Line Drawing Interpretation in a Multi-View Context

Jean-Dominique FAVREAU Florent LAFARGE Adrien BOUSSEAU
INRIA Sophia-Antipolis, France
firstname.lastname@inria.fr

Abstract

Many design tasks involve the creation of new objects in the context of an existing scene. Existing work in computer vision only provides partial support for such tasks. On the one hand, multi-view stereo algorithms allow the reconstruction of real-world scenes, while on the other hand algorithms for line-drawing interpretation do not take context into account. Our work combines the strength of these two domains to interpret line drawings of imaginary objects drawn over photographs of an existing scene. The main challenge we face is to identify the existing 3D structure that correlates with the line drawing while also allowing the creation of new structure that is not present in the real world. We propose a labeling algorithm to tackle this problem, where some of the labels capture dominant orientations of the real scene while a free label allows the discovery of new orientations in the imaginary scene. We illustrate our algorithm by interpreting line drawings for urban planing, home remodeling, furniture design and cultural heritage.

1. Introduction

Multi-view stereo greatly facilitates modeling of real-word scenes, but is of little help for modeling *imaginary* scenes. Our work is motivated by the observation that designers and architects often create imaginary objects as a complement to an existing scene. Common *modeling in context* tasks include remodeling or extending a building, designing custom furniture to fit in a leaving room, or creating archaeological reconstitutions from ruins. We describe an algorithm that assists such modeling tasks by estimating a 3D model from a single line drawing traced over one picture of a multi-view dataset.

Recovering a 3D object from a single line drawing is a long-standing problem in computer vision because of the infinity of shapes that can project on the same input [2]. Existing methods resolve such ambiguity by trying to enforce a variety of regularities (symmetry, parallelism, orthogonality, minimal standard deviation of angles) [16, 18, 6, 15, 17, 28]. However, identifying regularity cues from the drawing alone is difficult because typical line configurations often have multiple concurrent interpretations.

Instead of using arbitrary regularity cues, we propose to consider the existing scene as a flexible context-driven regularizer for the new content. In particular, we leverage the fact that man-made environments are often composed of a small set of dominant planes – although not necessarily orthogonal – and that parts of the extensions drawn by designers follow a similar structure.

Our algorithm takes as input an unordered set of photographs of a scene, along with a drawing composed of intersecting line segments. We first extract the dominant orientations of the scene using standard multi-view stereo techniques. Next, the core technical contribution of our work resides in a labeling algorithm that assigns each polygon in the drawing to either one orientation of the existing scene, or to an unknown orientation. While we require that at least some of the polygons in the drawing have the same orientation as some planes in the scene, our algorithm uses this 3D information to bootstrap the reconstruction of polygons with arbitrary orientations.

We demonstrate the flexibility of our algorithm with several reconstruction scenarios in architecture, furniture design and archeology.

2. Related work

Our work is related to object reconstruction and modeling as well as line drawing interpretation. In this section we detail the existing work that most inspired our approach, we refer the interested reader to surveys of these fields for a more general overview [19, 20, 7].

Multi-view reconstruction of man-made environments. Scenes often contain dominant structures that can greatly constrain the multi-view stereo problem. In particular, buildings are often composed of planar surfaces [5, 24, 4, 10] that are mutually orthogonal [12, 26]. Our method also exploits the inherent structure of man-made environments. However, while prior work detects planar surfaces to regularize the multi-view reconstruction, we use the dominant

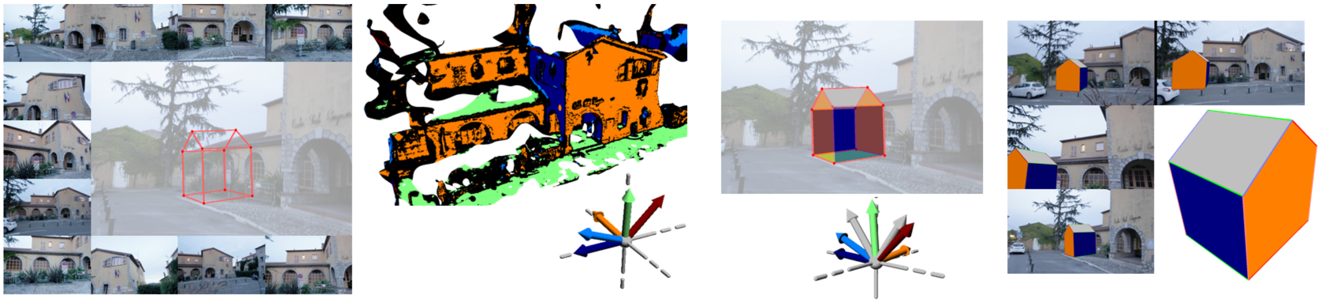

(a) Input (b) Dominant 3D orientations (c) Orientation labeling (d) Output

Figure 1. Overview of our approach. (a) Our algorithm takes as input multiple images of a scene along with a line-drawing traced over one of these images. (b) We first detect the dominant orientations of the existing scene from its multi-view stereo reconstruction. (c) Our labeling algorithm estimates the orientation of each facet of the drawing, favoring orientations already present in the scene. We visualize each dominant orientation with a random color, gray denotes new orientations. (d) We finally solve for the 3D model corresponding to the estimated orientations.

planes of the reconstruction to disambiguate the 3D interpretation of the drawing.

Image-based modeling. Several interactive systems have been proposed to allow users to model a man-made environment by tracing polygons over pictures of that scene. While early work relies on the user strokes to facilitate structure-from-motion [8], subsequent work first generates a sparse point cloud that then anchors the drawn polygons in 3D [27]. The reconstructed 3D structure can also provide snapping guidance during modeling [25, 1]. In contrast to this body of work, our goal is to model *new* content that is not present in the 3D reconstruction, and as such cannot be directly projected on the 3D point cloud. Nevertheless, our algorithm exploits the existing structure to constrain the interpretation of the user-provided drawing.

Line drawing interpretation. Inferring 3D shape from a single drawing is inherently ill-posed, as an infinity of shapes can project on the same drawing [2]. Existing algorithms make this problem tractable by assuming that the shape has a regular structure. The seminal work of Lipson and Shpitalni [16] and related methods [18, 6, 15, 17, 28] detect regularity relationships between lines in 2D that are likely to also apply in 3D. For instance, lines that are parallel in 2D should be parallel in 3D, and triplets of lines with adequate angle distributions are likely to correspond to cubic corners [21]. The reconstruction algorithm then solves for the 3D model that best satisfies these constraints. Lau et al. [14] use a similar algorithm to model in context by drawing over a picture. However, they ask users to annotate parallelism and orthogonality. The main intuition behind our approach is to relax these assumptions by relying instead on the geometric structure contained in the multi-view stereo reconstruction, which provides us with a strong data-driven regularization. As a result, our algorithm is capable of automatically recovering complex 3D shapes even in the absence of common regularity cues like symmetry and orthogonality.

3. Overview

Figure 1 illustrates the main steps of our algorithms. Starting from multiple photographs of a scene, we apply structure-from-motion and multi-view stereo to estimate a dense point cloud of the scene. We use Autodesk Recap360[1] for this purpose.

In addition to the multi-view dataset, our algorithm takes as input a single line drawing composed of segments connected at their extremities and traced over one of the photographs or over a rendering of the 3D reconstruction. This directly provides us with a registration of the drawing in the scene. We assume that the drawing represents a polyhedron surface, i.e. is composed of straight lines forming closed planar cycles (Figure 1(a)). We additionally assume that a drawing represents a single connected component.

We first estimate the dominant orientations of the existing scene (Figure 1(b)). A number of methods have been proposed in the literature to extract dominant orientations from noisy 3D reconstructions, such as [5, 23, 24, 26] to name a few. Given the relatively high density of our point clouds, we followed the normal clustering approach of Chen and Chen [5]. Our implementation accumulates all the point normals onto the Gauss sphere, which we discretize as an image using a stereographic projection. We then cluster the normals using Mean-Shift.

Our core contribution is to formulate the shape inference as a labeling problem that assigns one orientation to each surface component of the drawing (Section 4, Figure 1(c)). The assigned orientation can either be one of the dominant orientations of the scene, or be a new orientation only

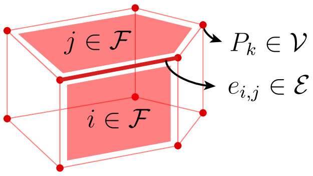
[1] http://recap360.autodesk.com

present in the imaginary object. In practice, while our algorithm needs some of the cycles of the drawing to align with existing orientations of the scene, a few such cycles is sufficient to bootstrap the inference of new orientations. Given a 2D drawing and the 3D orientations of its cycles, we solve for the 3D model that best satisfies the orientation constraints while minimizing reprojection error, as detailed in Section 4.3. Figure 1(d) shows the 3D polyhedron produced by our algorithm. We project this surface in the input photographs to visualize the imaginary object in the existing context.

4. Lifting the line drawing to 3D

Our ultimate goal is to position the lines of the drawing in 3D. While many prior methods solved such a problem by reasoning on the junctions of the lines, we follow Liu et al. [17] by reasoning on the facets that the lines delineate. Specifically, we solve for the 3D normal of each surface element of the drawing and recover the corresponding 3D position of the junctions in a subsequent step.

We denote by $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ the graph supporting the input drawing, where the edges $\mathcal{E}$ represent the lines of the drawing and the nodes $\mathcal{V}$ represent the junctions. We denote by $\mathcal{F}$ the set of simple cycles of the graph, namely facets.



4.1. Energy formulation

Our objective is to estimate the 3D normal of each facet of the drawing graph $\mathcal{G}$. The main idea behind our approach is to select part of these normals from the dominant orientations of the existing scene. We cast this selection as a labeling problem, where $l = (l_i)_{i \in \mathcal{F}} \in L$ denotes the configuration of labels that associates a normal to each facet and L represents the configuration space

$$L = \{d_1, .., d_m, d_{new}\}^{card(\mathcal{F})} \tag{1}$$

where $d_1, .., d_m$ are the m dominant orientations of the scene and d_{new} is a free 3D vector. This free label is critical to allow our algorithm to capture orientations that are not present in the existing scene.

We measure the quality of a configuration l by the energy

$$U(l) = U_{data}(l) + \beta U_{prior}(l) + \gamma U_{complexity}(l) \tag{2}$$

where $U_{data}(l)$ evaluates the coherence of a configuration l with respect to the input drawing, $U_{prior}(l)$ is a weak geometric prior to penalize flat interpretations, and $U_{complexity}(l)$ penalizes the use of free normals d_{new}. These three terms are balanced by the parameters $\beta > 0$ and $\gamma > 0$.

Data term. We assume for now that we can recover the 3D model $\mathcal{M}(l)$ corresponding to a given configuration l, we detail the estimation of this 3D model in Sec. 4.3. Our first term measures how well $\mathcal{M}(l)$ projects on the input drawing. Denoting $P_k(l)$ the 3D vertices of the reconstructed model, we measure the re-projection error

$$U_{data}(l) = \sum_{k \in \mathcal{V}} \|\Pi(P_k(l)) - p_k\|^2 \tag{3}$$

where p_k represents the 2D vertices of the input drawing and $\Pi(.)$ is the perspective projection to the drawing plane. In order to have a homogeneous energy, we scale the image domain to $[-1,\, 1]^2$.

Geometric prior. The data term above could be fully satisfied by simply assigning the same normal to each facet of the model. We prevent this naive solution by penalizing similar normals between adjacent facets:

$$U_{prior}(l) = \sum_{(i,j) \in \mathcal{E}^\star} |< l_i, l_j >| \tag{4}$$

where $\mathcal{E}^\star$ is the subset of $\mathcal{E}$ that excludes the edges adjacent to only one facet. Note that while this soft constraint implicitly favors orthogonality, it is more flexible than a strict Manhattan-world assumption.

Complexity. Finally, we encourage the interpretation of the drawing to be similar to the existing scene by giving a constant penalty to each facet labeled with a new orientation d_{new}:

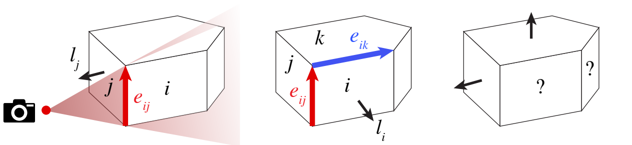
$$U_{complexity}(l) = \sum_{i \in \mathcal{F}} \mathbb{1}_{\{l_i = d_{new}\}} \tag{5}$$

where $\mathbb{1}_{\{.\}}$ is the characteristic function. Note that, because of this term, the energy is not Markovian.

4.2. Computing d_{new}

Evaluating our energy $U(l)$ requires knowledge of all the 3D orientations of a configuration l, including the unknown orientations of the facets labeled d_{new}. We estimate these orientations in a greedy manner, giving priority to the facets having the highest number of adjacent facets with a dominant or already resolved orientation.

For a given facet i, we first compute the 3D direction of each edge e_{ij} shared with a facet j of known orientation. This direction is given by the cross product between the normal of facet j and the normal of the plane supporting e_{ij} and the optical center of the camera, as illustrated in Figure 2(a). Assuming that the facet is planar, we compute its normal as the average cross product between the known directions of every pair of successive edges (e_{ij}, e_{ik}) (Figure 2(b)). While all pairs of edges could be used for this computation,



(a) Computing e_{ij} (b) Computing l_i (c) Minimal condition

Figure 2. Steps to compute the unknown orientation of a facet i. (a) We first compute the 3D direction of all the edges e_{ij} shared with facets of known orientations l_j. (b) We then compute l_i as the average cross product of pairs of known edges (e_{ij}, e_{ik}). (c) This greedy procedure requires that the facet with unknown orientation is adjacent to at least two facets with known non-collinear orientations.

we found that using solely the successive edges yields good results for a fraction of the computation.

This greedy procedure requires that at least two facets with known non-collinear orientations are adjacent to the facet with unknown orientation to be resolved at a given iteration (Figure 2(c)). When this is not the case, the configuration is rejected.

4.3. Computing $\mathcal{M}(l)$

We now describe how to recover the 3D model $\mathcal{M}(l)$ that minimizes the re-projection error U_{data} for a given configuration of facet orientations l.

A first challenge resides in the non-linearity of the perspective projection $\Pi(.)$, which we express as

$$\Pi(P_k) = \frac{RP_k}{SP_k} \tag{6}$$

where

$$\begin{cases} R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} M_p \\ S = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} M_p \end{cases} \tag{7}$$

and M_p is the projection matrix.

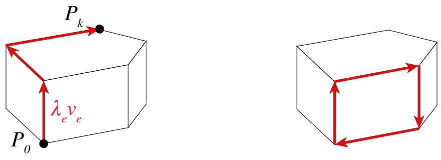
We linearize Equation (3) by multiplying $\Pi(P_k)$ and p_k by SP_k, which is equivalent to measuring the error in projective space rather than in image space:

$$\epsilon = \sum_{k \in \mathcal{V}} \| RP_k - SP_k p_k \|^2 \tag{8}$$

While this approximation over-penalizes points distant from the camera, it is satisfactory for small re-projection errors.

The second challenge is to deduce the 3D position of the vertices from the orientation of the facets. To do so, we express each vertex P_k by its shortest path in $\mathcal{G}$ to a reference point P_0 (Figure 3(a)). Since the facet orientations give us the 3D direction v_e of each edge e, we obtain

$$P_k = P_0 + \sum_{e \in \mathcal{E}} \delta_{ke} \lambda_e v_e \tag{9}$$



(a) Expressing P_k with respect to P_0 (b) Cycle closure

Figure 3. (a) We formulate 3D reconstruction as the optimization of the edge lengths λ_e by expressing each 3D vertex P_k with respect to a reference point P_0. (b) We additionally constrain each facet to form a closed cycle.

where λ_e is the unknown length of edge e, $\delta_{ke} = 0$ if e is not part of the shortest path between P_0 and P_k, and $\delta_{ke} \in \{-1, 1\}$ otherwise depending on the orientation of v_e.

In our implementation we assign the first drawn point to be P_0. We additionally assume that this point intersects the existing geometry, which positions it in depth. Substituting Equation (9) in Equation (8) yields an optimization problem on the edge lengths λ_e and the 2D coordinates p_0 of the reference point.

However, this formulation does not guaranty that all edges of the graph are traversed, and as such well constrained. We make the problem better constrained by additionally encouraging closure of the edge cycles that delineate each facet (Figure 3(b))

$$(\forall i \in \mathcal{F}) \sum_{e \in \mathcal{E}} c_{ei} \lambda_e v_e = 0 \tag{10}$$

where $c_{ei} = 0$ if the edge e is not part of facet i, $c_{ei} \in \{-1, 1\}$ otherwise depending on the orientation of v_e.

In supplementary material, we provide details on the resolution of this quadratic minimization problem under linear constraints.

4.4. Energy minimization

Our energy U is composed of a data term that does not respect the conditional independence hypothesis, and a complexity term that acts globally on the label configuration. As a result, U does not have the MRF form of standard labeling problems in vision and cannot be minimized by efficient graph-cut techniques [3]. Exhaustive search of the solution space is also not suitable with $(m+1)^{card(\mathcal{F})}$ possible configurations to compute. We thus chose to adopt the Metropolis-Hastings algorithm [13], a stochastic optimization technique known for its flexibility. This algorithm employs an update mechanism that randomly perturbs the current configuration l into a new configuration l'. The perturbation is local, which means in our case that each update only modifies the label of one facet. The configuration l' is accepted for the next iteration according to a probability depending on the energy variation between l and l' and a relaxation parameter T that geometrically decreases at a rate

C. Although the Metropolis-Hastings algorithm is guaranteed to converge to the global minimum of our energy when using a logarithmic decrease for T [9], we prefer using a geometric decrease to have reasonable running times. While this approximation removes the guarantee of reaching the global minimum, it finds solutions close to this optimum in practice. As a mean of evaluation, we exhaustively evaluated the energy of the $279,936$ possible configurations for the drawing in Figure 7(left). Over 1000 runs, the global solution was found in 96% of the trials, and the remaining 4% corresponded to local minima very close to the global one.

Algorithm 1 details the main steps of our optimization. Figure 4 shows the evolution of the configurations during the optimization.

Algorithm 1 Optimization procedure

```
Compute the dominant orientations $d_1 ... d_m$
Generate a random configuration $l$
Initialize relaxation parameter $T = T_{init}$
repeat
    Generate $l'$ by perturbing the label of a random facet
    Compute the orientations $d_{new}$ in $l'$            {Sec.4.2}
    Compute 3D model $\mathcal{M}(l')$                     {Sec.4.3}
    Compute the Metropolis ratio $R = \exp\left(\frac{U(l)-U(l')}{T}\right)$
    Draw a random value $p \in [0,1]$
    if $p < R$ then update $l \leftarrow l'$
    else update $l \leftarrow l$
    Update $T \leftarrow C \times T$
until $T < T_{end}$
```

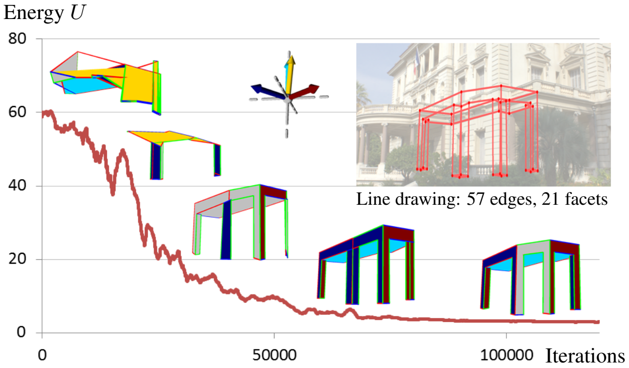
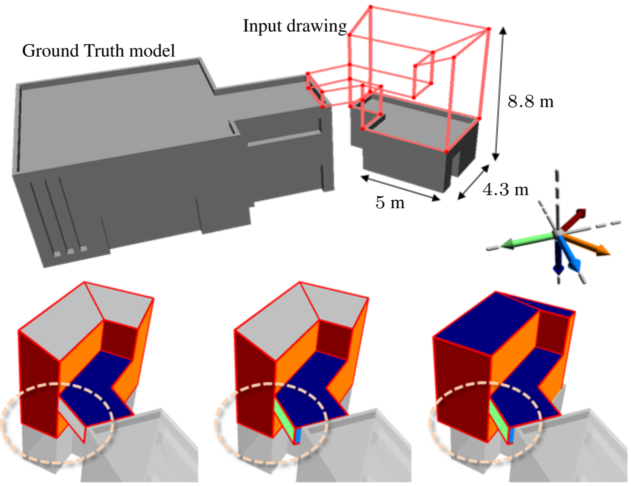


Figure 4. Energy minimization. In the first iterations, the current configuration l is of bad quality, represented by a high energy. As the relaxation parameter decreases, the update mechanism progressively becomes selective and moves the current configuration towards the optimal solution.

5. Experiments

We next describe a series of experiment to evaluate the flexibility, robustness and performance of our data-driven regularization. While our approach is complementary to other regularity cues used in prior work [16], we leave their unified treatment to future work.

We implemented our algorithm in C++, using the Computational Geometry Algorithms Library[2] for geometric operations. All timings were measured on an Intel Core i7 clocked at 2GHz. We used a standard digital camera to acquire our datasets. We also implemented a simple user interface to trace drawings over photographs. Users create segments by clicking their two extremities and we snap extremities together when they are within a 1% tolerance of the window diagonal. In what follows, we visualize each dominant orientation with a random color and new orientations with gray.



(a) $\gamma = 0.005$ (b) $\gamma = 0.1$ (c) $\gamma = 0.2$

Figure 5. The weight γ offers a trade-off between regularization and presence of new orientations. (a) Allowing too many new orientations does not provide enough regularization to correct for drawing inaccuracy. For example, while the bridge connects to the existing scene, it does not align well with its side wall. (b) Increasing γ encourages alignment of the bridge with the existing wall. (c) A high regularization prevents the use of new orientations, producing a flat roof.

Model parameters. The most important parameter of our algorithm is the weight γ, which penalizes the use of new orientations in Equation 2. This parameter offers a trade-off between robustness to imperfect line-drawings and regularity of the estimated model, as illustrated in Figure 5. A low γ favors the use of new orientations to interpret parts that do not align with existing geometry, while a high γ can correct drawing inaccuracy by approximating the 3D model with only dominant orientations. In practice we used the same value $\gamma = 0.005$ for all our datasets.

The other parameters were also kept fixed for all experiments. Specifically, the normal clustering (Section

[2] www.cgal.org

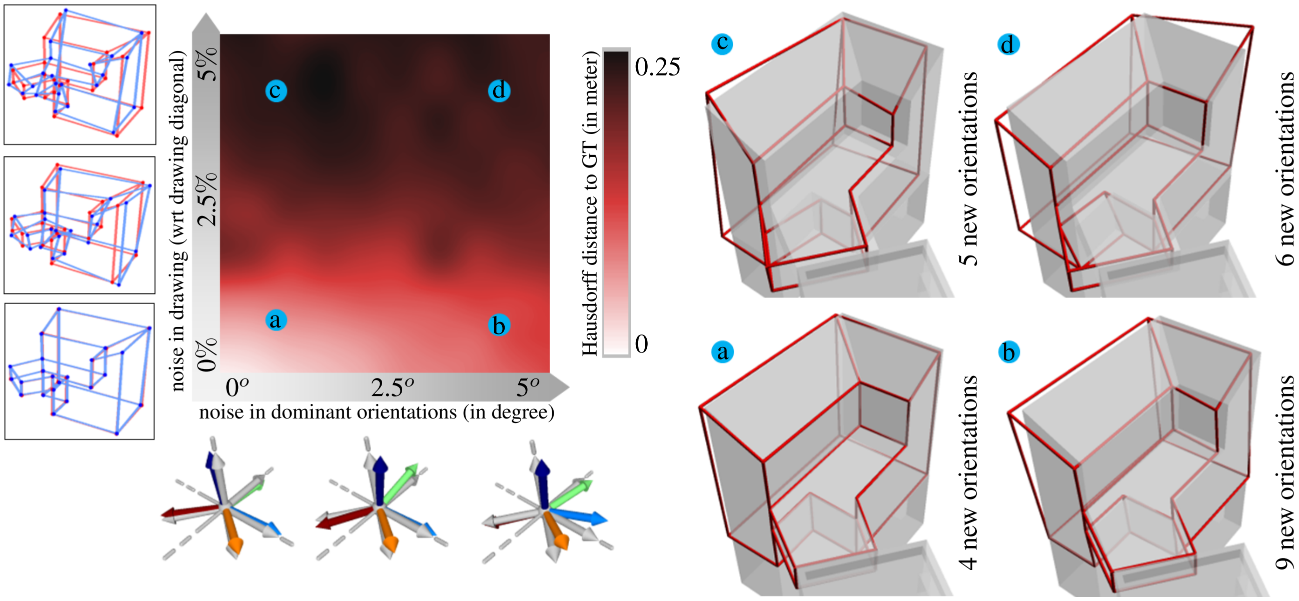


Figure 6. Robustness to noise. We use the Ground Truth (GT) model of Figure 5 to evaluate the robustness of our algorithm to noise in both the input drawing (rows of the error plot) and in the dominant orientations (columns of the error plot), with a fixed $\gamma = 0.005$. Because of the possibility of using new orientations, our algorithm is weakly affected by noise on the dominant orientations, as shown by configurations a and b that have a similar mean Hausdorff distance to GT. However, a low γ also makes our algorithm compensate for noise in the input drawing by resorting to new orientations.

3) is performed on a discretized stereographic projection of 360 × 360 pixels using a Mean-Shift Gaussian kernel with a 10 pixels standard-deviation. The geometric prior has a small weight of $\beta = 1/10$. Finally three parameters T_{init}, T_{end} and C control the speed and accuracy of the energy optimization. They are fixed by default to $1 \times card(\mathcal{F})$, $5 \times 10^{-4} \times card(\mathcal{F})$ and 0.99995 respectively.

Flexibility. Figure 10 illustrates various application scenarios of our method in urban planning (*Bridge*), furniture design (*Desk*), cultural heritage (*Ruins*) and architectural design (*House*). All these examples show how our system can be used to model existing content as well as imaginary objects, for instance to re-create destroyed buildings (*Ruins*) or extend existing ones (*House*).

Robustness. We evaluate the robustness of our algorithm with respect to three criteria: the presence of noise in the input drawing, the presence of noise in the extracted dominant orientations, and the presence of new orientations in the drawing.

The two first criteria are evaluated jointly on a Ground Truth dataset. We designed this dataset to contain the challenging cases we are interested in, such as non-Manhattan configurations and unknown orientations (see Figure 6). The error map shows that our algorithm is very robust to the presence of noise in dominant orientations, imperfect dominant orientations being replaced by new orientations. However, our algorithm also tends to resort to new orientations to compensate for strong noise in the input drawing.

We also evaluate the behavior of our algorithm as new orientations are inserted in the drawing (Figure 7). Our greedy estimation of new orientations can result in an accumulation of error for facets that are far from the ones labeled with dominant orientations. Nevertheless, Figure 7(d) shows successful reconstruction of 11 new orientations from only 5 dominant ones. Figure 7(e) shows a failure case where too few facets align with the existing geometry. Additional regularity cues would be needed to deal with such complex configurations.

Comparison with existing work. A strict comparison with previous work is not possible because of the novelty of our problem statement. In particular, our method leverages rich information provided by multi-view reconstructions, which is not available to existing methods. Existing algorithms also strongly rely on orthographic projection, and as such cannot be directly applied in our context. An exception is the algorithm of Masry [18], which we adapted to support a perspective camera model (Figure 8). This algorithm assumes that the drawing is dominated by three orthogonal directions and that a tree composed of lines with these directions can connect all vertices of the drawing. While this algorithm can reconstruct parts of our drawings, such as the chimney of the house in Figure 10, it fails for more complex objects. Our algorithm produces better results by exploit-

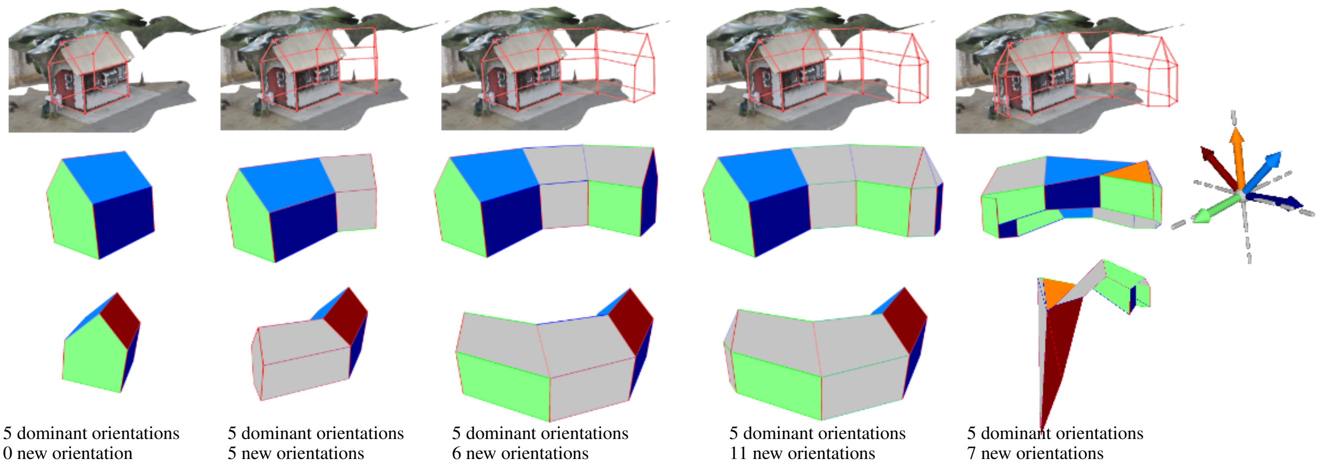


Figure 7. Robustness to new orientations. In this example, our algorithm produces consistent results despite adding up to 11 new orientations. However, the last column shows a failure case where two few known orientations were present to produce a valid result.

ing the orientations of the existing scene and by discovering new orientations.

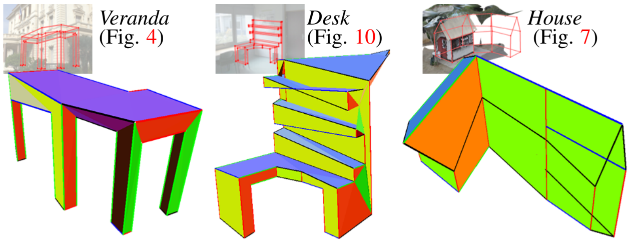


Figure 8. Comparison with [18]. This algorithm assumes that most lines in the drawing are aligned with one of three orthogonal directions. While this assumption is sufficient to reconstruct simple objects as the *Veranda* model (left), it is sensible to drawing accuracy and assigns erroneous directions to the near parallel lines at the top of the *Desk* model (middle). The algorithm fails on complex drawings where many lines do not align with the three orthogonal directions (right).

Performances. The drawings shown in this paper took between a few seconds (Figure 1) to a few minutes (Figure 10) to create with our interface. The running time of our algorithm depends on several factors, including the number of lines and facets in the drawing, the number of new orientations, the ambiguity between different orientations under a given viewpoint. Table 1 provides running times for the results presented in Figure 10 which are representative of these factors. Few minutes are necessary to both estimate dominant orientations and label a line-drawing with two dozens of facets. Note that the labeling recovers the 3D model as part of the optimization (Section 4.3).

	Bridge	*Desk*	*Ruins*	*House*
Dominant orientations	106	95	135	97
Labeling	347	876	271	483
Total	453	971	406	580

Table 1. Running time, expressed in second, of the algorithm for the datasets shown in Figure 10. Data loading and output saving are excluded from the timing.

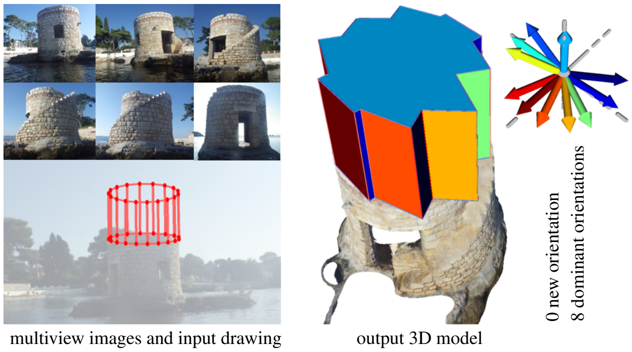
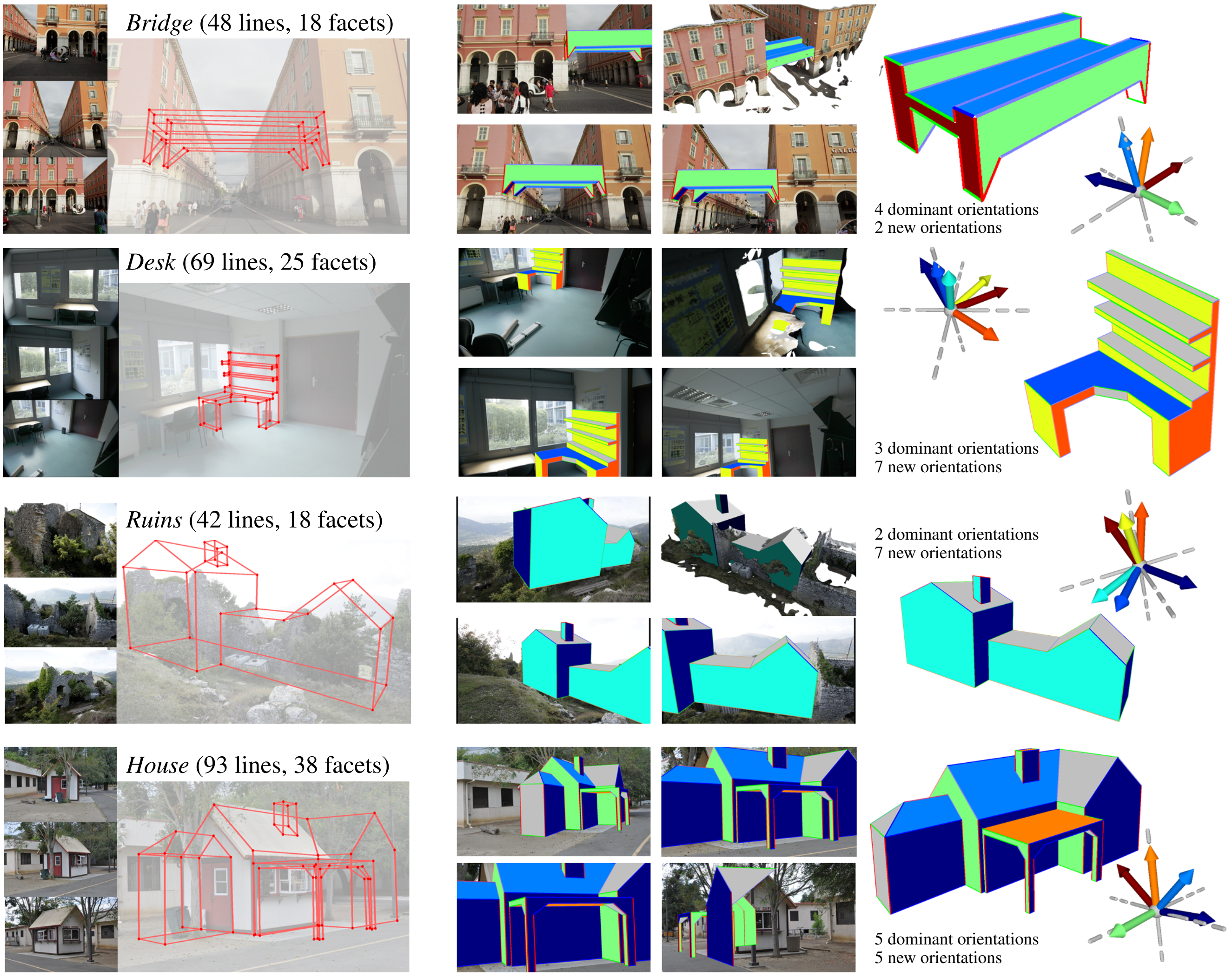


Figure 9. Failure case. Our system is not designed to model smooth surfaces. Attempting to approximate the shape with planar facets yields a jagged surface model, first because our geometric prior penalizes adjacent co-planar facets and second because the existing scene offers too many candidate orientations.

Limitations. Our algorithm is designed to reconstruct objects that can be represented by piecewise-planar surfaces. This assumption is reasonable for man-made environments, in particular for indoor and urban scenes, but can become too restricted for free-form objects. Figure 9 illustrates an attempt to extend a cylindrical tower, which our algorithm fails to interpret as a smooth surface.

Our algorithm also performs better when the drawing is traced from a viewpoint with minimal foreshortening. Fortunately, designers are trained to draw from such *informative* viewpoints [11].



(a) Input (b) Reconstruction in context (c) 3D model and orientations

Figure 10. Results of our method in different application scenarios. The *Ruins* and *House* examples were created with two and three drawings respectively, one per connected component.

6. Conclusion and future work

We have presented an approach to interpret line drawings of 3D objects when the drawing represents the extension of an existing scene. While little prior work has explored this application scenario, it is a common task in urban planning, furniture design and cultural heritage. At the core of our method is a labeling algorithm that combines the known dominant orientations of the scene with free orientations to offer a trade-off between regularization and discovery of new structures.

While we have demonstrated our approach using multi-view 3D reconstructions, our algorithm could equally apply to other sources of 3D information such as laser scanners and time-of-flight cameras. We also consider our approach to be a complement of existing regularity cues used in prior work such as orthogonality, parallelism and symmetry [16]. Finally, while we designed our algorithm to estimate 3D polyhedrons, in the future we plan to use these models as scaffolds for the interpretation of free-form shapes [22].

Acknowledgments

The authors thank Autodesk for the multi-view stereo reconstructions and Hyojin Kim for the *House* dataset. This work was partially supported by ANR-12-JS02-003-01 DRAO, the European Research Council (ERC Starting Grant Robust Geometry Processing, Grant agreement 257474), and software and research donations from Adobe.

References

[1] M. Arikan, M. Schwärzler, S. Flöry, M. Wimmer, and S. Maierhofer. O-snap: Optimization-based snapping for modeling architecture. *Trans. on Graphics*, 32(1), 2013. 2

[2] H. Barrow and J. Tenenbaum. Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17, 1981. 1, 2

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9), 2004. 4

[4] A.-L. Chauve, P. Labatut, and J.-P. Pons. Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data. In *CVPR*, 2010. 1

[5] J. Chen and B. Chen. Architectural modeling from sparsely scanned range data. *IJCV*, 78(2-3), 2008. 1, 2

[6] Y. Chen, J. Liu, and X. Tang. A divide-and-conquer approach to 3d object reconstruction from line drawings. In *ICCV*, 2007. 1, 2

[7] M. Cooper. *Line Drawing Interpretation*. Springer, 2008. 1

[8] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *SIGGRAPH*, 1996. 2

[9] X. Descombes. *Stochastic geometry for image analysis*. Wiley-ISTE, 2011. 5

[10] A. Dick, P. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *IJCV*, 60(2), 2004. 1

[11] K. Eissen and R. Steur. *Sketching: The Basics*. Bis Publishers, 2011. 7

[12] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Manhattan-world stereo. In *CVPR*, 2009. 1

[13] W. Hastings. Monte Carlo sampling using Markov chains and their applications. *Biometrika*, 57(1), 1970. 4

[14] M. Lau, G. Saul, J. Mitani, and T. Igarashi. Modeling-in-context: user design of complementary objects with a single photo. In *Proc. Sketch-Based Interfaces and Modeling*, 2010. 2

[15] S. Lee, D. Feng, and B. Gooch. Automatic construction of 3d models from architectural line drawings. In *Proc. Interactive 3D graphics & games*, 2008. 1, 2

[16] H. Lipson and M. Shpitalni. Optimization-based reconstruction of a 3d object from a single freehand line drawing. *Computer-Aided Design*, 28, 1996. 1, 2, 5, 8

[17] J. Liu, L. Cao, Z. Li, and X. Tang. Plane-based optimization for 3d object reconstruction from single line drawings. *PAMI*, 30(2):315–327, Feb 2008. 1, 2, 3

[18] M. Masry, D. Kang, and H. Lipson. A freehand sketching interface for progressive construction of 3d objects. *Computers & Graphics*, 29(4):563 – 575, 2005. 1, 2, 6, 7

[19] P. Musialski, P. Wonka, D. G. Aliaga, M. Wimmer, L. van Gool, and W. Purgathofer. A Survey of Urban Reconstruction. *Computer Graphics Forum*, 32(6), 2013. 1

[20] L. Olsen, F. F. Samavati, M. C. Sousa, and J. A. Jorge. Sketch-based modeling: A survey. *Computers & Graphics*, 33(1), 2009. 1

[21] D. Perkins. Cubic corners, oblique views of pictures, the perception of line drawings of simple space forms. geometry and the perception of pictures: Three studies. Technical report, Harvard Univ., Cambridge, MA. Graduate School of Education., 1971. 2

[22] R. Schmidt, A. Khan, K. Singh, and G. Kurtenbach. Analytic drawing of 3d scaffolds. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 28(5), 2009. 8

[23] R. Schnabel, R. Wahl, and R. Klein. Efficient ransac for point-cloud shape detection. *Computer Graphics Forum*, 26(2):214–226, June 2007. 2

[24] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009. 1, 2

[25] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3d architectural modeling from unordered photo collections. *Trans. on Graphics*, 27(5), 2008. 2

[26] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher III. A Mixture of Manhattan Frames: Beyond the Manhattan World. In *CVPR*, 2014. 1, 2

[27] A. van den Hengel, A. Dick, T. Thormählen, B. Ward, and P. H. S. Torr. Videotrace: Rapid interactive scene modelling from video. *Trans. on Graphics*, 26(3), 2007. 2

[28] L. Yang, J. Liu, and X. Tang. Complex 3d general object reconstruction from line drawings. In *ICCV*, 2013. 1, 2