Internet speed by wired connection type. If not specified, the downstream transmission and the upstream transmission share the same bandwidth

Types	Bit rate (bps)	Band (Hz)	Range (m)
Zigbee (802.15.4)	20 – 250 K	868 – 915 M/2.4 G	50
Bluetooth	732 K–3.0 M	2.4 G	100
3G HSPA	400 K–14.0 M	$\leq$ 3.5 G	N/A
HSPA+	5.76 M–44.0 M	$\leq$ 3.5 G	N/A
LTE	10 M–300 M	$\leq$ 3.5 G	N/A
WiFi (802.11a,b,g,n)	11 – 600 M	2.4 G/5 G	100

Table 44.2: Survey of wireless technologies in terms of bit rate and range

To deliver video and audio over the Internet, raw video and audio data from camera optical sensor and microphone must be compressed according to different video and audio compression standards to fit in the limited network bandwidth. Due to lack of bandwidth and computing power to encode streaming video, most early systems only transmit periodic snapshots of the remote scene in JPEG format at limited frame rate, i.e. 1–2 frames per second or less. Audio was rarely considered in the early system design. The rudimentary video delivery methods in the early system were mostly implemented using HTML and Javascript to reload the JPEG periodically.

Today, the expansion of HTML standards allow web browsers to employ plug-ins as the client end of streaming video. HTML5 even natively supports video decoding. Therefore, the server end of recent systems often employs streaming server software, such as Adobe Flash Media Encoder, Apple Quick Time Streaming Server, Oracle Java Media Framework, Helix Media Delivery Platform, Microsoft DirectX, SkypeKit, etc. to encode and deliver video. These streaming video sever packages often provide easy-to-use SDK to facilitate system integration.

It is worth noting that these different software packages are just different implementations of video/audio streaming protocols. Not every protocol is suitable for networked robots. Some protocols are designed to deliver video on demand while others are designed for live streaming for videoconferencing purposes. Networked robots use real time video as feedback information, which imposes strict requirements in latency and bandwidth similar to those of videoconferencing. One way latency of more than 150 ms can significantly degrade telepresence and hence the performance of the human operator.

Latency is often caused by bandwidth and video encoding/decoding time. Since audio data amount is negligible when comparing to that video data. We will focus the discussion on video compression standards. There is always a tradeoff between framerate and resolution for a given bandwidth. There is also a tradeoff between compression ratio and computation time for a given CPU. The computation time includes both CPU time and data-buffering time at both client and server ends. Video encoding is a very computationally intensive task. A long computation period introduces latency and significantly impair the system performance. It is possible to use hardware to cut down the computation time but not the data-buffering time, which are controlled by the video encoder.

There are many standards and protocols available but most of them are just variations of MJPEG, MPEG2, H.263 and MPEG4/AVC/H.264. We compare those standards in Table 44.3.4. Note that the comparison is qualitative and may not be the most accurate due to the fact that each video encoding standard has many parameters that affect the overall buffering time. From networked robot point of view, the buffering time determines the latency and the framerate determines the responsiveness of the system. An ideal videostream should have both high framerate and low buffering time. But if both cannot be achieved at the same time, low latency is preferred. From Table 44.3.4, H.264/MPEG4-AVC clearly outperforms other competitors and is the most popular video compression method.

44.4 Properties of Networked Robots

Networked robots have the following properties

- The physical world is affected by a device that is locally controlled by a network *server*, which connects to the Internet to communicate with remote human users, databases, agents, and off-board sensors, which are referred to as *clients* of the system.
- Human decision making capability is often an integral part of the system. If so, humans often access the robot via web browsers, such as Internet Explorer or Firefox, or apps in mobile device. As of 2012, the standard protocol for network browsers is the hypertext transfer protocol (HTTP), a stateless transmission protocol.
- Most networked robots are continuously accessible (online), 24 hours a day, 7 days a week.
- Networks may be unreliable or have different speed for clients with different connections.
- Since hundreds of millions of people now have access to the Internet, mechanisms are needed to handle client authentication and contention. System security and privacy of users are important in the networked robots.
- Input and output for human users for networked robots are usually achieved with the standard computer screen, mouse, and keyboard.
- Clients may be inexperienced or malicious, so online tutorials and safeguards are generally required.
- Additional sensing, databases and computing resources may be available over the network.

44.4.1 Overall Structure

As defined by *Mason*, *Peshkin*, and others [68, 69], in *quasistatic* robot systems, accelerations and inertial forces are negligible compared to dissipative forces. In quasistatic robot systems, motions are often modeled as transitions between discrete atomic *configurations*.

We adopt a similar terminology for networked telerobots. In quasistatic telerobotics (QT), robot dynamics and stability are handled locally. After each atomic motion, a new state report is presented to the remote user, who sends back an atomic command. The atomic state describes the status of the robot and its corresponding environment. Atomic commands refer to human directives, which are desired robotic actions.

Several issues arise

- *State-command presentation*: How should state and available commands be presented to remote human operators using the two-dimensional (2-D) screen display?
- *Command execution/state generation*: How should commands be executed locally to ensure that the desired state is achieved and maintained by the robot?

Standards	Feasible Minimum Buffering Time (FMBT)	Framerate
MJPEG	zero (<10 msecs)	Low
MPEG2	variable (i.e. 50 msec – video length), 2–10 secs are common	Moderate
H.263+	<300 msecs	High
H.264/MPEG4-AVC	zero (<10 msecs)	Highest

Table 44.3: *A comparison of existing videostreaming standards for the same resolution under the same fixed bandwidth. FMBT represents buffering time settings that would not significantly decrease compression ratio or video quality*

- *Command coordination*: How should commands be resolved when there are multiple human operators and/or agents? How to synchronize and aggregate commands issued by users/agents with different network connectivity, background, responsiveness, error rate, etc. to achieve best possible system performance?

- *Virtual Fixture: Error prevention and state correction*: How should the system prevent the wrong commands that may lead the robot to collision or other undesirable states?

Before we detail these issues, let us walk through how to build a minimum networked robot system. A reader can follow the below example to build his/her owner networked robot system as well as understand challenges in the issues.

44.4.2 Building a Networked Robot System

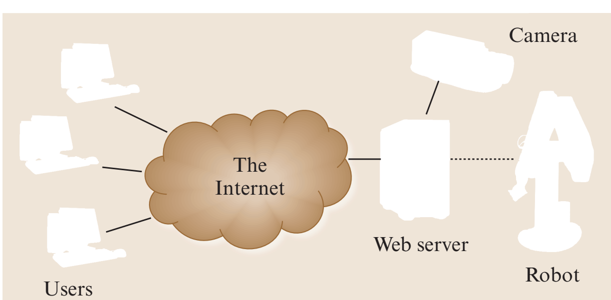

Figure 44.4: Typical system architecture for a networked telerobot

This minimal system is a networked telerobotic system which allows a group of users to access a robot via web browsers. As illustrated in Fig. 44.4, a typical or minimal networked telerobotic system typically includes three components:

- users: anyone with an Internet connection and a web browser or equivalent apps that understand HTTP.

- web server: a computer running a web server software

- robot: a robot manipulator, a mobile robot, or any device that can modify or affect its environment

Users access the system via their web browsers. Any web browser that is compatible with W3C's HTML standard can access a web server. In 2012, the most popular web browsers are Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Apple Safari, and Opera. New browsers and updated versions with new features are introduced periodically. All of these popular browsers issue the corresponding mobile appls to support mobile devices such as Apple iPads, Apple iPhones, and Google Andriod-based Tablets and smart phones.

A web server is a computer that responds to HTTP requests over the Internet. Depending upon the operating system of the web server, popular server software packages include Apache and Microsoft Internet Information Services (IIS). Most servers can be freely downloaded from the Internet.

To develop a networked telerobot, one needs a basic knowledge of developing, configuring, and maintain-

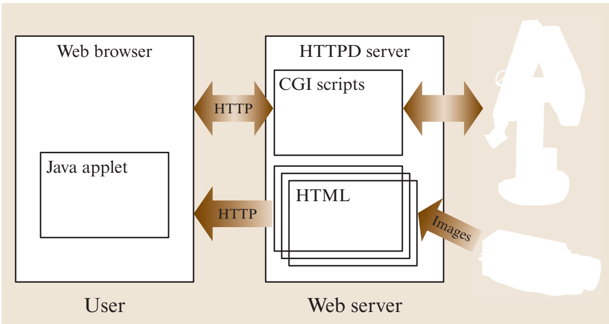

Figure 44.5: A sample software architecture of a networked telerobot

ing web servers. As illustrated in Fig. 44.5, the development requires knowledge of HTML and at least one local programming languages such as C, C#, CGI, Javascript, Perl, PHP, .Net, or Java.

It is important to consider compatibility with the variety of browsers. Although HTML is designed to be compatible with all browsers, there are exceptions. For example, Javascript, which is the embedded scripting language of web browsers, is not completely compatible between Internet Explorer and Firefox. One also needs to master the common HTML components such as forms that are used to accept user inputs, frames that are used to divide the interface into different functional regions, etc. An introduction to HTML can be found in [70].

User commands are usually processed by the web server using CGI, the common gateway interface. Most sophisticated methods such as PHP, Java Server Pages (JSP), and socket-based system programming can also be used. CGI is invoked by the HTTP server when the CGI script is referred in the Uniform Resource Locator (URL). The CGI program then interprets the inputs, which is often the next robot motion command, and sends commands to the robot via a local communication channel. CGI scripts can be written in almost any programming language. The most popular ones are Perl and C.

A simple networked telerobotic system can be constructed using only HTML and CGI. However, if the robot requires a sophisticated control interface, advanced plug-ins such as Java Applet, Silver Light, or Flash, is recommended. These plug-ins run inside the web browser on the client's computer. Information about these plug-ins can be found at home pages of Oracle, Microsoft, and Adobe, respectively. Java applet is highly recommended because it is the most widely supported by different browsers. Recently, the fast adoption of HTML5 also provide a new long term solution to solve the compatibility issue.

Most telerobotic systems also collect user data and robot data. Therefore, database design and data processing program are also needed. The most common used databases include MySQL and PostgresSQL. Both are open-source databases and support a variety of platforms and operation systems. Since a networked telerobotic system is online 24 hours a day, reliability is also an important consideration in system design. Website security is critical. Other common auxiliary developments include online documentation, online manual, and user feedback collection.

It is not difficult to expand this minimal networked telerobotic system into a full-fledged networked robot system. For example, some users can be replaced by agents that runs 24 hours a day and 7 days a week to monitor system states and co-perform tasks with humans or take over the system when nobody is online. These agents can be implemented using cloud computing. Such extensions are usually based on the need of the task.

44.4.3 State-Command Presentation

To generate a correct and high-quality command depends on how effectively the human operator understands the state feedback. The state-command presentation contains three subproblems: the 2-D representation of the true robot state (state display), the assistance provided by the interface to generate new commands (spatial reasoning), and the input mechanism.

State Displays Unlike traditional point-to-point teleoperation, where specialized training and equipment are available to operators, networked telerobots offer wide access to the general public. Designers cannot assume that operators have any prior experience with robots. As illustrated in Fig. 44.6, networked

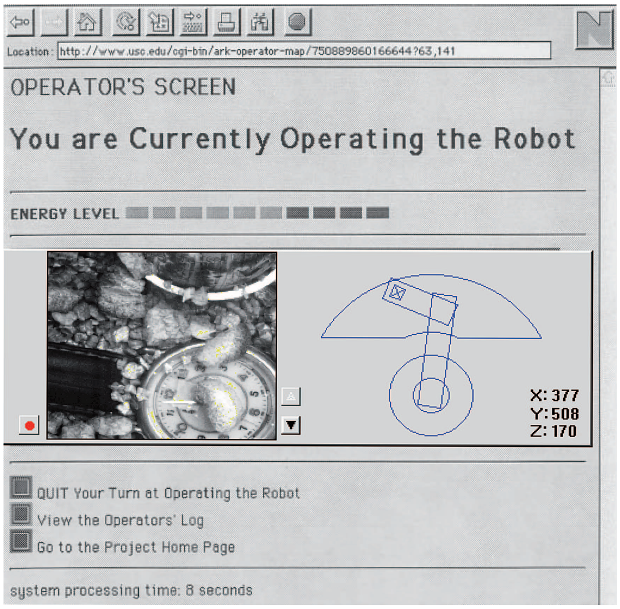

Figure 44.6: Browser's view of the first networked telerobot interface [71]. The schematic at *lower right* gives an overhead view of position of the four-axis robot arm (with the camera at the end marked with X), and the image at the *lower left* indicates the current view of the camera. The *small button marked with a dot at the left* directs a 1 s burst of compressed air into the sand below the camera. The Mercury Project was online from August 1994 to March 1995

telerobotic systems must display the robot state on a 2-D screen display.

The states of the teleoperated robot are often characterized in either world coordinates or robot joint configuration, which are either displayed in numerical format or through a graphical representation. Figure 44.6 lists robot XYZ coordinates on the interface and draws a simple 2-D projection to indicate joint configurations. Figure 44.7 illustrates another example of teleoperation interface that was developed by *Taylor* and *Trevelyan* [47]. In this interface, XYZ coordinates are presented in a sliding bar near the video window.

The state of the robot is usually displayed in a 2-D view as shown in Figs. 44.6 and 44.7. In some

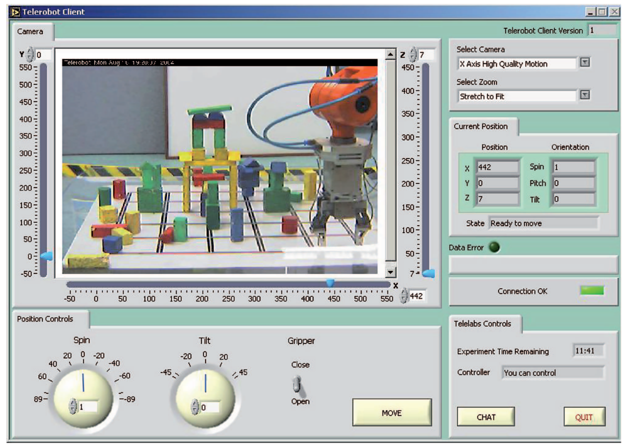

Figure 44.7: Browser interface to the Australian networked telerobot which was a six-axis arm that could pick up and move blocks [16]
.

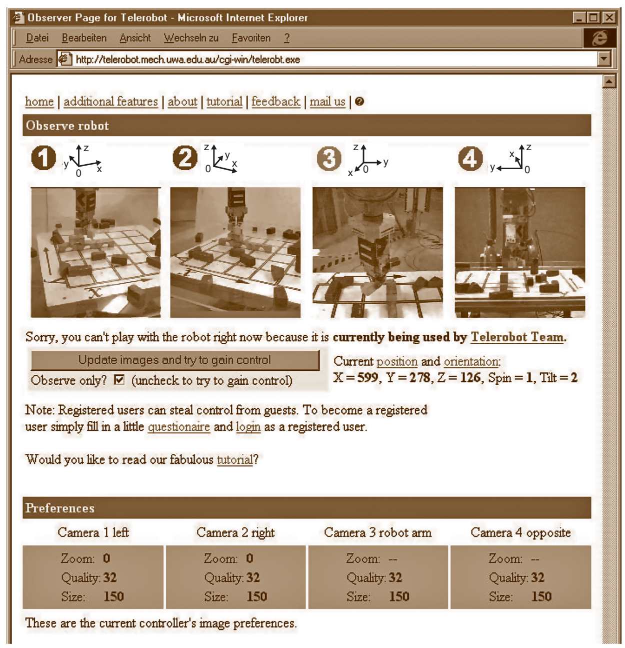

Figure 44.8: Use of a multicamera system for multi-viewpoint state feedback [72]

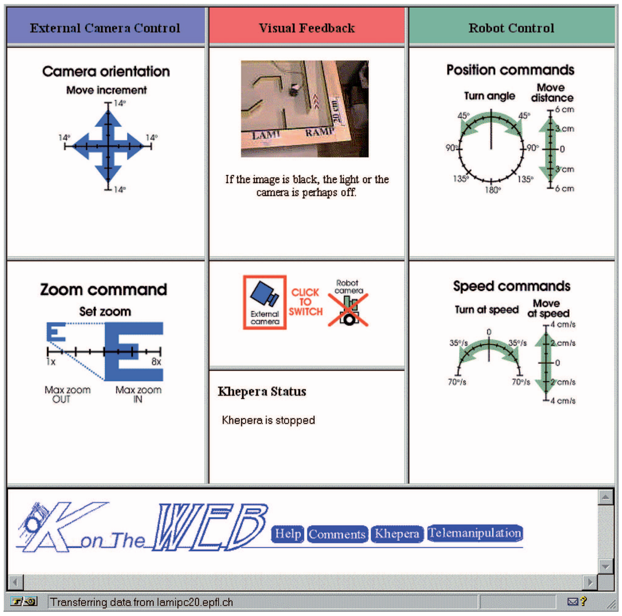

Figure 44.9: Camera control and mobile robot control in Patrick Saucy and Francesco Mondada's Khep on the web project

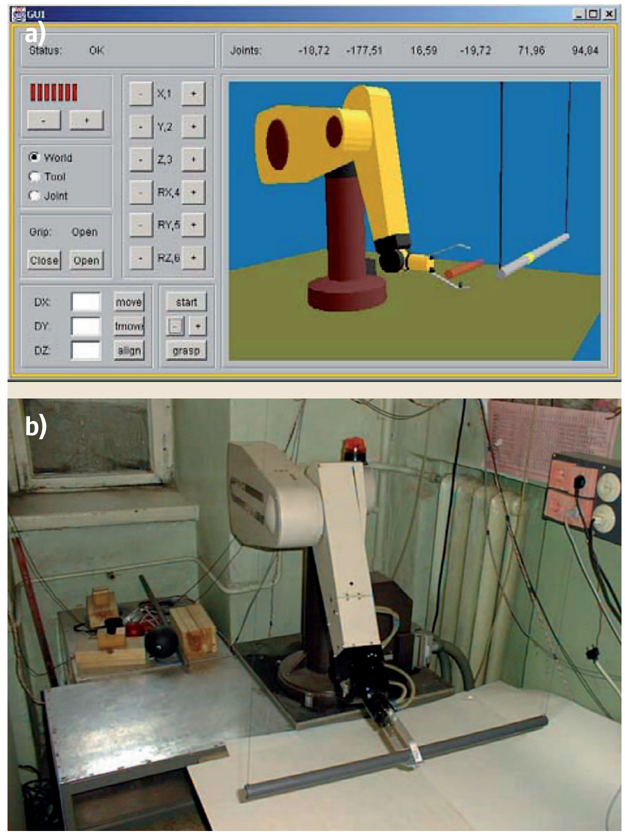

Figure 44.10: A web-based teleoperation system that allows a robot to capture a fast-moving rod [27] **(a)** User interface and **(b)** system setup

systems, multiple cameras can help the human operator to understand the spatial relationship between the robot and the objects in the surrounding environment. Figure 44.8 shows an example with four distinct camera views for a six-degree-of-freedom industrial robot.

Figure 44.9 demonstrate an interface with a pan–tilt–zoom robotic camera. The interface in Fig. 44.9 is designed for a mobile robot.

More sophisticated spatial reasoning can eliminate the need for humans to provide low-level control by automatically generating a sequence of commands after it receives task-level commands from the human operator. This is particularly important when the robotic system is highly dynamic and requires a very fast response. In this case, it is impossible to ask the human to generate intermediate steps in the robot control; for example, *Belousov* et al. adopt a shared autonomy model to direct a robot to capture a moving rod [27] as shown in Figure 44.10. *Fong* and *Thorpe* [73] summarize vehicle teleoperation systems that utilize these supervisory control techniques. *Su* et al. developed an incremental algorithm for better translation of the intention and motion of operators into remote robot action commands [32].

The fast development of sensing and display technology makes it possible to visualize robot and environment states in 3D displays or generate synthetic eco-centric views (a.k. a third person views). To achieve that, it often requires the robot is equipped with multiple cameras and laser range finders to quickly reconstruct the remote environment [74, 75]. Sometimes, the reconstructed sensory information can be superimposed on priorly known 3D information to form an augmented reality. This kind of display can drastically increase telepresence and performance.

Human Operator Input Most networked telerobotic systems only rely on mouse and keyboards for input. The design problem is what to click on in the interface. Given the fact that user commands can be quite different, we need to adopt an appropriate interface for inputs; for example, inputs could be Cartesian XYZ coordinates in world coordinate system or robot configurations in angular joint configurations.

For angular inputs, it is often suggested to use a round dial as a control interface, as illustrated in bottom left of Fig. 44.7 and the right-hand side of Fig. 44.9. For linear motion in Cartesian coordinate, arrows operated by either mouse clicks or the keyboard are often suggested. Position and speed control are often needed, as illustrated in Fig. 44.9. Speed control is usually controlled by mouse clicks on a linear progress bar for translation and a dial for rotation.

The most common control type is position control. The most straightforward way is to click on the video image directly. To implement the function, the software needs to translate the 2-D click inputs into three-dimensional (3-D) world coordinates. To simplify the problem, the system designer usually assumes that the clicked position is on a fixed plane; for example, a mouse click on the interface of Fig. 44.6 assumes the robot moves on the X–Y plane. The combination of a mouse click on the image can also allow abstract task-level command. The example in Fig. 44.12 uses mouse clicks to place votes on an image to generate a command that directs a robot to pick up a test agent at the task level.

44.4.4 Command Execution/State Generation

When a robot receives a command, it executes the command and a new state is generated and transmitted back to the human operator. However, commands may not arrive in time or may get lost in transmission. Also, because users are often inexperienced, their commands may contain errors. Over the limited communication channel, it is impossible to ask the human to control the manipulator directly. Computer vision, laser range finder, local intelligence, and augmented-reality-based displays [75] are required to assist the human operator.

Belousov and colleagues demonstrated a system that allowed a web user to capture a fast rod that is thrown at a robot manipulator [27]. The rod is on bifilar suspension, performing complicated oscillations. Belousov et al. designed a shared-autonomy control to implement the capture. First, an operator chooses the desired point for capture on the rod and the capture instant using a 3-D online virtual model of the robot and the rod. Then, the capturing operation is performed automatically using a motion prediction algorithm that is based on the rod's motion model and two orthogonal camera inputs, which perceive the rod's position locally in real time.

This *shared autonomy* approach is often required when the task execution require much faster response than the Internet can allow. Human commands have to remain at task level instead of directing the movements of every actuators. The root of this approach can be traced back to the "Tele-Autonomous" concept proposed by *Conway*, *Volz*, and *Walker* [76] in 1990. In the paper, two important notions including time clutch and position clutches are introduced to illustrate the shared autonomy approach. The time clutch disengages the time synchronization between the human operator and the robot. The human operator verifies his/her commands on a predictive display before sending a set of verified commands to remote robots. The robot can then optimize the intermediate trajectory proposed by the human operator and disengage the position correspondence, which is referred to as the position clutch. Recent work [77] use the similar idea to guide load-haul-dump vehicles in the underground mines by combining human inputs with tunnel following behavior.

44.4.5 Virtual Fixtures

Due to time delay, lack of background, and possible malicious behavior, human errors are inevitably introduced to system from time to time. Erroneous states may be generated from the incorrect commands. If unchecked, robots or objects in the environment may be damaged. Some times, users may have good intention but are not able to generate accurate commands to control the robot remotely. For example, it is hard to generate a set of commands to direct a mobile robot to move along the wall and maintain a distance of 1 meter to the wall at the same time.

Virtual fixtures are designed to cope with these challenges in teleoperation tasks. Proposed by Rosenberg [78], virtual fixtures are defined as an overlay of abstract sensory information on a robot workspace in order to improve the telepresence in a telemanipulation task. To further explain the definition, Rosenberg uses a ruler as an example. It is very difficult for a human to draw a straight line using bare hands. However, if a ruler, which is a physical fixture, is provided, then the task becomes easy. Similar to a physical fixture, a virtual fixture is designed to guide robot motion through some fictitious boundaries or force fields, such as virtual tubes or surface, generated according to sensory data. The virtual fixtures are often implemented using control laws [79, 80] based on a "virtual contact" model.

Virtual fixtures serve for two main purposes: avoiding operation mistakes and guide robots along the designable trajectories. This is also a type of shared autonomy that is similar to that in Section 44.4.4 where both the robot and the human share control in the system. Chapter 43 details the shared control scheme. It is worth noting that virtual fixtures should be visualized in the display to help operators understand the robot state to maintain situation awareness. This actually turns the display to *augmented reality* [81].

44.4.6 Collaborative Control and Crowd Sourcing

When more than one human is sharing control of the device, command coordination is needed. According to [82], multiple human operators can reduce the chance of errors, cope with malicious inputs, utilize operators' different expertise, and train new operators. In [83, 84], a collaboratively controlled networked robot is defined as a telerobot simultaneously controlled by many participants, where input from each participant is combined to generate a single control stream.

Figure 44.11: Spatial dynamic voting interface for the Tele-Actor system [52]: the spatial dynamic voting (SDV) interface as viewed by each user. In the remote environment, the Tele-Actor takes images with a digital camera, which are transmitted over the network and displayed to all participants with a relevant question. With a mouse click, each user places a color-coded marker (a *votel* or voting element) on the image. Users view the position of all votels and can change their votel positions based on the group's response. Votel positions are then processed to identify a *consensus region* in the voting image that is sent back to the Tele-Actor. In this manner, the group collaborates to guide the actions of the Tele-Actor

When group inputs are in the form of direction vectors, averaging can be used as an aggregation mechanism [85]. When decisions are distinct choices or at the abstract task level, voting is a better choice [52]. As illustrated in Fig. 44.11, *Goldberg* and *Song* develop the Tele-Actor system using spatial dynamic voting. The Tele-Actor is a human equipped with an audio/video device and controlled by a group of online users. Users indicate their intensions by positioning their votes on a 320×320 pixel voting image during the voting interval. Votes are collected at the server and used to determine the Tele-Actor's next action based on the most requested region on the voting image. (see `http://www.tele-actor.net`)

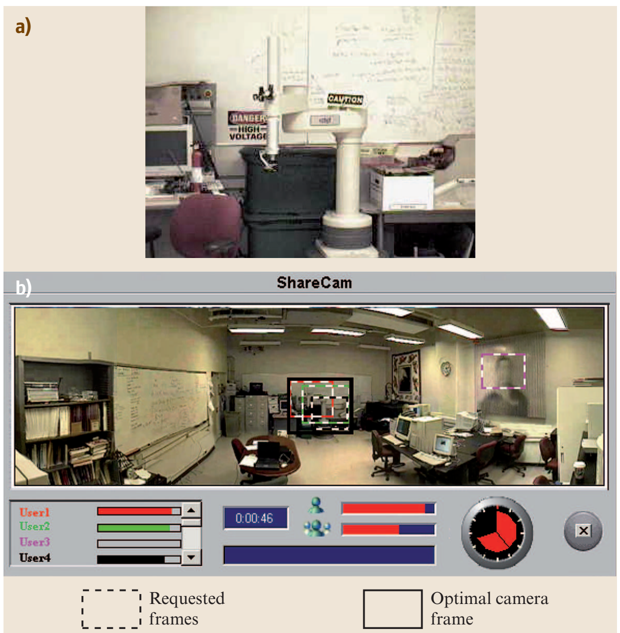

Figure 44.12: Frame selection interface [86]. The user interface includes two image windows. The lower window **(b)** displays a fixed panoramic image based on the camera's full workspace (reachable field of view). Each user requests a camera frame by positioning a *dashed rectangle* in **(b)**. Based on these requests, the algorithm computes an optimal camera frame (shown with a *solid rectangle*), moves the camera accordingly, and displays the resulting live streaming video image in the upper window **(a)**

Another approach to collaboratively control a networked robot is the employ a optimization framework. *Song* and *Goldberg* [86,87] developed a collaboratively controlled camera that allowed many clients to share control of its camera parameters, as illustrated in Fig. 44.12. Users indicate the area they want to view by drawing rectangles on a panoramic image. The algorithm computes an optimal camera frame with respect to the user satisfaction function, which is defined as the frame selection problem [88, 89].

Recent work by Xu et al. [39, 90] further the optimization framework to p-frames that allow multiple cameras to be controlled and coordinated whereas human inputs can also be replaced by autonomous agents and other sensory inputs. These developments have been applied to a recent project, the Collaborative Observatory for Nature Environments (CONE) project [91], which aims to design a networked robotic camera system to collect data from the wilderness for natural scientists.

One important issue in collaborative control is the disconnection between individual commands and the robot action, which may lead to loss of situation awareness, less participation, and eventual system failure. Inspired by engaging power in scoring systems in computer games, Goldberg et al. [92] design scoring mechanism for the collaborative control architecture by evaluating individual leadership level. The early results show great improvement in group performance. Furthermore, the recent development of social media, such as Blog and Twitter, can also be employed in the collaborative control to facilitate user interaction in real time, which can make the system more engaging and effective. The resulting new architecture can be viewed as a crowd sourcing [41,93] type approach to networked robots that combines human recognition and decision making capabilities to robot execution at a different scale and depth than a regular teleoperation system.

44.5 Cloud Robotics

As noted earlier, the term "Cloud Robotics" is increasingly common based on advances in what is now called "Cloud Computing". Cloud Robotics extends what were previously called "*Online Robots*" [54] and "*Networked Robots*" [55,56]. Cloud computing provides robots with vast resources in computation, memory, programming.

Here we review five ways that Cloud Robotics and

Automation can potentially improve robots and automation performance: 1) providing access to global libraries of images, maps, and object data, eventually annotated with geometry and mechanical properties, 2) massively-parallel computation on demand for demanding tasks like optimal motion planning and sample-based statistical modeling, 3) robot sharing of outcomes, trajectories, and dynamic control policies, 4) human sharing of "open-source" code, data, and designs for programming, experimentation, and hardware construction, and 5) on-demand human guidance ("call centers") for exception handling and error recovery. Updated information and links are available at: `http://goldberg.berkeley.edu/cloud-robotics/`

44.5.1 Big Data

The term "Big Data" describes data sets that are beyond the capabilities of standard relational database systems, which describes the growing library of images, maps, and many other forms of data relevant to robotics and automation on the Internet. One example is grasping, where online datasets can be consulted to determine appropriate grasps. The Columbia Grasp dataset [94] and the MIT KIT object dataset [95] are available online and have been widely used to evaluate grasping algorithms [96] [97] [98] [99].

Related work explores how computer vision can be used with Cloud resources to incrementally learn grasp strategies [100] [101] by matching sensor data against 3D CAD models in an online database. Examples of sensor data include 2D image features [102], 3D features [103], and 3D point clouds [104]. Google Goggles [105], a free network-based image recognition service for mobile devices, has been incorporated into a system for robot grasping [106] as illustrated in Figure 44.13.

Dalibard et al. attach "manuals" of manipulation tasks to objects [107]. The RoboEarch project stores data related to objects maps, and tasks, for applications ranging from object recognition to mobile navigation to grasping and manipulation (see Figure 44.15) [64].

As noted below, online datasets are effectively used to facilitate learning in computer vision. By leveraging Google's 3D warehouse, [108] reduced the need for manually labeled training data. Using community photo collections, [109] created an augmented reality application with processing in the cloud.

44.5.2 Cloud Computing

As of 2012, Cloud Computing services like Amazon's EC2 elastic computing engine provide massively-parallel computation on demand [110]. Examples include Amazon Web Services [111] Elastic Compute Cloud, known as EC2 [112], Google Compute Engine [113], Microsoft Azure [114]. These rovide a large pool of computing resources that can be rented by the public for short-term computing tasks. These services were originally used primarily by web application developers, but have increasingly been used in scientific and technical high performance computing (HPC) applications [115] [116] [117] [118].

Cloud computing is challenging when there are real-time constraints [119]; this is an active area of research. However there are many robotics applications that are not time sensitive such as decluttering a room or pre-computing grasp strategies.

There are many sources of uncertainty in robotics and automation [120]. Cloud computing allows massive sampling over error distributions and Monte Carlo sampling is "embarrassingly parallel"; recent research in fields as varied as medicine [121] and particle physics [122] have taken advantage of the cloud. Real-time video and image analysis can be performed in the Cloud [108] [123] [124]. Image processing in the cloud has been used for assistive technology for the visually impaired [125] and for senior citizens [126]. Cloud computing is ideal for sample-based statistical motion planning under uncertainty, where it can be used to explore many possible perturbations in object and environment pose, shape, and robot response to sensors and commands [127]. Cloud-based sampling is also being investigated for grasping objects with shape uncertainty [128] [129] (see Figure 44.14). A grasp planning algorithm accepts as input a nominal polygonal outline with Gaussian uncertainty around each vertex and the center of mass to compute a grasp quality metric based on a lower bound on the prob-

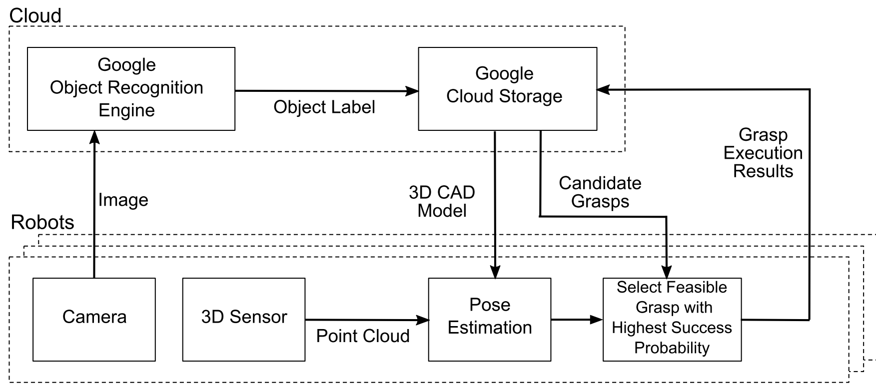

Figure 44.13: System Architecture for cloud-based object recognition for grasping. The robot captures an image of an object and sends via the network to the Google object recognition server. The server processes the image and returns data for a set of candidate objects, each with pre-computed grasping options. The robot compares the returned CAD models with the detected point cloud to refine identification and to perform pose estimation, and selects an appropriate grasp. After the grasp is executed, data on the outcome is used to update models in the cloud for future reference [106].

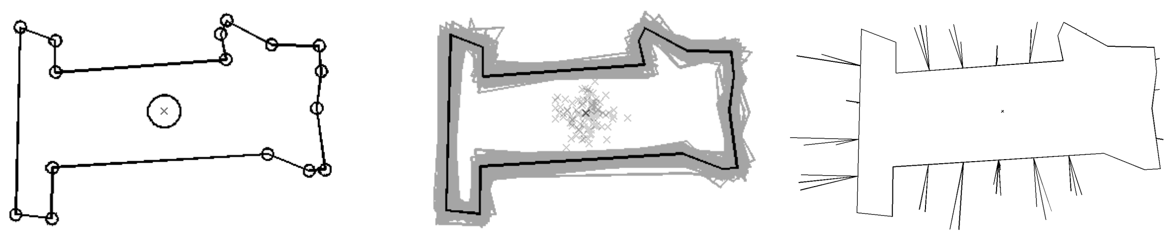

Figure 44.14: A cloud-based approach to geometric shape uncertainty for grasping [128] [129].

ability of achieving force closure.

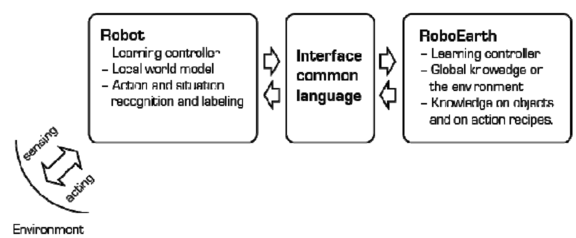

Figure 44.15: RoboEarth architecture [64].

44.5.3 Collective Robot Learning

The Cloud allows robots and automation systems to "share" data from physical trials in a variety of environments, for example initial and desired conditions, associated control policies and trajectories, and importantly: data on performance and outcomes. Such data is a rich source for robot learning.

One example is for path planning, where previously-generated paths are adapted to similar environments [130] and grasp stability of finger contacts can be learned from previous grasps on an object [97].

The MyRobots project [131] from RobotShop proposes a "social network" for robots: "In the same way humans benefit from socializing, collaborating and sharing, robots can benefit from those interactions too by sharing their sensor information giving insight on their perspective of their current state" [132].

44.5.4 Open-Source and Open-Access

The Cloud facilitates sharing by humans of designs for hardware, data, and code. The success of open-source software [133] [134] [135] is now widely accepted in the robotics and automation community. A primary example is ROS, the Robot Operating System, which provides libraries and tools to help software developers create robot applications [136] [137]. ROS has also been ported to Android devices [138]. ROS has become a standard akin to Linux and is now used by almost all robot developers in research and many in industry.

Additionally, many simulation libraries for robotics are now open-source, which allows students and researchers to rapidly set up and adapt new systems and share the resulting software. Open-source simulation libraries include Bullet [139], a physics simulator originally used for video games, OpenRAVE [140] and Gazebo [141], simulation environments geared specifically towards robotics, OOPSMP, a motion-planning library [142], and GraspIt!, a grasping simulator [143].

Another exciting trend is in open-source hardware, where CAD models and the technical details of construction of devices are made freely available [144] [145]. The Arduino project [146] is a widely-used open-source microcontroller platform, and has been used in many robotics projects. The Raven [147] is an open-source laparoscopic surgery robot developed as a research platform an order of magnitude less expensive than commercial surgical robots [148].

The Cloud can also be used to facilitate open challenges and design competitions. For example, the African Robotics Network with support from IEEE Robotics and Automation Society hosted the "$10 Robot" Design Challenge in the summer of 2012. This open competition attracted 28 designs from around the world including a winning entry from Thailand (see Fig. 44.16) that modified a surplus Sony game controller, adapting its embedded vibration motors to drive wheels and adding lollipops to the thumb switches as inertial counterweights for contact sensing, which can be built from surplus parts for US $8.96 [149].

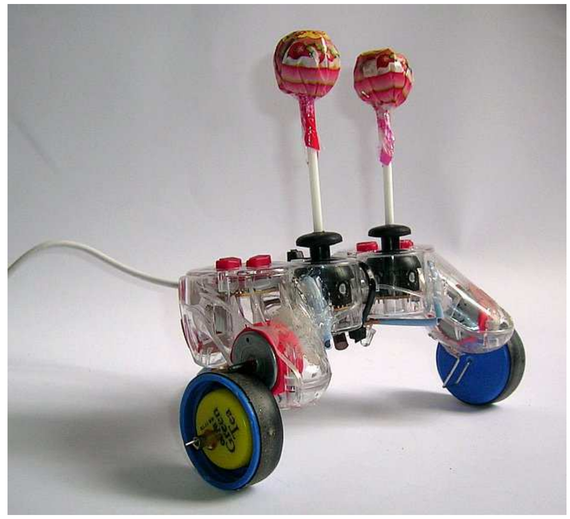

Figure 44.16: Suckerbot, designed by Tom Tilley of Thailand, a winner of the $10 Robot Design Challenge [149].

44.5.5 Crowdsourcing and Call Centers

In contrast to automated telephone reservation and technical support systems, consider a future scenario where errors and exceptions are detected by robots and automation systems, which then access human guidance on-demand at remote call centers. Human skill, experience, and intution is being tapped to solve a number of problems such as image labeling for computer vision [150] [100] [62] [53]. Amazon's Mechanical Turk is pioneering on-demand "crowdsourcing" that can draw on "human computation" or "social computing systems". Research projects are exploring how this can be used for path planning [151], to determine depth layers, image normals, and symmetry from images [152], and to refine image segmentation [153]. Researchers are working to understand pricing models [154] and apply crowdsourcing to grasping [155] (see Figure 44.17).

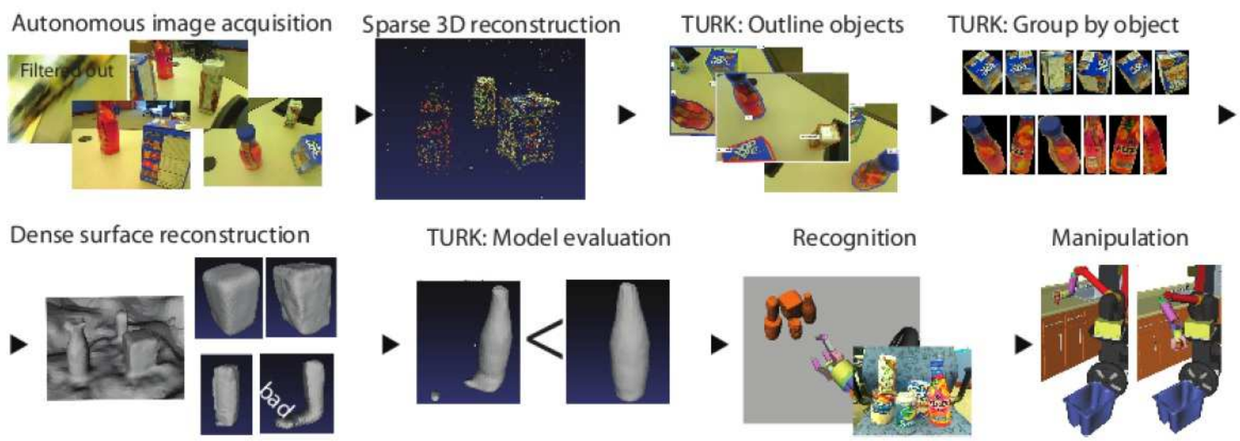

Figure 44.17: A cloud robot system that incorporates Amazon's Mechanical Turk to "crowdsource" object identification to facilitate robot grasping [155].

44.6 Conclusion and Future Directions

As this technology matures, networked robots will gradually go beyond university laboratories and find application in the real world.

As mentioned earlier in Sections 44.2.2 and 44.5, the new efforts in cloud robotics lead by Google and RoboEarth naturally bridge research and applications. The open source nature and ready-to-use APIs can quickly spread and deploy research results. Japan's Advanced Telecommunications Research Institute International (ATR) Intelligent Robotics and Communication Laboratory has also announced its networked robot project led by Norihiro Hagita (ATR). Its mission is to develop network-based intelligent robots for applications such as service, medical, and safety. Hideyuki Tokuda (Keio University) chaired the Networked Robot Forum in Spring 2005, which promotes research and development (R&D) and standardization on network robots through activities to support awareness campaigns and verification experiments in collaboration among wide-ranging parties, which includes over 100 industry and academic members. Korea's Ministry of Information and Communication has also announced the Ubiquitous Robotic Companion (URC) project to develop network-based intelligent robots.

Networked robots have allowed tens of thousands of nonspecialists around the world to interact with robots. The design of networked robots presents a number of engineering challenges to build reliable systems that can be operated by nonspecialists 24 hours a day, 7 days a week and remain online for years. Many new research challenges remain.

- *New interfaces*: As portable devices such as cellphones and tablet computers becomes grow in computation power, networked robotics should be able to adopt them as new interfaces. As computers becomes increasingly powerful, they become capable of visualizing more sophisticated sensor inputs. Designers of new interfaces should also keep track of new developments in hardware such as haptic interfaces and voice recognition systems. New software standards such as flash, extensible markup language (XML), extensible hyper text markup language (XHTML), virtual reality modeling language (VRML), and wireless markup language (WML) will also change the way we design interface.

  New interface technology arises as human computer interaction technology, mobile computing, and computer graghics areas progress. Recent progresses on brain-machine interaction explore the possibility of using brain wave, such as EEG signals, to control robot movements for ground robots [156] and UAVs [157]. Gesture [158] and multi-touch [159] are also used to generate control commands. Unlike the traditional mouse and keyboard interfaces, the new interfaces facilitate more natural interaction but suffers from precision issues, because these methods have large noise and require more research efforts in improving robustness and accuracy.

- *New algorithms*: Algorithms determine performance. Scalable algorithms that are capable of handing large amounts of data such as video/sensor network inputs and utilize fast-evolving hardware capability such as distributed and parallel computation will become increasingly important in the networked robotics, especially in cloud robotics.

- *New protocols*: Although we have listed some pioneering work in changing the network environment to improve teleoperation, there are still a large number of open problems such as new protocols, appropriate bandwidth allocation [160], QoS [161], security, routing mechanisms [28, 162], and many more. Network communication is a very fast-evolving field. The incorporation/modification of network communication ideas into networked telerobotic system design will continue to be an active research area. The common object request broker architecture (CORBA) or real-time CORBA [19, 20, 38, 163, 164] have great potential for networked robots.

- *New performance metrics*: As more and more robots enter service, it is important to develop metrics to quantify the performance of the robot-human team. As we are more familiar with metrics developed to assess robot performance or task performance [161], recent progresses on using the robot to assess human performance [165, 166] shed light on new metrics. Standardizing these metrics will also be an important direction.

- *Video for robotics*: Another interesting observation is that all of existing video compression and transmission standards try to rebuild a true and complete representation of camera field of view. However, it might not be necessary or infeasible due to bandwidth limit for a networked robot [167]. Sometimes, a high level abstraction is sufficient. For example, when a mobile robot is avoiding an moving obstacle, all the robot needs to know is the speed and bounding box of the moving object instead of knowledge that whether this object is human or other robots. We might want to control the level of details in video perception and transmission. This actually imposes a interesting problem: we need a new streaming standard that serves for networked robots.

- *Applications*: Recent successful applications include environment monitoring [42, 168], manufacturing [169, 170], and infrastructure inspection and maintenance [171, 172]. The fast development of networked robot systems is worldwide. Many new applications are emerging in areas such as security, inspection, education, and entertainment. Application requirements such as reliability, security, and modularity will continuous to pose new challenges for system design.

Bibliography

[1] K. Goldberg and R. Siegwart, editors. *Beyond Webcams: An Introduction to Online Robots*. MIT Press, 2002.

[2] N. Tesla. Method of and apparatus for controlling mechanism of moving vessels or vehicles. *http://www.pbs.org/tesla/res/613809.html*, 1898.

[3] Raymond Goertz and R. Thompson. Electronically controlled manipulator. *Nucleonics*, 1954.

[4] R. D. Ballard. A last long look at titanic. *National Geographic*, 170(6), December 1986.

[5] A. K. Bejczy. Sensors, controls, and man-machine interface for advanced teleoperation. *Science*, 208(4450), 1980.

[6] R. S. Mosher. Industrial manipulators. *Scientific American*, 211(4), 1964.

[7] R. Tomovic. On man-machine control. *Automatica*, 5, 1969.

[8] A. Bejczy, G. Bekey, R. Taylor, and S. Rovetta. A research methodology for tele-surgery with time delays. In *First International Symposium on Medical Robotics and Computer Assisted Surgery*, Sept. 1994.

[9] Matthew Gertz, David Stewart, and Pradeep Khosla. A human-machine interface for distributed virtual laboratories. *IEEE Robotics and Automation Magazine*, December 1994.

[10] T. Sato, J. Ichikawa, M. Mitsuishi, and Y. Hatamura. A new micro-teleoperation system employing a hand-held force feedback pencil. In *IEEE International Conference on Robotics and Automation*, May 1994.

[11] Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.

[12] FirstWebcam. `http://www.cl.cam.ac.uk/coffee/qsf/timeline.html`. 1993.

[13] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Robot teleoperation via www. In *IEEE International Conference on Robotics and Automation*, May 1995.

[14] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and Jeff Wiegley. Beyond the web: Manipulating the physical world via the www. *Computer Networks and ISDN Systems Journal*, 28(1), December 1995. Archives can be viewed at http://www.usc.edu/dept/raiders/.

[15] B. Dalton and K. Taylor. A framework for internet robotics. In *IEEE International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots, Victoria, Canada*, 1998.

[16] K. Taylor and J. Trevelyan. `http://telerobot.mech.uwa.edu.au/`. 1994.

[17] H. Hu, L. Yu, P. W. Tsui, and Q. Zhou. Internet-based robotic systems for teleoperation. *Assemby Automation*, 21(2):143–151, May 2001.

[18] R. Safaric, M. Debevc, R. Parkin, and S. Uran. Telerobotics experiments via internet. *IEEE Transactions on Industrial Electronics*, 48(2):424–31, April 2001.

[19] S. Jia and K. Takase. A corba-based internet robotic system. *Advanced Robotics*, 15(6):663–673, Oct 2001.

[20] S. Jia, Y. Hada, G. Ye, and K. Takase. Distributed telecare robotic systems using corba as a communication architecture. In *IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, United States*, 2002.

[21] J. Kim, B. Choi, S. Park, K.Kim, and S. Ko. Remote control system using real-time mpeg-4 streaming technology for mobile robot. In *IEEE International Conference on Consumer Electronics*, 2002.

[22] T. Mirfakhrai and S. Payandeh. A delay prediction approach for teleoperation over the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[23] K. Han, Y. Kim, J. Kim, and S.Hsia. Internet control of personal robot between kaist and uc davis. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[24] L. Ngai, W.S. Newman, and V. Liberatore. An experiment in internet-based, human-assisted robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[25] R.C. Luo and T. M. Chen. Development of a multibehavior-based mobile robot for remote supervisory control through the internet. *IEEE/ASME Transactions on Mechatronics*, 5(4):376–385, 2000.

[26] D. Aarno, S. Ekvall, and D. Kragi. Adaptive virtual fixtures for machine-assisted teleoperation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1151–1156, 2005.

[27] I. Belousov, S. Chebukov, and V. Sazonov. Web-based teleoperation of the robot interacting with fast moving objects. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 685–690, 2005.

[28] Z. Cen, A. Goradia, M. Mutka, N. Xi, W. Fung, and Y. Liu. Improving the operation efficiency of supermedia enhanced internet based teleoperation via an overlay network. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 691–696, 2005.

[29] N. P. Jouppi and S. Thomas. Telepresence systems with automatic preservation of user head height, local rotation, and remote translation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 62–68, 2005.

[30] B. Ricks, C. W. Nielsen, and M. A. Goodrich. Ecological displays for robot interaction: a new perspective. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 2855 – 2860, 2004.

[31] D. Ryu, S. Kang, M. Kim, and J. Song. Multi-modal user interface for teleoperation of robhaz-dt2 field robot system. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 168–173, 2004.

[32] J. Su and Z. Luo. Incremental motion compression for telepresent walking subject to spatial constraints. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 69–74, 2005.

[33] I. Toshima and S. Aoki. Effect of driving delay with an acoustical tele-presence robot, telehead. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 56–61, 2005.

[34] N. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, and K. Tanie. Remote coordinated controls in multiple telerobot cooperation. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3138–3343, April 2000.

[35] P. Cheng and V. Kumar. An almost communication-less approach to task allocation for multiple unmanned aerial vehicles. In *IEEE*

International Conference on Robotics and Automation (ICRA), Pasadena, California, pages 1384 –1389, may 2008.

[36] X. Ding, M. Powers, M. Egerstedt, S. Young, and T. Balch. Executive decision support. *IEEE Robotics Automation Magazine*, 16(2):73 –81, june 2009.

[37] J. Liu, L. Sun, T. Chen, X. Huang, and C. Zhao. Competitive multi-robot teleoperation. In *IEEE International Conference on Robotics and Automation Barcelona, Spain*, April 2005.

[38] Z. Zhang, Q. Cao, L. Zhang, and C. Lo. A corba-based cooperative mobile robot system. *Industrial Robot: An International Journal*, 36(1):36 – 44, 2009.

[39] Y. Xu and D. Song. Systems and algorithms for autonomous and scalable crowd surveillance using robotic ptz cameras assisted by a wide-angle camera. *Autonomous Robots*, 29(1):53–66, July 2010.

[40] D. Sanders, J. Graham-Jones, and A. Gegov. Improving ability of tele-operators to complete progressively more difficult mobile robot paths using simple expert systems and ultrasonic sensors. *Industrial Robot: An International Journal*, 37(5):431 – 440, 2010.

[41] S. Faridani, B. Lee, S. Glasscock, J. Rappole, D. Song, and K. Goldberg. A networked telerobotic observatory for collaborative remote observation of avian activity and range change. In *the IFAC workshop on networked robots, Oct. 6-8, 2009, Golden, Colorado*, Oct. 2009.

[42] R. Bogue. Robots for monitoring the environment. *Industrial Robot: An International Journal*, 38(6):560 – 566, 2011.

[43] R. Murphy and J. Burke. From remote tool to shared roles. *IEEE Robotics Automation Magazine*, 15(4):39 –49, Dec. 2008.

[44] Eric Paulos, John Canny, and Francesca Barrientos. Prop: Personal roving presence. In *SIGGRAPH Visual Proceedings*, page 99, August 1997.

[45] L. Takayama, E. Marder-Eppstein, H. Harris, and J. Beer. Assisted driving of a mobile remote presence system: System design and controlled user evaluation. In *IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, pages 1883 –1889, may 2011.

[46] D. Lazewatsky and W. Smart. An inexpensive robot platform for teleoperation and experimentation. In *IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China*, pages 1211 –1216, may 2011.

[47] K. Taylor and J.P. Trevelyan. Australias telerobot on the web. In *26th Symposium on Industrial Robotics, Singapore*, pages 39–44, Oct. 1995.

[48] A. Khamis, D. M. Rivero, F. Rodriguez, and M. Salichs. Pattern-based architecture for building mobile robotics remote laboratories. In *IEEE Internetional Conference on Robotics and Automation (ICRA), Taipei, ROC*, pages 3284–3289, Sep. 2003.

[49] C. Cosma, M. Confente, D. Botturi, and P. Fiorini. Laboratory tools for robotics and automation education. In *IEEE Internetional Conference on Robotics and Automation (ICRA), Taipei, ROC*, pages 3303–3308, Sep. 2003.

[50] K. W. Dorman, J. L. Pullen, W. O. Keksz, P. H. Eismann, K. A. Kowalski, and J. P. Karlen. The servicing aid tool: A teleoperated robotics system for space applications. In *The Seventh Annual Workshop on Space Operations Applications and Research (SOAR 1993), Johnson Space Center, Houston, TX*, volume 1, Jan. 1994.

[51] C. Pollak and H. Hutter. A webcam as recording device for light microscopes. *Journal of*

Computer-Assisted Microscopy, 10(4):179–83, 1998.

[52] K. Goldberg, D. Song, and A. Levandowski. Collaborative teleoperation using networked spatial dynamic voting. *The Proceedings of The IEEE*, 91(3):430–439, March 2003.

[53] James J. Kuffner. Cloud-Enabled Robots. In *IEEE-RAS International Conference on Humanoid Robots*, Nashville, TN, 2010.

[54] Ken Goldberg and Roland Siegwart, editors. *Beyond Webcams: An Introduction to Online Robots*. MIT Press, 2002.

[55] Ken Goldberg, Michael Mascha, Steve Gentner, Nick Rothenberg, Carl Sutter, and Jeff Wiegley. Desktop teleoperation via the World Wide Web. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 654–659. IEEE, 1995.

[56] G. McKee. What is Networked Robotics? *Informatics in Control Automation and Robotics*, 15:35–45, 2008.

[57] Eric Guizzo. Cloud Robotics: Connected to the Cloud, Robots Get Smarter, 2011.

[58] Moritz Tenorth, Alexander Clifford Perzylo, Reinhard Lafrenz, and Michael Beetz. The RoboEarth language: Representing and exchanging knowledge about actions, objects, and environments. In *IEEE International Conference on Robotics and Automation*, number 3, pages 1284–1289. Ieee, May 2012.

[59] Rajesh Arumugam, V.R. Enti, Liu Bingbing, Wu Xiaojun, Krishnamoorthy Baskaran, F.F. Kong, A.S. Kumar, K.D. Meng, and G.W. Kit. DAvinCi: A Cloud Computing Framework for Service Robots. In *IEEE International Conference on Robotics and Automation*, pages 3084–3089. IEEE, 2010.

[60] Zhihui Du, Weiqiang Yang, Yinong Chen, Xin Sun, Xiaoying Wang, and Chen Xu. Design of a Robot Cloud Center. In *International Symposium on Autonomous Decentralized Systems*, pages 269–275. IEEE, March 2011.

[61] Guoqiang Hu, WP Tay, and Yonggang Wen. Cloud Robotics: Architecture, Challenges and Applications. *IEEE Network*, 26(3):21–28, 2012.

[62] Koji Kamei, Shuichi Nishio, Norihiro Hagita, and Miki Sato. Cloud Networked Robotics. *IEEE Network*, 26(3):28–34, 2012.

[63] Dominique Hunziker, Mohanarajah Gajamohan, Markus Waibel, and Raffaello D Andrea. Rapyuta: The RoboEarth Cloud Engine. 2013.

[64] Markus Waibel, Michael Beetz, Javier Civera, Raffaello D'Andrea, Jos Elfring, Dorian Gálvez-López, Kai Häussermann, Rob Janssen, J.M.M. Montiel, Alexander Perzylo, Björn Schieß le, Moritz Tenorth, Oliver Zweigle, and René De Molengraft. RoboEarth. *IEEE Robotics & Automation Magazine*, 18(2):69–82, June 2011.

[65] What is RoboEarth? `http://www.roboearth.org/what-is-roboearth`.

[66] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A Survey. *Computer Networks*, 54(15):2787–2805, October 2010.

[67] J. Walrand and P. Varaiya. *High-Performance Communication Networks, 2nd Edition*. Morgan Kaufmann Press, 2000.

[68] Michael A. Peshkin and Art C. Sanderson. Minimization of energy in quasi-static manipulation. *IEEE Transactions on Robotics and Automation*, 5(1), February 1989.

[69] Matthew T. Mason. On the scope of quasi-static pushing. In O. Faugeras and G. Giralt, editors, *The Third International Symposium on Robotics Research*. MIT Press, 1986.

[70] Eric Ladd and Jim O'Donnell. *Using Html 4, Xml, and Java 1.2*. QUE Press, 1998.

[71] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Desktop tele-operation via the world wide web. In *IEEE International Conference on Robotics and Automation, Nagoya, Japan.*, May 1995.

[72] H. Friz. *Design of an Augmented Reality User Interface for an Internet based Telerobot using Multiple Monoscopic Views*. PhD thesis, Institute for Process and Production Control Techniques, Technical University of Clausthal Clausthal-Zellerfeld, Germany, 2000.

[73] T. Fong and C. Thorpe. Vehicle teleoperation interfaces. *Autonomous Robots*, 11:9–18, 2001.

[74] A .Birk, N. Vaskevicius, K. Pathak, S. Schwertfeger, J. Poppinga, and H. Buelow. 3-d perception and modeling. *IEEE Robotics Automation Magazine*, 16(4):53 –60, Dec. 2009.

[75] A. Kellyo, N. Chan, H. Herman, D. Huber, R. Meyers, P. Rander, R. Warner, J. Ziglar, and E. Capstick. Real-time photorealistic virtualized reality interface for remote mobile robot control. *The International Journal of Robotics Research*, 30(3):384–404, 2011.

[76] L. Conway, R. A. Volz, and M. W. Walker. Teleautonomous systems: Projecting and coordinating intelligent action at a distance. *IEEE Transactions on Robotics and Automation*, 6(20):146–158, Appl. 1990.

[77] J. Larsson, M. Broxvall, and A. Saffiotti. An evaluation of local autonomy applied to teleoperated vehicles in underground mines. In *IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska*, pages 1745 –1752, may 2010.

[78] Louis B. Rosenberg. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *IEEE Virtual Reality Annual International Symposium (VRAIS), Seattle, Washington*, pages 76–82, Sep. 1993.

[79] P. Marayong, M. Li, A. Okamura, and G. Hager. Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1954 – 1959, sept. 2003.

[80] A. Bettini, P. Marayong, S. Lang, A. Okamura, and G. Hager. Vision-assisted control for manipulation using virtual fixtures. *IEEE Transactions on Robotics*, 20(6):953 – 966, dec. 2004.

[81] R. Azuma. A survey of augmented reality. *Presence*, 6(4):355 – 385, Aug. 1997.

[82] K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon, and G. Smith. Collaborative teleoperation via the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 2019–2024, April 2000.

[83] D. Song. *Systems and Algorithms for Collaborative Teleoperation*. PhD thesis, Department of Industrial Engineering and Operations Research, University of California, Berkeley, 2004.

[84] D. Song. *Sharing a Vision: Systems and Algorithms for Collaboratively-Teleoperated Robotic Cameras*. Springer, 2009.

[85] K. Goldberg and B. Chen. Collaborative teleoperation via the internet. In *International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2001.

[86] D. Song, A. Pashkevich, and K. Goldberg. Sharecam part II: Approximate and distributed algorithms for a collaboratively controlled robotic webcam. In *IEEE/RSJ International Conference on Intelligent Robots (IROS), Las Vegas, NV*, volume 2, pages 1087 – 1093, Oct. 2003.

[87] D. Song and K. Goldberg. Sharecam part I: Interface, system architecture, and implementation of a collaboratively controlled robotic webcam. In *IEEE/RSJ International Conference*

on Intelligent Robots (IROS), Las Vegas, NV, volume 2, pages 1080 – 1086, Oct. 2003.

[88] D. Song and K. Goldberg. Approximate algorithms for a collaboratively controlled robotic camera. *IEEE Transactions on Robotics*, 23(5):1061–1070, Nov. 2007.

[89] D. Song, A. F. van der Stappen, and K. Goldberg. Exact algorithms for single frame selection on multi-axis satellites. *IEEE Transactions on Automation Science and Engineering*, 3(1):16–28, January 2006.

[90] Y. Xu, D. Song, and J. Yi. An approximation algorithm for the least overlapping p-frame problem with non-partial coverage for networked robotic cameras. In *IEEE International Conference on Robotics and Automation (ICRA), Pasadena, CA*, May 2008.

[91] D. Song, N. Qin, and K. Goldberg. Systems, control models, and codec for collaborative observation of remote environments with an autonomous networked robotic camera. *Autonomous Robots*, 24(4):435–449, May. 2008.

[92] K. Goldberg, D. Song, I. Y. Song, J. McGonigal, W. Zheng, and D. Plautz. Unsupervised scoring for scalable internet-based collaborative teleoperation. In *IEEE International Conference on Robotics and Automation (ICRA), New Orleans, LA, US*, April 2004.

[93] J. Rappole, S. Glasscock, K. Goldberg, D. Song, and S. Faridani. Range change among new world tropical and subtropical birds. In K.-L. Schuchmann, editor, *Tropical vertebrates in a changing world*, Bonner Zoologische Monographien, Nr 57, pages 151–167. Bonn: Zoologisches Forschungsmuseum Alexander Koenig, Germany, 2011.

[94] C. Goldfeder, M. Ciocarlie, and P.K. Allen. The Columbia Grasp Database. In *IEEE International Conference on Robotics and Automation*, pages 1710–1716. IEEE, May 2009.

[95] A. Kasper, Z. Xue, and R. Dillmann. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, May 2012.

[96] Hao Dang, Jonathan Weisz, and Peter K. Allen. Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics. In *IEEE International Conference on Robotics and Automation*, pages 5917–5922. Ieee, May 2011.

[97] Hao Dang and Peter K. Allen. Learning grasp stability. In *IEEE International Conference on Robotics and Automation*, pages 2392–2397. IEEE, May 2012.

[98] Jonathan Weisz and Peter K. Allen. Pose error robust grasping from contact wrench space metrics. In *IEEE International Conference on Robotics and Automation*, pages 557–562. IEEE, May 2012.

[99] Mila Popovic, Gert Kootstra, Jimmy Alison Jorgensen, Danica Kragic, and Norbert Kruger. Grasping unknown objects using an Early Cognitive Vision system for general scene understanding. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 987–994. IEEE, September 2011.

[100] Matei Ciocarlie, Caroline Pantofaru, Kaijen Hsiao, Gary Bradski, Peter Brook, and Ethan Dreyfuss. A Side of Data With My Robot. *IEEE Robotics & Automation Magazine*, 18(2):44–57, June 2011.

[101] M.A. Moussa and M.S. Kamel. An experimental approach to robotic grasping using a connectionist architecture and generic grasping functions. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 28(2):239–253, May 1998.

[102] Kai Huebner, Kai Welke, Markus Przybylski, Nikolaus Vahrenkamp, Tamim Asfour, and Danica Kragic. Grasping Known Objects

with Humanoid Robots: A Box-Based Approach. In *International Conference on Advanced Robotics*, 2009.

[103] Corey Goldfeder and Peter K. Allen. Data-Driven Grasping. *Autonomous Robots*, 31(1):1–20, April 2011.

[104] Matei Ciocarlie, Kaijen Hsiao, E. G. Jones, Sachin Chitta, R.B. Rusu, and I.A. Sucan. Towards Reliable Grasping and Manipulation in Household Environments. In *Intl. Symposium on Experimental Robotics*, pages 1–12, New Delhi, India, 2010.

[105] Google Goggles. `http://www.google.com/mobile/goggles/`.

[106] B. Kehoe, A. Matsukawa, S. Candido, J. Kuffner, and K. Goldberg. Cloud-Based Robot Grasping with the Google Object Recognition Engine. 2013. Submitted to IEEE International Conference on Robotics and Automation, 2013.

[107] Sebastien Dalibard, Alireza Nakhaei, Florent Lamiraux, and Jean-Paul Laumond. Manipulation of documented objects by a walking humanoid robot. In *IEEE-RAS International Conference on Humanoid Robots*, pages 518–523. Ieee, December 2010.

[108] K. Lai and D. Fox. Object Recognition in 3D Point Clouds Using Web Data and Domain Adaptation. *The International Journal of Robotics Research*, 29(8):1019–1037, May 2010.

[109] Stephan Gammeter, Alexander Gassmann, Lukas Bossard, Till Quack, and Luc Van Gool. Server-side Object Recognition and Client-side Object Tracking for Mobile Augmented Reality. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, number C, pages 1–8. Ieee, June 2010.

[110] Michael Armbrust, Ion Stoica, Matei Zaharia, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, and Ariel Rabkin. A View of Cloud Computing. *Communications of the ACM*, 53(4):50, April 2010.

[111] Amazon Web Services. `http://aws.amazon.com`.

[112] Amazon Elastic Cloud (EC2). `http://aws.amazon.com/ec2/`.

[113] Google Compute Engine. `https://cloud.google.com/products/compute-engine`.

[114] Microsoft Azure. `http://www.windowsazure.com`.

[115] Gideon Juve, Ewa Deelman, G. Bruce Berriman, Benjamin P. Berman, and Philip Maechling. An Evaluation of the Cost and Performance of Scientific Workflows on Amazon EC2. *Journal of Grid Computing*, 10(1):5–21, March 2012.

[116] Piyush Mehrotra, Jahed Djomehri, Steve Heistand, Robert Hood, Haoqiang Jin, Arthur Lazanoff, Subhash Saini, and Rupak Biswas. Performance evaluation of Amazon EC2 for NASA HPC applications. In *Proceedings of the 3rd workshop on Scientific Cloud Computing Date - ScienceCloud '12*, page 41, New York, New York, USA, 2012. ACM Press.

[117] Radu Tudoran, Alexandru Costan, Gabriel Antoniu, and Luc Bougé. A performance evaluation of Azure and Nimbus clouds for scientific applications. In *Proceedings of the 2nd International Workshop on Cloud Computing Platforms - CloudCP '12*, pages 1–6, New York, New York, USA, 2012. ACM Press.

[118] TOP500. `http://www.top500.org/list/2012/06/100`.

[119] Nitesh Kumar Jangid. Real Time Cloud Computing. In *Data Management & Security*, 2011.

[120] Jared Glover, Daniela Rus, and Nicholas Roy. Probabilistic Models of Object Geometry for Grasp Planning. In *Robotics: Science and Systems*, Zurich, Switzerland, 2008.

[121] Henry Wang, Yunzhi Ma, Guillem Pratx, and Lei Xing. Toward real-time Monte Carlo simulation using a commercial cloud computing infrastructure. *Physics in medicine and biology*, 56(17):N175–81, September 2011.

[122] Martin Sevior, Tom Fifield, and Nobuhiko Katayama. Belle monte-carlo production on the Amazon EC2 cloud. *Journal of Physics: Conference Series*, 219(1):012003, April 2010.

[123] D Nister and H Stewenius. Scalable Recognition with a Vocabulary Tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168. IEEE, 2006.

[124] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object Retrieval with Large Vocabularies and Fast Spatial Matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. Ieee, June 2007.

[125] Bharat Bhargava, Pelin Angin, and Lian Duan. A Mobile-Cloud Pedestrian Crossing Guide for the Blind. In *International Conference on Advances in Computing & Communication*, 2011.

[126] JJS Garc´ia. Using Cloud Computing as a HPC Platform for Embedded Systems. 2011.

[127] Jur van den Berg, Pieter Abbeel, and Ken Goldberg. LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, June 2011.

[128] Ben Kehoe, D Berenson, and K Goldberg. Estimating Part Tolerance Bounds Based on Adaptive Cloud-Based Grasp Planning with Slip. In *IEEE International Conference on Automation Science and Engineering*. IEEE, 2012.

[129] Ben Kehoe, Dmitry Berenson, and Ken Goldberg. Toward Cloud-based Grasping with Uncertainty in Shape: Estimating Lower Bounds on Achieving Force Closure with Zero-slip Push Grasps. In *IEEE International Conference on Robotics and Automation*, pages 576–583. IEEE, May 2012.

[130] Dmitry Berenson, Pieter Abbeel, and Ken Goldberg. A Robot Path Planning Framework that Learns from Experience. *IEEE International Conference on Robotics and Automation*, pages 3671–3678, May 2012.

[131] MyRobots.com. `http://myrobots.com`.

[132] What is MyRobots? `http://myrobots.com/wiki/About`.

[133] Laura Dabbish, Colleen Stuart, Jason Tsay, and Jim Herbsleb. Social coding in GitHub: transparency and collaboration in an open software repository. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, page 1277, New York, New York, USA, 2012. ACM Press.

[134] Alexander Hars. Working for free? Motivations of participating in open source projects. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, volume 00, page 9. IEEE Comput. Soc, 2001.

[135] Daniel Nurmi, Rich Wolski, Chris Grzegorczyk, Graziano Obertelli, Sunil Soman, Lamia Youseff, and Dmitrii Zagorodnov. The Eucalyptus Open-Source Cloud-Computing System. In *IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 124–131. IEEE, 2009.

[136] ROS (Robot Operating System). `http://ros.org`.

[137] Morgan Quigley and Brian Gerkey. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, number Figure 1, 2009.

[138] rosjava, an implementation of ROS in pure Java with Android support. `http://cloudrobotics.com`.

[139] Bullet Physics Library. http://bulletphysics.org.

[140] OpenRAVE. http://openrave.org/.

[141] Gazebo. http://gazebosim.org.

[142] Erion Plaku, Kostas E. Bekris, and Lydia E Kavraki. OOPS for Motion Planning: An Online, Open-source, Programming System. In *IEEE International Conference on Robotics and Automation*, number April, pages 3711–3716. IEEE, April 2007.

[143] A.T. Miller and P.K. Allen. GraspIt! A Versatile Simulator for Robotic Grasping. *IEEE Robotics & Automation Magazine*, 11(4):110–122, December 2004.

[144] S. Davidson. Open-source hardware. *IEEE Design and Test of Computers*, 21(5):456–456, September 2004.

[145] Erik Rubow. Open Source Hardware. Technical report, 2008.

[146] Arduino. http://www.arduino.cc.

[147] H Hawkeye King, Lei Cheng, Philip Roan, Diana Friedman, Sina Nia, Ji Ma, Daniel Glozman, Jacob Rosen, and Blake Hannaford. Raven II: Open Platform for Surgical Robotics Research. In *The Hamlyn Symposium on Medical Robotics*, 2012.

[148] An open-source robo-surgeon. *The Economist*, 2012.

[149] The African Robotics Network (AFRON). "Ten Dollar Robot" Design Challenge Winners. http://robotics-africa.org/design_challenge.html.

[150] Luis von Ahn. Human Computation. In *Design Automation Conference*, page 418, 2009.

[151] Juan Camilo Gamboa Higuera, Anqi Xu, Florian Shkurti, and Gregory Dudek. Socially-Driven Collective Path Planning for Robot Missions. *2012 Ninth Conference on Computer and Robot Vision*, pages 417–424, May 2012.

[152] Yotam Gingold, Ariel Shamir, and Daniel Cohen-Or. Micro perceptual human computation for visual tasks. *ACM Transactions on Graphics*, 31(5):1–12, August 2012.

[153] Matthew Johnson-Roberson, Jeannette Bohg, Gabriel Skantze, Joakim Gustafson, Rolf Carlson, Babak Rasolzadeh, and Danica Kragic. Enhanced visual scene understanding through human-robot dialog. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3342–3348. IEEE, September 2011.

[154] Alexander Sorokin and David Forsyth. Utility Data Annotation with Amazon Mechanical Turk. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, number c, pages 1–8. IEEE, June 2008.

[155] A Sorokin, D Berenson, S S Srinivasa, and M Hebert. People helping robots helping people: Crowdsourcing for grasping novel objects. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2117–2122, October 2010.

[156] C. Escolano, J. Antelis, and J. Minguez. Human brain-teleoperated robot between remote places. In *IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pages 4430 –4437, may 2009.

[157] A. Akce, M. Johnson, and T. Bretl. Remote teleoperation of an unmanned aircraft with a brain-machine interface: Theory and preliminary results. In *IEEE International Conference on Robotics and Automation (ICRA), Anchorage, Alaska*, pages 5322 –5327, may 2010.

[158] J. Roselln, R. Suárez, C. Rosales, and A. Pérez. Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures. *Auton. Robots*, 31(1):87–102, July 2011.

[159] K. Onda and F. Arai. Parallel teleoperation of holographic optical tweezers using multi-touch

user interface. In *IEEE International Conference on Robotics and Automation (ICRA), St. Paul, Minnesota*, pages 1069 –1074, may 2012.

[160] P. X. Liu, M. Meng, and S.X. Yang. Data communications for internet robots. *Autonomous Robots*, 15:213–223, 2003.

[161] W. Fung, N. Xi, W. Lo, B. Song, Y. Sun, Y. Liu, and I. H. Elhajj. Task driven dynamic qos based bandwidth allcoation for real-time teleoperation via the internet. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, US*, October 2003.

[162] F. Zeiger, N. Kraemer, and K. Schilling. Commanding mobile robots via wireless ad-hoc networks x2014; a comparison of four ad-hoc routing protocol implementations. In *IEEE International Conference on Robotics and Automation (ICRA), Pasadena, California*, pages 590 –595, may 2008.

[163] M. Amoretti, S. Bottazzi, M. Reggiani, and S. Caselli. Evaluation of data distribution techniques in a corba-based telerobotic system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, NV, US*, October 2003.

[164] S. Bottazzi, S. Caselli, M. Reggiani, and M. Amoretti. A software framework based on real time COBRA for telerobotics systems. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, EPFL, Lausanne, Switzerland*, October 2002.

[165] J. Chen, E. Haas, and M. Barnes. Human performance issues and user interface design for teleoperated robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1231 –1245, nov. 2007.

[166] Y. Jia, N. Xi, Y. Wang, and . Li X. Online identification of quality of teleoperator (qot) for performance improvement of telerobotic operations. In *IEEE International Conference on Robotics and Automation (ICRA), St. Paul, Minnesota*, pages 451 –456, may 2012.

[167] S. Livatino, G. Muscato, S. Sessa, C. Koffel, C. Arena, A. Pennisi, D. Di Mauro, and F. Malkondu. Mobile robotic teleguide based on video images. *IEEE Robotics Automation Magazine*, 15(4):58 –67, dec. 2008.

[168] G. Podnar, J. Dolan, A. Elfes, S. Stancliff, E. Lin, J. Hosier, T. Ames, J. Moisan, T. Moisan, J. Higinbotham, and E. Kulczycki. Operation of robotic science boats using the telesupervised adaptive ocean sensor fleet system. In *IEEE International Conference on Robotics and Automation (ICRA), , Pasadena, California*, pages 1061 –1068, may 2008.

[169] Y. Kwon and S. Rauniar. E-quality for manufacturing (eqm) within the framework of internet-based systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1365 – 1372, nov. 2007.

[170] Lihui Wang. Wise-shopfloor: An integrated approach for web-based collaborative manufacturing. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(4):562 –573, july 2008.

[171] P. Debenest, M. Guarnieri, K. Takita, E. Fukushima, S. Hirose, K. Tamura, A. Kimura, H. Kubokawa, N. Iwama, and F. Shiga. Expliner - robot for inspection of transmission lines. In *IEEE International Conference on Robotics and Automation (ICRA), Pasadena, California*, pages 3978 –3984, may 2008.

[172] N. Pouliot and S. Montambault. Linescout technology: From inspection to robotic maintenance on live transmission power lines. In *IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan*, pages 1034 – 1040, may 2009.