UMCP VIDEO CAMERA SYSTEM
&
BLUE LIGHT POLICE EMERGENCY REPORTING TELEPHONES (PERT)

Students: Maysaa Almonayer and Dan Fitzgerald
Instructor: Dr. Mark Austin
Teaching Assistant: Nazanin Poorfarhani

Table of Contents

1. Introduction

The issue of security is one of the most important issues in our daily life, whether in our homes, at the office, in the airport, or on college campuses. As a result, the usage of security systems has increased. As consumers, we must have the knowledge of what requirements are necessary for a particular application because there are a variety of security systems in the market.

The University of Maryland Department of Public Safety (DPS) decided to use a video camera system to enhance proactive police patrols on campus. Security video cameras have been installed in certain high-traffic exterior and interior public spaces on campus, in locations that afford them the greatest field of view, as well in high security areas such as elevator and stairwell entrance, ATM machines and on the Blue Light Phones. The cameras are recorded 24 hours a day, 7 days a week. The Security Operation Center (SOC), which is part of the DPS, personnel monitors them during specific hours throughout the day and night.

In addition to the cameras there are the Blue Light Police Emergency Reporting Telephones (PERT), also known as the blue light phones, all over the campus. These phones provide a direct line to the Security Operations Center throughout the campus both inside and outside many academic and administrative buildings and residence halls. These phones are either yellow or encased within a blue cylindrical column and marked "Emergency". Frequently they have blue lights overhead making their locations easier to find. Individuals may contact the DPS directly and without charge by activating these phones. This phone system will inform the dispatcher of the caller's exact location.

Why are the video cameras used?

- To enhance security on campus.
- To utilize the technology of today to enhance proactive police on campus.
- It is a cost effective method of increasing patrol coverage at a fraction of the cost of adding additional officers. One camera can cover the same area of approximately three officers.
- Police can gain reliable and valuable information about what happened during a particular incident because the system automatically records present patterns in each area of coverage.
- Police will be able to respond more quickly to areas with detailed information about incidents which have just occurred in the camera's view.

What can video cameras do ?

- Provide electronic escort for faculty/staff and students.
- Provide monitoring for designated high security areas (i.e. ATM, sporting events).
- Provide monitoring for closed areas of the campus during night hours.
- Provide remote patrol for designated high crime areas (i.e. remote parking lots, isolated buildings).
- Notify police when problematic traffic situations occur or when disabled vehicles may need assistance.
- Pan 360 degree, Tilt, and utilize optical/digital zoom.

Both the cameras and the PERT poles are under the full control of the SOC, through monitors and computers with high speed network connections to process and report information rapidly and efficiently. For communications, the SOC has multiple telephones available to contact police dispatch and police radio units for constant and immediate communication with officers on duty. Also available to the operations center is equipment and software for reviewing incidents that have been recorded on camera.

2. Scope and Objective

During the last semester for ENPM641, we chose to study the Security System on campus (blue poles). We approached the study in a general way and we considered our system from the functional point of view. Our system was analyzed to be composed of camera, blue pole, telephone, and the communications & Security Operations Center. We developed our high level requirements and we used these requirements to generate the UML diagram, use case, activity and statechart for each use case, and the class diagram. We developed also general equations for the trade off analysis which was based on economics and performance.

This semester we still studied the security system on campus, but we approached this system from a different point of view. We analyzed the spatial positions of the blue poles in order to know where we should place our poles with respect to the existing structure of the campus with its buildings and other objects. The goal is to minimize the total number of poles that should be placed on campus while ensuring security for all. These poles should be easily accessible and in the line of sight of any potential user.

This semester we illustrated this problem as a system engineering case study. First, we revised our UML diagrams to reflect the change in scope of our project. Next, we mapped behavior model to our structural model to verify consistency in our UML diagrams. Then, we explored different methods for solving this problem, providing analysis of the advantages and disadvantages of each. Next semester, we will implement and verify a solution method discussed in this report. Once the method is verified, we will use optimization to determine the minimum number of blue poles needed for a certain area.

3. Goals, Scenarios, and Use cases

<u>Goals and Scenarios</u>

Goal 1. System must be easy to locate in case of emergency.

- **Scenario 1.1** Blue poles must be on the line of sight of the user.
- **Scenario 1.2** Blue poles must be within a reasonable distance from the user, which is < 50 ft.

Goal 2. System must be cost effective.

- **Scenario 2.1** Minimize the number of cameras used to satisfy safety requirements on campus.

System Boundary

The system boundary is defined by the location of the poles themselves and the surrounding spatial area includes the buildings.

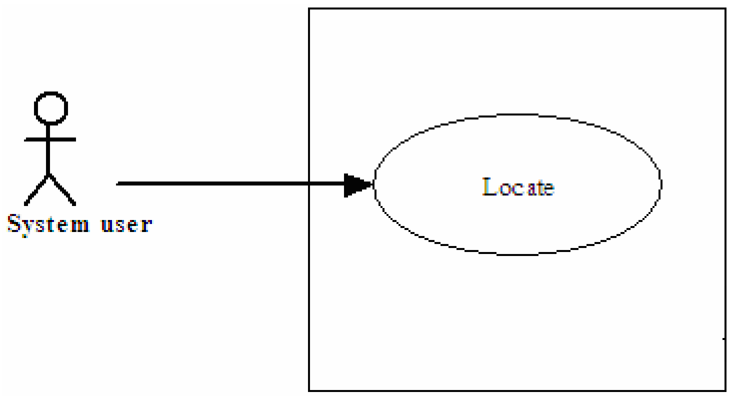


Figure 1. Initial Use Case Diagram

Use Cases Text and Activity Diagram

Use Case: Locate

- **Description**: The system user must be able to locate the blue pole.
  The user maybe any person on campus, and the user maybe one person or multiple people.
  Primary Actor(s): System user and his spatial location.
  Preconditions: We consider portion of the campus area with 1 building and few blue poles.
  Flow of Events: When the system user will experiences an incident or any suspicious behavior he will try to locate the nearest blue pole or emergency system.
  Alternative Flow of Events: None
  Post Condition:
  Requirement:
    - **1.1** Blue pole is within line of sight of the user.
    - **1.2** Blue pole is within reasonable distance of the user, < 50 ft.

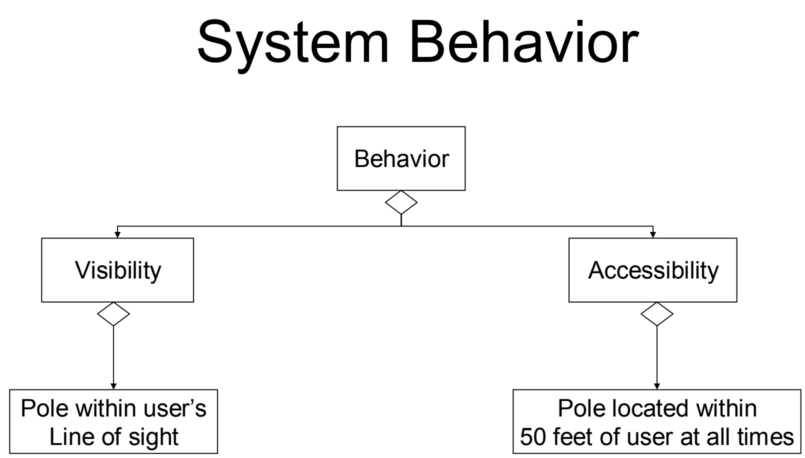
5. Generation of Requirements

The requirements for our system are based on the visibility and accessibility of the blue poles. Users must be able to locate and reach a pole in order to use it.

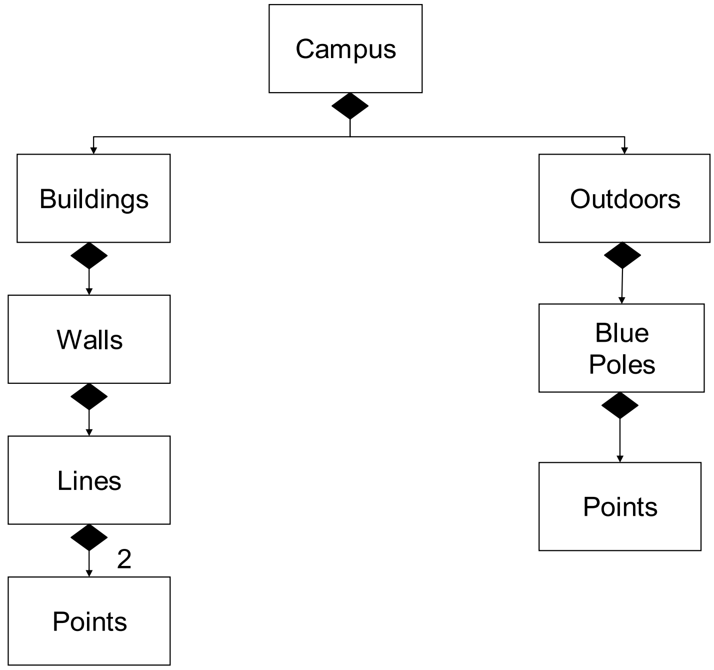
Requirement	Structure	Behavior
Req. 1.0: A user must always be able to see a blue pole.	Camera, cameraLocation Building, buildingLocation	lineOfSight()
Req. 2.0: A user must always be within a reasonable distance of a blue pole.	Camera, cameraLocation	Distance ≤ 50 feet

6. Models of System Behavior and Structure

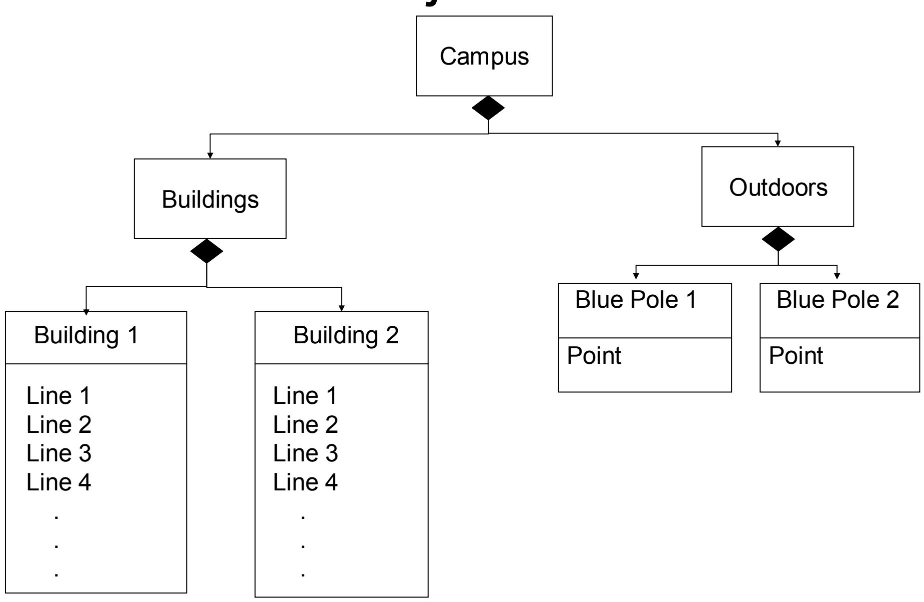
Since the scope of our project changed from last semester, it was necessary to create new models for system behavior and structure. An object diagram was also created to describe how our objects can be geometrically represented. Then, we mapped our behavior diagram to our structural diagram in order to verify our models.



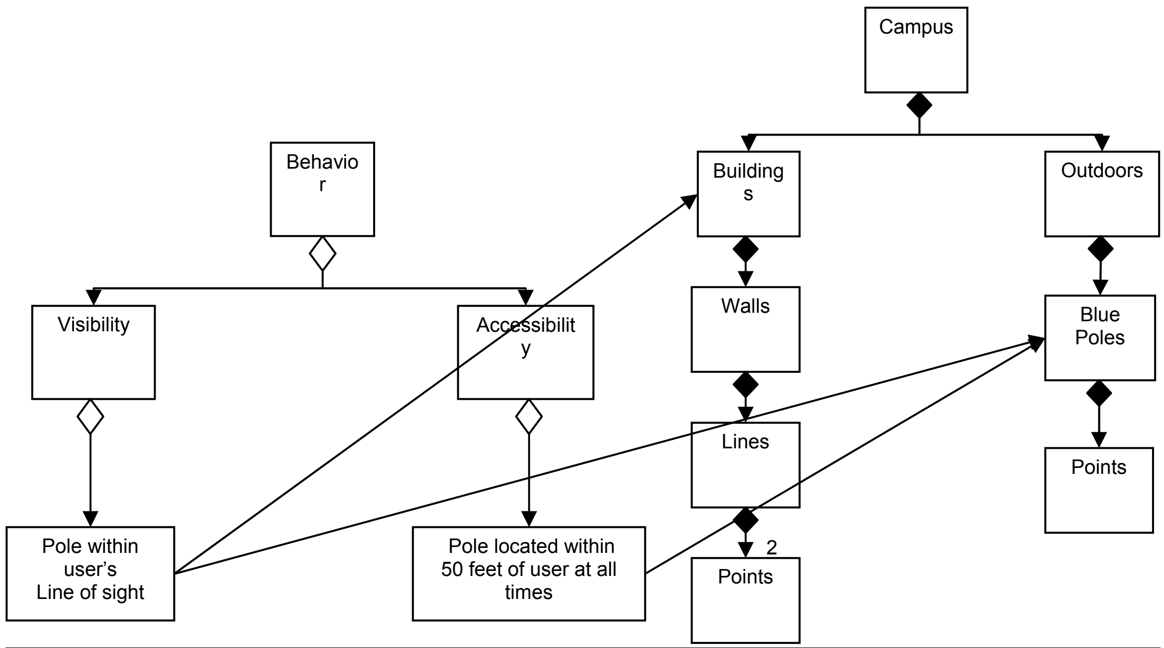
System Structure



Objects



Mapping Behavior to Structure



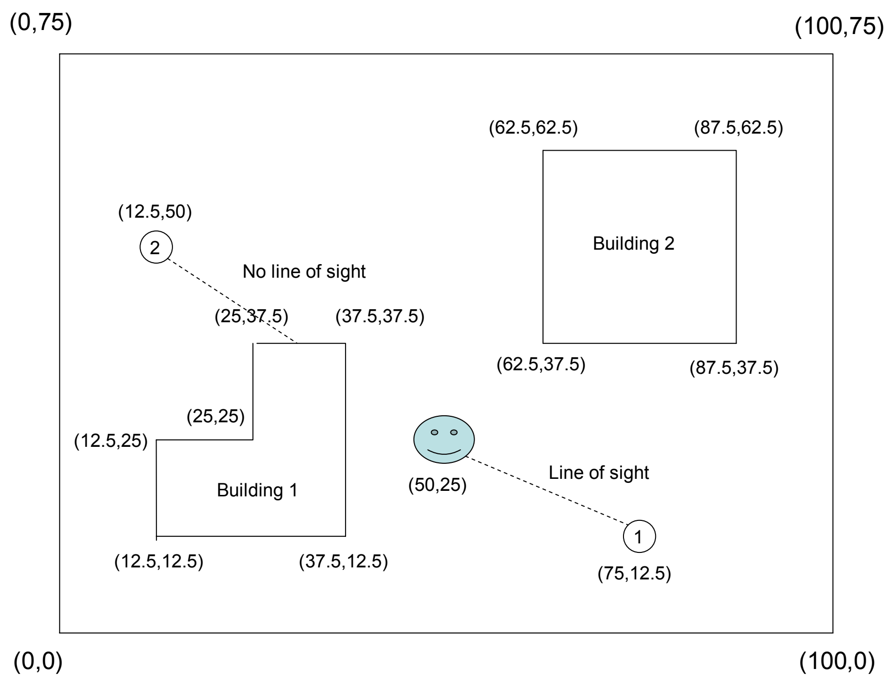
7. System Validation

To validate this system, we considered different methods in order to find the most efficient way to validate our requirements. The methods we explored are explained below and our proposed method of validation is outlined in the flowchart at the end of this section. This final diagram is a combination of many ideas and trials. This proposed solution will be implemented, tested, and optimized next semester.

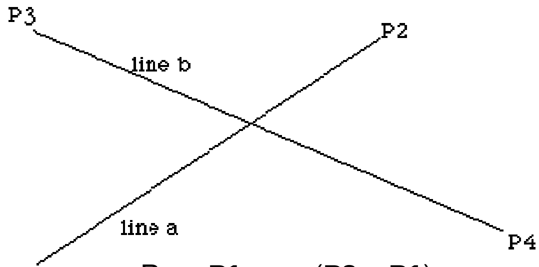
First Method
The first attempt to validate this model resulted in a MATLab program that represented the buildings and blue poles in a matrix. The program would explore and test every point on campus against the requirements. If all of the points met the requirements, then the proposed placement of the poles would be valid. The disadvantage of this method is that it requires a lot of computational power to complete. The figures below help explain the method.

1^{st} trial of Validation

- Use MATLab to create a matrix that represents the buildings, cameras, and open space within the sample area.
- Test the line of sight and distance requirements at every point that isn't within a building or a camera.
- If all points satisfy both requirements, camera setup is valid.
- After initial model is verified, a random number generator can be used to place cameras and multiple iterations can be run to determine a minimum number of cameras needed and their placement.



Line of Sight Algorithm



Pa = **P1** + ua (**P2** – **P1**)
Pb **= P3 +** ub **(P4 - P3)**

Solve for Pa=Pb:
x1 + ua (x2 - x1) = x3 + ub (x4 - x3)
y1 + ua (y2 - y1) = y3 + ub (y4 - y3)

$$u_a = \frac{(x4 - x3)(y1 - y3) - (y4 - y3)(x1 - x3)}{(y4 - y3)(x2 - x1) - (x4 - x3)(y2 - y1)}$$
$$u_b = \frac{(x2 - x1)(y1 - y3) - (y2 - y1)(x1 - x3)}{(y4 - y3)(x2 - x1) - (x4 - x3)(y2 - y1)}$$

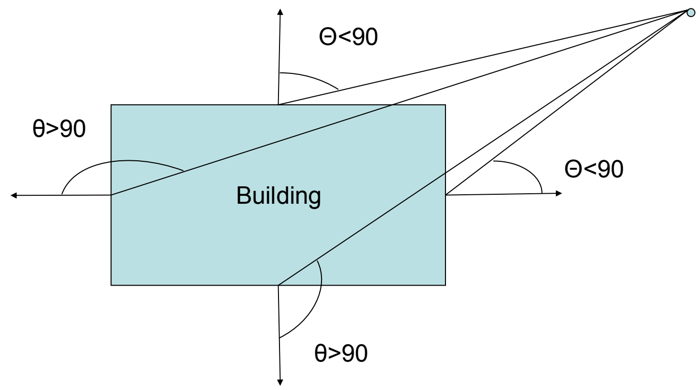
The equations apply to lines, if the intersection of line segments is required then it is only necessary to test if ua and ub lie between 0 and 1. Whichever one lies within that range then the corresponding line segment contains the intersection point. If both lie within the range of 0 to 1 then the intersection point is within both line segments.

Distance Algorithm

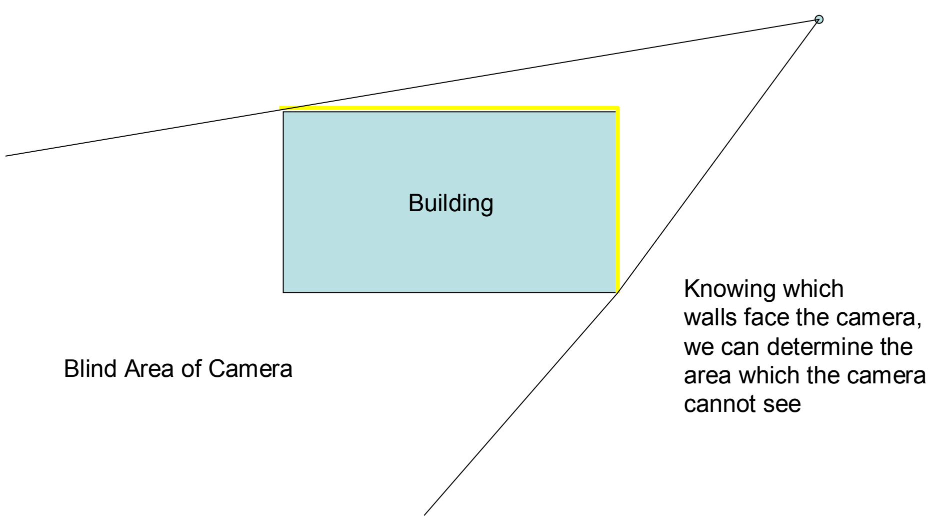
- Xd=x2-x1
- Yd=y2-y1
- Distance=sqrt(xd^2+yd^2)
- Req2=(Distance≤50)

Second Method
After realizing that the first method is not efficient, we decided to explore the problem in a geometric fashion. We wanted to see if we could determine which walls faced the camera in order to determine the blind spot of the camera. We learned that if we drew perpendicular lines outward from the center of each wall and connected them with a line from the camera, the angle formed provided us with information on whether or not the wall faces the camera. If the angle formed is less than ninety degrees, the wall faces the camera. The figures below further explain this method.

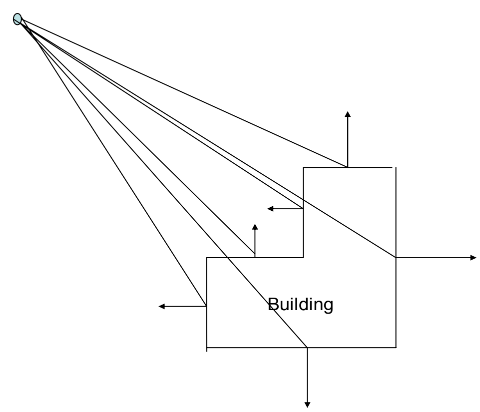
2^{nd} trial of validation



If the angle between the camera line and perpendicular is
Smaller than 90 degrees, the wall faces the camera



More Complicated Building

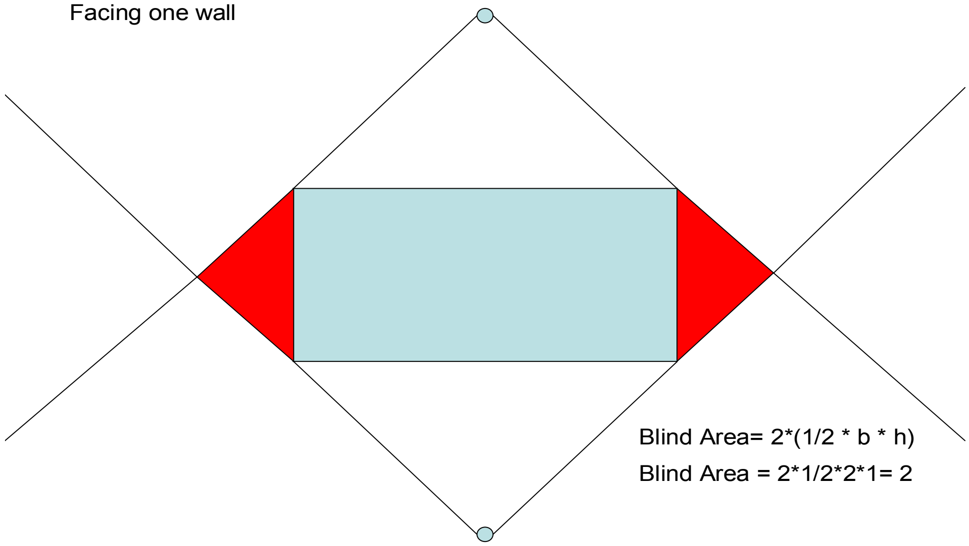


If more than two sides face the camera, then the two largest angles that are less than 90 degrees contain the blind area constraint lines.

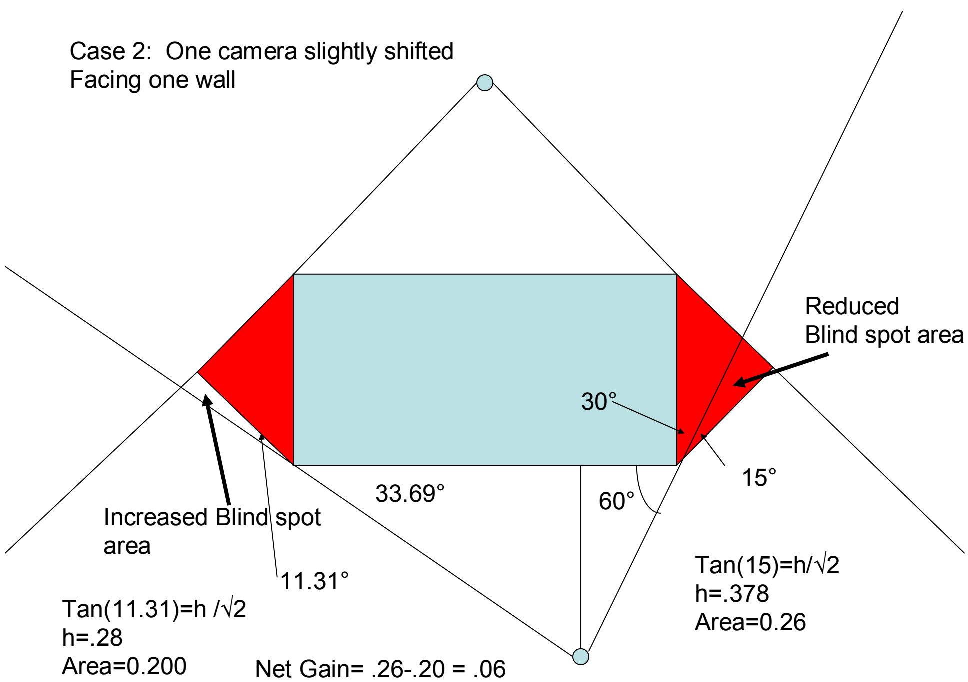
Third Method

After exploring which walls faced the camera, we were interested in determining how the blind spot of multiple cameras change when repositioned. We explored many shaped buildings to determine a methodology for placing the cameras based on building shape. We found that if we extended the wall lines of the building out, it forms sub areas on the outside of the building. The good sub areas have the maximum number of walls facing it. From the second method, we are able to determine how many walls face a point so we are able to calculate how many walls face an area by just taking a point from the area. The figures below help explain the work completed for this method.

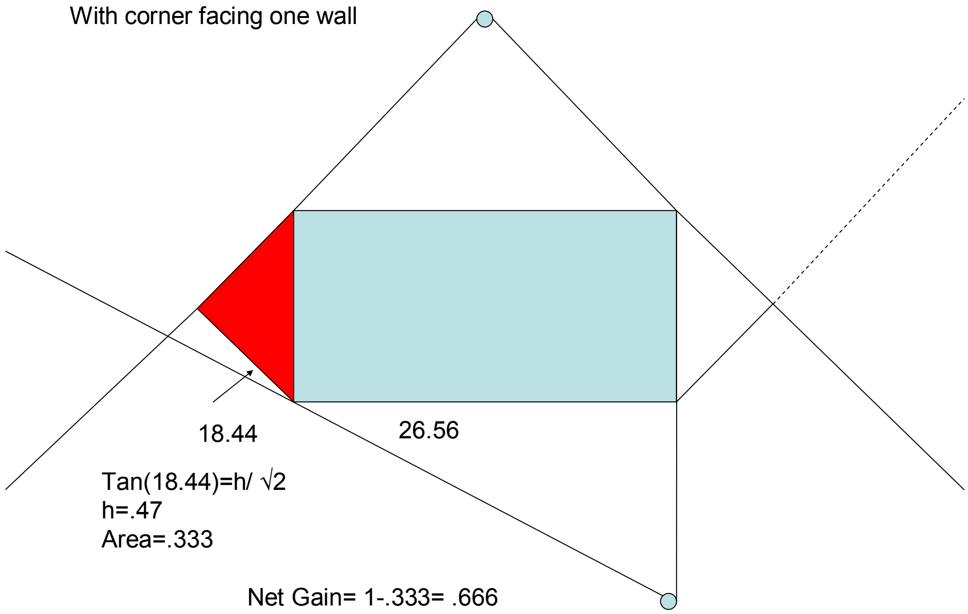
Case 1: Cameras are at the midpoint
Facing one wall



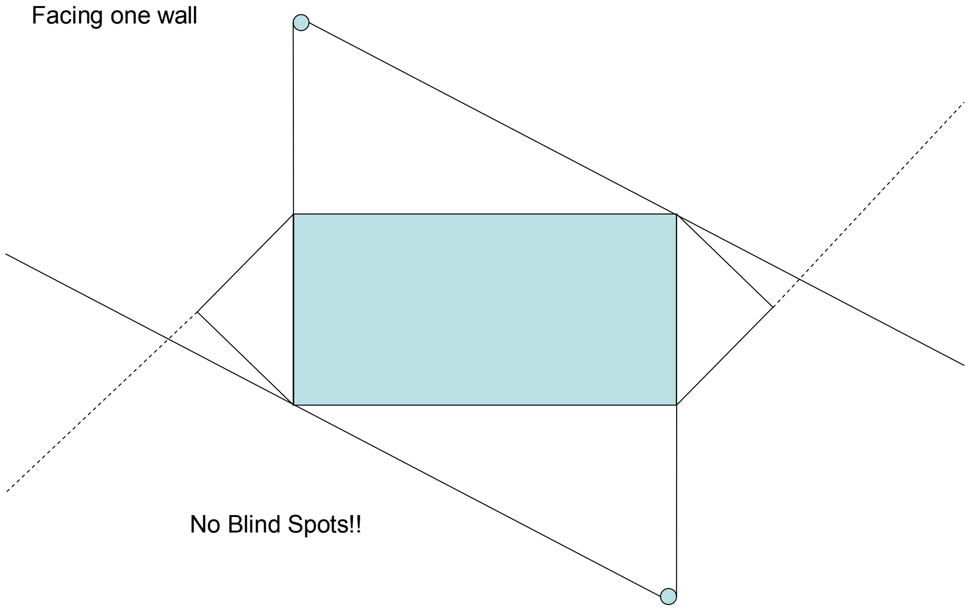
Case 2: One camera slightly shifted
Facing one wall

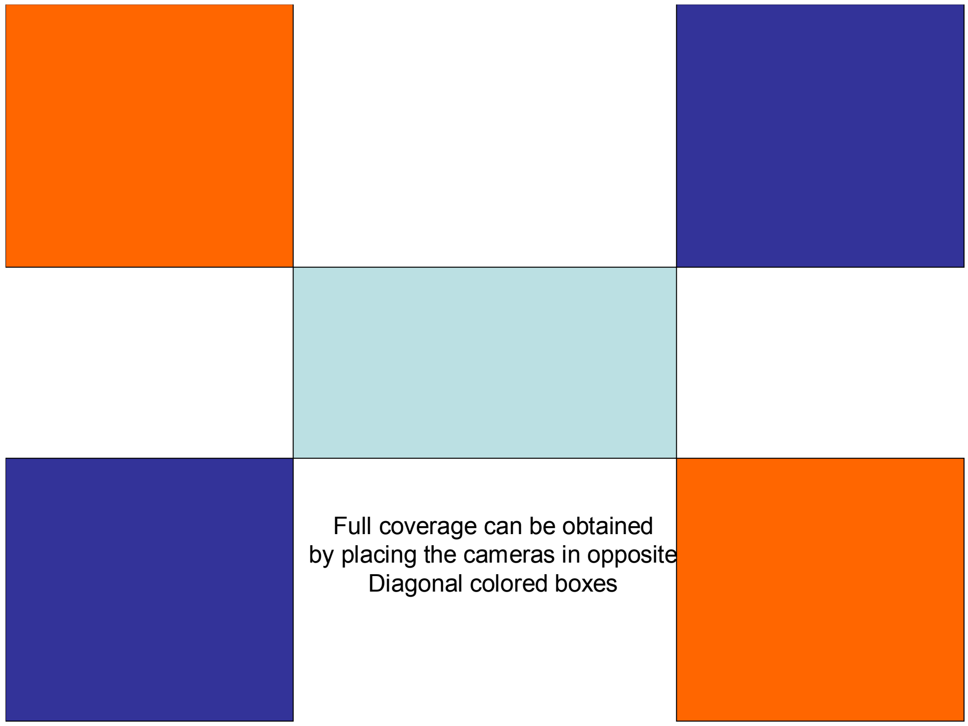


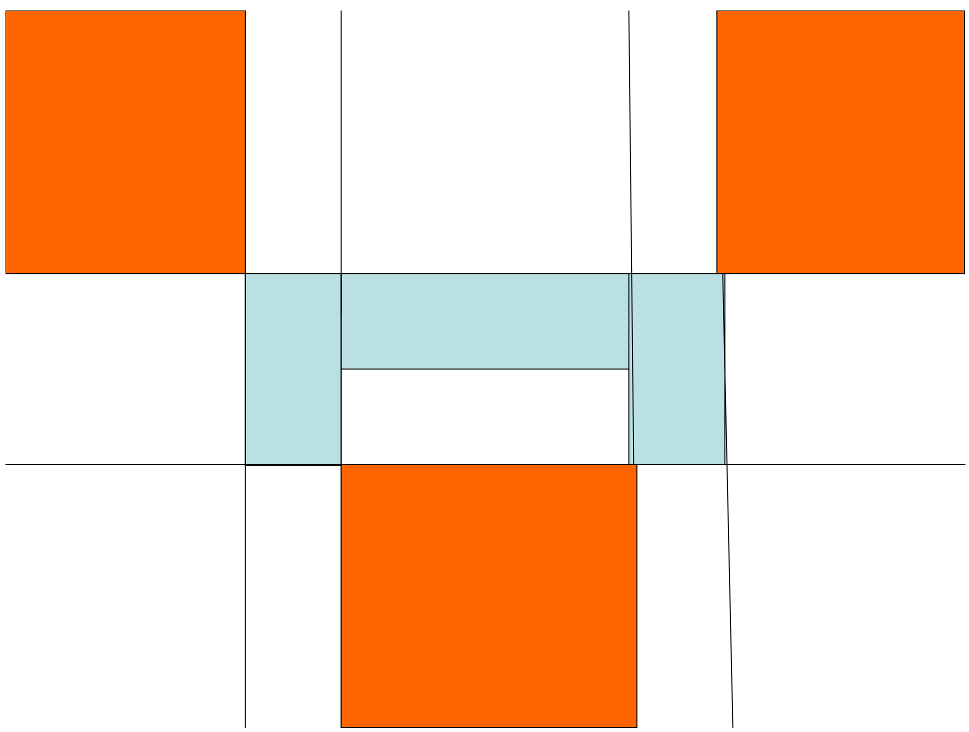
Case 3: One camera in line
With corner facing one wall

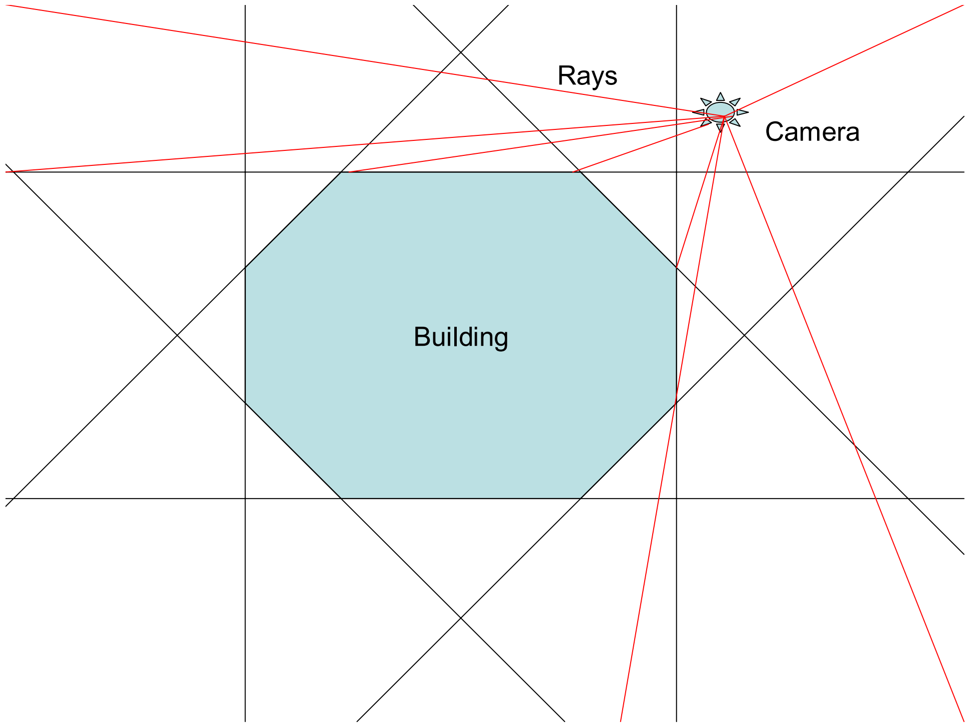


Case 4: Cameras are at the opposite corners
Facing one wall



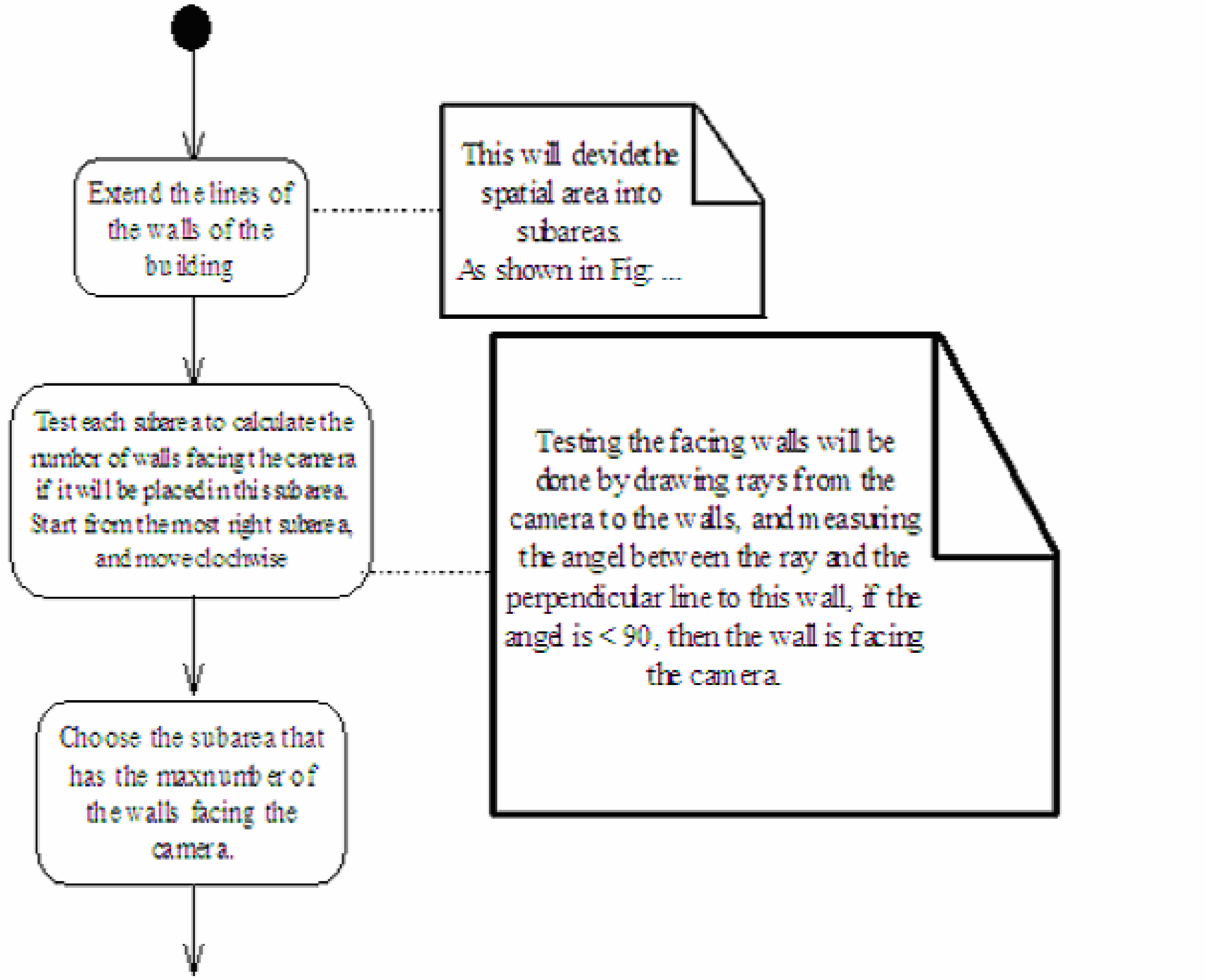

Full coverage can be obtained
by placing the cameras in opposite
Diagonal colored boxes



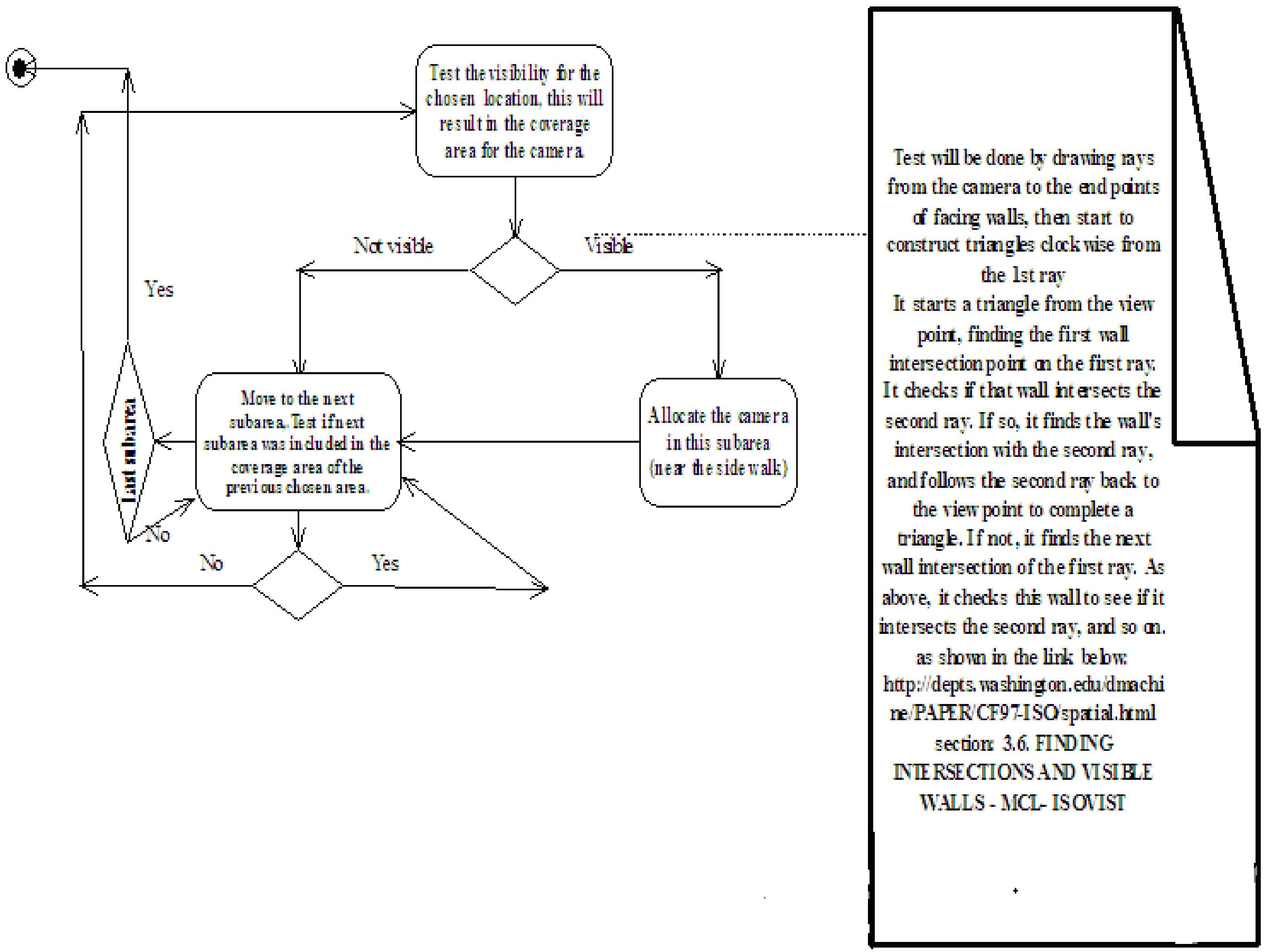


Final Result and Future Work

Our final proposed solution combines ideas from the methods described above and various research papers. The flowchart below outlines the steps that will be taken next semester to validate the system. The tasks for next semester include implementing the solution, validating the solution, and optimizing the solution.



Bottom arrow connects to box that starts with: “Test the visibility”.


Test the visibility for the chosen location, this will result in the coverage area for the camera
Not visible
Visible
Yes
Last subarea
Move to the next subarea. Test if next subarea was included in the coverage area of the previous chosen area.
Allocate the camera in this subarea (near the side walk)
No
No
Yes
Test will be done by drawing rays from the camera to the end points of facing walls, then start to construct triangles clock wise from the 1st ray
It starts a triangle from the view point, finding the first wall intersection point on the first ray. It checks if that wall intersects the second ray. If so, it finds the wall's intersection with the second ray, and follows the second ray back to the view point to complete a triangle. If not, it finds the next wall intersection of the first ray. As above, it checks this wall to see if it intersects the second ray, and so on. as shown in the link below:
http://depts.washington.edu/dmachine/PAPER/CF97-ISO/spatial.html
section: 3.6. FINDING INTERSECTIONS AND VISIBLE WALLS - MCL- ISOVIST

8. References

Austin, Mark. ENSE 621: System Modeling and Analysis Lecture Notes. Fall 2004.

Security Operations Center. Accessed Fall 2004. http://www.umpd.umd.edu/organization/TSB/socwebpage/vidfaq.htm

Yi-Luen Do, Ellen and Mark D. Gross. Tools for visual and spatial analysis of CAD models: implementing computer tools as a means to thinking about architecture. Proceedings of CAAD Futures 1997 Conference. Munich, Germany. August 4-6, 1997. http://dmg.caup.washington.edu/xmlSiteEngine/browsers/stylin/publication50.html