

Community Seismic Network

Robert W. Clayton^{1,*}, Thomas Heaton², Mani Chandy³, Andreas Krause³, Monica Kohler², Julian Bunn⁴, Richard Guy¹, Michael Olson³, Mathew Faulkner³, MingHei Cheng², Leif Strand⁴, Rishi Chandy³, Daniel Obenshain³, Annie Liu³, Michael Aivazis⁴

¹ California Institute of Technology, Seismological Laboratory, Pasadena, CA, USA

² California Institute of Technology, Earthquake Engineering, Pasadena, CA, USA

³ California Institute of Technology, Computer Science, Pasadena, CA, USA

⁴ California Institute of Technology, Center for Advanced Computing Research, Pasadena, CA, USA

Article history

Received June 30, 2011; accepted October 26, 2011.

Subject classification:

Seismology/Ground motion, Instruments and techniques, Seismic risk, Computational geophysics/Algorithms and implementation, Data dissemination/Seismological data.

ABSTRACT

The article describes the design of the Community Seismic Network, which is a dense open seismic network based on low cost sensors. The inputs are from sensors hosted by volunteers from the community by direct connection to their personal computers, or through sensors built into mobile devices. The server is cloud-based for robustness and to dynamically handle the load of impulsive earthquake events. The main product of the network is a map of peak acceleration, delivered within seconds of the ground shaking. The lateral variations in the level of shaking will be valuable to first responders, and the waveform information from a dense network will allow detailed mapping of the rupture process. Sensors in buildings may be useful for monitoring the state-of-health of the structure after major shaking.

Introduction

The most important information that can be provided to emergency responders in the minutes to hours following an earthquake is an assessment of damage, and the best proxy we have to produce estimates on this time scale are measurements of ground shaking. Products such as ShakeMap [Wald et al. 1999a] do this with near real-time sensors that are available from traditional seismic networks such the Southern California Seismic Network (SCSN). However, the sensors are sparsely distributed and the resulting maps have low-resolution and require model-driven interpolation, which means they need to know the earthquake hypocenter and magnitude. Sensors are typically separated by several kilometers (approximately 10 km in the case of the SCSN). Increasing the density of seismic networks beyond that needed for their basic function of earthquake location is cost prohibitive under their current paradigm of operation of closed networks with high-quality (expensive) sensors. The capital and maintenance costs,

permitting and data analysis are the usual limiting factors.

Another approach to providing this information is to use crowd-sourcing to obtain the measurements. The “Did You Feel It” (DYFI) product of the US Geological Survey (USGS) [Wald et al. 1999b] does this with a simple post-earthquake web interface, where users enter intensity observations with postal codes providing the location information. The sheer numbers of reports largely overcome the simple and subjective measurement scale and crude location mechanism. Recent mild but widely-felt, earthquakes in the Los Angeles region have produced over 40,000 entries. When the inverse-distance dependency is removed from these maps such as in Figure 1, there is surprising level of detail revealed about the lateral variations in shaking intensity. One drawback to this form of sensing is the human responders in the areas of heavy shaking usually do not make the data entry their first priority, and hence information from the most critical areas is usually late in arriving.

In this paper we describe an alternative way to achieve the goal of providing detailed and rapid assessment of ground shaking in urban areas. The method is based on an open-network of low-cost sensors that are hosted by volunteers and the telemetry is provided by the internet. The Community Seismic Network (CSN) (<http://www.communityseismicnetwork.org>) described here can be viewed as a quantitative version of DYFI. The primary product of the CSN is a map of ground shaking that can be delivered within seconds of major shaking. With a dense network this can be generated before accurate estimates of the location and magnitude are obtained. The measurement of moderate earthquakes will provide maps of anomalous ground amplification. The waveforms from the network will provide the information to determine the dynamic slip on the fault.

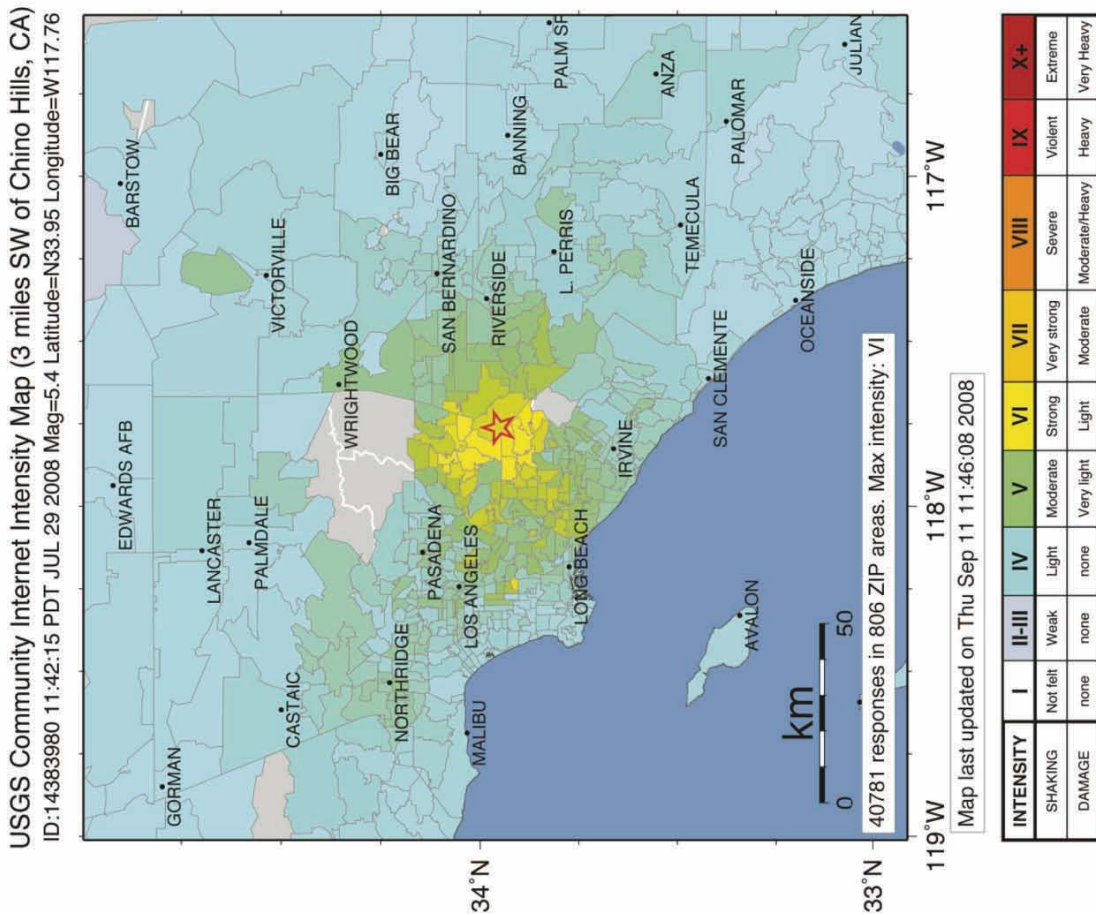
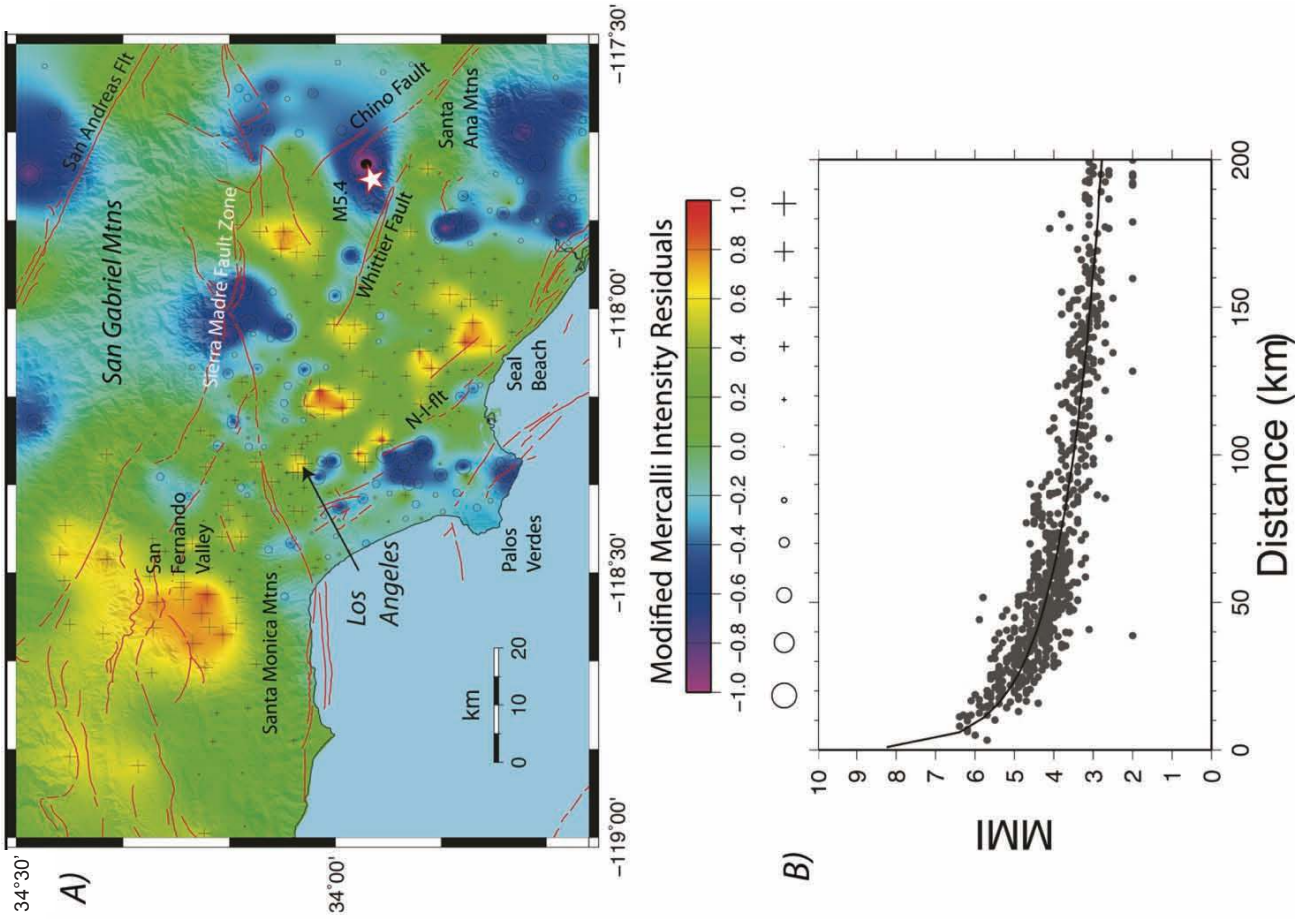


Figure 1. A crowd-sourcing example. The event is magnitude 5.4 on July 29, 2008, in the Los Angeles region. The “Did You Feel It” intensity map is constructed by the USGS from over 40,000 responses. When the $1/r$ factor is removed, the residual intensity map shown in the upper right is obtained. The DYFI is available from the USGS website, and the residual map was constructed by Susan Hough of the USGS.

The concept of basing a seismic network on micro-electromechanical systems (MEMS) sensors was proposed by Evans et al. [2005]. In the time since that suggestion, the sensors have become much more sensitive (and cheaper). The Quake-Catcher Network [Cochran et al. 2009a, Cochran et al. 2009b, Chung et al. 2011] and the Home Seismometer Network [Horiuchi et al. 2009] are also based on low-cost MEMS sensors. The oil-industry has been increasing the use of MEMS accelerometers in their surveys, driven by the need for low-cost compact three-component sensors [Hons et al. 2008, Mougnot et al. 2011].

The CSN described herein is under development with approximately 100 sensors deployed, and thus far no felt earthquakes have occurred. Consequently, the system has not been tested under real event conditions. The CSN is embedded in the reporting region of the SCSN, which is jointly operated by Caltech and the USGS. The CSN is not intended as a replacement for traditional networks, but rather as a supplement to increase the resolution of ground shaking measurements. The MEMS sensors that are currently used by the CSN are not sensitive enough to detect regional or small local earthquakes.

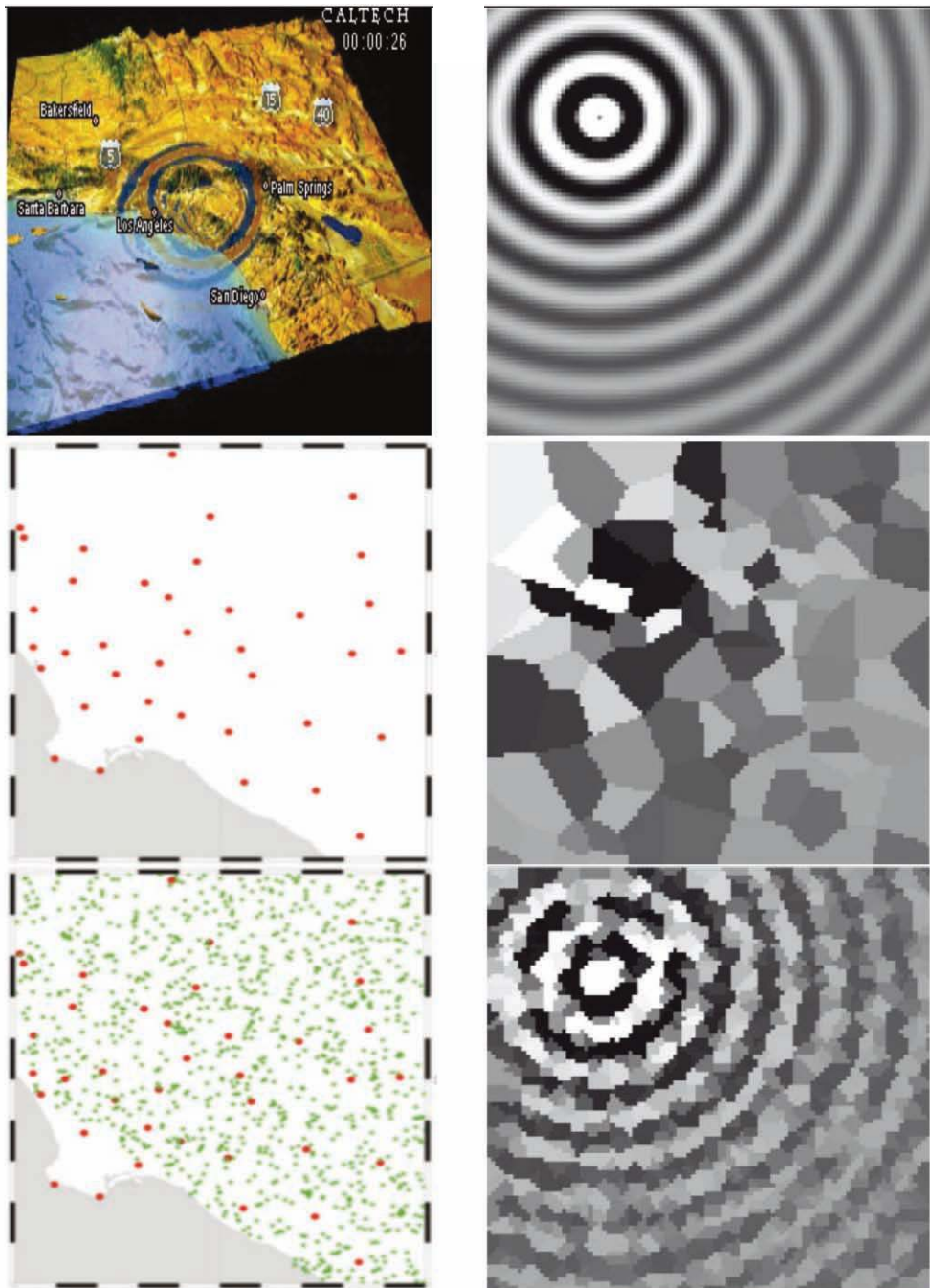


Figure 2. Effect of a dense array. The upper row shows the output of simulated motion in southern California, and a perfect point source. The middle row shows the current seismic network density (the SCSN) and the reconstruction of the point source. The bottom row shows the results for a dense array of 1000 stations.

Advantages of a dense network

With the CSN, the quality of an individual sensor is traded-off against the density of the network. The tradeoff is common in the seismic exploration industry, where the goal is to measure the wavefield in an unaliased manner. For earthquake monitoring, a dense network is important when there are significant lateral variations in the intensity of ground shaking [Field and Hough 1997]. This is likely the case when there are near surface structures such as micro-basins or when there are variations in soil conditions. In Figure 2, a synthetic example shows the effect of a sparse and dense network on an interpolated wavefield. Analysis methods that utilize sampled wavefields will not work with sparse networks.

One advantage of a dense network is that the real-time processing is generally simpler. The first group of responding stations provides a fairly accurate location of the hypocenter of the event if it occurs within the network. Also a map of shaking can be produced directly from the observations rather than from a model-based interpolation that depends on knowing the epicenter of the event.

In Figure 3, a real example of a dense network is shown. In this case, it is an exploration network that happened to capture a nearby earthquake. The recorded wavefield is unaliased, and it is clear that there are significant and rapid lateral variations in the peak accelerations.

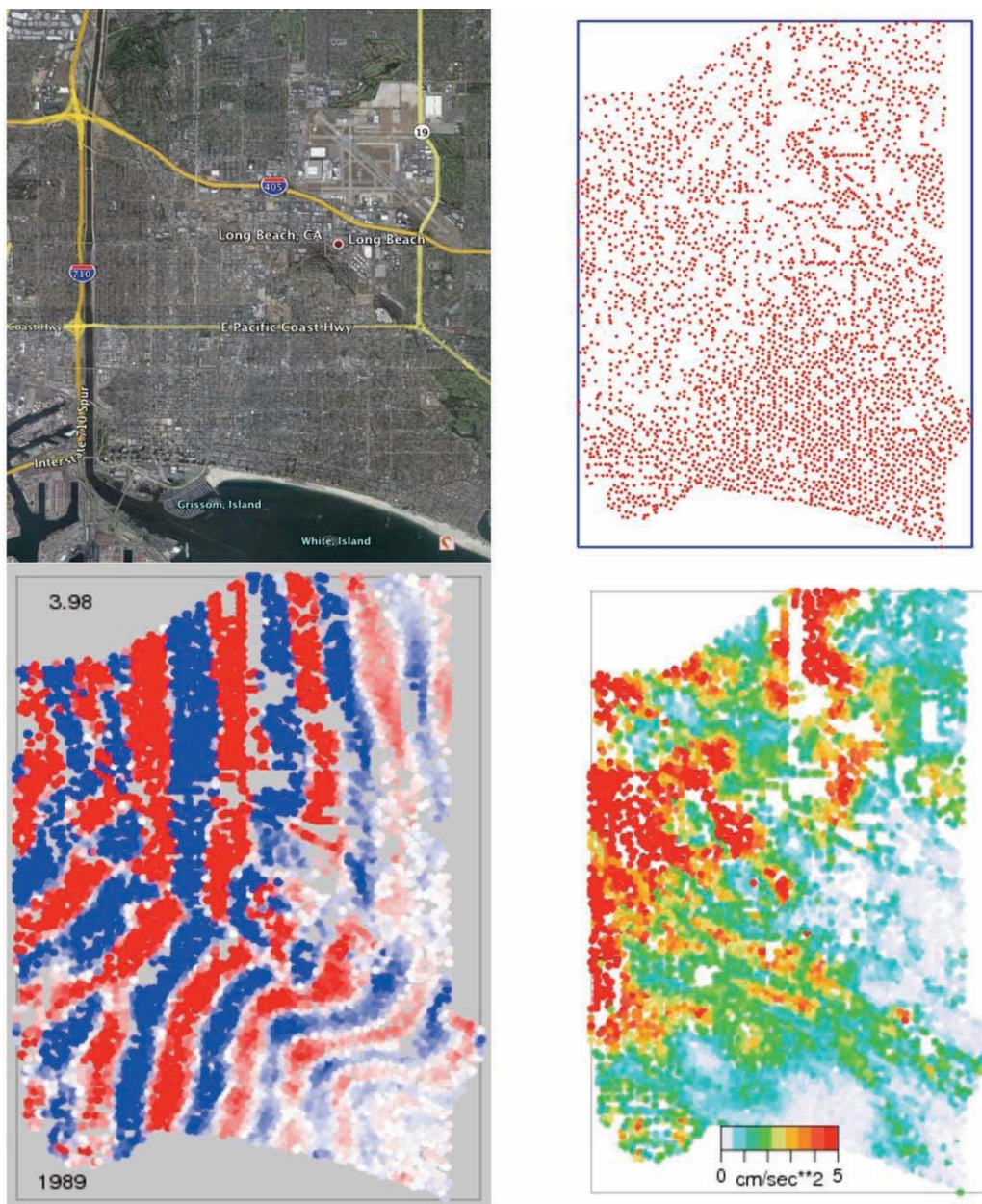


Figure 3. A real example of a dense array. The array consists of over 5,000 sensors in a 7×10 km area. It was deployed by NodalSeismic Inc. on behalf of Signal Hill Oil Co. and was designed for active source imaging of the Signal Hill oil field. The upper panels show the location of the stations. The lower left panel shows a time-slice of the S-wave wave field due to a magnitude 2.5 earthquake located approximately 5 km to the west of the array (red is positive and the numbers refer to the frame and time of the slice). The lower right panel shows the variations in peak acceleration due to this earthquake.

Low-cost sensors

The technology change that has produced low-cost sensors is MEMS. The accelerometer version of these ‘sensor-on-a-chip’ devices uses capacitance variations induced by motions and were developed in 1960’s. Their development as low-cost devices was driven by their widespread use in air-bag systems, disk-drive protection devices and computer game controls. They are now commonly included in smart phones and mobile computers.

MEM accelerometers vary from very low-cost low-resolution high-noise devices to sensors that have performance that is comparable to expensive force-feedback sensors [Holland 2003]. Sensors with 70 mgal sensitivity at 1 Hz are available for US\$100. Cell phones are typically equipped with sensors that are about ¼ of this sensitivity. In the CSN, the initial deployment of Phidget sensors (www.phidgets.com) have a sensitivity of 70 mgal with a 16-bit digitizer and a dynamic range of ± 2g. The packaged sensor is shown in Figure 4, along with its noise curve. For comparison, the response of a smart phone sensor is also shown.

Communications with the sensors

Communications with the sensors is generally over the Internet. This reduces costs but does introduce some security and robustness issues. To connect the sensors to the network, there are a number of options, but for most sites a host computer is used. The sensor package connects to the host via a USB port, and a client program analyzes the samples and communicates with the central server. The client program has minimal impact on the performance of the

host, but it does require the host be functioning all the time to provide real time monitoring. The advantage of using a host computer is that network connectivity problems are solved by others. This allows the client installation program and process to be fairly simple.

Ideally, the sensors would not use a host computer, but would rather directly connect to the Internet. We have successfully ported the client software to a small single board computers (SBC) but this does not solve the Internet connectivity problem. The SBC’s, whether using wired or wireless connections, need to negotiate a variety of protocols that often require passwords. This makes volunteer (i.e. non-expert) installation problematic. Using the cell-phone infrastructure would obviate a number of these problems, but with the current price structure for this type of communication, this is not practical. It may be suitable in countries outside of the USA. Some form of the SBC-based sensor will likely become the preferred configuration as always-on desktop computers become less popular.

Software design

The CSN software is divided into client and server components. At the moment, the server part is also subdivided into real-time and archiving sub-systems, although our longer-term plan is to join these.

Client software

The purpose of the client software is to retrieve the sampled data from the sensor system, perform a limited set of processing on the data, and send the results (and

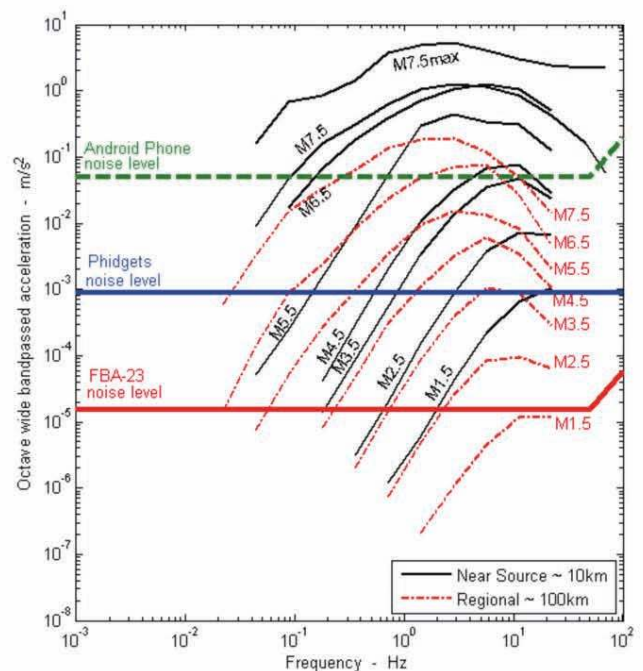


Figure 4. Low-cost MEM sensor used in the CSN. The sensor used is a Phidget 1056. The response of this sensor and that of a typical smart phone are compared to a standard high-quality force-balance accelerometer in the right panel. The response for various sized earthquakes at two distances is shown for reference, which is adapted from Clinton and Heaton [2002].

possibly the data) to the central server. In the current implementation, the data samples are decimated to a data rate of 50 samples per second (sps) and placed in a ring buffer. A detector algorithm that uses a variation of the standard ratio of short-term-average over long-term average [Earle and Shearer 1994] is used to pick events. The adjustable parameters in the detector are the lengths of the short- and long-term averages and the threshold of detection. We plan to dynamically adjust these parameters through machine learning, which will be discussed later. Time for the hosted sensors is determined by a local Network Time Protocol (NTP) server [Mills 1990, Frassetto et al. 2003]. We tested the local clocks on the host computers, which were supposed to sync to an NTP server and to generally available NTP servers, but found that neither were sufficiently accurate. We are currently running our own NTP server and this has stabilized most of the clocks, but we still occasionally have 1-10 second jumps that appear to be introduced by some older operating systems.

When the client has detected an event, it measures the peak amplitude in the next second and sends this information along with the time of the detection to the server. We believe that it is important that this initial information be sent as quickly as possible in order to precede a possible (maybe likely) failure of the network infrastructure. The client continues to look for the peak amplitude and send updates as the event proceeds.

The clients will send their raw data to the server when they are requested to do so. For some sensors, the parameters are set such that all of the data are sent every few minutes, forming a continuous stream of data. These data streams are very important for research on the detection and processing algorithms and for scientific research on the earthquakes themselves. In other cases, when bandwidth is an issue, the sensors will send a time window of the data when requested by the server. The request is communicated as part of the ‘call-home’ procedure.

When initially downloaded, the client software performs the installation task and requests certain information from the volunteer such as location as determined by a Google map, floor of the building, building type, and contact information. An identification key is also obtained from the server to authenticate future data exchange.

The final task for the client is to contact the server at regular intervals (e.g. hourly) to report its state of health and request any software or parameter updates. With this ‘call-home’ or ‘heartbeat’ mechanism, the entire client code can be changed, and various parameters updated. This is also the mechanism whereby the server can request that the client send the raw waveform data for a particular time segment. This task also fills the important role of letting the server know which sensors are functioning at any given time.

Server software

The role of the server is to receive detection and parametric data from the sensors and to process it to produce a map of the peak ground motion, along with other related products. This information will then be broadcast to the general public and emergency responders. To handle the very impulsive load generated by earthquakes, the server needs to be able to dynamically add computational and I/O resources. It also needs to be robust with respect to failure by the very event that it is attempting to report on, which means it probably needs to be located outside the region of concern.

To achieve the dynamic and robust qualities, it was decided to use distributed cloud computing [Armbrust et al. 2010, Mell and Grace 2011] for the server. In our initial configuration, we are using the Google cloud, with the Google App Engine as a development platform. The robustness is achieved through the global distribution of redundant ground servers, and the dynamic loading is through the structure of the database of the Google App Engine. The database is highly denormalized (i.e. one table instead of several separate tables) which is a change for most seismic software. Tutorials on the Google database (Big Table) and programming the App Engine can be found at <http://labs.google.com/papers/>. In theory the cloud will automatically handle the redundant storage of data, so we only need to deal with one system (i.e. no backup system). In measurements provided by Google engineers, but otherwise undocumented, the redundant storage will occur very rapidly. One other features of the cloud is the ability to create additional networks anywhere in the world by simply replicating the particular instance. The downside of the cloud is that some aspects of the environment change with time as is to be expected with an evolving system.

In standard (sparse) network software the picks are normally associated with events with a fairly complex code that is often termed an ‘associator’. With the CSN, the density of the network allows us to use a simpler system based on ‘geo-cells’ (latitude–longitude boxes). The system simply counts the numbers of detections within the geo-cells within time slices, and when that number exceeds a threshold number, the geo-cell declares an event. When a sufficient number of geo-cells detect an event, a map of the ground motion for the whole region is produced. Thresholds are determined in order to maximize the detection probability, while minimizing the number of false alarms. Since this map is not dependent on the location of the event, only that its effects are within the area of interest, there is no need to initially do the association and location steps. The geo-cells can be used in a telescoping multi-scale approach to determine how widespread the shaking it is. The size of the geo-cells and time slices are tunable parameters of the network. In the initial Pasadena test the geo-cells are the size of a few blocks, and the time slice is the transit time of a

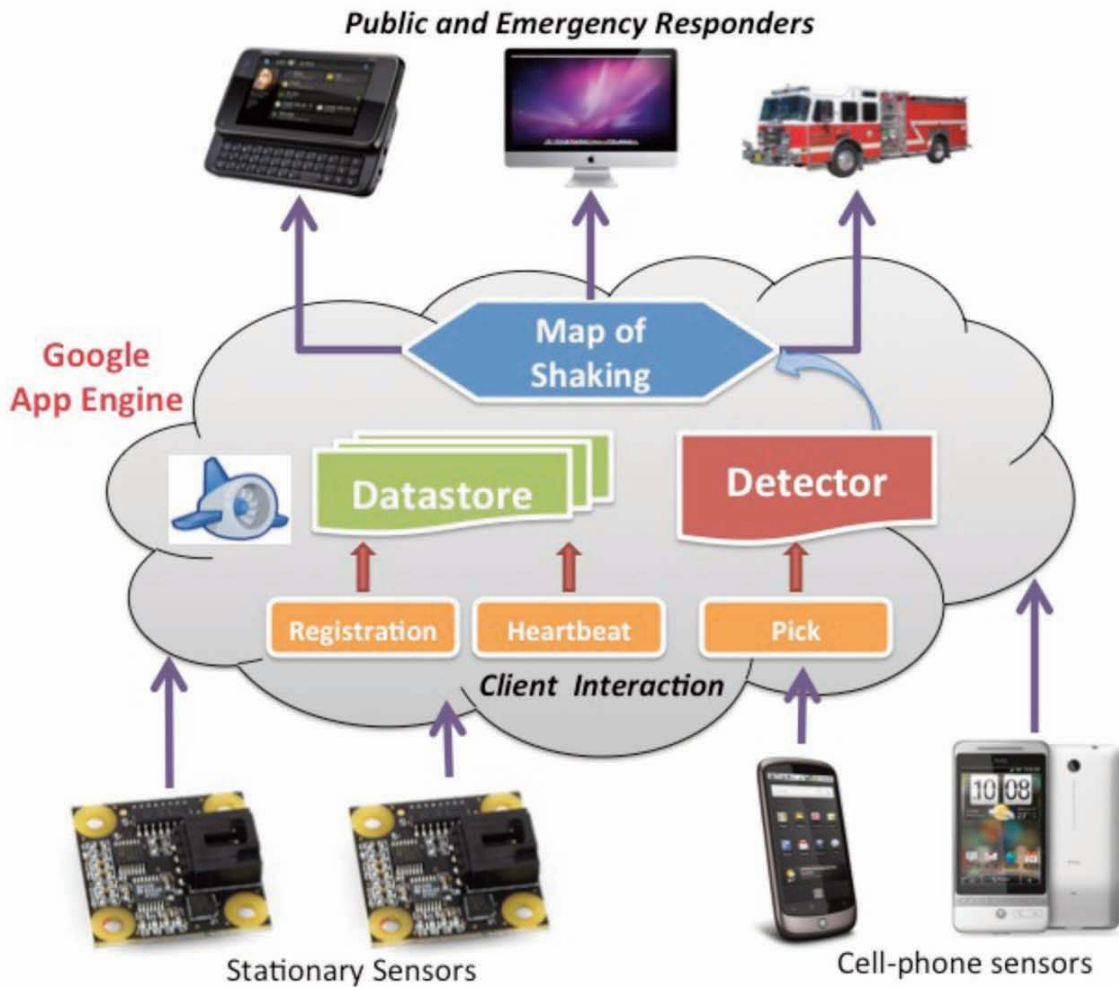


Figure 5. A schematic of the client/server interaction. The functionality of the server is shown. The event detection is done by geo-cells as described in the text. The heartbeat is database of regular check-ins by the sensors. The datastore is the cloud database. The advantage of a ‘cloud-based’ server is robustness and extensible load capability.

P-wave across the cell. The threshold count is also a parameter that is likely to vary laterally across the network.

The system can respond very quickly, and the goal is produce the map within seconds of the shaking, so it has the potential of reaching emergency responders before the network connectivity fails. However, even if this is not the case, the use of the cloud environment should allow the information to be sent to outside responders. A schematic of the server implementation is shown in Figure 5.

The server also requests and receives the waveform data, which is archived. Presently a conventional land-based server handles this, because cloud-based storage is too expensive. However, it is expected that this will evolve so that the waveform archive is also maintained in the cloud.

Data archive

The waveform and other parametric data are sent to a standard archive shortly after they are received and processed. The waveform data are entered into an archive wave pool and the metadata into a database. They are then available to scientists working on the CSN project for further

analysis. Due to privacy concerns of the sensor-hosting entities, these waveform data are not generally available, however, the waveforms for detected events will be made available to the scientific community. One challenge of the archive is the mobility of the sensors and the ‘naming’ of stations. With the use of mobile devices as the sensing platform, we need to be able to handle rapidly changing station coordinates, which also has implications for the naming of stations. In traditional networks, the stations are re-named when they are moved, but that is not practical where the stations are frequently on the move, such as the case with cell phones with MEMS sensors that are discussed below. At the moment, the station location is kept as an attribute of a particular waveform and not the station (i.e. the database is denormalized), but the station name is kept fixed. This approximates the station location, by its position at the start of the waveform. Keeping consistent names for stations in a network where the sensors move around (and disappear) is a challenge that we have not completely solved. The solution may be to abandon names and use the coordinates as the station tag.

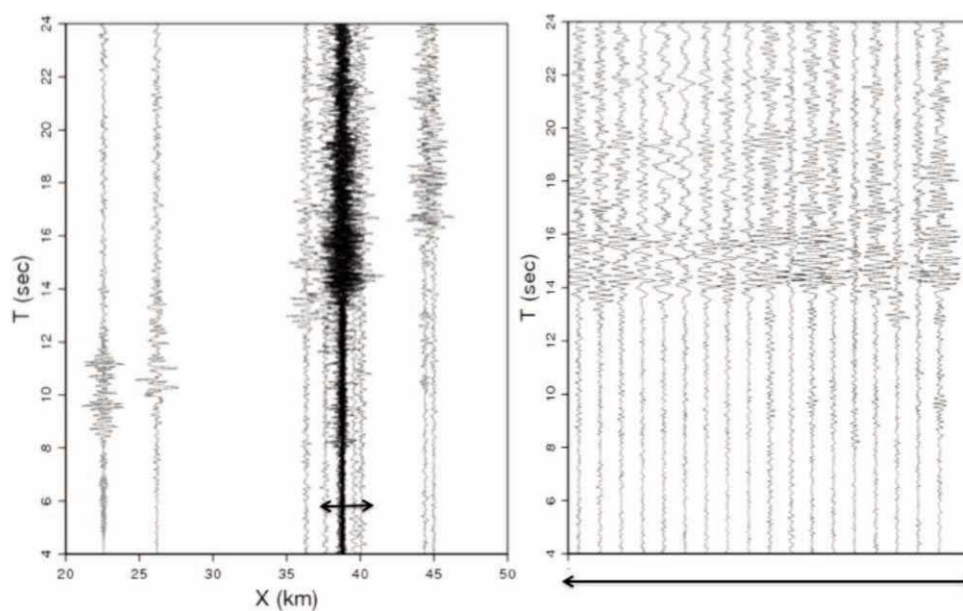


Figure 6. Sample recording from CSN. The earthquake is a Ml 4.1 near Newhall, CA, USA, approximately 38 km from the center of the CSN network. The east component of data is shown. The left panel shows the data in distance from the earthquake, while the right panel shows a blowup of the region denoted by arrows. The traces in the left panel are self-normalized, while those in the right panel have a single scale factor. The data are band-passed at 1-10 Hz. The amplitude variations in the right panel are the main effect that the CSN is trying to measure. There is clearly one station in the right panel that has a timing error of about one second.

Security issues

There are a number of security and privacy issues that are not normally encountered in a closed seismic network. The first is ensuring that the host computers are not compromised by entities spoofing as the server. To minimize this issue, all communications in the CSN system are initiated by the clients themselves and only with a trusted host. This means that software updates and request for the waveforms wait until the client makes a regular state-of-health contact. To minimize spoofing the server with bogus earthquakes, the server only accepts information from clients that have a key, which is assigned at the time of registration (software installation) of client.

Privacy issues

The privacy of information on the sensor host is increasingly becoming a major issue. It is relatively straightforward to keep the contact information separate from the seismic database and in a secure place available only to the administrators of the network, but the precise location information is a necessary part of the meta data for the seismic system. To minimize the broadcast of this information, public displays of the network show only sensor locations at the resolution of a geo-cell (about one block in area), and within a geo-cell only the number of sensors is shown, not the individual locations. This still gives an overall impression of the distribution of the network, but doesn't give precise locations. Note that we chose this procedure after experimenting with adding a small random component to the displayed locations, which seemed to confuse people more than it helped. The geo-cell display concept is further enhanced for mobile devices such

that only geo-cells that have two or more mobile devices will be displayed, which should make tracking of people through their sensors practically impossible.

To minimize the issues of distributing the real time waveform (with locations), we plan to only make available extracts of the data for felt earthquakes, and only after a delay. This appears to be an unfortunate consequence of an open network in a privacy-concerned environment.

Prototype network

A prototype of the CSN network is currently being installed in the vicinity of Pasadena, California, USA. The purpose of this network is to demonstrate the viability of the community-hosted sensors and the open network design of the CSN. To date, approximately 100 sensors have been distributed. A map of the distribution and the current size of the network is at <http://map.communityseismicnetwork.org>. The sensing, detecting and archiving aspects of the network have been implemented and are being tested. In Figure 6, one of the few events recorded by the CSN is shown. The event is a magnitude 4.1 at a distance of 38 km from the center of the network. The data show the general noise level on the sensors that can be expected on this type of network. The timing error that is evident on one of the stations is due to the problems mentioned above. The sensors did not detect the April 4, 2010, 7.2 El Mayor Cucapah earthquake in Baja, Mexico some 300 km away.

Future directions

The prototype CSN network represents a small part of the envisioned scope of the planned CSN network. The

future plan is to expand over the entire urban Los Angeles region, and eventually other urban areas that are subject to significant seismic hazard. In terms of developing the network software and sensors, we are working on tuning the network parameters with machine learning, detecting the state of health of buildings, and adding the sensors in cell-phones.

Machine learning

The sensors for the CSN are installed by the volunteers themselves and as a result it is expected they will be placed in a wide variety of noise and vibrational environments. The sensors themselves have some self-configuration ability for orientation. They can detect the vertical component because they can sense the acceleration of gravity and do a software rotation to align it with the z-component. Currently, the horizontals are not oriented, other than they are in the horizontal plane and are perpendicular to each other. In the future, if an onboard magnetic compass is available (as it is with the Phidget sensor), it can be used for determining the compass directions of the accelerometer horizontal axes.

The network will use machine learning in order to optimize the detection parameters. This will be implemented when the network has densified. The algorithm for setting the thresholds or sensitivity of picking takes into account the detection history for a particular sensor. It adjusts the threshold up or down to bring the sensor in line with the average for the whole network. This could even be made time-of-day dependent as its performance characteristics become more evident with time. Adjusting the parameters of the picking algorithm (such as the short and long-term filters) will be more complex because these will depend on the performance relative to the background noise and on the history of detecting earthquakes of different sizes (hence frequency content). A first approach towards using machine learning to optimize detection performance in the CSN is described in Faulkner et al. [2011]. In addition to tuning the sensors themselves, machine learning can also be exploited to optimize the network parameters such as the size of the geo-cells and the thresholds within each cell.

One planned feature of the network is to incorporate the ability of the clients to initiate a test earthquake to perform end-to-end test of the system; something that is not possible with most seismic networks. To accomplish this, the details of the event, including synthetic waveforms will be downloaded into the clients the day before the scheduled time as part of their regular call-home communications. Then at the scheduled time the synthetic data will replace the real data stream, and the test earthquake will be simulated. The entire system including the detection functions on the clients and the capacity of the server will be tested.

Networks in buildings

One of the proposed applications of the CSN is in providing dense instrumentation of buildings. This is to measure the level of acceleration that the individual floors have experienced in an earthquake, and also to monitor the state of health of the building by looking at the variations in the modes of the building before, during and after an earthquake [Clinton et al. 2006]. This is a new area of research that will require a dense array such as the CSN to provide the necessary observations. We have installed such a network in the Millikan Library (a ten story building at Caltech) for structural monitoring. The sensors are capable of detecting the fundamental modes of the building with both forced vibrations and ambient noise. Communications within buildings can be challenging for a variety of technical and practical reasons, so we are currently investigating using the electrical wiring system as a communication network (IP over power), but at the moment, noise appears to significantly limit the range. If this can be made to work, it will greatly simplify the placement of sensors, and their communications in buildings.

Cellphones and mobile devices

Smartphones are ubiquitous in today's society, and most are equipped with a motion sensor. While the quality of this sensor is not as good as that of the stationary sensors, its use for gaming and other applications is demanding more precision. The newest generation of phones and mobile devices allow programs to be run in the background, which allows a modified version of the client described above to function on these devices.

The obvious problem with mobile devices is that they routinely generate signals that are much larger than earthquakes through regular activities. One straightforward solution is to detect when the phone is at rest, and only use data from the sensors when the phone is in this state. This is the approach of the iShake project [Dashti et al. 2011]. A more challenging problem is to separate the human-generated motion from the earthquake signals. We have developed a prototype algorithm to make this separation and discriminate between earthquake motion and acceleration due to regular activities. The algorithm and its evaluation are described in Faulkner et al. [2011]. A CSN app for Google Android phones implementing these algorithms is available as a free download through the Android Market store (<https://market.android.com/details?id=edu.caltech.android>).

Examples from a dense network

During the first half of 2011, NodalSeismic Inc., installed a dense seismic network in Long Beach for the Signal Hill Oil Company. This network consisted of approximately 5,000 autonomous sensors distributed over a 7×10 km area. This network recorded a few small

earthquakes that were within a few kilometers of its center.

While the sensors on this network are the standard seismic exploration velocity sensors, this deployment gives some indication of what might be observable with a dense earthquake network such as CSN. Figure 3 shows the distribution of receivers and shows a snapshot of the S-wave as it crosses the network. The peak accelerations are also shown. It is the variability that we see in this figure that confirms the premise of the CSN – that there is significant lateral variability of ground shaking and that it needs to be measured on a fine scale. This network has also been an excellent test bed for the picking algorithms and the geo-cell concept for detecting events.

Acknowledgements. The development of the Community Seismic Network is supported by the Betty and Gordon Moore Foundation. This research is also supported by NSF Award CNS0932392. We thank NodalSeismic Inc. for the samples of data used in Figure 3. The manuscript was significantly improved by suggestions from the associate editor, Paul Earle, and a reviewer, Jesse Lawrence.

References

- Armbrust, M., A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia (2010). A view of cloud computing, *Communication of the ACM*, 53 (4), 50-58; doi: 10.1145/1721654.1721672.
- Clinton, J. and T. Heaton (2002). Potential advantages of a strong-motion velocity meter as opposed to a strong motion accelerometer, *Seismol. Res. Lett.*, 73, 332-342.
- Clinton, J., S. Bradford, T. Heaton and J. Favela (2006). The Observed Wander of the Natural Frequencies in a Structure, *B. Seismol. Soc. Am.*, 96 (1), 237-257; doi: 10.1785/0120050052.
- Chung, A.I., C. Neighbors, A. Belmonte, M. Miller, H.H. Sepulveda, C. Christensen, R.S. Jakka, E.S. Cochran and J.F. Lawrence (2011). The Quake-Catcher Network Rapid Aftershock Mobilization Program Following the 2010 M 8.8 Maule, Chile Earthquake, *Seismol. Res. Lett.*, 82 (4), 526-532.
- Cochran, E.S., J.F. Lawrence, C. Christensen and R. Jakka (2009a). The Quake-Catcher Network: Citizen science expanding seismic horizons, *Seismol. Res. Lett.*, 80, 26-30.
- Cochran, E.S., J.F. Lawrence, C. Christensen and A.I. Chung (2009b). A novel strong-motion seismic network for community participation in earthquake monitoring, *IEEE Instru. Meas. Mag.*, 12 (6), 8-15.
- Dashti, S., J. Reilly, J. Bray, A. Bayen, S. Glaser and E. Mari (2011). iShake: Using Personal Devices to Deliver Rapid Semi-Qualitative Earthquake Shaking Information, USGS Final Technical Report, G10AP00006, February 28, 2011.
- Earle, P.S. and P.M. Shearer (1994). Characterization of global seismograms using an automatic-picking algorithm, *B. Seismol. Soc. Am.*, 84, 366-376.
- Evans, J., R. Hamstra, C. Kundig, P. Camina and J. Rogers (2005). TREMOR: A Wireless MEMS Accelerograph for Dense Arrays, *Earthq. Spectra*, 21 (1), 91-124.
- Field, E.H. and S.E. Hough (1997). The variability of PSV response spectra across a dense array deployed during the Northridge aftershock sequence, *Earthq. Spectra*, 13 (2), 243-257.
- Faulkner, M., M. Olson, R. Chandy, J. Krause, K. Chandy and A. Krause (2011). The Next Big One: Detecting Earthquakes and Other Rare Events from Community-Based Sensors, *IPSN'11*, ACM 978-1-4503-0512-9/11/04.
- Frassetto, A., T.J. Owens and P. Crotwell (2003). Evaluating the Network Time Protocol (NTP) for timing in the South Carolina Earth Physics Project (SCEPP), *Seismol. Res. Lett.*, 74, 649-652.
- Holland, A. (2003). Earthquake data recorded by the MEMS accelerometer: Field testing in Idaho, *Seismol. Res. Lett.*, 74 (1), 20-26.
- Hons, M.S., R.R. Stewart, D.C. Lawton and M.B. Bertram (2008). Field data comparisons of MEMS accelerometers and analog geophones, *The Leading Edge*, 896-903.
- Horiuchi, S., Y. Horiuchi, S. Yamamoto, H. Nakamura, C. Wu, P. Rydelek and M. Kachi (2009). Home seismometer for earthquake early warning, *Geophys. Res. Lett.*, 36, L00B04; doi: 10.1029/2008GL036572.
- Mell, P. and T. Grace (2011). The NIST Definition of Cloud Computing. National Institute of Science and Technology. Recommendations of the National Institute of Standards and Technology, Publication 800-145.
- Mills, D.L. (1990). On the accuracy and stability of clocks synchronized by the Network Time Protocol in the Internet system, *ACM Computer Communication Review*, 20, 65-75.
- Mougenot, D., A. Cherepovskiy and L. Junjie (2011). MEMS-bases accelerometers: expectations and practical achievements, *First Break*, 29 (2), 85-90.
- Wald, D.J., V. Quitoriano, T.H. Heaton, H. Kanamori, C.W. Scrivner and C.B. Worden (1999a). TriNet "ShakeMaps": Rapid Generation of Peak Ground Motion and Intensity Maps for Earthquakes in Southern California, *Earthq. Spectra*, 15 (3), 537-556.
- Wald, D.J., V. Quitoriano, L.A. Dengler and J.W. Dewey (1999b). Utilization of the Internet for Rapid Community Intensity Maps, *Seismol. Res. Lett.*, 70 (6), 680-697.

*Corresponding author: Robert W. Clayton,
California Institute of Technology, Seismological Laboratory,
Pasadena, CA, USA; email: clay@gps.caltech.edu.