Artificial Intelligence for Engineering Design, Analysis and Manufacturing

cambridge.org/aie

What shape grammars do that CAD should: the 14 cases of shape embedding

Tzu-Chieh Kurt Hong and Athanassios Economou

School of Architecture, College of Design, Georgia Institute of Technology, Atlanta, GA, USA

Research Article

Cite this article: Hong T-CK, Economou A (2022). What shape grammars do that CAD should: the 14 cases of shape embedding. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **36**, e4, 1–20. https://doi.org/10.1017/S0890060421000263

Received: 14 May 2021
Revised: 1 September 2021
Accepted: 1 September 2021

Key words:
Shape; shape embedding; shape grammar interpreters; shape grammars

Author for correspondence:
Athanassios Economou,
E-mail: economou@gatech.edu

Abstract

Shape queries based on shape embedding under a given Euclidean, affine, or linear transformation are absent from current CAD systems. The only systems that have attempted to implement shape embedding are the shape grammar interpreters albeit with promising but inconclusive results. The work here identifies all possible 14 cases of shape embedding with respect to the number of available registration points, four for determinate cases and ten for indeterminate ones, and an approach is sketched to take on the complexities underlying the indeterminate cases. All visual calculations are done with shapes consisting of straight lines in the Euclidean plane within the algebra U_{ij} for $i = 1$ the dimension of lines and $j = 2$ the dimension of space in which the lines are defined, transformed and combined. Aspects of interface design and integration to current work design workflows are deliberately left aside.

Introduction

A shape query in computer-aided design (CAD) systems is enabled by a database query requesting the retrieval of instances of shapes from a CAD database. The list of the commands available to a designer to perform a query for a particular shape is impressive and involves multiple ways of interfacing with the CAD software including typing, pointing, and pictorially interacting with the model in a variety of ways including clockwise and counterclockwise window-selecting, cross-selecting shapes, drop-down menus, listboxes, checkboxes, toggles, steppers, and so on, all with ready-made or *ad hoc* lists of shapes, combinations of shapes, and properties of shapes; and yet, regrettably, all are ultimately useless when it comes to enabling the designers retrieve shapes that cannot be derived from the ones inserted in the database.

Surprisingly, shape query under a given Euclidean, affine, or linear transformation (shape embedding) is entirely absent from current CAD systems: This might come at a surprise when CAD geometry is intimately related to computational geometry – an ever-growing field of algorithms dealing with geometric problems involving operations on points, lines, planes, and solids (see, e.g., Preparata and Shamos, 1990; Goodman and O'Rourke, 1997). An impressive list of formal approaches on comparison of shapes (shape matching) in computational geometry include diverse matching techniques such as tree pruning, pose clustering, geometric hashing, deformable templates, Fourier descriptors, wavelet transforms, and neural networks (Besl and Jain, 1985; Foley *et al.*, 1997; Loncaric, 1998; Campbell and Flynn, 2001; Cardone *et al.*, 2003); and an expanding universe of classes of shapes to be compared including sketch-based shapes, non-rigid shapes, relief surface patches, multi-domain protein shapes, gesture sequence shapes, watertight models, and so forth (Tangelder and Vetlkamp, 2008). And yet, the single most important characteristic in all these approaches, the measurement of how similar or dissimilar a shape is with another shape, using some appropriately constructed similarity or dissimilarity measure (Veltkamp and Hagedoorn, 2001), is an indifferent measure when it comes to the notion of shape embedding under a given Euclidean, affine, or linear transformation (Stiny, 2006).

The only systems that have attempted to implement shape embedding are the shape grammar interpreters – computer applications that process shape rules encoded in shape grammars (Krishnamurti, 1981, 1982, 1992; Krishnamurti and Giraud, 1986; Stouffs, 1994; Tapia, 1999; Trescak *et al.*, 2009; Jowers *et al.*, 2010; Jowers and Earl, 2011; Grasl and Economou, 2013; Ruiz-Montiel *et al.*, 2014; Stouffs and Li, 2020). A shape grammar consists of shape-rewriting rules cast in the form $u \rightarrow v$, for a shape u is rewritten as a shape v (Stiny, 1975). A shape rule $u \rightarrow v$ is applied to a shape W, when there is a geometric transformation f that makes the shape $f(u)$ part of the shape W – or alternatively, when there is a transformation f that embeds the shape $f(u)$ in W. The resulting computation identifies the instance of the shape $f(u)$ in the shape W and replaces it with the corresponding instance of the shape $f(v)$ to generate a new shape $W' = W - f(u) - f(v)$. Clearly, any application of a shape rule requires that the shape u can be embedded in a design. And if there is no need for the specification of a design action (shape v), these the shape rewrite rules are modeled after the identity shape rule $u \rightarrow u$ (Stiny, 1996; Economou *et al.*, 2021). Shape grammars do offer a generous formalism for

© The Author(s), 2022. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

CAMBRIDGE UNIVERSITY PRESS

design supporting ambiguity, redescription, and variability – and since their introduction in the mid-seventies have always offered a radical counterpart to mainstream research in CAD (McCullough, 2006; Knight, 2015).

All shape grammar interpreters approach the problem of implementing shape embedding in two ways, namely, by (a) the database query of the maximal elements that the shapes are made of, see for example, GRAPE (Grasl and Economou, 2013, 2018) and Sortal (Stouffs, 2018; Stouffs and Li, 2020) and (b) the derivation of the transformation matrix that embeds one shape into another, see for example, SGI (Krishnamurti, 1982; Krishnamurti and Giraud, 1986); SGS (Chase, 1989); GRAIL (Krishnamurti, 1992); and GEdit (Tapia, 1999). Both approaches rely on the formal descriptions of the maximal representation of shape and its registration marks or points (Stiny, 1975, 2006; Krishnamurti, 1981). The implementation of shape embedding through database query, despite its success in well-structured cases and design workflows, fails to provide a general solution to the subshape problem (McKay *et al.*, 2012). The implementation of shape embedding through the derivation of the transformation matrices is still the most promising method, but it is plagued by a series of problems across several fronts (Stouffs, 2019; Hong and Economou, 2021).

The work here focuses on the systematic exposition of the challenges underlying the implementation of the derivation of the transformation matrices for shape embedding, and the opportunities residing in the reworking of the problem. More specifically, 14 cases of shape embedding are identified with respect to the number of available registration points, four for determinate cases and ten for indeterminate ones, and an approach is sketched to take on the complexities underlying the indeterminate cases. All shape calculations are done with shapes consisting of straight lines in the Euclidean plane, that is, the algebra U_{ij} for $i = 1$ the dimension of lines, and $j = 2$ the dimension of space in which the lines are defined, transformed, and combined (Stiny, 1991, 2006; Krstic, 2014). Aspects of interface design and integration to current work design workflows are deliberately left aside.

Calculating embedding

Shape embedding is the process of embedding a shape u into a shape W under a transformation f. The two main parts of the visual match to consider are (a) the relation between the shape u and its instance $f(u)$ and (b) the relation between the shape $f(u)$ and W. In the first case, we are interested in the transformation f that relates the shapes u and $f(u)$: Do we want to search for congruent shapes with the exact size and shape with the shape u? Or similar shapes differing in size? Or affine congruent ones? Or projective congruent ones? The invariance of particular characteristics between the shapes u and $f(u)$ for Euclidean, affine, and linear transformations f require radically different geometrical constructions (Veblen and Young, 2007). In the second case, we are interested in the part relation between the shapes $f(u)$ and W, that is, the case that $f(u) \leq W$ holds true. Does a particular transformation f make the shape $f(u)$ part of W? Are there more instances $f(u)$ of the shape u that can be embedded in W under a particular group of transformations? Are there none? It might be that a visual search or embedding for a congruent instance $f(u)$ of a shape u gives null results but a more relaxed search for an affine congruent shape $f(u)$ does; or it might be that the visual match is indeterminate, that is, additional instructions and calculations have to be performed. A brief account of both steps of the visual match is given below confined for shapes made of lines in the algebra U_{12}.

The first step for a visual embedding of a shape u into a shape W is the appropriate specification of the geometric relation between the shape u and the shape $f(u)$ to be matched in W: More broadly, for a shape u, the shape $f(u)$ can be modeled by four types of transformations: (a) isometric transformations (or isometries) including translations, reflections, and rotations; (b) similarity transformations (or similarities) including isometric transformations, scale transformations and their combinations; (c) affine transformations (or affinities) including similarities, stretch, compress, and shear transformations and their combinations; and (d) linear transformations (or linearities) including affine transformations, one-point and two-point perspective transformations and their combinations. Each of these types of transformations vary some aspects of the shape and leave other aspects invariant: The isometric transformations preserve lengths and angles and vary position; the similarity transformations preserve angles and vary length and position; the affine transformations preserve parallelism and vary angles, lengths, and position; and the linear transformations preserve cross ratio and vary parallelism, angles, lengths, and position (March and Steadman, 1974). Any finite sequence of the above transformations specify the type of geometric congruence between the two shapes u and $f(u)$: two shapes are congruent if there is a finite sequence of isometric transformations mapping one shape to another; two shapes are similar if there is a finite sequence of similarity transformations mapping one shape to another; two shapes are affine congruent if there is a finite sequence of affine transformations mapping one shape to another; and two shapes are projective congruent if there is a finite sequence of linear transformations mapping one shape to another. The rising hierarchy of the transformations f is given in Figure 1 for a shape u in the form of a lowercase k. The initial instance of the shape u is taken here as the identity transformation $f(u)$.

The second step for a visual embedding of a shape u into a shape W is the specification of the appropriate registration mechanism to capture the geometric congruence of the shape $f(u)$ with parts of the shape W including the complete shape W. The selection of these registration points has been somewhat of a moving target in the literature and various approaches have been proposed (see, e.g., Stiny, 1975, 2006; Krishnamurti, 1981; Krishnamurti and Giraud, 1986; Krishnamurti and Earl, 1992; Stouffs, 1994; Earl, 1997; Tapia, 1999). Here, the registration points are solely defined as the intersections of the hyperplanes (see, e.g., Alexanderson and Wetzel, 1978) that the maximal lines are embedded in Stouffs and Krishnamurti (2019). More specifically, a registration point can be incident with (a) two maximal lines; (b) a maximal line and a boundary of maximal line; (c) two boundaries of maximal lines; (d) a maximal line and a hyperplane upon which a maximal line is embedded; (e) a boundary of a maximal line and a hyperplane upon which a maximal line is embedded; and (f) two hyperplanes upon which two maximal lines are embedded. A visual illustration of the six types of registration points and an instance of a shape u featuring all types of registration points are both given in Figure 2.

The selection of the particular type of geometric congruence between the shape u and $f(u)$ – namely, the choice whether the shape $f(u)$ is a congruent, similar, affine congruent or projective congruent instance of u – and the available registration points R_u of the shape u and the points R_w of the shape W determine (a) whether there is indeed an embedding of the shape $f(u)$ in

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

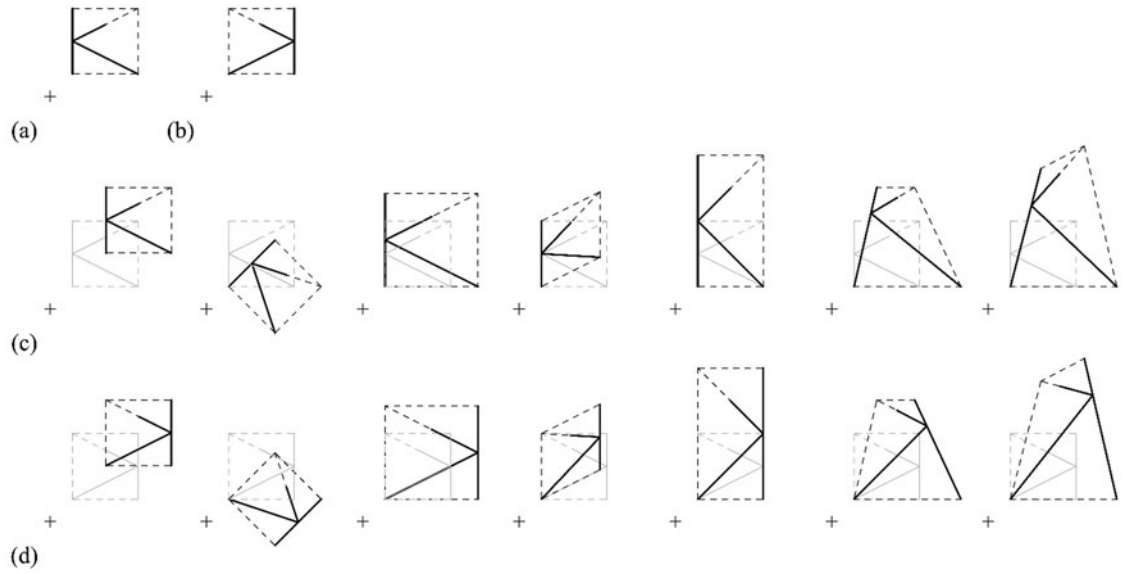


Fig. 1. Types of linear transformations. (a) Identity; (b) reflection; (c) direct transformations (from left to right): translation; rotation; scale; shear; stretch; one-point perspective; two-point perspective; (d) indirect or handed versions of the transformations in (c).

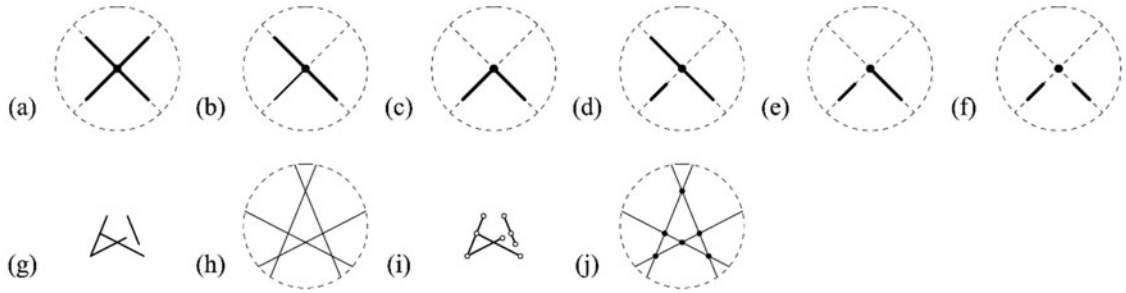


Fig. 2. Congruence of shapes and registration points. (a–-(f) Six types of registration points; (g) a shape exhibiting all six types of registration points; (h) four construction lines (hyperplanes) of shape *u*; (i) seven endpoints of shape *u*; and (j) six registration points of shape *u*.

W, and if there is one, (b) whether this embedding can be done in a determinate or indeterminate way. A comprehensive overview for all possible cases of determinate and indeterminate embeddings under Euclidean and affine transformations for all algebras U_{ij}, $0 \le i \le j \le 3$ has been given by Krishnamurti and Stouffs (1997). Interestingly, shapes made up of lines can be embedded in a determinate way under a particular kind of transformation and in an indeterminate way under some other. Most of the current shape grammar implementations use algorithms that can calculate determinate embeddings under isometric and similarity transformations (Euclidean transformations), and there is no general way to treat indeterminate embeddings. The goal of this work is to trace these cases and identify the conditions that need to be resolved for a general implementation of embedding. The work focuses on the outstanding issues associated with this method and illustrates them through a series of iterative visual calculations against a single shape *W* to make them as manifest as possible. The shape *W* pays homage to the familiar shape of the two nested squares (or four triangles, four pentagons, two hexagons, and so forth) that has characterized the field of the shape grammars since its first appearance in Stiny (1975) and Gips (1975) and afterwards (see, e.g., Stiny, 1980; Knight and Stiny, 2001; Mitchell, 2001; Stouffs and Krishnamurti, 2019).

Implementation of determinate embedding using transformation matrices

The general algorithm to the derivation of the transformation *f* for shape embedding by calculating transformation matrices has been given in Krishnamurti (1981, 1982) and Krishnamurti and Giraud (1986): A 3×3 matrix is used to represent all linear transformations, including isometries, similarities, affinities, and linearities. The calculation of the transformation between two shapes in 2D space is done by the derivation of the 3×3 matrix by plugging in the coordinates of the required number of registration points between the two shapes. This method is based on the fact that any point on the Cartesian plane is assigned with a coordinate with three real numbers *x*, *y*, and *z* to represent an absolute location in the space. For the four types of linear transformations and $\mathrm{Num}(R_u)$ the number of registration points of the shape *u*, there are four cases of determinate embedding to consider: (a) $\mathrm{Num}(R_u) = 1$ under isometric transformations; (b) $\mathrm{Num}(R_u) = 2$ under similarity transformations; (c) $\mathrm{Num}(R_u) = 3$ under affine transformations; and (d) $\mathrm{Num}(R_u) = 4$ under linear transformations. Clearly, any embedding with a larger set of available registration points should be determined, calculable, and otherwise straightforward – but see some further notes on this topic in the end of this work.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

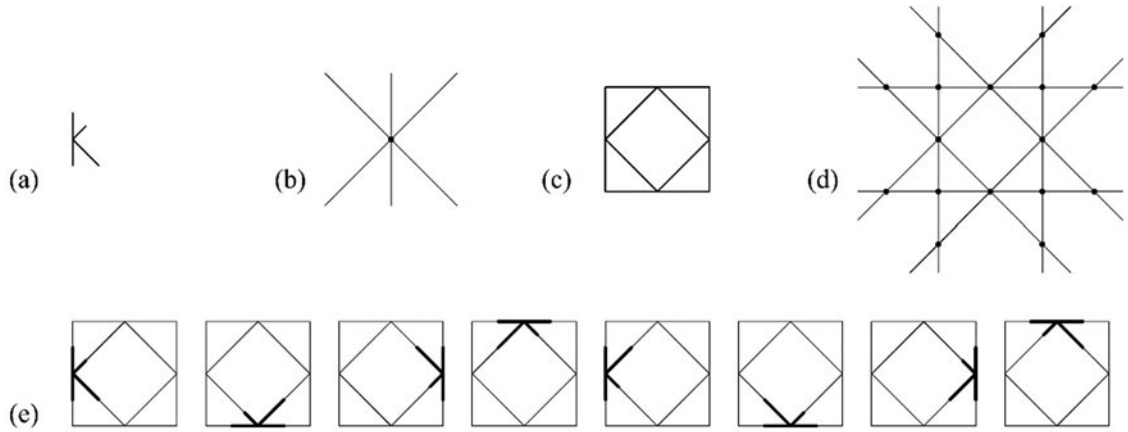


Fig. 3. An example of determinate congruent embedding with one registration point that cannot be implemented by the derivation of the isometry matrix. (a) Shape u; (b) registration points R_u; (c) shape W; (d) registration points R_W; and (e) eight results of determinate embedding under isometric transformations.

Determinate embedding under isometric transformations

An embedding of a congruent instance of a shape u in a shape W is determinate when the shape u has one or more registration points. An example of a determinate congruent embedding of a shape u having one registration point is shown in Figure 3. Note that the limiting case that the shape u has only one registration point – that is, one point coincident with the intersection of its hyperplanes cannot be calculated by the current algorithm of the derivation of the isometry matrix because the algorithm requires two points from each shape to resolve the three unknown variables of the isometric matrix, namely, the rotation angle θ, the translation in the x-direction t_x, and the translation in the y-direction t_y. In this sense, the embedding is visually possible but operationally impossible in terms of the given algorithm. The manual addition of "distinguishable" points in the set of registration points of the shape u is not readily deployable in automated applications of the algorithm, and the automated sampling of endpoints of maximal lines, regrettably, brings more problems than actual solutions – a pictorial example is offered in the final part of this work along with a series of cases that need to be considered for the automated implementation of visual matching. The eight possible congruent embeddings of the shape $f_i(u)$ are all determined by the interactions of the symmetry elements of the symmetry groups of the shapes u and W (Stiny, 1991) – here, the cyclic group C_1 of order 1 and the dihedral group D_4 of order 8, respectively (Armstrong, 1997).

Determinate embedding under similarity transformations

An embedding of a similar instance of a shape u in a shape W is determinate when the shape u has two or more registration points. An example of a determinate embedding under similarity transformations of a shape u having two registration points that is successfully implemented by the derivation of the similarity matrix is shown in Figure 4. The automated derivation of the similarity matrix is successful because all possible pairs of points sampled from the sets of registration points of the shapes u and W can be used to resolve the four unknown variables of the similarity matrix, that is, translation in the x-direction t_x, translation in the y-direction t_y, rotation angle θ, and scaling factor s. More specifically, the shape u has two registration points, both coincident with the intersections of two pairs of hyperplanes, and the shape

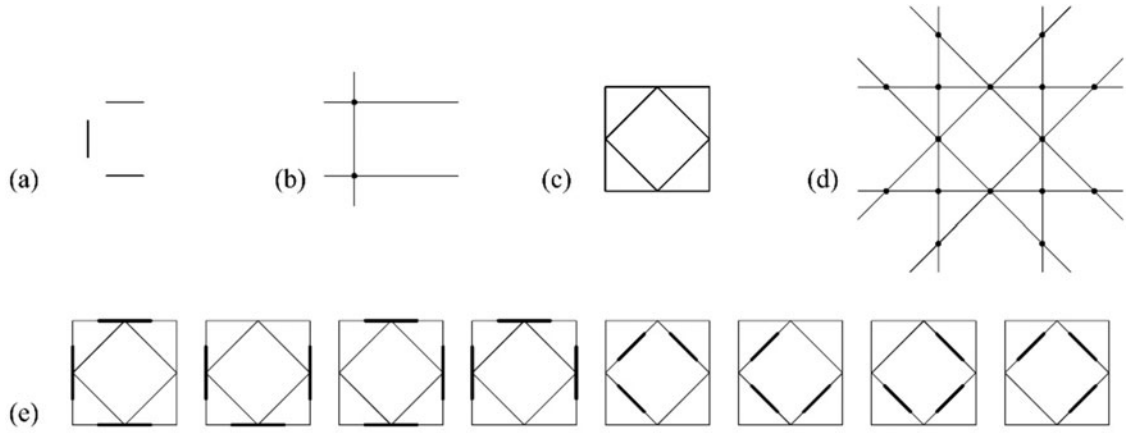


Fig. 4. An example of determinate similar embedding with two registration points that is successfully implemented by the automated derivation of the similarity matrix. (a) Shape u; (b) registration points R_u; (c) shape W; (d) registration points R_W; and (e) eight results of unrestricted embedding under similarity transformations.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

W has 16 registration points, all coincident with the intersections of the hyperplanes of its maximal lines. The automated sampling of two points out of the set of two registration points in the shape u and the set of 16 registration points in the shape W yields $C_2^{16} = 120$ pairs. The algorithm of the derivation of the similarity matrix checks whether any of the 120 pairs of pairs of points resolves the four unknown variables and the calculation yields eight pairs that lead to eight pictorially nonequivalent matches. Note that the derivation of the similarity matrix for a shape u having two or more registration points is the only one that currently works well among all possible cases of embedding, see, for example, SGI (Krishnamurti, 1982; Krishnamurti and Giraud, 1986); SGS (Chase, 1989); GRAIL (Krishnamurti, 1992); GEdit (Tapia, 1999); and SGI-RF (Trescak *et al.*, 2009).

Determinate embedding under affine transformations

An embedding of an affine instance of a shape u in a shape W is determinate when the shape u has three or more registration points – provided that at least three registration points in the shapes u and W are not collinear. The three noncollinear registration points provide sufficient information to resolve six unknown variables of affine matrix including translation in the x-direction t_x, translation in the y-direction t_y, rotation angle θ, and stretching factor in the x-direction α_x, stretching factor in the y-direction α_x, and a shearing angle φ. An example of a determinate embedding under affinity transformations of a shape u having three registration points that is successfully – albeit partially – implemented by the derivation of the affine matrix is shown in Figure 5. The shape u has three registration points, all coincident with the intersections of three pairs of hyperplanes, and the shape W has 16 registration points arranged as discussed above. The automated derivation of the affine matrix is partially successful here because some triples sampled from the set of registration points of the shapes W are collinear and in these cases the computation fails to resolve the six unknown variables. The automated sampling of three points out of the set of 16 registration points in the shape W should limit itself in the set comprised by the complete set of registration points minus the triples of points incident in the four pairs of hyperplanes above, that is, $C_3^{16} - (4 \times C_3^5 + 4 \times C_3^4) = 560 - 54 = 504$ triples of registration points. The algorithm of the automated derivation of the affine matrix should check whether any of the 504 pairs of triples of points resolves the six unknown variables and the calculation should yield eight pairs of triples that lead to eight pictorially nonequivalent matches.

Determinate embedding under linear transformations

An embedding of a projective instance of a shape u in a shape W is determinate when the shape u has four or more registration points – provided that at least three registration points in the shapes u and W are not collinear. An example of a determinate embedding under linearity transformations of a shape u having four registration points that is successfully – albeit partially – implemented by the derivation of the homography matrix (Hartley and Zisserman, 2004) is shown in Figure 6. The four noncollinear registration points provide sufficient information to resolve the nine unknown variables of a homography matrix h_0 to h_8, including the combinations of the rotation angles θ_x, θ_y, and θ_z; translation in the x-direction t_x, translation in the y-direction t_y, translation in the z-direction t_z, and a projection parameter p. Note that the rotation and the translation parameters represent the transformation of the camera (viewer), that is, the transformation of the world coordinates. With the projection parameter p, the two-dimensional perspective transformation can be represented as a homography matrix with nine variables. The shape u has six registration points, two coincident with the intersections of two pairs of maximal lines and four more coincident with the intersections of four pairs of hyperplanes; and the shape W has 16 registration points arranged as discussed above. The automated derivation of the homography matrix is partially successful because some quadruples and triples of points sampled from the set of registration points of the shape W are collinear, and thusly, the computation fails to resolve the nine unknown variables for these particular subsets. The problem here is that the computation of the derivation of the homography matrix needs to exclude: (a) the four sets of collinear triples of registration points in the shape u and (b) the eight sets of collinear triples in the shape W. The automated sampling of four points out of the set of six registration points in the shape u should limit itself to a pair of registration points for each hyperplane, that is, $C_4^6 - 4 \times C_3^3 \times 3 = 3$ quadruples (as opposed to $C_4^6 = 15$), and the corresponding automated sampling of four points out of the set of 16 registration points in the shape W should limit itself to a pair of registration points for each hyperplane too, that is, $C_4^{16} - 4 \times C_3^5 \times 13 - 4 \times C_3^4 \times 13 = 1092$ quadruples as opposed

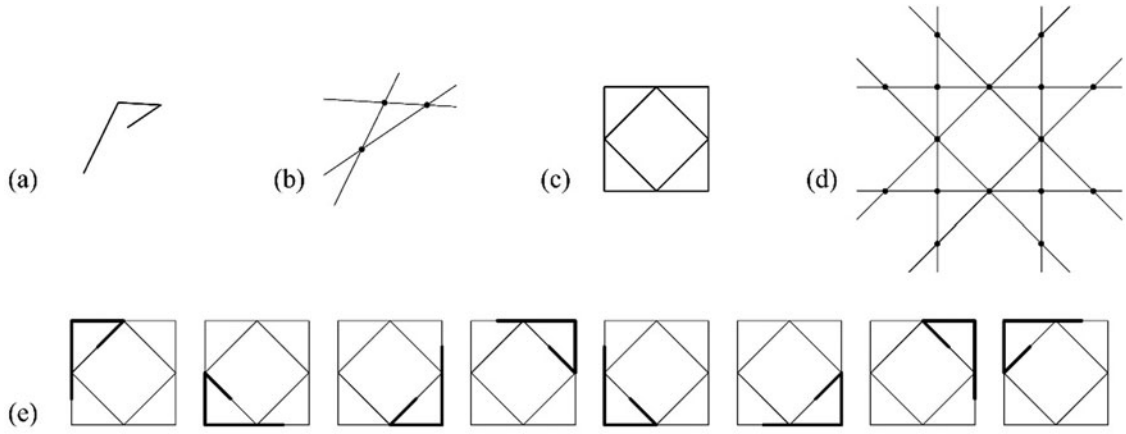


Fig. 5. An example of determinate affine embedding using three registration points that is partially implemented by the automated derivation of the affine matrix. (a) Shape u; (b) registration points R_u; (c) shape W; (d) registration points R_W; and (e) eight results of determinate embedding under affinity transformations.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

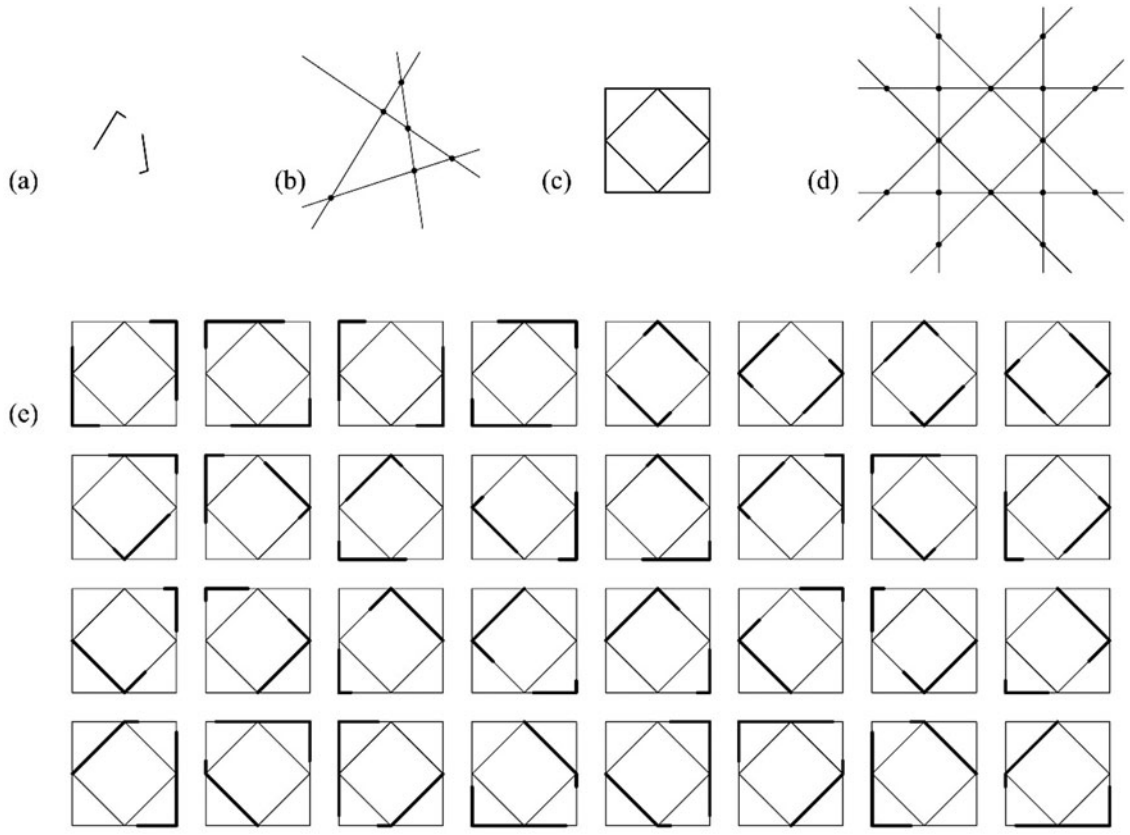


Fig. 6. An example of a determinate projective embedding using four registration points that is partially implemented by the automated derivation of the homography matrix. (a) Shape u; (b) registration points R_u; (c) shape W; (d) registration points R_W; and (e) 32 results of determinate embedding under linearity transformations.

to $C_4^{16} = 1820$. The algorithm of the automated derivation of the homography matrix should check whether any of the 1092 pairs of quadruples of points resolves the nine unknown variables and the calculation should yield 32 pairs of quadruples that lead to 32 pictorially nonequivalent matches.

Implementation of indeterminate embedding using transformation matrices

The review of the implementation of the derivation of the transformation matrices for determinate embeddings foregrounded several problems with the implementation of the current algorithm and its ability to provide robust calculations for visual embeddings. Intuitively, the review of the implementation of the derivation of the transformation matrices for indeterminate embeddings will foreground even more challenges that will need to be properly addressed. In general, the embedding is indeterminate when the shape u has less registration points than required to derive the transformation matrices. For the four types of linear transformations and $\text{Num}(R_u)$ the number of registration points of the shape u, there are ten cases of indeterminate embedding to consider: (a) $\text{Num}(R_u) = 0$ under isometric transformations; (b) $\text{Num}(R_u) = 1$ or 0 under similarity transformations; (c) $\text{Num}(R_u) = 2$, 1, or 0 under affine transformations; and (d) $\text{Num}(R_u) = 3$, 2, 1, or 0 under linear transformations. Clearly, none of these cases can be resolved by current implementations of automated derivations of transformation matrices, and yet, the fundamental problem remains the same as the one already encountered in the problems discussed on determinate embeddings: the selection of the right kind of families of registration points for the automated sampling to derive the transformation matrices. A possible way to address this issue is to identify ways to pick up "distinguishable" points (Krishnamurti, 1981) in the shape u, test their spatial relations to other points in the shape u as well as to corresponding pairs of points in the shape W, and then determine gradually the transformation and the derivation of the matrix. Put differently, a possible way to address the issue is to view the shape u as an instance of a parameterized shape U comprised by variables x – typically, the lines and the angles between the lines of the shape u and their associated midpoints and endpoints (intersections are already collected in the set of registration points) – assign real values determined by a function g to satisfy explicit conditions and constraints to produce the shape $g(U)$ (Stiny, 1989) and check whether there is a transformation f of the shape $g(U)$, such that the shape $f(g(U))$ is part of the design W. For every match of one of the variables of the shape $f(g(U))$ in W, there is an emergent set of distinguishable registration points that may be selected to determine the visual match. This multi-stepped process can indeed address all possible cases of indeterminate embedding and it always involves a number of steps equal to Num(required registration points) − $\text{Num}(R_{g(U)}) + 1$. Even more, the spatial relation between the shape $f(g(U))$ and the shape W partitions the symmetry group of the shape $f(g(U))$ into q classes in terms of W, and in congruence with the determinate matchings, it specifies how the shape rule can be used in q distinct ways (Stiny, 1991).

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

Significantly, the sequence of transformations for the resolution of the indeterminate does matter and may involve loops too. For example, the sequence for a shape embedding under isometric transformations consists of a translation, rotation, and a reflection, if required. Any transformation f derived in this manner is the product (multiplication) of the sequence of the applicable transformations, that is, $t_1t_2\ldots t_n$. Note that not every transformation t_i is effective. A trivial example is when the shape $g(U)$ is equal to shape W where shape embedding is processed without applying any transformation, regardless shape embedding is processed under isometric, similar, affine or linear transformation. More importantly though, not all transformations t_1, t_2,…, t_n, within the sequence of transformations can be determined. In the same example of, say, a shape embedding under isometric transformations, if the translation cannot be determined, then the next transformation in the sequence, rotation, is determined by calculating the angle between the hyperplanes of $g(U)$ and W – even at the absence of a registration point found in $g(U)$ – and the translation is determined afterwards based on some value selected through a user response from an admissible range or some predefined criterion. The section below provides an overview of the ten possible cases of indeterminate embedding that feature less registration points than the ones required for the derivation of the transformation matrices and fills with some detail the sketch outlined above.

Indeterminate embedding under isometric transformations

An example of an indeterminate embedding under isometric transformations of a shape u having no registration points is shown in Figure 7. The shape u does not have the required registration points for the derivation of the isometric matrix and the computation fails. To resolve the three unknown variables of the isometric matrix including the translation in the x-direction t_x, the translation in the y-direction t_y, and the rotation angle θ, a multi-stepped process can be adopted to address this problem. In this case, a two-stepped process is used to determine the embedding results. The first step is processed automatically by taking two endpoints of the shape $g(U)$ to determine the rotation θ and confine the translation range to a one-dimensional space, that is, any maximal line of W that has a larger length than $g(U)$. After the first step, the indeterminacy of embedding occurs in translation within any maximal line which is longer than $g(U)$. The second step requires an additional input (manual visual inspection or an automated optimization calculation) for a translational parameter t with a range lower bound $\leq t \leq$ upper bound,

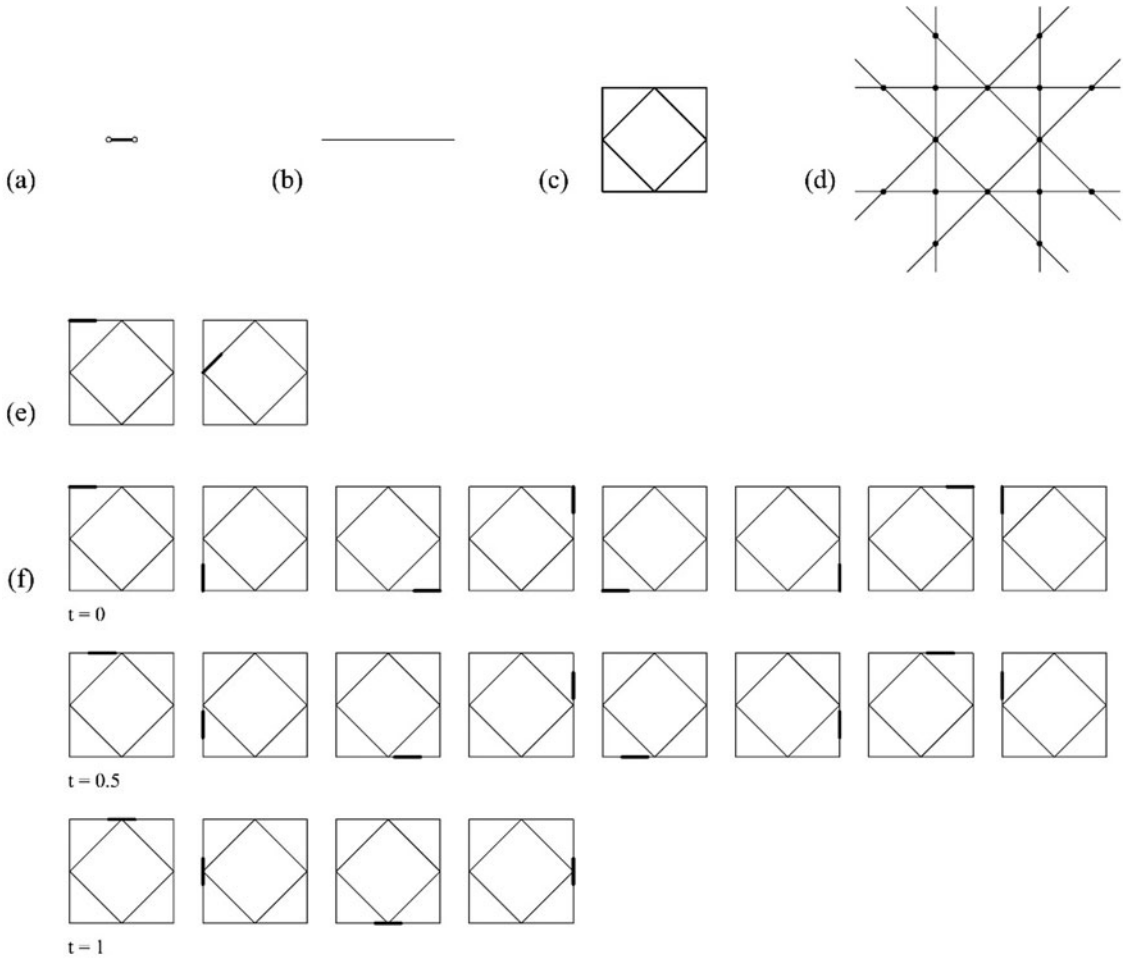


Fig. 7. An example of indeterminate congruent embedding using no registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) two embedding schemas, one per set of lines of equal lengths; (f) three instances of one of the two embedding schemas with assignments t: 0, 0.25, and 0.5, one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

here, the lower bound = 0 and upper bound = $(l_i - a)/2$, where l_i is the lengths of the sides of the two squares and a is the length of the line of $g(U)$ – the two parametric values are different for each square. Once the value of t is provided, in some absolute or relative sense, the system can determine the final embedding results. Note that the calculation of the range of translations for each embedding uses the midpoint of the shape $f(g(U))$ as a distinguishable point so that it can calculate specifically the nonequivalent matchings that will be then permuted by the actions of the symmetry group of each line in the shape W. In this particular case, the translation of the shape $g(U)$ in any of the two types of embedding schemas in the shape W produces eight matchings but when the midpoint of the shape $f(g(U))$ coincides with the midpoint of any of the edges of the shape W, one symmetry element of the shape $g(U)$ coincides with one of the symmetry elements of the shape W and the shape matches reduce to four per square.

Indeterminate embedding under similarity transformation

An example of an indeterminate embedding under similarity transformations of a shape u having one registration point is given in Figure 8. In this example, two registration points are required to resolve the four unknown variables of a similarity matrix including the rotation angle θ, the translation in the x-direction t_x, the translation in the y-direction t_y, and the scaling factor s. The shape u does not have the required registration points for the derivation of the similarity matrix and the computation fails. To address this problem, a two-stepped process is used to determine the embedding results. The first step is processed automatically by taking one registration point and one endpoint of $g(U)$ to derive θ, t_x, and t_y. The second step requires an additional input for the scaling factor s because the indeterminacy occurs in scaling. The process of dealing with indeterminate embedding in scaling is similar to the indeterminate embedding in translation and it can be similarly resolved by manual visual inspection or an automated optimization calculation. The shape embedding can be parameterized with a scaling parameter s that has a scaling range lower bound $\leq s \leq$ upper bound, in this example, the lower bound = 0 and the upper bound = 1. The calculation of the lower bound and upper bound is context-sensitive, and the indeterminacy might show up as multiple parameters. In this example, there is a single embedding schema of the shape $f(g(U))$ in the shape W, eight matches under the symmetry group of the shape W, and for each one of them there is an assignment g parameterized with a scaling factor s with a range $0 \leq s \leq 1$.

An example of an indeterminate embedding under similarity transformations of a shape u having no registration points is

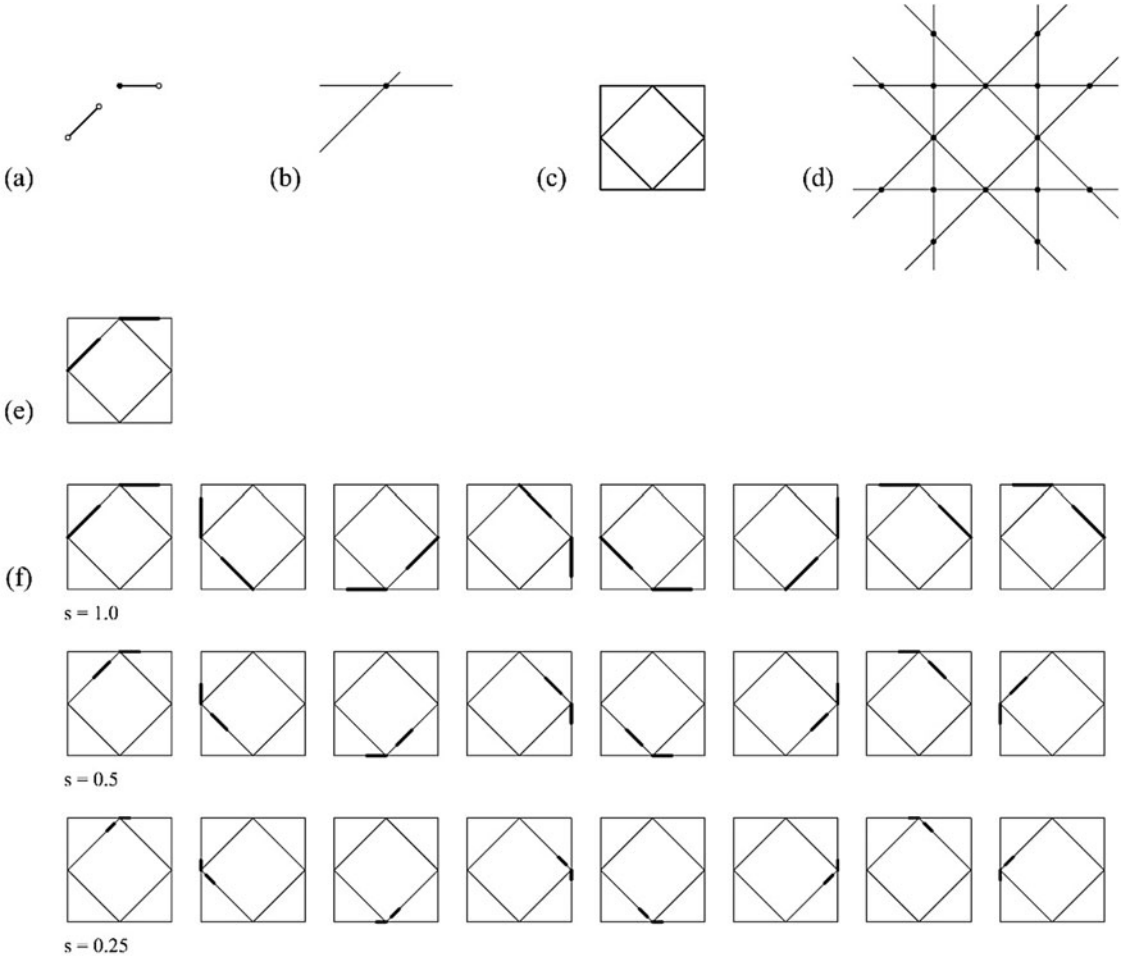


Fig. 8. An example of indeterminate similar embedding using one registration point. An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) the single embedding schema; (d) three instances of the single embedding schema with assignments s: 1, 0.5, and 0.25, one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

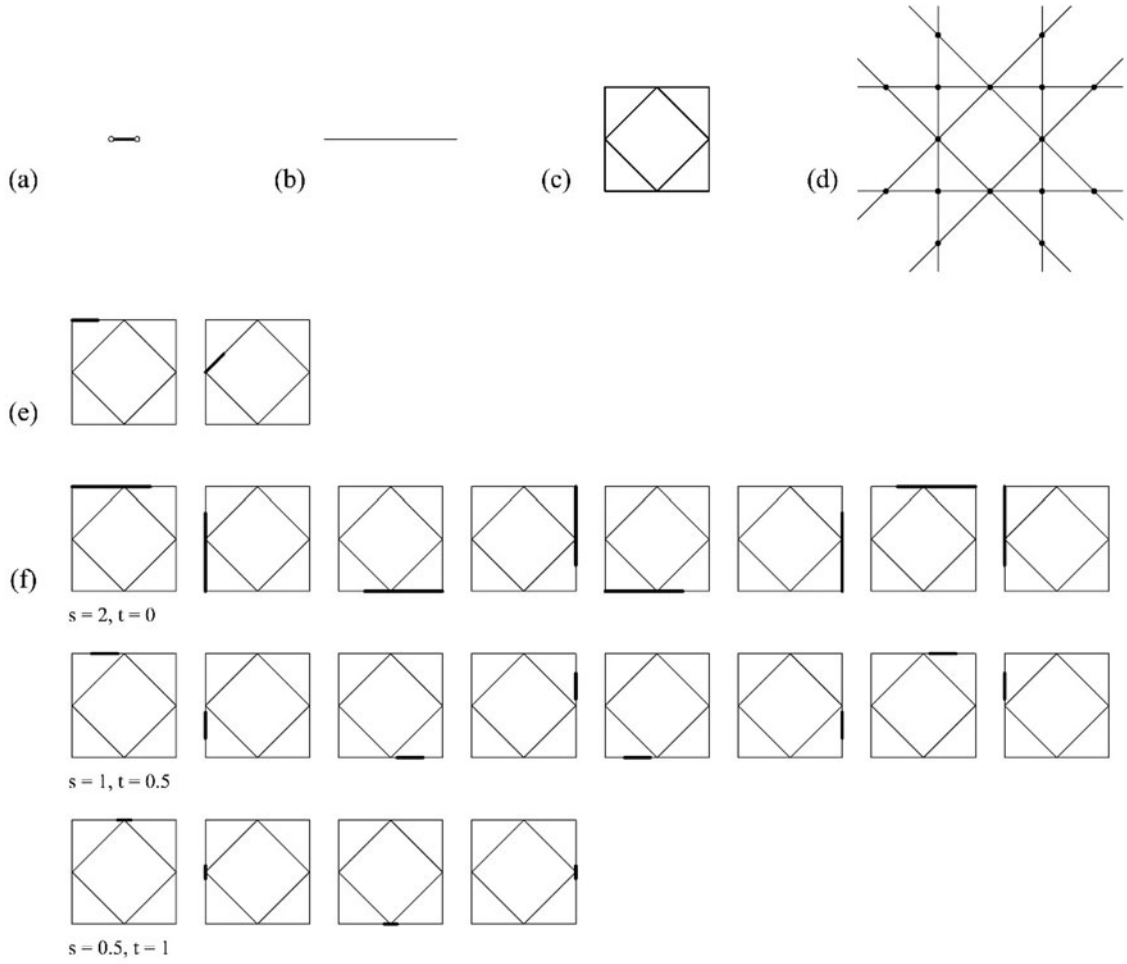


Fig. 9. An example of indeterminate similar embedding using no registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) two embedding schemas, one per set of lines of equal lengths; (f) three instances of one of the two embedding schemas with assignments s, t: (2, 0), (1, 0.5), and (0.5, 1), one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

given in Figure 9. In this example, two registration points are required to resolve the four unknown variables of the similarity matrix including the rotation angle θ, the translation in the x-direction t_x, the translation in the y-direction t_y, and the scaling factor s. The shape u does not have the required registration points for the derivation of the similarity matrix and the computation fails. To address this problem, a three-stepped process is used to determine the embedding results. The first step is processed automatically by taking two endpoints of $g(U)$ to derive θ, and confine the translation range in one-dimensional space, that is, any maximal line of W. The second step requires an additional input for the scaling parameter s as the indeterminacy occurs in both scaling and translation. The third step requires an input for the translation parameter t. In this case, the scaling parameter s has a range lower bound$_s \le s \le$ upper bound$_s$, here, the lower bound$_s = 0$ and upper bound$_s = l_i/a$, where l_i is the lengths of the sides of the two squares and a is the length of the line of $g(U)$. The translational parameter t has a range lower bound $\le t \le$ upper bound, here, the lower bond $= 0$ and upper bound $= (l_i - s \times a)/2$, where l_i the lengths of the sides of the two squares and a is the length of the line of $g(U)$. To simplify the notation, the translation parameter can be normalized as a parameter t' with a normalized range $0 \le t' \le 1$. In this example, there are two embedding schemas of the shape $f(g(U))$ in the shape W, eight matches for each shape under the symmetry group of the shape W, four matches when the midpoint of the shape $f(g(U))$ coincides with the midpoint of any of the edges of the shape W, and for each one of them, there is an assignment g parameterized with a translation factor t with a range $0 \le t' \le 1$.

Indeterminate embedding under affine transformations

An example of an indeterminate embedding under affine transformations of a shape u having two registration points is given in Figure 10. In this example, three registration points are required to resolve the six unknown variables of an affine matrix including a rotation angle θ, translation in the x-direction t_x, translation in the y-direction t_y, stretching in the x-direction α_x, stretching in the y-direction α_y, and a shearing angle φ. The shape u does not have the required registration points for the derivation of the affine matrix and the computation fails. To determine the embeddings, a two-stepped process is used to determine the embedding results. The first step is processed automatically by taking two registration points and two endpoints of $g(U)$ to derive θ, t_x, t_y, α_y, and φ. The second step requires an input for the stretching parameter in the x-direction α_x as the indeterminacy occurs in stretching in the

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

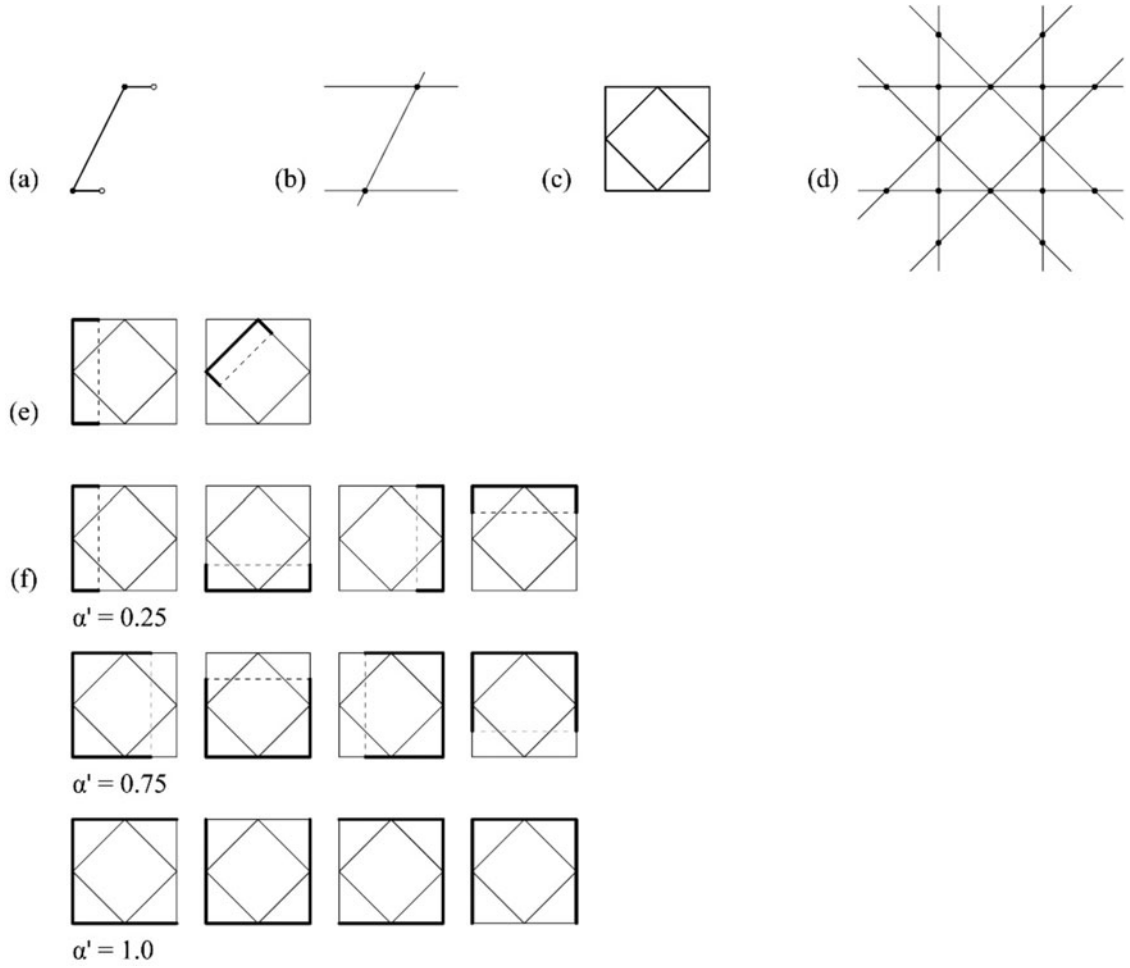


Fig. 10. An example of indeterminate affine embedding using no registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) two embedding schemas, one per two pairs of parallel lines; (f) three instances of one of the two embedding schemas with assignments α': 0.25, 0.75, and 1.0, one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

x-direction in this particular instance (see Fig. 10f, first column). Note that the stretching direction varies from instance to instance, and the system should automatically detect the stretching direction and provide a valid range of stretching. Furthermore, the stretching parameters α_x and α_y are one-dimensional scaling parameters, that is, the uniform scaling is presented when $\alpha_x = \alpha_y$. In this case, in total, there are $C_2^{16} = 120$ pairs of registration points, and among these 120 pairs, there are eight pairs in the configuration where two lines are in parallel and one line is intersecting with the other two, in which the shape u can be embedded under an affine transformation. These eight pairs of embeddings represent two families and each family is parameterized by a stretching parameter α_i with two ranges lower bond $\leq \alpha_i \leq$ upper bond, for α_i representing the stretching factor l_i/a of two squares, where l_i is the length of the side of the two squares and a is the length of the open-ended line of $g(U)$. One of the embedding schemas has a range $0 \leq \alpha_1 \leq 4$, and the other a range $0 \leq \alpha_2 \leq 2\sqrt{2}$. To simplify the notation of parameters, the parameter α_i can be normalized with a range $0 \leq \alpha_i' \leq 1$ for each family.

An example of an indeterminate embedding under affine transformations of a shape u having one registration point is given in Figure 11. In this example, three registration points are required to resolve the six unknown variables of an affine matrix including a rotation angle θ, translation in the x-direction t_x, translation in the y-direction t_y, stretching in the x-direction α_x, stretching in the y-direction α_y, and a shearing angle φ. The shape u does not have the required registration points for the derivation of the affine matrix and the computation fails. To determine the embeddings, a three-stepped process is used to determine the embedding results. The first step is processed automatically by taking one registration point and two endpoints of $g(U)$ to derive θ, t_x, t_y, and φ. Since the indeterminacy occurs in stretching both in the x-direction and y-direction, the second step requires an input for the stretching parameter in the x-direction α_x and the third step requires an input for the stretching parameter in the y-direction α_y for this particular instance (see Fig. 11f, first column). In this example, two ranges of stretching are provided: $0 \leq \alpha_x \leq 2$ in the x-direction and $0 \leq \alpha_y \leq 2$ in the y-direction. Among these 16 registration points in the shape W, there are four registration points where two maximal lines meet, four registration points where three maximal lines meet, and eight registration points where no maximal lines meet. Among those, the first two sets represent registration points in which the shape $f(g(U)$ can be embedded, and therefore, there are 24 pairs of embedding classified into four schemas. Each schema is

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

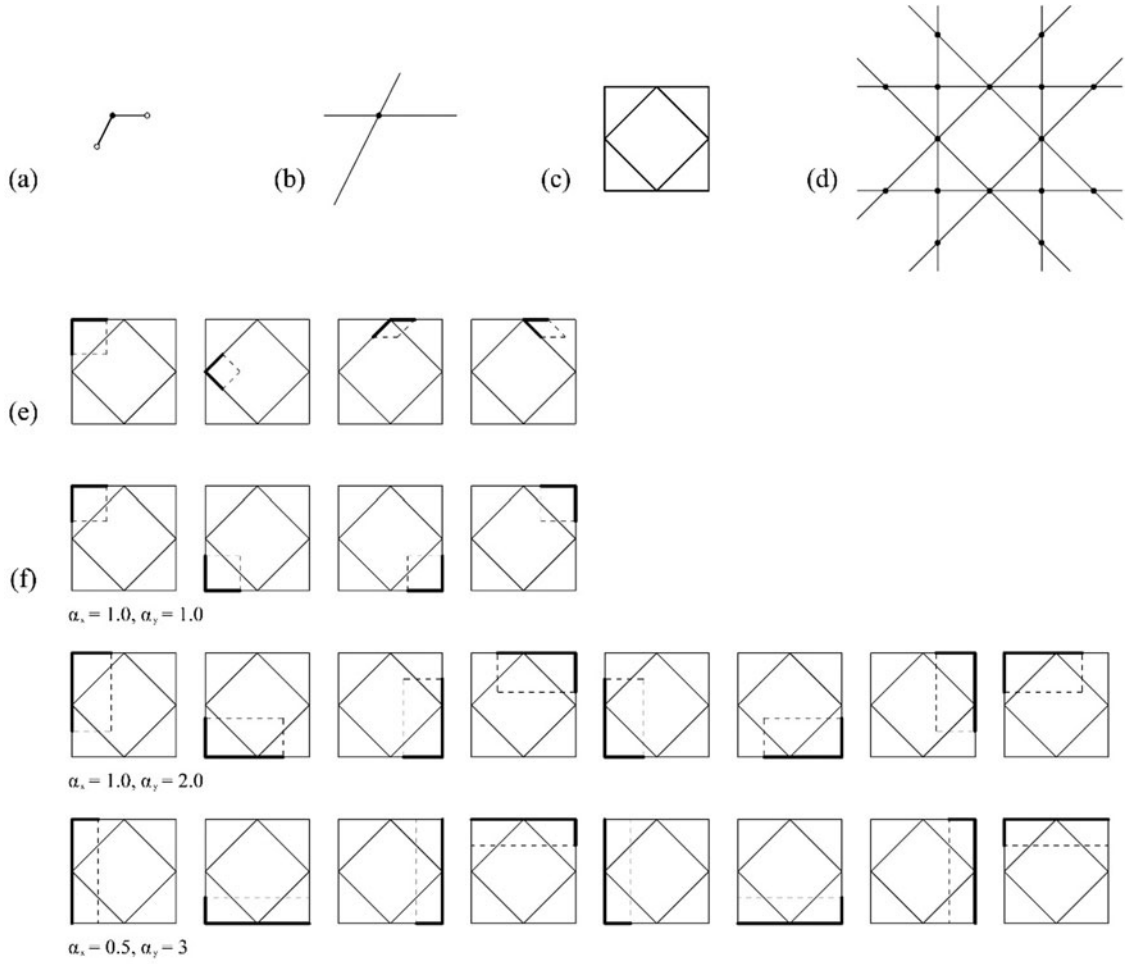


Fig. 11. An example of indeterminate affine embedding using one registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) four embedding schemas, one per two pairs of converging lines; (f) three instances of one of the four embedding schemas with assignments α_x, α_y: 1, 1, (1, 2), and (0.5, 3), one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

parameterized with two parameters α_x and α_y representing the stretching factors in the x-direction and y-direction, with ranges $0 \le \alpha_x \le l_i/a_x$ and $0 \le \alpha_y \le l_i/a_y$, and l_i representing the length of the sides for each square, a_x is the length of the horizontal line of $g(u)$, and a_y is the length of the other line of $g(u)$. To avoid double counting the results of embedding, a constraint of α_x and α_y is applied: $a_x \le a_y$.

An example of an indeterminate embedding under affine transformations of a shape u having no registration point is given in Figure 12. In this example, three registration points are required to resolve the six unknown variables of an affine matrix including a rotation angle θ, translation in the x-direction t_x, translation in the y-direction t_y, stretching in the x-direction α_x, stretching in the y-direction α_y, and a shearing angle φ. The shape u does not have the required registration points for the derivation of the affine matrix and the computation fails. To determine the embeddings, a four-stepped process is used to determine the embedding results. The first step is processed automatically by taking four endpoints of $g(U)$ to derive θ. Since the indeterminacy occurs in shear, stretch, and translation, the four endpoints can be used to confine the shear range, stretch range, and translation range to one-dimensional space in the y-direction for this particular instance (see Fig. 12f, first column). Here, the indeterminacy of the embedding occurs in three parameters so their ranges are correlated in a complex parametric function. More specifically, the shear transformation is parameterized with an angle parameter φ with a range $-\cos^{-1}(2\sqrt{5}/5) \le \varphi \le \cos^{-1}(4/5) - \cos^{-1}(2\sqrt{5}/5)$, which can be normalized as $0 \le \varphi' \le 1$; the stretch transformation, where l_i is the length of sides of the two squares and a is the length of the lines of $g(U)$, is parameterized with a range

$$0 \le \alpha \le \left(\frac{l_i}{a} \cdot \left(\frac{\tan\left(\left(\cos^{-1} \frac{\sqrt{5}}{5} \right) - \varphi \right) - 1}{\tan\left(\left(\cos^{-1} \frac{\sqrt{5}}{5} \right) - \varphi \right)} \right) - 1 \right),$$

which can be normalized as $0 \le \alpha' \le 1$; and the translation parameter has a range

$$0 \le t \le \frac{l_i}{2} \cdot \left(\frac{\tan\left(\left(\cos^{-1} \frac{\sqrt{5}}{5} \right) - \varphi \right) - 1}{\tan\left(\left(\cos^{-1} \frac{\sqrt{5}}{5} \right) - \varphi \right)} \right) - \alpha \cdot a,$$

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

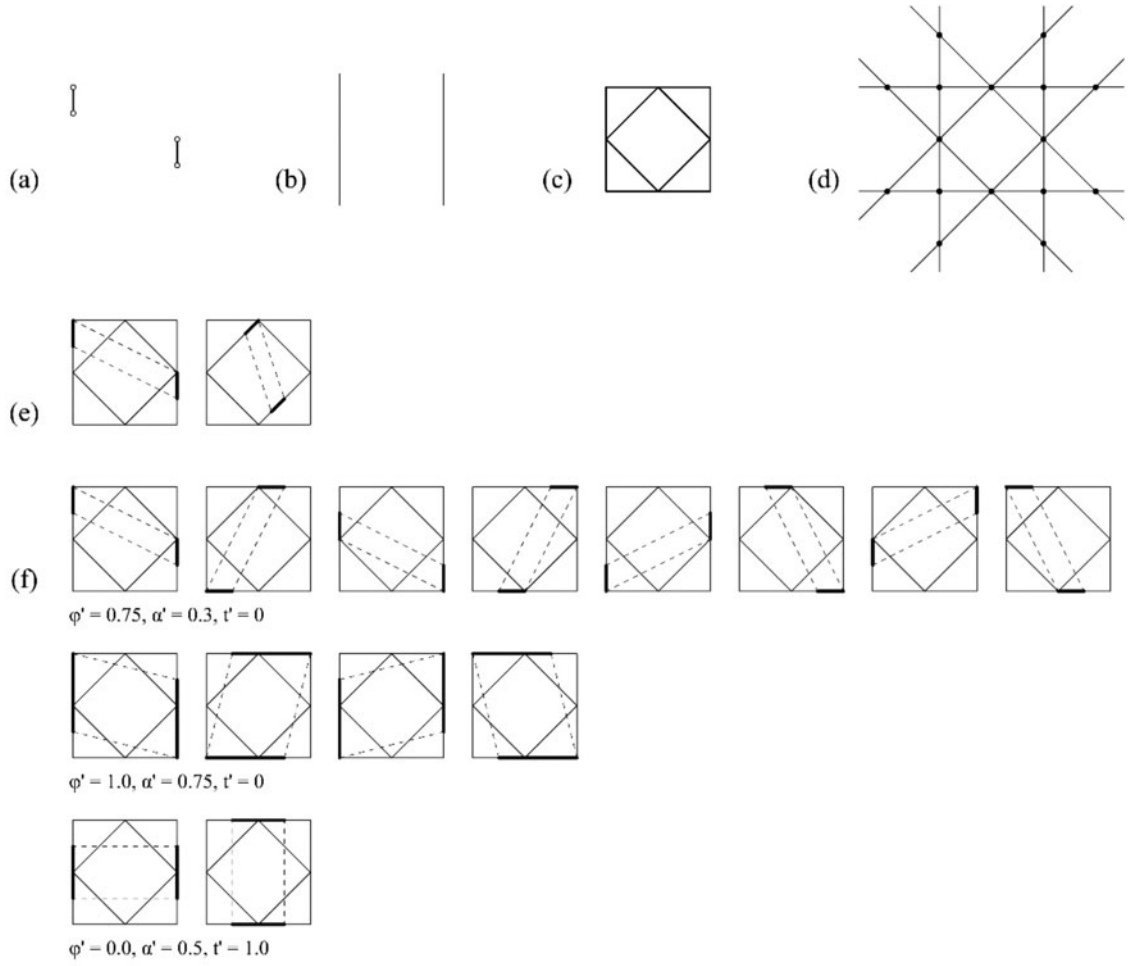


Fig. 12. An example of indeterminate affine embedding using no registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) two embedding schemas, one per two pairs of parallel lines; (f) three instances with assignments φ', α', t': (0.75, 0.3, 0), (1, 0.75, 0), and 0, 0.5, 1, one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

which can be normalized as $0 \leq t' \leq 1$. Note that the range of translation is constrained to avoid double counting. In this case, there are two embedding schemas and each schema is parameterized with three normalized parameters φ', α', and t'.

Indeterminate embedding under linear transformation

An example of an indeterminate embedding under linear transformations of a shape u having three registration points is shown in Figure 13. For linear transformation, four registration points are required to resolve the nine unknown variables of a homography matrix. The shape u does not have the required registration points for the derivation of the homography matrix and the computation fails. To determine the embeddings, a two-stepped process is used. The first step is processed automatically by taking three registration points which can be viewed as the process that the shape $g(U)$ is translated and rotated so that the three registration points are aligned with any three registration points of W. Once the three registration points are embedded into W, the second step requires an input for the fourth point to determine the final embeddings. In this case, the valid range of the additional point lies in a convex quadrilateral bounding box added on the shape $g(U)$. In this case, there are eight registration points of W, where two or more lines meet, in which the shape $g(U)$ can possibly be embedded. The single embedding schema is parameterized by the additional registration point represented with a pair of normalized parameters (p_x, p_y) with two normalized ranges $0 < p_x < \infty$ and $0 < p_y < \infty$. Note that p_x and p_y is constrained by a relation $p_x + p_y > l_i/2$ to keep the bounding box as a convex quadrilateral and constrained by a relation $p_x \geq p_y$ to avoid double counting.

An example of an indeterminate embedding under linearity transformations of a shape u having two registration points is shown in Figure 14. Similar to the previous case, four registration points are required to resolve the nine unknown variables of a homography matrix h_0 to h_8. The shape u does not have the required registration points for the derivation of the homography matrix and the computation fails. To determine the embeddings, a three-stepped process is used. The first step is processed automatically by taking two registration points of the shape $g(U)$, map them with any two registration points of W, and have the two short lines of $g(U)$ rotated to align with two maximal lines of W. Once the two registration points are embedded into W and the two short lines are embedded in the two maximal lines of W, the second step requires an input for the third point. In general, the third point can be any point on the world plane, however, the selection of the third point can be confined in a certain range, in this example, as the translation of the endpoint of the upper

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

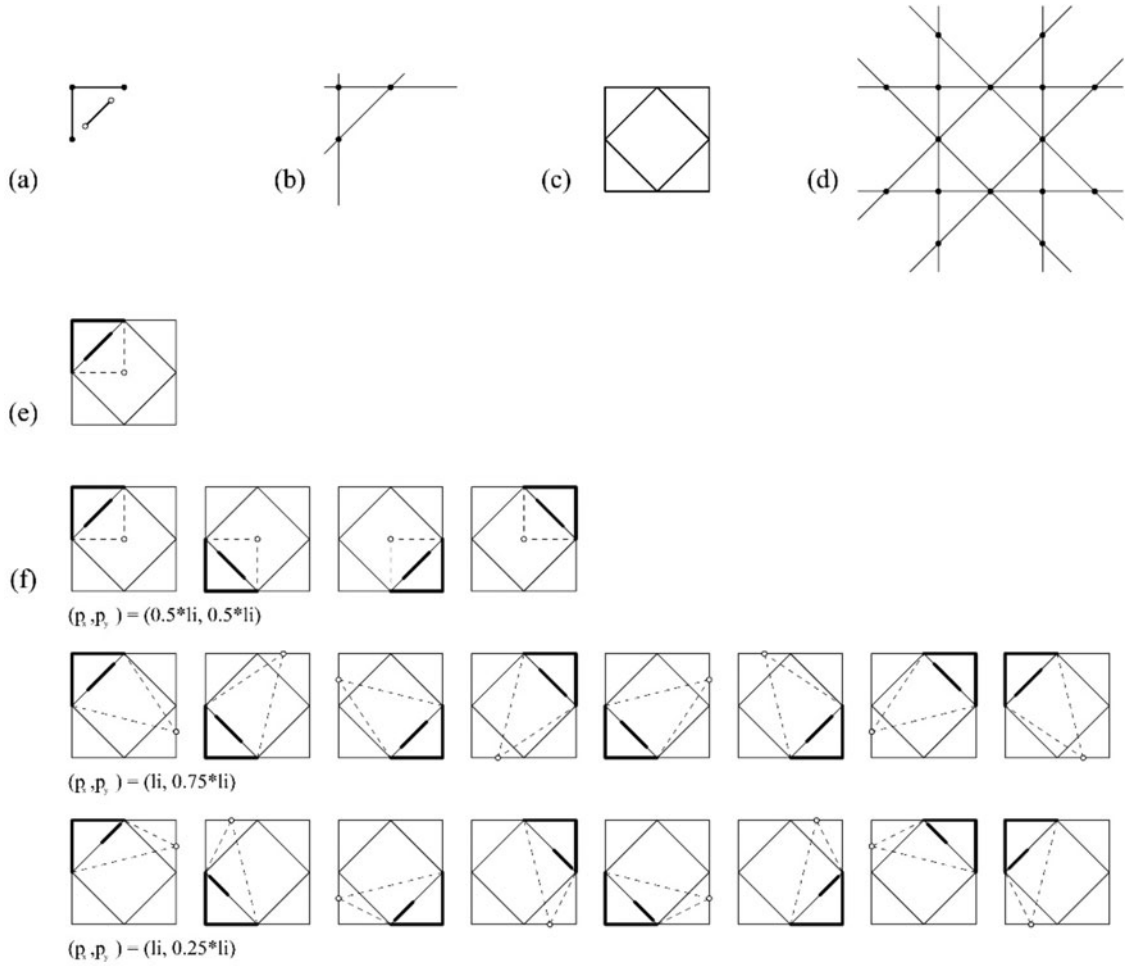


Fig. 13. An example of indeterminate projective embedding using three registration points. An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) one embedding schema; (f) three instances of the single embedding schema with assignments of p'_x, p'_y: $0.5l_i$, $0.5l_i$, (l_i, $0.75l_i$), and l_i, $0.25l_i$ one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

short line of the shape $g(U)$ within the maximal line of W. Similarly, the third step requires an input for the fourth point, and the endpoint of the lower short line of $g(U)$ as the fourth point. In total, there are $C_3^8 = 56$ triplets of lines in triangular configurations and the configurations where two lines are in parallel and the other line is nonparallel to another two. Among these 56 triplets, there are 40 triplets where $f(g(U))$ can be embedded for the bonding boxes of the triplets are convex quadrilaterals. There 40 triplets are classified into seven embedding schemas, and each schema is parameterized by two additional registration points represented as two normalized parameters p'_1 and p'_2 with the normalized ranges $0 < p'_1 < 1$ and $0 < p'_2 < 1$. Note that a constraint $p'_1 < p'_2$ is applied to avoid double counting. This case shows a strategy of inputting additional points to avoid infinite possibilities. Note that the indeterminacy theoretically occurs in one-point perspective transformation, but the indeterminacy can be mapped to the translation of the two endpoints because the bounding box of $g(U)$ is a parallelogram which is defined by the two registration points and two endpoints. Hence, the perspective transformation can be viewed as the transformation from a parallelogram to a trapezoid and the two endpoints must occur on the two maximal lines of W. In the cases later in this section, the selection of the additional points follows this strategy demonstrated in this example.

An example of an indeterminate embedding under linearity transformations of a shape u having one registration point is shown in Figure 15. Similar to the previous cases, four registration points are required to resolve the nine unknown variables of a homography matrix h_0 to h_8. The shape u does not have the required registration points for the derivation of the homography matrix and the computation fails. To determine the embeddings, a four-stepped process is used. The first step is processed automatically by taking one registration point and two endpoints of $g(U)$, and this step can be viewed as the process that the shape $g(U)$ is translated and rotated so that the registration point is aligned with a registration point of W, and the two lines of $g(U)$ are rotated to align with the two maximal lines of W. Once the registration point is embedded into W and the two lines are embedded in the two maximal lines of W, the second step requires an input for the second point. In this case, the second point is suggested by the system as the endpoint of the longer line because this endpoint must be embedded in the maximal line of W (in this particular instance, the upper line of the large square of W) and can only translate along the maximal line if $f(g(U)) \leq W$ holds true. Thusly, the required input for this step is the position of this endpoint along the maximal line in which this endpoint is embedded. Similarly to the previous step, the third step requires an input for the third point which

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

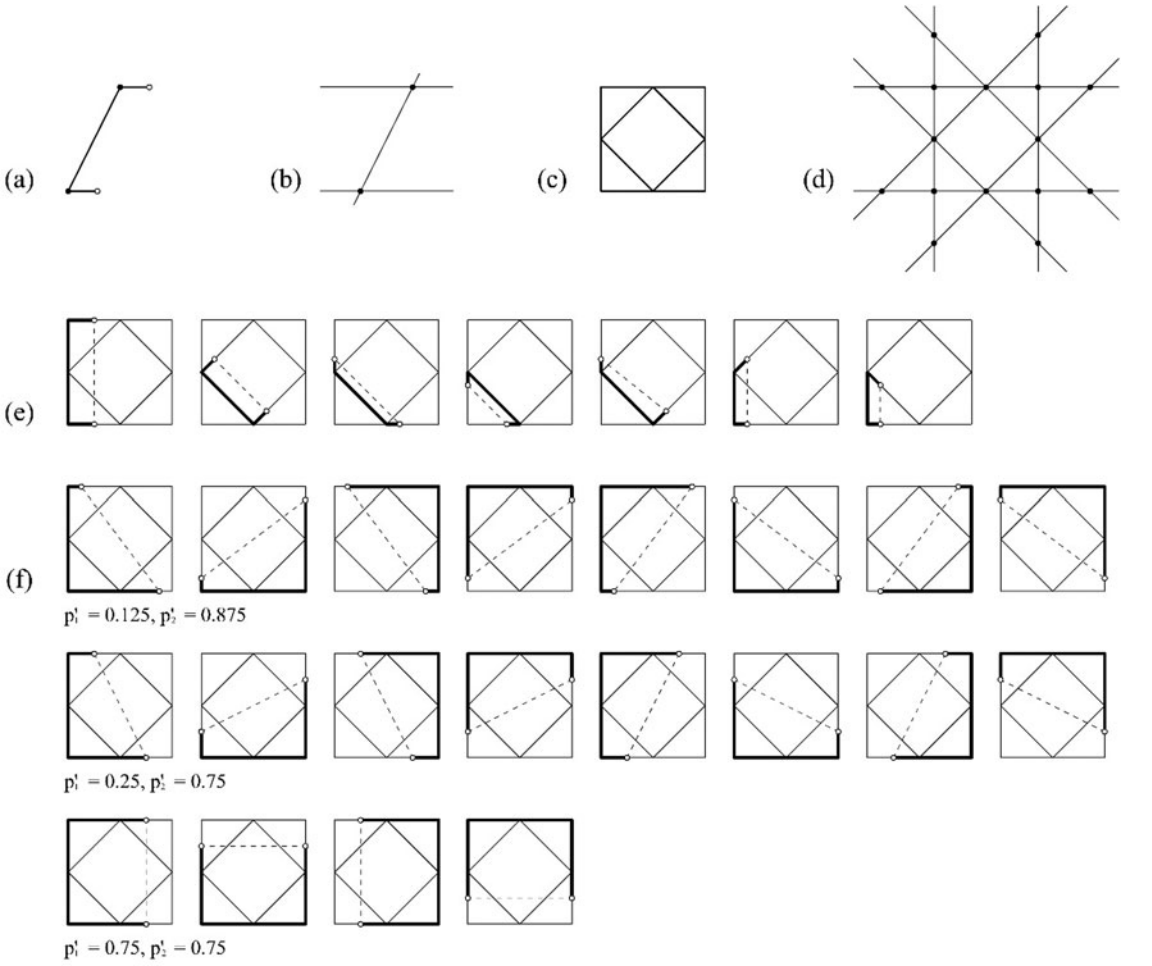


Fig. 14. An example of indeterminate projective embedding using two registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) seven embedding schemas, one per a U-shape configuration of lines; (f) three instances of one of the seven embedding schemas with assignments p'_1, p'_2: (0.125, 0.875), (0.25, 0.75), and 0.75, 0.75, one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

is suggested by the system as the farther endpoint (from the registration point) of the shorter line for the same reason. Thusly, the required input for this step is the position of this endpoint along the maximal line (in this particular instance, the left line of the large square of W) in which this endpoint is embedded. The fourth step requires an input for the fourth point, in this case, any point on the world plane with the constraint, that is, its selection should keep the bonding box of the shape $g(U)$ as a convex quadrilateral. The valid ranges of the three registration points are all open-ended and can be calculated by manual visual inspection or some automated optimization. Among the 16 registration points in W, there are four registration points where two maximal lines meet and four registration points where three maximal lines meet and therefore, there are six embedding schemas for the shape $g(U)$. Each schema is parameterized with three additional registration points represented as two normalized parameters p'_1 and p'_2 with normalized ranges $0 < p'_1 < 1$ and $0 < p'_2 < 1$, and one pair of normalized parameters p'_{3x} and p'_{3y} with a normalized range $0 < p'_{3x} < \infty$ and $0 < p'_{3y} < \infty$. Note that the parameters p'_{3x} and p'_{3y} are constrained by a relation $p'_{3x} + (p'_2/p'_1)p'_{3y} - p'_2 > 0$.

An example of an indeterminate embedding under linear transformations of a shape u having no registration points is shown in Figure 16. Similarly to the previous cases, four registration points are required to resolve the nine unknown variables of a homography matrix h_0 to h_8. The shape u does not have the required registration points for the derivation of the homography matrix and the computation fails. To determine the embeddings, a five-stepped process is used. The first step is processed automatically by taking all the four endpoints of $g(U)$, and this step can be viewed as the process that the two lines of $g(U)$ are translated and rotated so that the two lines are embedded in any two maximal lines of W. Once the two lines are embedded into W, the second step requires an input for the first point. Similar to the previous cases the selection of the first point is the left endpoint of the upper line of $g(U)$ and it can be translated within the maximal line of W on which it occurs. The third step requires an input for the second point, which is the right endpoint of the upper line of $g(U)$ and it can be translated within the maximal line of W on which it occurs. The fourth step requires an input for the third point, which is the left endpoint of the lower line of $g(U)$ and it can be translated within the maximal line of W on which it occurs. The last step requires an input for the fourth point which is the right endpoint of the lower line of $g(U)$ and it can be translated within the maximal line of W on which it occurs. Theoretically, any four points on the world plane can be used

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

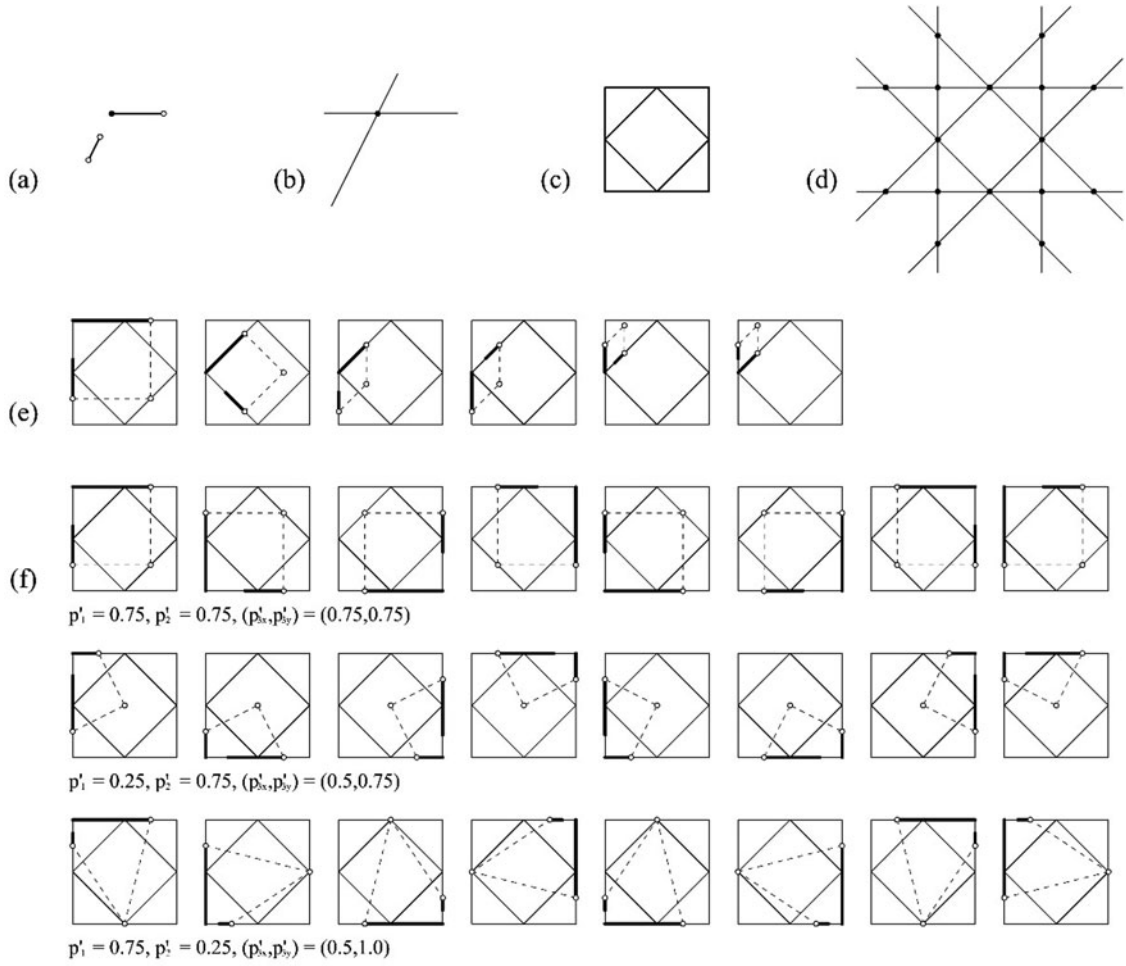


Fig. 15. An example of indeterminate projective embedding using one registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) six embedding schemas one per pair of converging lines; (f) three instances of one of the six embedding schemas with assignments p'_1, p'_2, p'_{3x}, p'_{3y}: (0.75, 0.75, 0.75, 0.75), (0.25, 0.75, 0.5, 0.75), and (0.75, 0.25, 0.5, 1.0), one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

to calculate the linear transformation, nevertheless, the selection of endpoint helps system to reduce the sampling space. Note that the four points form the bonding box of $g(U)$ so that the input coordinates of the four points should keep the bonding box a convex quadrilateral. In total, there are $C_2^8 = 28$ parallel and convergent pairs of lines in the shape W, in which the shape $f(g((U))$ can be embedded and seven embedding schemas, each parameterized with four registration points p'_1, p'_2, p'_3, and p'_4 with normalized ranges $0 < p'_1 < p'_2$, $p'_1 < p'_2 < 1$, $0 < p'_3 < p'_4$, and $p'_3 < p'_4 < 1$. To avoid double counting, a constraint $p'_2 - p'_1 \leq p'_4 - p'_3$ is applied. Note that the indeterminacy is also involved with translation when the four registration points are all determined. The range of t is

$$0 \leq t \leq \frac{l_i(1 - \max(p'_2, p'_4))}{2},$$

which can be normalized as $0 \leq t' \leq 1$.

Discussion

Shape embedding continues to be one of the most difficult tasks for a shape grammar interpreter and a *tabula incognita* for CAD systems. The title of the paper playfully underscores this state by using a pan on George Stiny's "What Designers Do That Computers Should" a paper written in the late 1980s foregrounding the need for computation to support ambiguity and variability in design (Stiny, 1990). In the years that have passed since, work on variability has produced an impressive array of tools and processes all in support of design, parametric representations, optimizations, building information modeling, and so forth (see, e.g., Eastman *et al.*, 2018; Woodbury, 2010; Woodbury *et al.*, 2017). Work on ambiguity (embedding) not so much. The opening lines in this work reaffirmed just that shape embedding under a given Euclidean, affine, or linear transformation is entirely absent from current CAD systems. Be it as it may, the problem is tractable and it is strongly suggested that the work here provides a firm step toward the solution.

The review of the implementation of the derivation of the transformation matrices algorithm for determinate and indeterminate embeddings revealed a complex problem in shape embedding that does not lend itself easily to a unified solution. A series of the specific challenges that the algorithm faces for the successful derivation of particular transformation matrices includes the calculation of the numbers of registration points incident in each hyperplane per shape, for each shape u and W separately, and one against the other.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

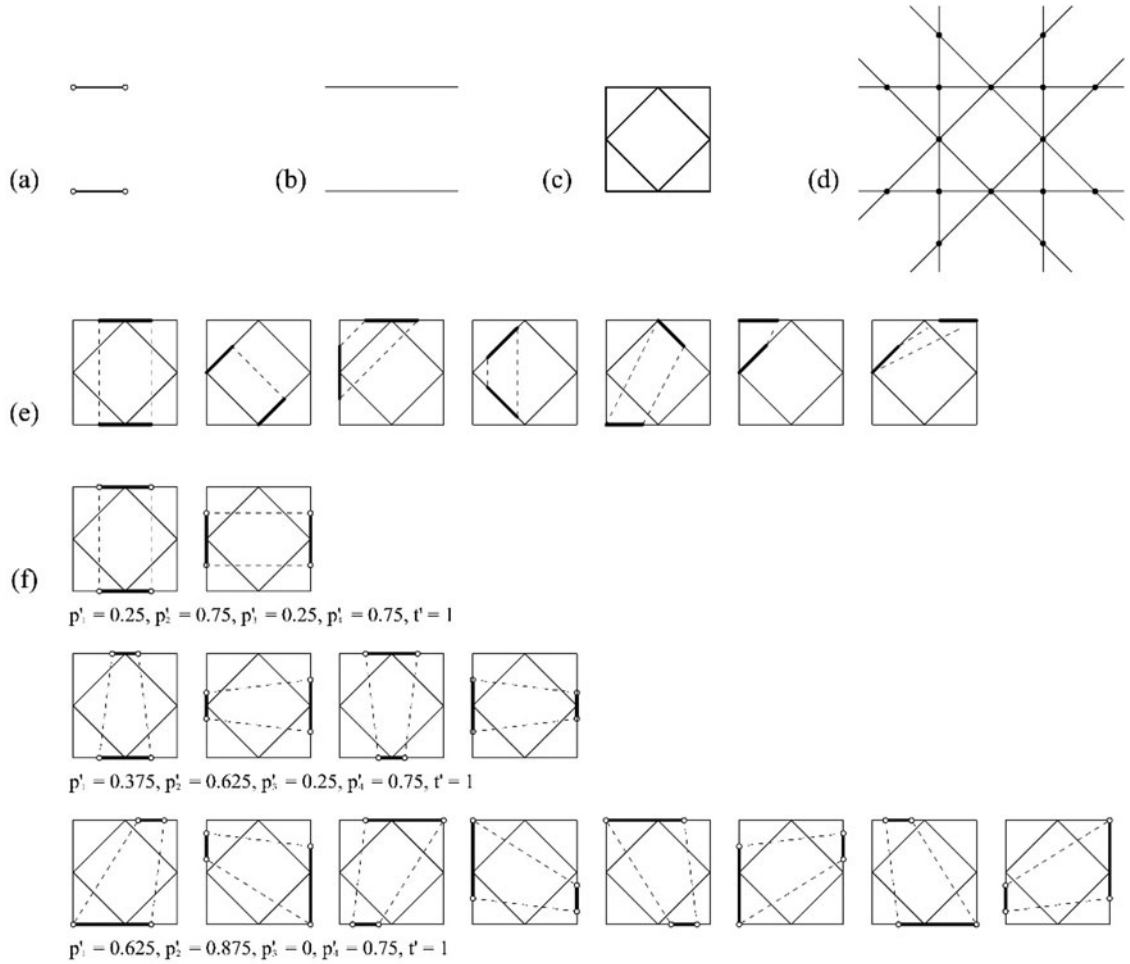


Fig. 16. An example of indeterminate projective embedding using no registration points. (a) An instance of a parameterized shape $g(U)$; (b) registration points $R_{g(U)}$; (c) shape W; (d) registration points R_W; (e) seven embedding schemas one per pair of parallel or converging lines; (f) three instances of one of the seven embedding schemas with assignments $p'_1, p'_2, p'_3, p'_4, t'$: 0.25, 0.75, 0.25, 0.75, 1 (0.375, 0.625, 0.25, 0.75, 1), and (0.625, 0.875, 0, 0.75, 1), one per row, and their equivalent embeddings by the actions of the symmetry group of the shape W.

Clearly, the greatest challenge in the automated implementation of the derivation of the transformation matrix is the appropriate selection of the required registration points to resolve the variables. The automated sampling of missing registration points to include the endpoints of maximal lines of the two shapes (Stiny, 1975; Krishnamurti, 1981) produces several failed computations. An example of a failed sampling of endpoints to derive the isometric matrix for an isometry embedding of a shape u with one registration point is given in Figure 17. The algorithm requires two points to resolve all three unknown variables and the endpoint sampling leads to an incorrect derivation of the isometry matrix.

A second challenge in the automated implementation of the derivation of the affine and homography matrices is the appropriate exclusion of the all triples of collinear registration points to resolve the variables. An example of a problematic automated selection of collinear triples of registration points from the shape W has already been given earlier in this work. An example of a failed sampling of registration points from the shape u to derive the affine matrix is briefly discussed here and illustrated in Figure 18. In this case, the shape u has seven registration points and five hyperplanes, and there are $C_3^7 = 35$ ways to sample three registration points from it, however, five of them are collinear and provide failed calculations when they are fed in the derivation algorithm, that is, there is a chance of $5/35 \approx 14.28\%$ to have an invalid triplet if the system blindly processes the sampling. Hence, the collinear triples should be filtered out before processing the embedding. It has already been discussed above that there are 504 applicable triples of registration points in the shape W to be compared to derive the affine matrix with an extra computational load filtering out the collinear triples.

A third challenge in the automated implementation of the derivation of the transformation matrices appears in special cases of geometrical constraints of the shape W that condition whether an embedding will be determinate or indeterminate, irrelevant of the number of required registration points in the shape u. This problem is significant to the extent that current implementations of the derivation of the transformation matrices miss determinate embeddings that an experienced designer can see. A case of determinate embedding under a similarity transformation where the shape u has no registration point is shown in Figure 19. Typically, the embedding of a shape u under similarity and no registration point is indeterminate, and yet, in this particular example, one of the four pairs of the endpoints of the two maximal lines of the shape u is incident with eight pairs of registration points in the shape W under similarity, and thusly, the shape u can be embedded in W in eight determinate ways.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

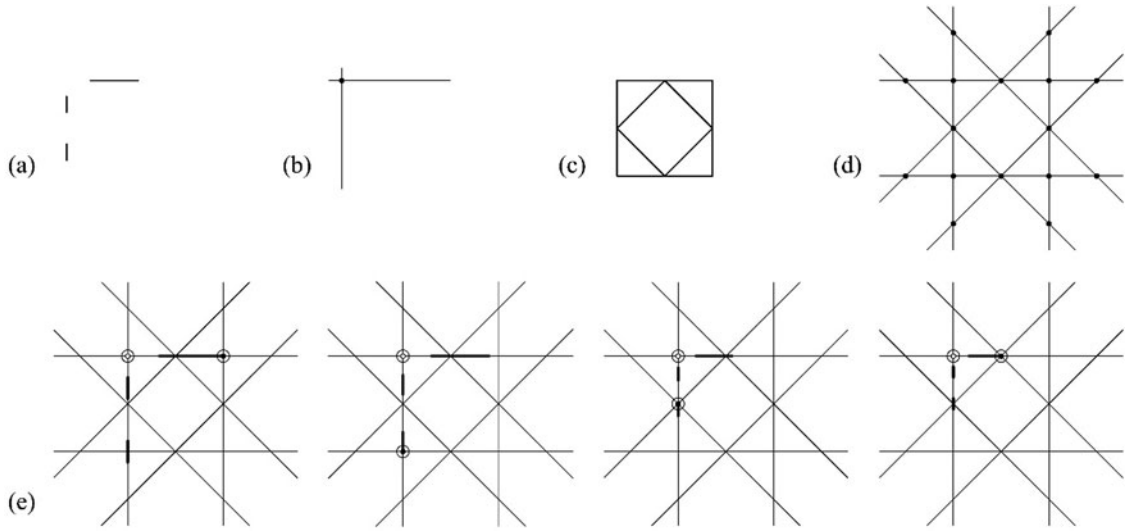


Fig. 17. Automated point sampling for the derivation of an isometric matrix. (a) failed and (b–-(d) correct.

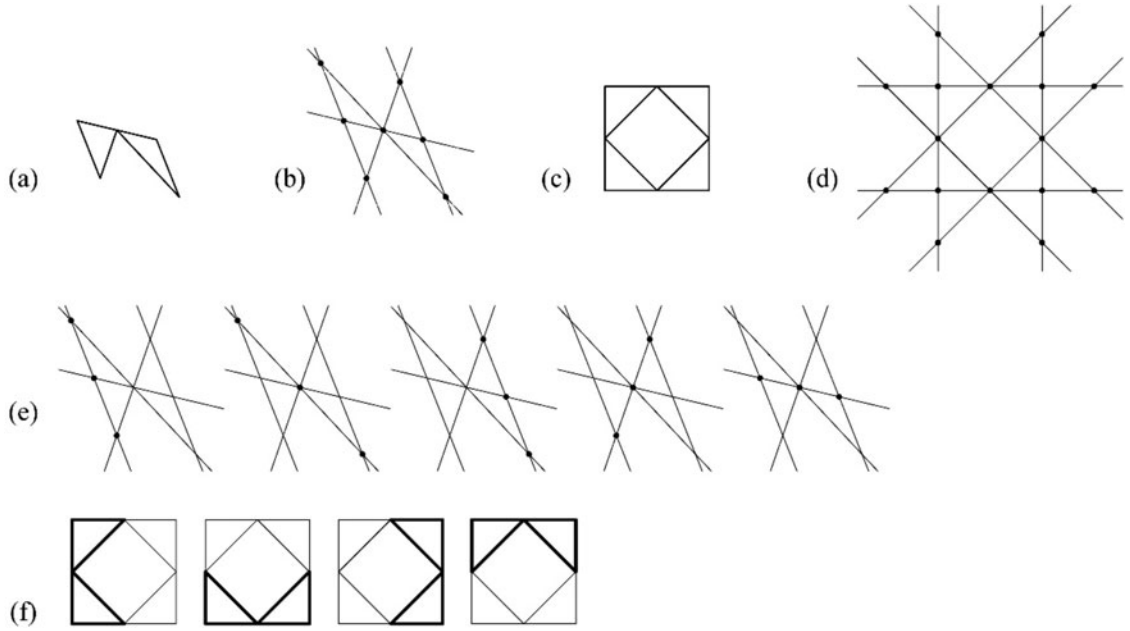


Fig. 18. An example of determinate embedding under affine transformation that cannot be completely determined by the affine matrix. (a) Shape u and the registration points R_u; (b) shape W and the registration points R_W; (c) five failed intersection samplings from the shape u; and (e) four results of determinate embedding under affinity transformation.

A fourth challenge in the automated implementation of the derivation of the transformation matrices is the seamless integration of determinate and indeterminate embeddings in one unified modeling workflow. In determinate cases of embedding, the registration points are appropriately selected from a given set of points; in indeterminate cases, the registration points have to be defined in some other alternative way. It is suggested here that transformation f that maps a shape onto another shape can be decomposed into a finite sequence of applicable transformations, that is, $f_1 f_2 \ldots f_n$. Among these applicable transformations, some of them can be first derived with the given registration points. For the rest of the transformations, they are indeterminate. As the following step, the ranges of the indeterminate transformations are calculated (as shown in previous sections) one by one, and the requests are sent to users to input the values of the transformation factors step by step to determine the transformations so that the embedding can be processed.

Lastly, and certainly not least, a major challenge in the automated implementation of the derivation of the transformation matrices is related with aspects of performance in shape recognition and with the calculations involved in the various procedures of embedding outlined in this work: The complexity of embedding follows the number of the registration points available in the current design W. The method of the 3×3 matrix allows the system to use two registration points to achieve a Euclidean transformation, thus, the complexity of the embedding under

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

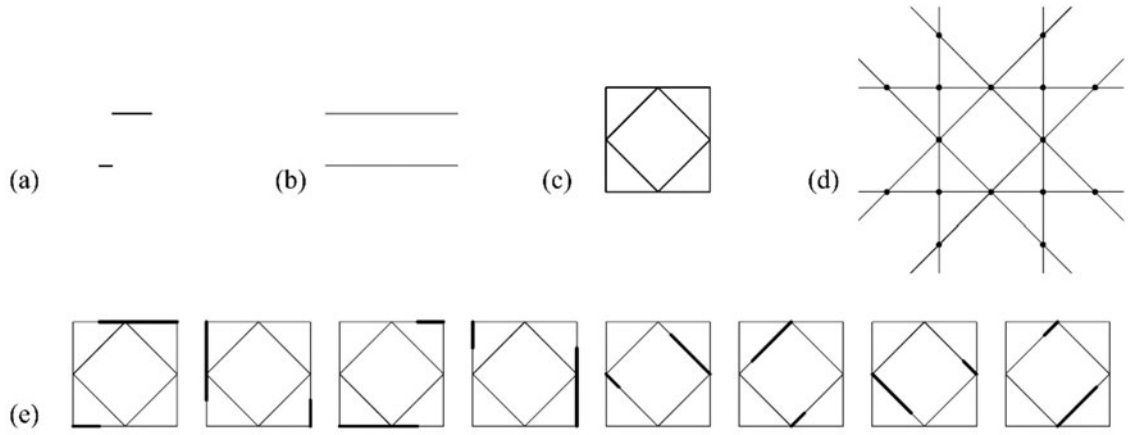
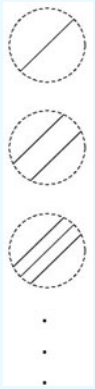
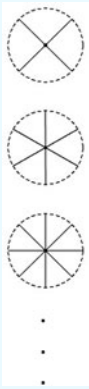
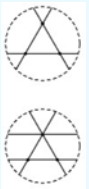
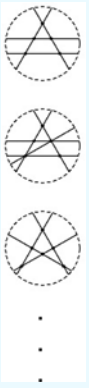


Fig. 19. An example of determinate similar embedding using no registration points. (a) Shape u and the registration points R_u; (b) shape W and the registration points R_W; and (c) eight results of determinate embedding under similarity transformations.

Table 1. 14 cases of shape embedding

	Number of Registration Points of ***u***					
Transformation	0	1	2	3	4	>4
Isometry	Indeterminate	Determinate				
Similarity	Indeterminate	Indeterminate	Determinate			
Affinity	Indeterminate	Indeterminate	Indeterminate	Determinate		
Linearity	Indeterminate	Indeterminate	Indeterminate	Indeterminate	Determinate	
Hyperplane Configurations						

Euclidean transformations is $O(n^2)$ for n the number of the registration points of W. For affine transformations, the complexity increases to $O(n^3)$, and for linear transformations, the complexity increases to $O(n^4)$. For more on algorithmic complexity and the Big-O notation, see, for example, Cormen *et al.* (2009).

The results of all possible 14 cases of shape embedding, including four determinate ones and ten indeterminate ones, are tabulated in Table 1 according to the number of intersection of hyperplanes and illustrated by the corresponding hyperplane arrangements upon which maximal lines are embedded: solid lines denote the hyperplanes and a dashed circle denotes the Euclidean plane that the hyperplanes are embedded in. There is an infinite number of hyperplane arrangements that have zero or one intersection point, namely, those consisting of parallel hyperplanes or hyperplanes that pass through a point, respectively. There is only one hyperplane arrangement with two registration points produced by three hyperplanes; two hyperplane arrangements with three registration points produced by the intersection of three and four hyperplanes, respectively; two hyperplane arrangements of four registration points produced by the intersection of three and four hyperplanes, respectively; and an infinite number of hyperplane arrangements with more than four registration points produced by the intersection of four or more hyperplanes (Economou et al., 2019).

The goal of this study is to rework the problem of shape embedding in the algebra U_{12} and explore a general solution for a shape grammar interpreter implementation for any number of registration points and for all linear transformations, including isometries, similarities, affinities, and projectivities. An initial effort to put some of the 14 calculations outlined above in practice is evidenced in the making of the Shape Machine, a new shape grammar interpreter designed from scratch (Hong and

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

Economou, 2019). An initial set of shape calculations in Shape Machine pertaining to shapes consisting of lines, arcs and their combinations in the plane, has successfully met the requirements for all determinate cases as well as for two cases of indeterminate embedding under isometric and similarity transformations of shapes having one registration point (Economou *et al.*, 2021). Interestingly, the structured sequence of the steps required to resolve the indeterminate cases reported in the work, and the guided choices made by the designer through the process – suggest a different way to tackle ambiguity in shape embedding. This process requires additional information typically given in the form of written instructions or numerical values to generate the candidate shape to be embedded. These decisions along the steps generate different instances of shapes, different embeddings into the design, and ultimately a different understanding of the design in terms of the shapes that become parts of it. It is hoped that the extension of these findings for all indeterminate cases for shapes made up of lines in the plane will make a coherent and unified framework for queries based on shape embedding, and a visual tool to provide insight in highly unstructured visual searches and design actions (shape replacements).

Clearly, there is lots of ground that needs to be covered including the expansion of the range of geometry descriptors for various types of shapes in different dimensions, implementation of corresponding maximal representations of shapes embedded in these descriptors, Boolean operations for these kinds of geometries, resolution of indeterminant visual embeddings for all such types, and of course, extensions to parametric definitions and rule schemata. Additional directions pertaining to the design of the interfaces of these systems and their seamless integration with current modes of practice provide indeed a bright future for their development.

References

Alexanderson GL and Wetzel JE (1978) Simple partitions of space. *Mathematics Magazine* **51**, 220–225.

Armstrong MA (1997) *Groups and Symmetry*. New York; London: Springer.

Besl PJ and Jain RC (1985) Three-dimensional object recognition. *Computing Surveys* **17**, 75–145.

Campbell RJ and Flynn PJ (2001) A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding* **81**, 166–210.

Cardone A, Gupta SK and Karnik M (2003) A survey of shape similarity assessment algorithms for product design and manufacturing applications. *Journal of Computing and Information Science in Engineering* **3**, 109–118.

Chase SC (1989) Shapes and shape grammars: from mathematical model to computer implementation. *Environment and Planning B: Planning and Design* **16**, 215–242.

Cormen TH, Leiserson CE, Rivest R and Stein C (2009) *Introduction to Algorithms*, 3rd Edn. Cambridge, MA: MIT Press.

Earl C (1997) Shape boundaries. *Environment and Planning B: Planning and Design* **24**, 669–687.

Eastman C, Teicolz P, Sacks R and Liston K (2018) *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Hoboken, NJ: Wiley.

Economou A, Yu J and Park J (2019) Shape Signature. Available at https://shape.design.gatech.edu/Research/Projects/2019_ShapeSignature/index.html (accessed January 6, 2021).

Economou A, Hong TK, Ligler H and Park J (2021) Shape machine: a primer in visual composition. In Lee J-H (ed.), *A New Perspective of Cultural DNA*. KAIST Research Series. Singapore: Springer Nature, pp. 65–92.

Foley J, Van Dam A, Feiner S and Hughes J (1997) *Computer Graphics: Principles and Practice*. Reading, MA: Addison-Wesley.

Gips J (1975) *Shape Grammars and Their Uses: Artificial Perception, Shape Generation and Computer Aesthetics*. Basel: Birkhäuser.

Goodman JE and O'Rourke J (1997) *Handbook of Discrete and Computational Geometry*. Boca Raton, FL: CRC Press.

Grasl T and Economou A (2013) From topologies to shapes: parametric shape grammars implemented by graphs. *Environment and Planning B: Planning and Design* **40**, 905–922.

Grasl T and Economou A (2018) From shapes to topologies and back: an introduction to a general parametric shape grammar interpreter. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **32**, 208–224.

Hartley R and Zisserman A (2004) *Multiple View Geometry in Computer Vision*. Cambridge: Cambridge University Press.

Hong TK and Economou A (2019) Shape Machine. Available at https://shape.design.gatech.edu/Research/Projects/2019_ShapeMachine/index.html (accessed January 6, 2021).

Hong TK and Economou A (2021) Five criteria for shape grammar interpreters. In Gero JS (ed.), *Design Computing and Cognition*. Cham: Springer, pp. 189–208.

Jowers I and Earl C (2011) Implementation of curved shape grammars. *Environment and Planning B: Planning and Design* **38**, 616–635.

Jowers I, Hogg DC, McKay A, Chau HH and Pennington A (2010) Shape detection with vision: implementing shape grammars in conceptual design. *Research in Engineering Design* **21**, 235–247.

Knight T (2015) Shapes and other things. *Nexus Network Journal* **17**, 963–980. doi:10.1007/s00004-015-0267-3

Knight T and Stiny G (2001) Classical and non-classical computation. *Architectural Research Quarterly* **5**, 355–372.

Krishnamurti R (1981) The construction of shapes. *Environment and Planning: Planning and Design B* **8**, 5–40.

Krishnamurti R (1982) *SGI: a shape grammar interpreter*. Milton Keys, UK : The Open University

Krishnamurti R (1992) The arithmetic of maximal planes. *Environment and Planning B: Planning and Design* **19**, 431–464.

Krishnamurti R and Earl C (1992) Shape recognition in three dimensions. *Environment and Planning B: Planning and Design* **19**, 585–603.

Krishnamurti R and Giraud C (1986) Towards a shape editor: the implementation of a shape generation system. *Environment and Planning B: Planning and Design* **13**, 391–404.

Krishnamurti R and Stouffs R (1997) Spatial change: continuity, reversibility, and emergent shapes. *Environment and Planning B: Planning and Design* **24**, 359–384.

Krstic D (2014) Algebras of shapes revisited. In Gero JS (ed.), *Design Computing and Cognition '12*. Dordrecht: Springer.

Loncaric S (1998) A survey of shape analysis techniques. *Pattern Recognition* **31**, 983–1001.

March L and Steadman P (1974) *The Geometry of Environment: An Introduction to Spatial Organization in Design*. Cambridge, MA: MIT Press.

McCullough M (2006) 20 years of scripted space. *Architectural Design* **76**, 12–15.

McKay A, Chase S, Shea K and Chau HH (2012) Spatial grammar implementation: from theory to usable software. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **26**, 143–159.

Mitchell W (2001) Vitruvius redux. In Antonsson E and Kagan J (eds), *Formal Engineering Design Synthesis*. Cambridge: Cambridge University Press, pp. 1–19.

Preparata FP and Shamos MI (1990) *Computational Geometry: An Introduction*. New York: Springer-Verlag.

Ruiz-Montiel M, Belmonte MV, Boned J, Mandow L, Millán E, Badillo AR and Pérez JL (2014) Layered shape grammars. *Computer-Aided Design* **56**, 114–119.

Stiny G (1975) Pictorial and formal aspects of shape and shape grammars. In *Interdisciplinary Systems Research*. Basel: Birkhäuser.

Stiny G (1980) Introduction to shape and shape grammars. *Environment and Planning B: Planning and Design* **7**, 343–351.

Stiny G (1989) Formal devices in design. In Newsome SA Spillers WR and Finger S (eds), *Design Theory 8*. New York: Springer-Verlag, pp. 173–188.

Stiny G (1990) What designers do that computers should. In McCullough M Mitchell WJ and Purcell P (eds), *The Electronic Design Studio: Architectural Knowledge and Media in the Computer Era*. Cambridge, MA: MIT Press, pp. 17–30.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press

Stiny G (1991) The algebras of design. *Research in Engineering Design* **2**, 171–181.
Stiny G (1996) Useless rules. *Environment and Planning B: Planning and Design* **23**, 235–237.
Stiny G (2006) *Shape: Talking about Seeing and Doing*. Cambridge, MA: MIT Press.
Stouffs R (1994) *The Algebra of Shapes* (Ph.D. thesis). Computational Design, School of Architecture, Carnegie Mellon University.
Stouffs R (2018) Where associative and rule-based approaches meet. In Fukuda T, Huang W, Janssen P, Crolla K and Alhadidi S (eds), *Proceedings of CAADRIA 2018*, V2, pp. 453–462.
Stouffs R (2019) Shape rule types and spatial search. *CAAD Futures, CCIS* **1028**, 474–488.
Stouffs R and Krishnamurti R (2019) A uniform characterization of augmented shapes. *Computer-Aided Design* **110**, 37–49.
Stouffs R and Li A (2020) Learning from users and their interaction with a dual-interface shape-grammar implementation. *Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia, CAADRIA 2020*, **2**, 153–162.
Tangelder H and Veltkamp RC (2008) A survey of content based 3D shape retrieval methods. *Multimedia Tools Applications* **39**, 441–471.
Tapia M (1999) A visual implementation of a shape grammar system. *Environment and Planning B: Planning and Design* **26**, 59–73.
Trescak T, Rodriguez I and Esteva M (2009) General shape grammar interpreter for intelligent designs generations. In *Proceedings of the 2009 6th International Conference on Computer Graphics, Imaging and Visualization: CGIV2009*, pp. 235–240.
Veblen O and Young JW (2007) *Projective Geometry*. Seaside, OR: Merchant Books.
Veltkamp RC and Hagedoorn M (2001) State of the art in shape matching. *Principles of Visual Information Retrieval*. London: Springer, pp. 87–119.
Woodburry R (2010) *Elements of Parametric Design*. New York: Routledge.
Woodbury R, Mohiuddin A, Cichy M and Mueller V (2017) Interactive design galleries: a general approach to interacting with design alternatives. *Design Studies* **52**, 40–72.

Tzu-Chieh Kurt Hong is a lecturer in the School of Architecture at the Georgia Institute of Technology teaching courses in media modeling, parametric modeling and design scripting. His research focuses on the areas of shape grammars, generative modeling, parametric design, and CAD software development. Kurt is a research scientist at the Shape Computation Lab working with Dr. Athanassios Economou on the Shape Machine, a new shape-rewrite computational technology, designing and implementing its geometry kernel, data representation and system architecture. Before entering the Shape Computation Lab, Kurt was a research assistant at the Digital Building Lab working with Dr. Dennis Shelden, and his main contribution to Digital Building Lab included the implementation of an automatic routing system in architecture and an interactive web-based data visualization platform, Smart 3D Atlanta. Kurt holds a BSEE, MSEE, and an M.Arch from the National Chiao Tung University (Taiwan), an MS.Arch from the University of Michigan, and a PhD in Architecture from Georgia Institute of Technology.

Athanassios Economou is Professor at the School of Architecture in the College of Design, and Adjunct Professor at the School of Interactive Computing at Georgia Institute of Technology. Dr. Economou's teaching and research are in the areas of shape grammars, computational design, computer-aided design and design theory, with over sixty published papers in these areas. He is the Director of the Shape Computation Lab, a research group that explores how the visual nature of shape can be formally implemented with new technologies to enable new paradigms in visual computation, design automation, and creative design. Recent projects include the Shape Machine, a new computational technology that successfully implements direct shape rewriting algorithms in CAD systems, and CourtsWeb, the most significant visual database on Federal Courthouses, sponsored by GSA and US.Courts. Design projects from his studios at Georgia Tech have received prestigious awards in international and national architectural competitions. He has been invited to give talks, seminars, and workshops at several universities including MIT, Harvard, TU Vienna, U. Michigan, KAIST, Chiao Tung U Taiwan, Emory, Seoul National U, Cambridge U, Tsinghua U, UCLA, NTUA, U.Thessaly, U.Aegean, among others. Dr. Economou holds a Diploma in Architecture from NTUA, Athens, Greece, an M.Arch from USC, and a PhD in Architecture from UCLA.

https://doi.org/10.1017/S0890060421000263 Published online by Cambridge University Press