Proceedings of TMCE 2012, May 7–11, 2012, Karlsruhe, Germany, Edited by I. Horváth, Z. Rusák, A. Albers and M. Behrendt

UBIQUITOUS CURTAIN-WALL ASSEMBLY-SUPPORT BY MOBILE COMPUTING AND TOTAL STATIONFIRST

Leon Kos
Computer Aided Design lab. LECAD
Faculty of Mech. Eng., University of Ljubljana
Slovenia
leon.kos@lecad.fs.uni-lj.si

Simon Kulovec
Joˇzef Duhovnik
Computer Aided Design lab. LECAD
Faculty of Mech. Eng., University of Ljubljana
Slovenia
{fsimon.kulovec, jozef.duhovnikg}@lecad.fs.uni-lj.si

Viktor Zaletelj
Research Department, Trimo d.d.
Slovenia
viktor.zaletelj@trimo.si

ABSTRACT

Curtain-wall systems have become a classic approach for covering facades in different structures. Pushing aesthetics to the limit requires tight intercurtain gaps and thus hard to assemble process that presents a challenge in technological and organizational terms. To support different information flow requirements effectively, ubiquitous computing must accompany such a process from production to the assembly on site. Support consists of specialized software that can be used for planning, simulations, premeasurements, analysis, assembly tracking with wall alignment, progress planning and animation of sensor carrying wall on site. A unification of the above principles is presented for use in office and duringassembly

KEYWORDS

Curtain-wall, ubiquitous assembly”.

1. INTRODUCTION

Curtain-wall systems should fulfill architectural and aesthetic requirements, sound and thermal insulation requirements and they have to be durable. The façade system is basically a mesh of rectangles with uniform facade elements with various corner joints. Without effective measurement support assembly of such structures can be a tedious task This is especially true for situations with tight tolerances, i.e. small interpanel gaps, large non/uniform structures and complex joints. It means that there are different types of basic elements with different positions of measurement points to be tracked. As the relative position of mounting represents the basis for measurements, it is necessary to take account of differences in the types and positions of elements already in the mesh model. Together with curtain-wall elements with properties assigned from real-world measurements it is necessary to introduce additional (global) points for the support of measurements. They represent reference prisms, designating the coordinate system of a mobile total station that is used for measurements on site. The objective of the application presented – besides the manipulation and optimization of orthogonal mesh structures – is to provide a basis for optimum installation of fac¸ade elements for free-form structures (Kos et al., 2010; Kos et al., 2009) that are even harder to assemble due to non-uniform elements, large variations in form and consequently with larger deviations due to complex assembly and load bearing.

A vast majority of patents in the area of the curtainwall systems concentrates in frame construction alone, without notion to the installation complexity. Usually, measurements with total station is used for surveying of fac¸ades (Ord´o˜nez et al., 2010) where accuracy is not a very important aspect when representing building construction site (Shih, 2003). Cloud scanning and inspection presented by (Shih & Huang, 2006) is another application where accuracy of total stations is favorable, but not used in a sense of precise measurements that can be acquired when using prisms as targets. For rough project planover accuracy and inspection (Shih et al., 2004) goals largely differ from our intent of "ubiquitous" support of curtain-wall assembly.

We propose a system of ubiquitous computing approach to the common goal (assembly) at the construction site and believe that this is "a way to go" for similar applications. Within ubiquitous (Gerritsen & Horv´ath, 2010) approach to process control we reuse well-established computing devices like smart phones, laptops, desktops and specialized devices (theodolite, sensors, actuators) that are interconnected with secure communication over WiFi. Such approach is so versatile that we cannot simply underpin it as mechatronics or refurbished intelligent system with inter(net)-device communication. As mentioned before, our further developments in free-form buildings is required to be be assembled with ubiquitous support of measurements at production and at the construction site as the overall complexity is much higher than curtain-wall presented here.

2. CURTAIN WALL PLANNING FOR ASSEMBLY

Assembling curtain walls triggers the problem of curtain wall tolerance and load-bearing structure tolerance, as well as the problem of taking account of acceptable loads that affect the effectiveness of assembly and reaching target construction requirements. The problem of reaching target tolerances is increases with decreasing gaps between elements. One of the primary goals of program application for planning of the curtain wall assembly onto any given structure is the control of tolerances through the entire process. Another aim is identification of required structure with the goal of increasing the size of panels and simultaneously reducing gaps between them.

Within the planning process it is also vital to pay attention to the comparison of mesh structures, particularly the analysis of larger mesh assembly as the errors can stack-up. Assembly simulation is primarily in order to reduce errors before the physical installation of a curtain wall system. Solving all assembly aspects is provided by developing specialized software as shown in Fig. 1. According to measured structure parameters, it enables planning, simulation and optimization of necessary modifications on the structure. In order to allow for

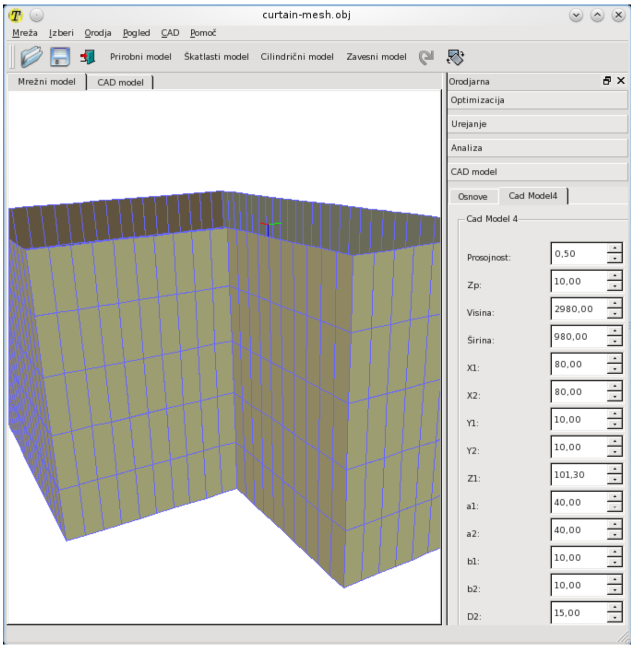

Figure 1 Desktop application for planning and analysys of curtain wall systems. Mesh carries basic assembly information and provides coordinates for parametric CAD model generation shown on right toolbox.

all predicted possibilities, specialized software with three different functionalities is developed;

Firstly, the functionality of the software package is made which allows realization of any mesh model physically, with planes. There is no need for a connection between building blocks but if it exists, it means that they are topologically compatible and it is easy to build parametric models with them. For the purpose of reading, various polygonal mesh formats are provided and specialized converters and modelling filters are made in order to allow importing mesh structures.

The next software package functionality involves adding attribute information on the mesh. Attributes are necessary for a comprehensive description of installing curtain walls in space. Parametric generation of individual panels is predicted, taking account of production and assembly tolerances. Additionally, it is possible to import STEP and IGES files of the load-bearing structure into a geometric model with the option of translation, scaling and rotation of the whole model for the precise alignment with the generated model. Fig. 2 shows a detail of generated CAD model with the fixture elements. Panels are hanged on support plates that allow position calibration of the position after initial installation. Positional calibration is carried out in

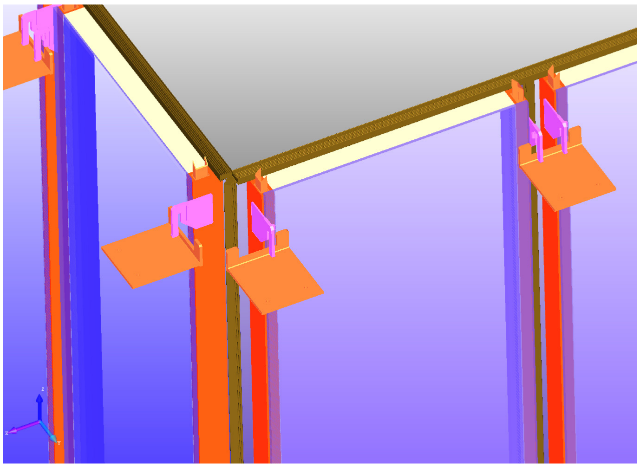

Figure 2 Detail of curtain wall model generated from mesh model.

two directions with screws that allow independent directional movements. As curtain wall elements are stacked and joined vertically, there is no need for calibration at the bottom of the panel. All adjustment can be made at the top of the panel. Nevertheless, real world practice shows that tolerances can significantly impact assembly as some of the dimensions like load-bearing structure is usually not controlled by curtain wall installers. To compensate for this, simulation of deviations is carried out by means of varying panel's basic parameters. It also allows assemblability simulations with the Monte Carlo method, for which production and assembly variations are additionally assumed.

The last planning software functionality provides support to measurements and basis for monitoring deviations during the assembly. Desktop use for planning is a way different experience that assembly at the construction site where limited user comfort regarding the information access is available. To accommodate difference between desktop and mobile use the assembly plan is exported into database that allows various kinds of implementations when using all kinds of devices at the assembly location.

3. UBIQUITOUS COMPUTING FOR ASSEMBLY

In practice different approaches are used for assembly. Influencing factors are dependent on building height, overall construction site size, panel details, accessibility, time pressure, logistics, etc. It is an imminent goal to optimize assembly process to the extent possible that will ensure cost effectiveness in a timely manner. Curtain wall assembly is often a

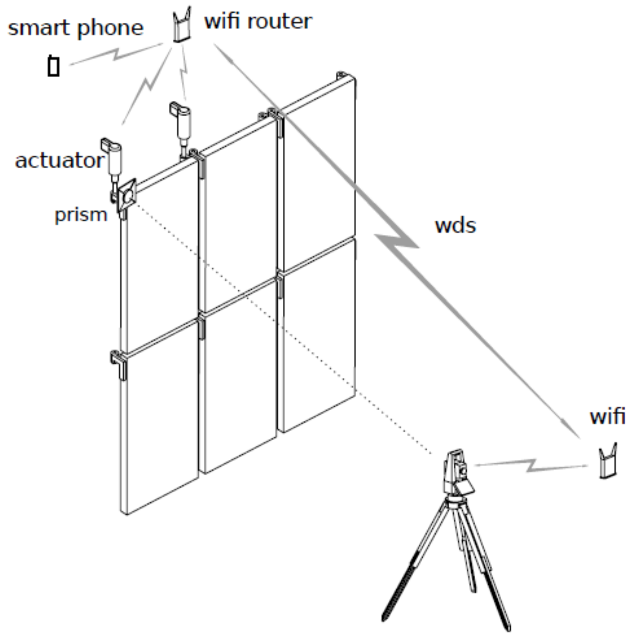

Figure 3 Communication support for curtain-wall assembly is carried with WiFi interconnect.

bottleneck of the entire project as it is delicate to be applied in concurrent while building is still growing. Thus it is normally planned as the last stage performed from outside. At the construction site we propose a set of personal and specialized devices that are capable enough for the following tasks:

1. Ensuring that the panel holders are within tolerances before assembly takes place.
2. Real-time supervision of the panel assembly.
3. Fast and accurate identification of the panel position.
4. Remote control of the total station without the need of the operator.
5. Concurrent work of several teams with crane support for panel transport to final position.

To date most common communication protocol that has fair coverage range is WiFi. Open air range is within the limits of the construction site. Fig. 3 shows basic communication properties of the proposed interconnect. WiFi routers carry communication on a long range and provide nearby communication to other devices. Wireless Distribution System (WDS) is aWiFi protocol that provides a "backbone" for WiFi range extension. This means that WiFi signal can be extended with several WiFi routers if there are additional requirements besides assembly that works in visible range that doesn't impose obstacles in communication.

For nearby communication other wireless protocols are more practical to implement when specialized control software development is required for non-standard devices. An example of devices shown in Fig. 3 is:

1. Total station (motorized theodolite) with USB communication connected directly to
2. WiFi router that extends range with WDS to another
3. WiFi router with Bluetooth dongle that provide adjustment data directly to
4. modified standard battery-powered wrench actuators.
5. Other devices like smart phones, laptops and tablets can additionally be used for accessing measurement data.

Such approach benefits over others with versatility that at the end means ubiquitous computing. Namely, specialized radios that are provided as total station (TS) addons with remote controls are just range extenders that still require development of licensed software that is run within TS. This possess limitations by TS operating system and restrains general usage. When using TS in operator-less manner a precise measurement plan is required as in needs to find prism located on mounting panel. TS with automatic target recognition (ATR) need to be pointed to given prism position, to be able to return measurement. Additional benefit of the proposed solution is power management of the battery powered TS as shown in Appendix code example. In principle TS operates internally with polar coordinate system that consists of two axes. Geo-spatial coordinates normally used are just recalculated out of internal angles and measured distance. Fig. 4 shows TS internal coordinate system.

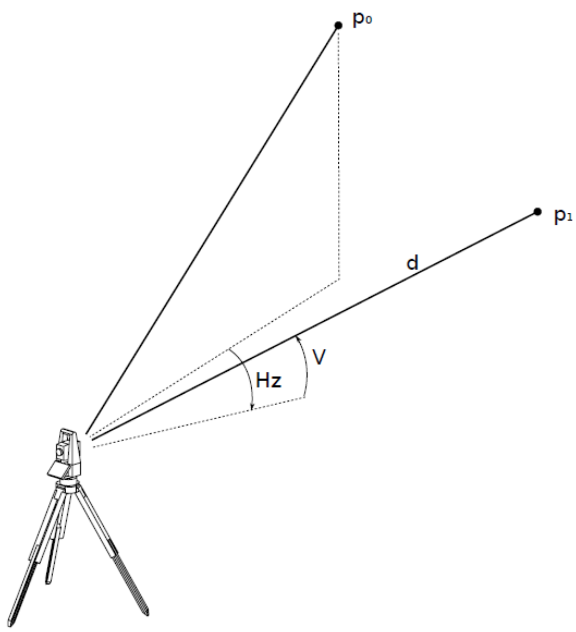

Figure 4 Resection of the coordinate system.

It turns out that interface for driving i.e. pointing TS to selected direction requires providing Hz and V angles. So, various coordinate systems provided by TS are more or less useless for our application. But, this could not be regarded as an obstacle as we already agreed to take measurement control outside of theTS.

Suitable devices that can control TS are variousWiFi routers. Surplus of such devices that are usually used as domestic routers can be converted into computing devices by modifying factory firmware. Several open-source projects (OpenWRT, NSLU2, DDWRT, ...) already recognized WiFi routers as a potential target for extending their hiden processing capabilities into different "embedded" applications. When changing Linux operating system that all supported routers run we can convert router to perform standard server-like tasks and develop our specialized applications in programming languages like Python and C. The easiest thing to do with is to provide HTTP server. This enables smart phones to connect viaWiFi and access required (measurement) data concurrently. WWW pages can be prepared especially for this "limited" devices. For communication between TS and WiFi router USB cable is used, so proper router with USB is required. With WDS communication between two or more routers computing power can be shared but usually it is just balanced to provide different services. For example, SIP protocol for voice communication can be used by virtually all GSM mobile phones with WiFi. Site with WDS backbone can thus provide free voice communication between workers. With PBX gateway this can be extended even further. Powering routers with battery does not pose major obstacle in terms of power consumption and mobility. Getting TS measurements concurrently with various "terminals" from router acting as measuring server is therefore inherently supported. This means that with prepared measurement plan and enumerated prism location (i.e. measuring points) assembly team can follow tolerances when installing panels. With limited outside connectivity one can use USB key to install site measurement plan to router database server. Tablet PCs can be used when higher graphics capabilities are required.

Measurement differences are just information for calibration to be performed to hangers mounted on the floor slabs. As the panels built from glass are heavy it is difficult to perform manual adjustments on calibration screws with wrench. Each panel uses two hangers that needs to be simultaneously adjusted. That's why manual adjustments, although possible are not favorable as the can introduce additional stress to the panels. To compensate this we propose electrical wrench that are remotely controlled. Driving both wrenches requires two construction workers that drive calibration screws. Standard cordless wrenches were updated with motor control and rotational encoders for precise turning. For rotational synchronization interconnect is used. Communication to nearby WiFi router is implemented with simple one-way wireless protocol with transceivers within ISM band. Required adjustment in each direction is calculated from TS measurement. With know adjustment screw pitch and tight loop that is provided by fast communication between TS and wrenches this semi-automatic concept is fast and precise enough for effective assembly process.

Driving TS completely with software installed on WiFi router means that functionality provided by TS for establishing coordinate system is not used any more. Custom software is responsible for creating (global) coordinate system. For initial setup three or more prism positions are required. Usually this coordinate reference prisms are constantly installed at visible positions even when TS is required to move to better position. The procedure of recalculation new local coordinate system when TS is moved is called resection (see Fig. 4). Theory of resection from two points (p0, p1) and calculating TS position from angles (Hz, V) and distance d is fairly simple. So, normally we establish site coordinate system and recalculate local resections to the global one. To compensate for mechanical errors TS provides measurements from two opposite sides (I and II). As this substantially increases time required for measurement it is used for coordinate system and reference measurements only. All compensations that influence TS measurements still remain in the TS domain. Accuracy of the modern TSs is good enough when proper positioning is performed.

4. IMPLEMENTATION DETAILS

Overall process presented concentrates in describing ubiquitous technology for supporting assembly. Engineering side and transportation logistics also contribute to the success of each project. Implementation of software components described were done with various programming languages and concepts of communication. Most applications were prepared for Linux as this is usual platform for embedded devices. Vendor libraries are usually nonexistent when Linux is in question. For communication between router and TS we were forced to develop whole protocol library from scratch. In appendix we show just an example with protocol library calls that demonstrates on shot measurement from TS power on.

Python interpreter is used as speed requirements is well bellow router capabilities. Most wireless routers with N have 400+MHz CPU with 128+MB of RAM are enough for running database and recalculation. When using large applications (e.g. Python) on router internal FLASH is usually too small. As an external storage SD or USB cards can be attached. We restrained from creating applications that could run on mobiles phones and rather provided WWW server that is easier to maintain. The only exception is desktop software shown in Fig. 1 that is multiplatform code based on OpenMesh and OpenCASCADE libraries with Qt interface. WWW server is used for general interaction with TS like prism positioning, calibration, resection and assembly plan review.

Simple graphics representations are also built within WWW server. The only time critical component developed is cordless wrench controller that is based on micro-controller drive with optical encoder loopback. As DC motors are used there, driving electronics and communication presently is done with external control box that is controlled wirelessly from router. Justification for such prototype can be given with high energy consumption of the wrenches that are intensively used during the installation. When tight gaping is requested additional linear encoder is installed during the assembly to measure and equalize errors over entire row.

5. CONCLUSION

While production quality can be controlled by standard procedures, assembly is still considered an experience task. Tight gaping can be achieved at "the final" assembly only if the whole process is under constant monitoring. The approach proposed fulfills such requirements. Supporting software should be brought forth and back from office to the site, carefully matching required functionality to support assembly. Proposed WiFi based ubiquitous process control gives possibilities for access to vital assembly data from multiple parties. This ensures better project tracking and planning. Planning in the office can also simulate variations of dimensions and thus specify appropriate assembly plan for optimal efforts. Future applications with free-form structures require foundations of a presented approach even more. Panel alignment there is far from trivial as the support structure is a moving target that is deformed under the load during installation. Simulation of such structures is therefore a must to avoid unpleasant surprises and project delays; due to fitting problems at the planing stage.

REFERENCES

[1] Bosche, F. & Haas, C. (2008). Automated 3D data collection (A3DDC) for 3D building information modeling. In The 25th International Symposium on Automation and Robotics in Construction Vilnius, Lithuania.

[2] Gerritsen, B. & Horváth, I. (2010). The upcoming and proliferation of ubiquitous technologies in products and processes. Strojniˇski vestnik - Journal of Mechanical Engineering, 56(11), pp. 765–783.

[3] Kos, L., Kulovec, S., & Duhovnik, J. (2010). Support structure optimization for freeform architectural design. In I. Horvath, F. Mandorli, & Z. Rus´ak (Eds.), Eighth International Symposium on Tools and Methods of Competitive Engineering - TMCE 2010 Ancona, Italy: Delft: University of Technology pp. 829– 840.

[4] Kos, L., Kulovec, S., Zaletelj, V., & Duhovnik, J. (2009). Structure generation for free-form architectural design. Advanced engineering, 3(2), pp. 187–194.

[5] Ordoñez, C., Martínez, J., Arias, P., & Armesto, J. (2010). A software program for semi-automated measurement of building fac¸ades. Measurement, 43(9), pp. 1197– 1206.

[6] Shih, N. (2003). The Application of 3D Scanner in the Representation of Building Construction Site. NIST special publication, pp. 337–342.

[7] Shih, N., Wu, M., & Kunz, J. (2004). The inspections of as-built construction records by 3D point clouds. Center for Integrated facility engineering (CIFE).

[8] Shih, N.-J. & Huang, S.-T. (2006). 3D scan information management system for construction management. Journal of Construction Engineering and Management, 132(2), pp. 134–142.

[9] Van Gassel, F. & Schrijver, P. (2006). Assembling wall panels with robotic technologies. In The 23rd International Symposium on Automation and Robotics in Construction Japan: ISARC 2006 pp. 728–733.

APPENDIX

Main Python code that demonstrates basic communication with total station and sample output. Driving protocol between TS and router is based on serial communication over USB.

```
ser = serial.Serial('/dev/ttyUSB0', 19200,
rtscts=0, dsrdtr=1)
COM_SwitchOnTPS()
ser.timeout = 20 # Necessary for long meas.cmds.
IOS_BeepOn(100)
(RC, Name) = CSV_GetInstrumentName()
print "Instrument:", Name
(RC, nRelease, nVersion, nSubVersion) =
COM_GetSWVersion()
print "GeoCOM Release: %d.%d.%d " %
(nRelease, nVersion, nSubVersion)
(RC, unCapacity, eActivePower, ePowerSuggest)
= CSV_CheckPower()
print "Power source status:", RETURN_CODES[RC],
"Capacity:", unCapacity,"%"
(RC, Lambda, Pressure, DryTemperature,
WetTemperature) = TMC_GetAtmCorr()
RC, Temp = CSV_GetIntTemp()
print "Pressure:", Pressure, " kpa, Temp:",
Temp, "C"
(RC, eTargType) = BAP_GetTargetType()
print "Target type:", BAP_TARGET_TYPE[eTargType]
(RC, ePrismType) = BAP_GetPrismType()
print "Prism type:", BAP_PRISMTYPE[ePrismType]
(RC, eMeasPrg) = BAP_GetMeasPrg()
print "Actual distance measurement program:",
BAP_USER_MEASPRG[eMeasPrg]
(RC,) = BAP_SearchTarget()
print "Search target result:", RETURN_CODES[RC]
RC, dHz, dV, dDist, DistMode =
BAP_MeasDistanceAngle(BAP_DEF_DIST)
print "Horizontal angle:", dHz, "rad"
print "Vertical angle:", dV, "rad"
print "Slope distance:", dDist, "m"
IOS_BeepOff()
COM_SwitchOffTPS()
ser.close()

OUTPUT:
Waiting for theodolite to power up:.......OK
Instrument: "TS30+ R1000"
GeoCOM Release: 1.50.0
Power source status: OK Capacity: 35 %
Pressure: 1013.25 kpa, Temperature: 26.304 C
Target type: with reflector
Prism type: Leica Circular Prism
Actual distance measurement program: IR Standard
Search target result: OK
Horizontal angle: 5.43119552542 rad
Vertical angle: 4.74131622831 rad
Slope distance: 3.6323 m

Power OFF
```