# Exploiting Automatically Inferred Constraint-Models for Building Identification in Satellite Imagery

Martin Michalowski, Craig A. Knoblock
University of Southern California
Information Sciences Institute

*Research funded by the AFSOR,*
*grant numbers FA9550-04-1-0105 and FA9550-07-1-0416.*


Kenneth M. Bayer, Berthe Y. Choueiry
University of Nebraska-Lincoln
Constraint Systems Lab

*Research funded by NSF CAREER Award No. IIS-0324955.*

# Problem Statement

- **Goal**: Annotating satellite imagery with addresses
- Addresses can be assigned by exploiting sets of addressing “rules”
- Many traditional and non-traditional data sources available online
- How can we combine our knowledge of addressing with the available data?

# Building Identification Process

*Traditional Sources*

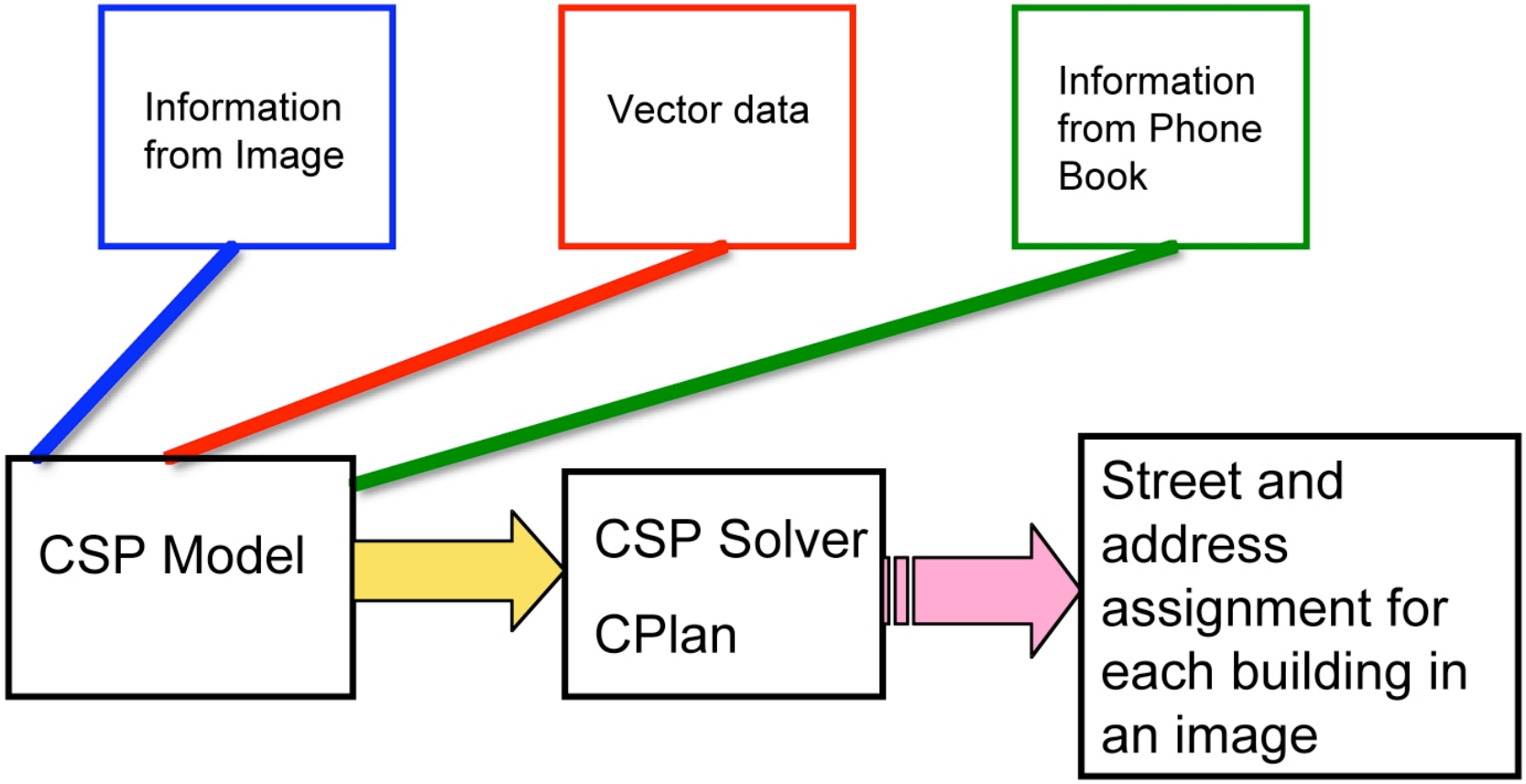
*Non-traditional Sources*

# Challenges

- Integrating heterogeneous data
- Modeling data and addressing characteristics
- Supporting various addressing schemes
  - One model tailored & stored per area ← BAD
  - Non-homogenous addressing within one area
- Efficiently solving the constructed problem

# Initial Approach

[Michalowski & Knoblock, 2005]



# Building Identification as a CSP [Michalowski+, 2005]

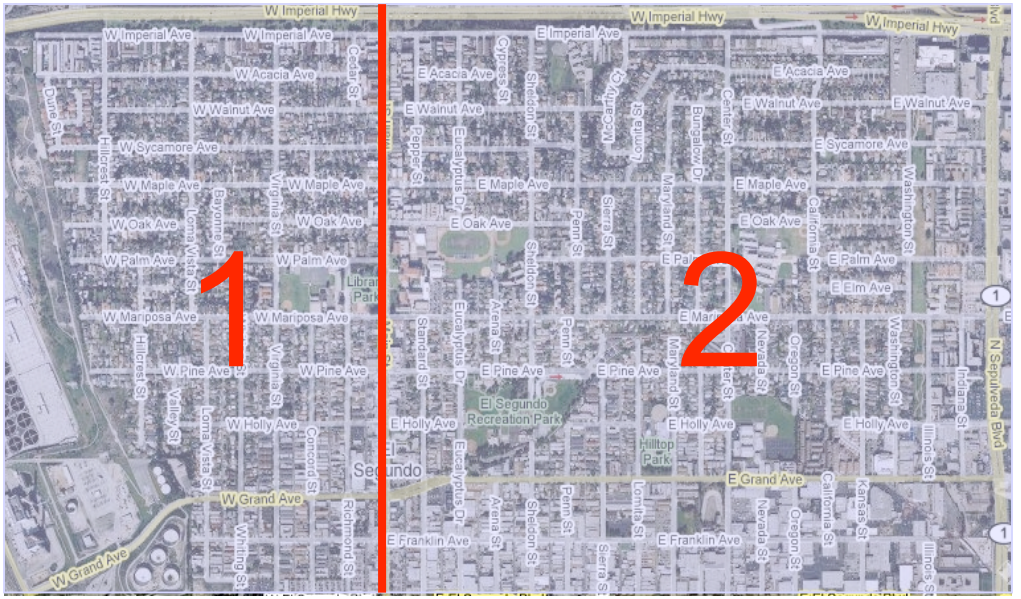
- **Constraint Satisfaction Problem**
  - **Variables**: Buildings
  - **Variable Domains**: Potential street addresses
  - **Constraints**: Global addressing characteristics (parity, ascending direction, etc.)
- Demonstrated the feasibility of modeling data integration for building identification as a CSP
- **Limitations**
  - Relied on a ‘single-model’ approach
  - Limited to small homogeneous areas
  - Did not scale

# Why a Single Model Doesn't Work

Block Numbering

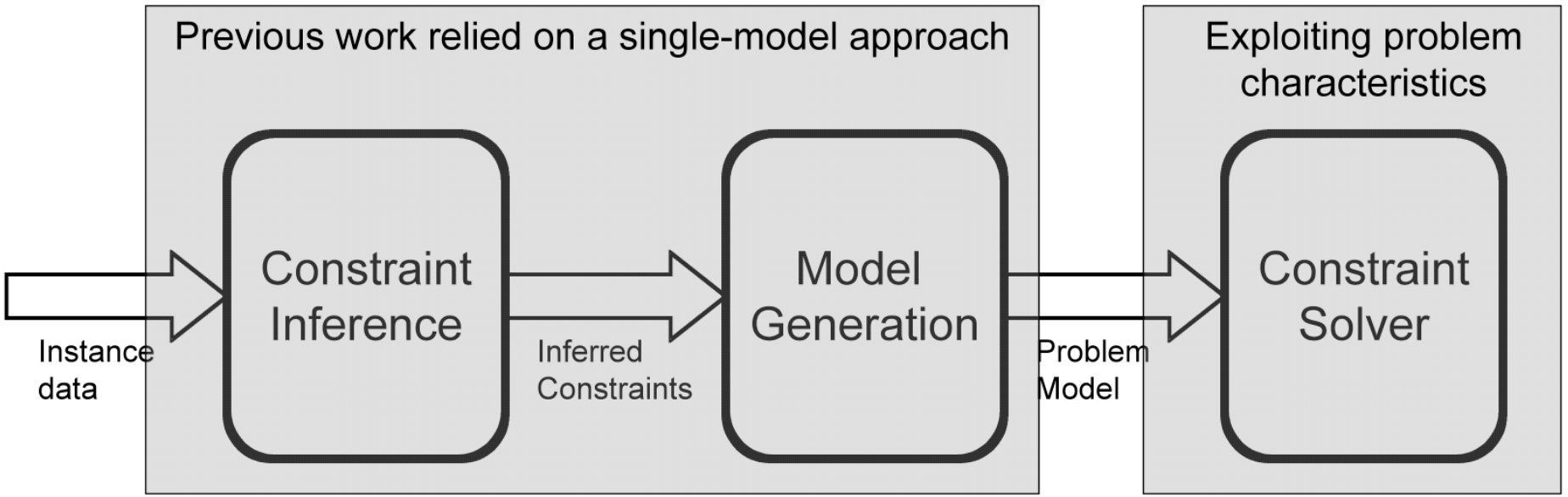


Constraints apply in different *contexts*

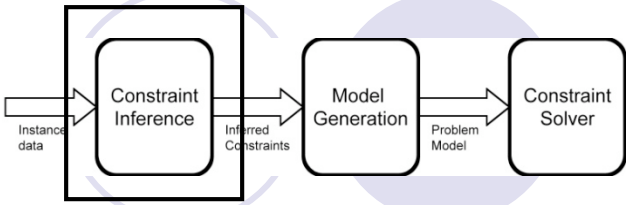



**Addresses increase West** **Addresses increase East**

# Our Solution



# Constraint Inference




**Problem Instance**

Input information

$F = \{F_1, F_2, \ldots, F_n\}$

Generic model

$C_B = \{ C_1, C_2, \ldots, C_i\}$

**Constraint Library**

User-defined (& learned) constraints

$C_L = \{C_{l1}, C_{l2}, \ldots, C_{lz}\}$

**Inference Engine**

Inference rules

$R_t = \{R_1, R_2, \ldots, R_z\}$

$R_k$: $F_i \in F \rightarrow C_l \in C_L$

**Refined model**: $C_{new} = C_B \cup C_I$

# Example

- Data points
  - Landmark points that describes a particular instance
    - Obtained from any online point repository (e.g. gazetteers)
  - Features: Address Number, Street Name, Lat, Lon…

***852 Hillcrest St***

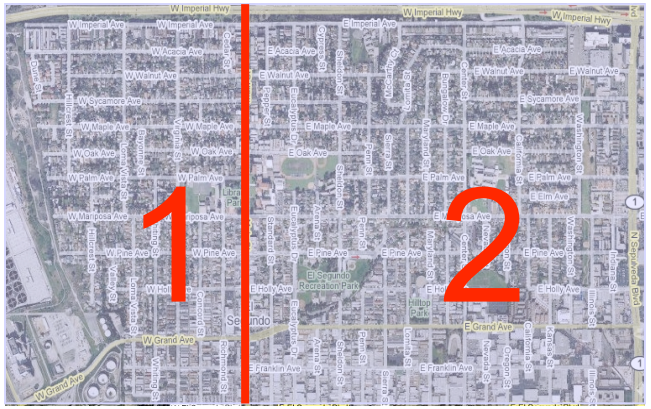
***859 Loma Vista St.***

***834 Hillcrest St***

- Constraints

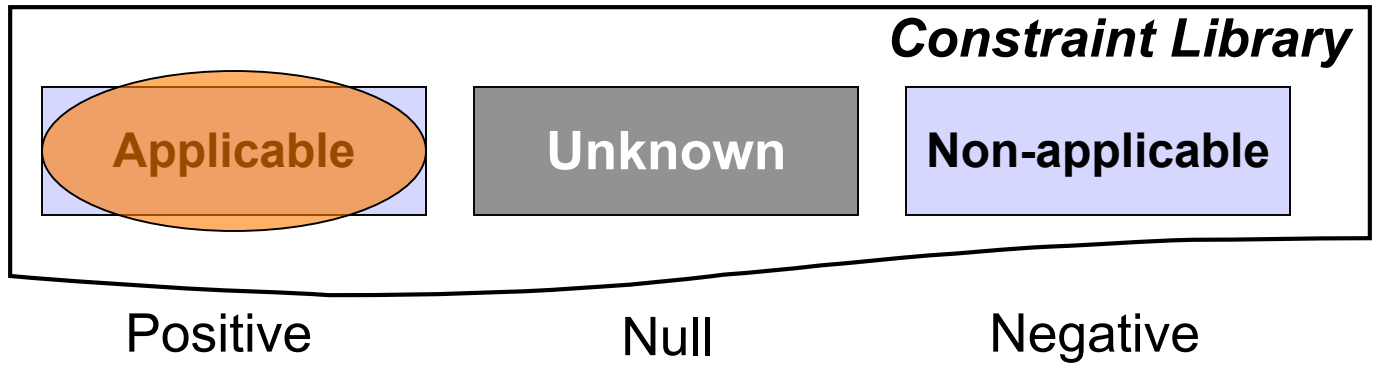
| Name | Description |
|---|---|
| Parity (odd/even) | Addresses on the same side of a street have the same parity |
| Continuous | Addresses increment continuously by a fixed number $n$ |
| Block Numbering (Grid) | Addresses increment by a factor of $k$ across grid lines |
| Ordering | Addresses increase monotonically along a given street |
| ... | |

- Context (El Segundo)

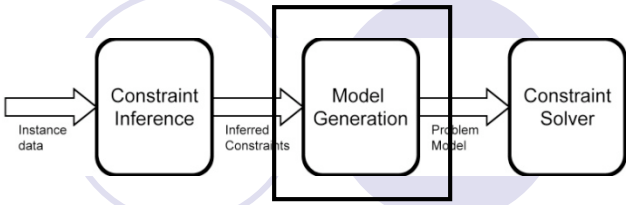


# Inferring Constraints

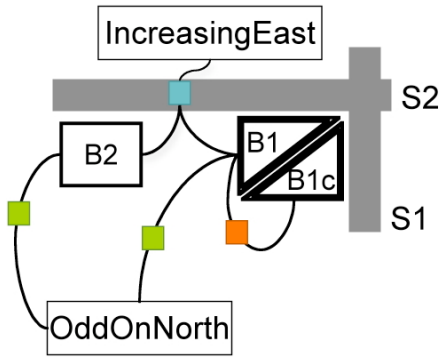
- Inference rules are evaluated using data points
  - Supports (+,-) provided for the constraints
- Constraints are partitioned based on *support level*
  - Status: Applicable, Unknown, Non-applicable
- Applicable constraints added to generic model

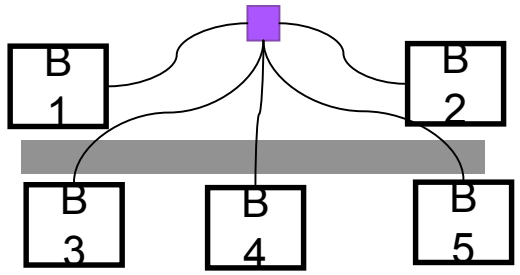


# Model Generation

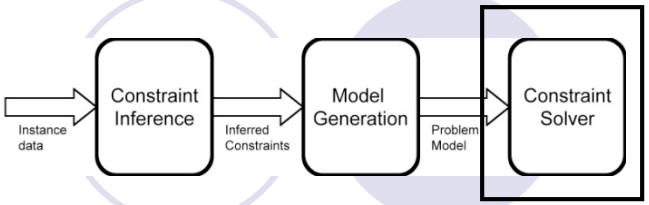



- Generates constraint model from variables and *inferred* constraints
- Model improvements over previous work
  - Reduces total number of variables and constraints' arity
  - Reflects topology: Constraints can be declared locally & in restricted 'contexts'



# Constraint Solver




- Backtrack-search with nFC3 and conflict-directed back-jumping
- Exploits structure of problem (backdoor variables)
- Implements domains as (possibly infinite) intervals
- Incorporates new reformulations that increase the scalability by large factors
  - Details available in [Bayer+, 2007]

# Case Studies

| Case study | Phone-book completeness | Number of... bldgs | blocks | building-address combinations |
|---|---|---|---|---|
| NSeg125-c | 100.0% | 125 | 4 | 4160 |
| NSeg125-i | 45.6% | | | 1857 |
| NSeg206-c | 100.0% | 206 | 7 | 10009 |
| NSeg206-i | 50.5% | | | 4879 |
| SSeg131-c | 100.0% | 131 | 8 | 3833 |
| SSeg131-i | 60.3% | | | 2375 |
| SSeg178-c | 100.0% | 178 | 12 | 4852 |
| SSeg178-i | 65.6% | | | 2477 |

- All cases are beyond what our initial work could solve

# Experimental Results

**CSP Search Solver**

| | W/o orientation cons | | W/ orientation cons | | | |
|---|---|---|---|---|---|---|
| | Runtime (sec) | Domain size | Runtime (sec) | Domain size | Runtime reduction | Domain reduction |
| NSeg125-c | 22397.08 | 1.22 | 1962.53 | 1.0 | 11.41x | 1.22x |
| NSeg125-i | 22929.49 | 6.11 | 3987.73 | 4.18 | 5.75x | 1.46x |
| NSeg206-c | 198169.43 | 1.21 | 10786.33 | 1.0 | 18.37x | 1.21x |
| NSeg206-i | 232035.89 | 7.91 | 12900.36 | 4.99 | 17.99x | 1.59x |
| SSeg131-c | 173565.78 | 1.56 | 125011.65 | 1.41 | 1.39x | 1.11x |
| SSeg131-i | 75332.35 | 12.56 | 17169.84 | 3.92 | 4.39x | 3.20x |
| SSeg178-c | 523100.80 | 1.41 | 284342.89 | 1.31 | 1.84x | 1.08x |
| SSeg178-i | 334240.61 | 8.24 | 62646.91 | 3.23 | 5.34x | 2.55x |
| | | | | Average | 8.31x | 1.68x |

- 26 points used to infer correct model (inference time < 2 secs)
- Inferred model greatly reduces runtime
- Domain reduction leads to higher precision by a significant factor
- Additional results show an even greater improvement (see paper)

# Observations

- Constraint inference provides framework for data integration
- Inferred models lead to more precise results
- Ability to solve more complex instances
- Dynamic modeling makes global coverage possible and easier

# Related Work

- Geospatial
  - Geocoding

    [Bakshi+, 2004]
  - Computer Vision

    [Agouris+, 1996; Doucette+, 1999]
- Modeling
  - Learning constraint networks from data

    [Coletta+, 2003; Bessière+, 2005]

# Current Work

- Eliminating incorrect constraint inference
  - *Support levels* associate confidence with inferences
- Dealing with a lack of expressiveness in data points
  - *Iterative algorithm* with constraint *propagation*
- Generalizing context-inference mechanism
  - Classification in the *variable* space using *SVMs*
- Learning constraints to populate library
  - *Agglomerative clustering* combined with *set covering*

# Thank you!!!