



Structural Filtering for Object Recognition: A Deterministic Alternative to Optimization-Based Training

Huseyin Murat Cekirge

Department of Mechanical Engineering, The City College of New York (CUNY), New York, USA

Citation: Huseyin Murat Cekirge (2026) Structural Filtering for Object Recognition: A Deterministic Alternative to Optimization-Based Training. *J of Poin Artf Research* 2(2), 1-15 WMJ-JPAIR-128

Abstract

This paper records a structural observation obtained during recent deterministic object recognition experiments. When an object is represented on a finite grid and normalized by centroid alignment, the spatial location variable becomes intrinsic to the representation rather than an external parameter. Extending the representation to include color values per pixel introduces a natural color-space coordinate, which can be interpreted as an additional geometric dimension. Under this structured representation, rotational displacement does not require brute-force search. The relative shift angle emerges directly from the second-order spatial statistics (covariance structure) of the normalized object. Translation, color distribution, and rotation therefore appear as components of a unified finite geometric state rather than independent search variables. The result is a compact deterministic framework in which translation, color distribution, and orientation emerge directly from the structural representation itself. Recognition emerges through structural elimination rather than through parameter optimization, using deterministic normalization, pixel elimination search, and bounded angle search. The proposed framework is not intended to replace modern statistical learning systems. Rather, it demonstrates that in structured environments object recognition can also emerge from deterministic structural filtering. The proposed method can therefore be viewed as a complementary paradigm to optimization-based learning. In this sense, recognition is interpreted not as a process of parameter learning, but as the deterministic revelation of structural compatibility within a finite geometric representation.

***Corresponding author:** Huseyin Murat Cekirge, Department of Mechanical Engineering, The City College of New York (CUNY), New York, USA. **E-mail:** hmcekirge@usa.net

Submitted: 13.03.2026

Accepted: 17.03.2026

Published: 30.03.2026

Keywords: Deterministic Object Recognition, Structural Filtering, Finite Grid Representation, Pixel-Wise Elimination, Geometric Normalization, Non-Learning Recognition

Introduction

Object recognition is traditionally formulated as a statistical learning or optimization problem. In modern computer vision systems, recognition models are

typically trained through iterative optimization procedures in which parameters are adjusted to minimize a loss function defined over large training datasets. Gradient-based methods and neural architectures domi-

nate many large-scale recognition tasks because of their ability to model complex visual variability [1-6].

Earlier work in pattern recognition explored deterministic feature representations derived from geometric descriptors and spatial statistics. Invariant features such as moment invariants were introduced to characterize objects independently of translation, scale, and rotation [3]. These approaches demonstrated that structural properties of objects could provide stable recognition cues without requiring statistical training. Comprehensive reviews of pattern recognition methods describe the historical transition from rule-based systems to data-driven learning frameworks [4].

Despite the success of statistical learning, deterministic structural representations remain relevant for structured recognition tasks. When objects are represented on a finite grid with consistent geometric normalization, many degrees of freedom commonly treated as search variables can be eliminated directly through deterministic preprocessing. In particular, translation can be removed by centroid alignment, scale differences can be corrected through normalization of spatial spread, and orientation can be derived from second-order spatial statistics of the object representation.

Under this perspective, object recognition can be interpreted not as an optimization process but as a structural compatibility problem. Once objects are expressed in a common normalized coordinate system, recognition reduces to verifying structural consistency between a query object and entries stored in a library. The comparison process can therefore be implemented through deterministic pixel-wise evaluation combined with progressive candidate elimination.

The framework explored in this study follows this structural viewpoint. Objects are represented on finite grids and undergo deterministic normalization through centroid alignment, scale normalization, and orientation correction. After normalization, recognition proceeds through a pixel-wise tally mechanism in which incompatible library objects are progressively eliminated as structural constraints accumulate. Because the representation is finite and

normalized, the search process remains bounded and computationally inexpensive.

This approach does not attempt to replace large-scale statistical learning systems that operate on highly variable natural image datasets. Instead, the goal is to demonstrate that for structured object libraries, recognition can emerge directly from deterministic structural filtering without training procedures, gradient-based optimization, or iterative parameter estimation.

The experiments presented in this work show that once objects are represented within a normalized finite-grid framework, translation, orientation, and structural compatibility can be handled through deterministic operations. Recognition thus arises as a consequence of accumulated structural constraints rather than numerical optimization. Classical recognition methods relied on structural descriptors and template matching techniques, while modern approaches increasingly employ statistical learning frameworks and deep neural architectures [1-5, 7-12].

Deterministic Recognition Paradigm

The recognition framework proposed in this study follows a deterministic structural paradigm rather than a statistical learning paradigm. In many contemporary recognition systems, the identification of objects is formulated as an optimization problem in which model parameters are adjusted through iterative learning procedures. In contrast, the deterministic paradigm assumes that the structural compatibility between an observed object and a stored representation already contains the information necessary for recognition.

Under this perspective, recognition does not emerge from parameter estimation or loss minimization but from the progressive application of structural constraints. When objects are expressed within a normalized finite-grid representation, each pixel contributes a deterministic compatibility condition. As additional structural constraints are evaluated, incompatible candidates are eliminated from the library until a single consistent object remains. The recognition process is therefore not an optimization trajectory, but a monotone reduction over a finite compatibility space.

Deterministic Angle Vicinity Search Sequence

The angular refinement step is performed using a deterministic vicinity search. A fixed angular increment

of $\Delta\theta = 10^\circ$ is used as the search step. Starting from the reference orientation, the algorithm evaluates candidate orientations by expanding symmetrically in both rotational directions.

The search begins at 0° , followed by $+10^\circ$ and -10° . Negative angles are mapped to the interval $[0^\circ, 360^\circ)$, so that -10° corresponds to 350° . The procedure continues by testing $+20^\circ$ and -20° (340°), followed by $+30^\circ$ and -30° (330°), and so on. The symmetric expansion continues until a satisfactory match is obtained.

In general, the tested orientations follow the sequence

$$\theta_k = \pm k \Delta\theta \quad (1)$$

where k denotes the step index and $\Delta\theta$ is the angular increment. $[0^\circ, 360^\circ]$ using modulo 360 arithmetic. The deterministic symmetric search sequence is illustrated in Figure 1.

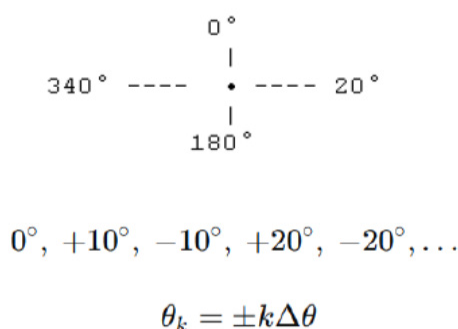


Figure 1: Deterministic symmetric angle vicinity search used to compensate for small residual rotation after structural normalization.

Efficiency Consideration

If real rotation is small: Match found early. If rotation is large: Search expands symmetrically, for $\Delta\theta = 10^\circ$: $N = 18$ steps (worst case). The search is finite, deterministic, and bounded.

Deterministic Conditional Search Logic

For $k = 0$ to N :

test $\theta = +k \Delta\theta$
 if match \rightarrow stop

if $k > 0$:
 test $\theta = -k \Delta\theta$
 if match \rightarrow stop

This is not a brute-force sweep over the entire $0-360^\circ$ range. Instead, it performs a center-first symmetric expansion, which minimizes the average search time while maintaining deterministic bounds.

Pixel-Wise Tally and Deterministic Elimination

After geometric normalization and invariant-based candidate reduction, the final recognition stage is performed through a deterministic pixel-wise comparison between the query object and the remaining library candidates. The procedure relies on a simple tally mechanism combined with progressive elimination.

Let the normalized object representation be defined on a fixed $N \times N$ grid. For the experiments considered in this work, $N=7$, resulting in 49 pixels per object.

Each object is flattened into a vector $v = [v_1, v_2, \dots, v_{49}]^T$ (2)

where

- $v_1, v_2 \dots v_{49} \rightarrow$ pixel values
- $\dots \rightarrow$ continuation
- $^T \rightarrow$ transpose (superscript T)

7x7 GRID

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ \dots & & & & & & \\ \end{bmatrix}$$

↓ row-wise flattening

$$v = [v_1 \ v_2 \ v_3 \ \dots \ v_{49}]^T$$

Figure 2: Transformation of the 7×7 grid representation into a structural vector by row-wise flattening.

For a query vector q and a candidate library vector $v(k)$, a match score is computed using a pixel-wise tally.

$$S_k = \sum_{i=1}^{49} \delta(q_i, v_i^{(k)}) \quad (3)$$

- $S_k \rightarrow$ tally score for candidate object k
- $\delta(\cdot) \rightarrow$ Kronecker delta function
- $q_i \rightarrow$ query pixel value
- $v_i^{(k)} \rightarrow$ i -th pixel of the k -th library object and $\sum_{i=1}^{49} \rightarrow$ summation over all pixels of the $N \times N$ grid

Delta function definition:

$$\delta(q_i, v_i^{(k)}) = \begin{cases} 1, & q_i = v_i^{(k)} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where $\delta(\cdot)$ denotes the Kronecker delta function, and each matching pixel contributes one unit to the tally score. Recognition proceeds sequentially through the pixels. During the comparison process, candidates whose remaining potential score cannot exceed the current best candidate are immediately eliminated. This early elimination mechanism drastically reduces the number of active candidates during the scan.

The computational operations required in this stage are minimal. The algorithm involves only:

- binary comparisons,
- integer counting, and
- conditional elimination.

No matrix inversion, optimization procedure, or iterative equation solving are required.

Rotational uncertainty is handled through a limited orientation search. Instead of evaluating all possible orientations, the algorithm examines a small angular vicinity sequence, that is, $(0^\circ, 10^\circ, 350^\circ, 20^\circ, 340^\circ, \dots)$ or, $(0^\circ, +10^\circ, -10^\circ, +20^\circ, -20^\circ, \dots)$ (5)

and,

$$\theta_k = \pm k \Delta\theta \quad (6)$$

until a stable maximum tally score is obtained. Because the candidate set has already been strongly reduced by invariant filtering, the additional computational cost remains small.

This deterministic tally-based comparison provides an efficient final discrimination stage for structured low-resolution pattern libraries.

Structural Elimination Recognition (SER) Paradigm

The Structural Elimination Recognition (SER) paradigm interprets object recognition as a deterministic structural filtering process in which candidate objects are progressively eliminated through accumulated compatibility constraints within a normalized finite-grid representation. Recognition is therefore achieved not through parameter optimization but through the deterministic

revelation of structural compatibility between the query object and the stored library. The method is built on three things:

- finite grid representation
- pixel elimination
- deterministic normalization

Interpretation in the Context of Autonomic Systems

The proposed deterministic framework can also be interpreted within the conceptual framework of autonomic systems. The solution is not discovered through optimization; rather, it is revealed through the elimination of incompatible structures. This viewpoint emphasizes representation, normalization, and structural compatibility as the fundamental mechanisms of recognition.

The proposed framework can be interpreted as a Structural Elimination Recognition (SER) paradigm. The deterministic recognition mechanism proposed in this study can be interpreted within the conceptual framework of autonomic systems. Classical autonomic computing describes systems that are capable of self-management through properties such as self-configuration, self-optimization, self-healing, and self-protection, typically achieved via feedback loops, adaptive control, and learning-based optimization [13-15]. In contrast, the present framework exhibits a form of deterministic autonomic behavior that does not rely on learning, parameter adaptation, or iterative optimization procedures. Instead, the system operates through a finite sequence of structural consistency evaluations. At each stage of the recognition process, incompatible candidates are eliminated based on invariant and geometric constraints, resulting in a progressive reduction of the feasible solution set.

From this perspective, the recognition process may be viewed as a self-regulating mechanism in which the system internally enforces structural compatibility without external supervision. The evolution of the candidate set is not governed by gradient-based updates or probabilistic inference, but by explicit deterministic filtering. As a consequence, the system converges in a finite number of steps, bounded by the size of the discrete representation and the library.

This interpretation highlights a fundamental distinction from conventional autonomic systems. While classical approaches achieve self-regulation through adaptive

feedback and optimization dynamics, the proposed method achieves a functionally comparable form of internal regulation through deterministic elimination over a finite compatibility space. No training phase, parameter tuning, or stochastic process is required.

Therefore, the framework may be understood as an instance of non-learning autonomic behavior, in which self-organization emerges from structural constraints rather than adaptive mechanisms. This viewpoint does not aim to replace existing autonomic paradigms, but rather to complement them by demonstrating that self-regulating behavior can arise in fully deterministic and finite settings.

This interpretation is consistent with the MAPE-K (Monitor–Analyze–Plan–Execute over Knowledge) model of autonomic computing, where structural compatibility evaluation replaces adaptive feedback and learning mechanisms. The deterministic elimination process can also be interpreted in control-theoretic terms as a finite-step stability mechanism, where the candidate set size acts as a discrete Lyapunov-like measure that decreases monotonically under structural constraints until convergence is achieved.

Related Work

Object recognition has been studied extensively within the broader field of pattern recognition and computer vision. Early work focused on extracting geometric descriptors and structural features capable of distinguishing objects under transformations such as translation, scaling, and rotation. One of the earliest influential contributions was the introduction of moment invariants, which provide descriptors that remain unchanged under common geometric transformations [3]. Moment-based approaches demonstrated that structural properties of shapes could serve as stable identifiers even when objects appear under varying orientations.

Subsequent developments in pattern recognition expanded the use of geometric and statistical descriptors. Structural and statistical classification methods were widely studied during the late twentieth century, where objects were represented through feature vectors and classified using distance measures or decision rules [4]. These approaches emphasized feature extraction and similarity comparison rather

than large-scale parameter optimization.

Template matching methods also provided an early deterministic approach to object recognition. In these methods, an observed object is compared directly with stored reference templates using similarity metrics such as correlation or Hamming distance [2]. While conceptually simple, classical template matching typically requires exhaustive comparison across translations and rotations, which increases computational cost as image resolution grows.

With the rapid growth of computational power and large datasets, modern object recognition research has increasingly shifted toward statistical learning frameworks. Machine learning techniques, particularly convolutional neural networks, learn hierarchical feature representations through gradient-based optimization procedures [6]. These models have demonstrated remarkable performance in large-scale visual recognition tasks but typically require extensive training data, significant computational resources, and complex parameter tuning.

Despite the success of deep learning, deterministic recognition strategies remain relevant in structured environments where objects can be represented within constrained geometric frameworks. In such cases, explicit structural normalization and direct compatibility checks may provide efficient alternatives to learning-based approaches. Several studies in computer vision continue to emphasize the importance of geometric normalization, invariant descriptors, and structured representations for reducing search complexity [8].

The approach explored in this work follows this deterministic structural perspective. Instead of relying on statistical training procedures, objects are represented on finite grids and normalized through deterministic geometric operations. Recognition is then performed through pixel-wise structural comparison combined with progressive candidate elimination. In this formulation, the recognition process emerges from accumulated structural constraints rather than from optimization-based learning.

The method therefore differs from classical template matching by incorporating structural normalization and deterministic elimination, while also differing

from statistical learning systems by avoiding training procedures and gradient-based optimization. The goal is not to compete with large-scale learning architectures but to demonstrate that deterministic structural filtering can provide an efficient recognition mechanism for structured object libraries.

The objective of the present work is not to replace statistical learning systems, but to demonstrate that in structured and geometrically constrained environments, deterministic structural filtering can provide a simple, interpretable, and computationally efficient recognition mechanism, while remaining fundamentally distinct from classical template matching approaches.

Deterministic Structural Representation on Finite Grids

In the proposed framework, objects are represented on a finite spatial grid that provides a compact and deterministic structural description. The representation converts a visual object into a fixed-dimensional structure that can be processed through direct comparison and elimination rather than statistical inference.

Let the observation domain be discretized into an $N \times N$ grid. Each cell of the grid corresponds to a pixel location in the normalized object representation. The object structure is therefore described by N^2 spatial elements.

For a binary representation, each grid cell takes one of two possible values
 $v_i \in \{0, 1\}$ (7)

where
 0 represents absence of the object at that pixel location

1 represents presence of the object.

The object can therefore be written as a vector obtained by flattening the grid row by row.
 $v = [v_1, v_2, \dots, v_{N^2}]^T$ (8)

For example, in a 7×7 grid the object contains $N^2 = 49$ pixel elements, producing a 49-dimensional structural vector.

Because every object is represented using the same normalized grid, the resulting vectors share a common coordinate system. This allows objects to be stored in a deterministic structural database where each candidate is represented by a fixed-length vector.

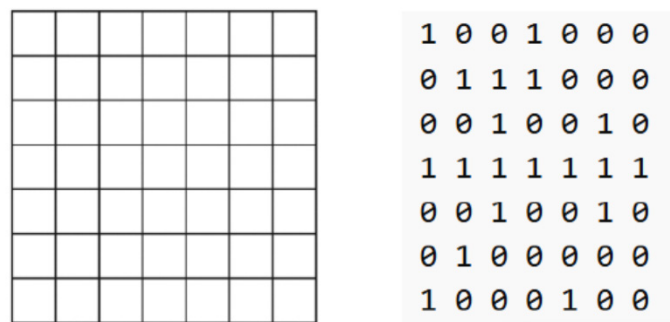


Figure 3: Example object represented on a 7×7 binary grid. Active cells correspond to object pixels.

This representation is deterministic because each pixel corresponds to a fixed spatial coordinate within the normalized grid. Once normalization is completed, every object in the library and every query object share the same spatial reference frame.

The finite-grid representation provides several important structural properties.

First, the dimensionality of the object description is fixed and small compared with the dimensionality of raw image data. This allows direct structural comparison between objects without requiring dimensionality reduction procedures.

Second, the spatial relationships between pixels remain explicitly encoded in the grid structure. Unlike feature extraction methods that transform the data into abstract feature spaces, the proposed representation preserves the physical spatial organization of the object.

Third, the representation is directly compatible with deterministic comparison operations. Since each pixel corresponds to a known location, structural compatibility between two objects can be evaluated through pixel-wise comparison.

Let q denote the vector representation of a query object and $v^{(k)}$ denote the vector representation of the k -th object in the library. Structural comparison

between these two objects can be expressed as in Equation (3). The quantity S_k represents the number of structurally compatible pixels between the query object and the candidate object.

Because the representation is finite, the recognition process becomes a bounded structural filtering problem. Each pixel introduces an additional compatibility constraint. As the number of evaluated pixels increases, incompatible objects are progressively eliminated from the candidate set. This progressive elimination mechanism allows recognition to emerge from accumulated structural constraints rather than from statistical estimation or optimization procedures.

The deterministic nature of the finite-grid representation therefore provides the foundation for the elimination-based recognition process described in the following sections.

Structural Normalization of the Blind Object

Before structural comparison can be performed, the observed object must be transformed into the same geometric reference frame as the objects stored in the library. Because library objects are assumed to be centered, scale-consistent, and represented on a fixed $N \times N$ grid, the blind object must undergo an equivalent normalization process.

The purpose of normalization is to remove variations caused by translation, scale differences, and orientation so that structural comparison becomes meaningful. After normalization, the blind object and the library objects share the same structural coordinate system.

The normalization process consists of three deterministic steps: centroid alignment, scale normalization, and orientation correction.

Centroid Alignment

The blind object may initially appear displaced within the observation frame. To remove this translation effect, the centroid of the object is computed from the active pixels.

$$\mu_x = \sum w_i x_i / \sum w_i \quad (9)$$

$$\mu_y = \sum w_i y_i / \sum w_i \quad (10)$$

where w_i denotes the weight associated with pixel i . For binary representations $w_i = 1$ for active pixels and $w_i = 0$ otherwise.

The coordinates of the pixels are then shifted relative to the centroid

$$x_i' = x_i - \mu_x \quad (11)$$

$$y_i' = y_i - \mu_y \quad (12)$$

This translation aligns the centroid of the blind object with the origin of the normalized coordinate system, ensuring that the structural position of the object becomes independent of its original location in the observation frame. After this transformation the object becomes centered within the coordinate system. Translation differences between objects are therefore eliminated.

Scale Normalization

Objects may appear at different physical sizes within the observation frame. To ensure compatibility with the library representation, the spatial spread of the centered coordinates is normalized.

The spatial dispersion of the object is measured by

$$S = \sum_i w_i (x_i'^2 + y_i'^2) \quad (13)$$

where

\sum_i denotes the summation over all pixel indices i . (14)

The quantity S corresponds to the second spatial moment of the normalized object distribution and therefore measures the intrinsic spatial dispersion of the object independently of its absolute position.

The normalized coordinates are then defined as

$$x_i'' = x_i' / \sqrt{S}, \quad (15)$$

$$y_i'' = y_i' / \sqrt{S}. \quad (16)$$

This operation rescales the object while preserving its geometric proportions. Consequently, objects that originally appear at different sizes become structurally comparable within the normalized coordinate system.

In the proposed framework no predefined fixed scale factor is assumed between the blind object and the objects stored in the library. Instead, scale compatibility

is achieved through structural normalization derived directly from the internal spatial distribution of the object.

After centroid alignment, the spatial dispersion of the object is measured using Equation (13). This normalization removes scale differences by mapping the object into a unit structural space. As a result, both the blind object and the library objects are expressed within the same normalized coordinate system.

Because the normalization depends only on the intrinsic spatial structure of the object, no external scale parameter needs to be specified. Scale compatibility therefore emerges naturally from the normalization process.

The overall normalization procedure can therefore be summarized as

translation → centroid alignment

scale → dispersion normalization

rotation → covariance-based orientation correction

Orientation Correction

Even after centering and scale normalization, the object may still be rotated relative to the orientation of the stored library objects. The orientation of the object can be estimated from the covariance structure of the normalized coordinates.

The covariance components are computed as

$$S_{xx} = \sum w_i x_i^2 \quad (17)$$

$$S_{yy} = \sum w_i y_i^2 \quad (18)$$

$$S_{xy} = \sum w_i x_i y_i \quad (19)$$

$$C = \begin{bmatrix} S_{xx} & S_{xy} \\ S_{xy} & S_{yy} \end{bmatrix} \quad (20)$$

The principal orientation angle is obtained from the covariance structure

$$\theta = \left(\frac{1}{2}\right) \arctan\left(\frac{2S_{xy}}{(S_{xx} - S_{yy})}\right) \quad (21)$$

The blind object is then rotated by $-\theta$ so that its dominant axis aligns with the reference orientation of the library objects.

This operation removes rotational displacement deterministically and avoids exhaustive rotational search over the full angular range.

Structural Compatibility

After centroid alignment, scale normalization, and orientation correction, the blind object and the library objects share the same structural reference frame. The normalized object can therefore be embedded into the fixed $N \times N$ grid representation.

At this stage the object becomes structurally compatible with the library representation, allowing direct pixel-wise comparison and deterministic elimination during the recognition stage.

Deterministic Pixel Elimination Search

After structural normalization, the blind object and the objects stored in the library share the same spatial coordinate system. Recognition can therefore be performed through direct structural comparison.

The proposed method performs recognition through deterministic pixel-wise comparison combined with progressive candidate elimination. The procedure does not rely on statistical learning, optimization procedures, or iterative parameter estimation.

Instead, incompatible objects are progressively removed from the candidate set as structural mismatches are detected.

Pixel Vector Representation

Let the normalized object be represented on an $N \times N$ grid. The object is flattened into a structural vector

$$v = [v_1, v_2, \dots, v_{N^2}]^T \quad (22)$$

Similarly, the blind object is represented as

$$q = [q_1, q_2, \dots, q_{N^2}]^T \quad (23)$$

where each element corresponds to the value of a pixel at a fixed spatial location. For the experiments considered in this work, $N = 7$, and therefore $N^2 = 49$.

Pixel-Wise Tally

The structural compatibility between the blind object and a library object k is measured by a pixel-wise tally score. Each matching pixel contributes one unit to the tally score.

The maximum possible score is therefore

$$S_{\max} = N^2. \quad (24)$$

The candidate object with the highest tally score represents the object that is most structurally compatible with the blind object.

Because the comparison is performed directly on the normalized structural grid, recognition emerges from the accumulation of matching structural elements rather than from probabilistic inference or optimization-based learning.

```
Pixel 1 → {Leopard, Apple, Pear}
Pixel 2 → {Leopard, Apple}
Pixel 3 → {Leopard}
```

Figure 4: Progressive elimination of incompatible library objects during pixel comparison.

Progressive Candidate Elimination

Recognition proceeds sequentially through the pixels of the grid. As each pixel is evaluated, objects that are incompatible with the observed pixel value are removed from the candidate set.

This process can be interpreted as a progressive intersection of compatible object identifiers.

Illustrative example:

```
pixel 1 → {leopard, apple, pear}
pixel 2 → {leopard, apple}
pixel 3 → {leopard}
```

As additional pixels are examined, the candidate set becomes progressively smaller until a single compatible object remains.

Because incompatible objects are eliminated early, the number of active candidates typically decreases rapidly during the comparison process.

Early Elimination Principle

A further efficiency improvement is obtained through an early elimination rule. During the scan, the remaining maximum achievable score for each candidate object can be estimated. If the maximum possible score that a candidate could obtain from the remaining pixels cannot exceed the current best score, that candidate is immediately removed from the candidate set. This early elimination principle

significantly reduces the number of comparisons required in practice.

The operations required by the algorithm are limited to:

- binary pixel comparison
- integer tally accumulation
- logical candidate elimination

No matrix inversion, optimization procedure, or gradient-based learning is required.

Deterministic Decision

The recognition process continues until one of the following conditions is satisfied:

- only one candidate object remains
- all pixels have been evaluated

The remaining object identifier corresponds to the recognized object. Because the representation is finite and the number of pixels is bounded, the recognition procedure is deterministic and guaranteed to terminate after a finite number of comparisons.

Structural Interpretation

The deterministic elimination search interprets recognition as a structural compatibility process. Each pixel introduces an additional constraint on the candidate objects stored in the library.

As the number of evaluated pixels increases, incompatible objects are progressively removed from the candidate set. Recognition therefore emerges from the accumulation of structural constraints rather than from statistical estimation or optimization procedures.

Deterministic Angle Vicinity Search

Although structural normalization significantly reduces orientation differences between the blind object and the library objects, small rotational deviations may still remain due to discretization effects or imperfect alignment. These small deviations can reduce the pixel-wise tally score even when the correct object is present in the library.

To address this issue, the proposed method applies a deterministic angular vicinity search. Instead of performing an exhaustive search over all possible rotations, the algorithm evaluates a small set of orientations around the reference orientation.

Symmetric Angle Expansion

The search begins from the reference orientation and expands symmetrically in both rotational directions.

The search order follows the sequence $0^\circ, +\Delta\theta, -\Delta\theta, +2\Delta\theta, -2\Delta\theta, +3\Delta\theta, -3\Delta\theta, \dots$ (25)

This sequence ensures that small rotational deviations are evaluated first while maintaining deterministic bounds on the search process.

The general form of the tested angles is

$$\theta_k = \pm k \Delta\theta \quad (26)$$

where $\Delta\theta$ denotes the angular step size and k is a non-negative integer.

To maintain valid angles within the circular range, the values are mapped to the interval $[0^\circ, 360^\circ)$ (27)

using modulo arithmetic.

Search Bound

Because the angular search expands symmetrically around the reference orientation, the maximum number of steps required to cover the full angular range is

$$N = 180 / \Delta\theta \quad (28)$$

For example, when

$$\Delta\theta = 10^\circ \quad (29)$$

the maximum number of evaluated orientations is $N = 18$. (30)

In practice, recognition is typically achieved much earlier because small orientation deviations are corrected within the first few steps.

Recognition Procedure

For each tested orientation the blind object is rotated accordingly and the deterministic pixel elimination search described in the previous section is executed. The procedure for each angular step consists of

- the normalized object,
- pixel-wise tally comparison,
- applying candidate elimination.

If a stable maximum tally score is obtained or a single candidate remains in the candidate set, the

search terminates.

Deterministic Nature of the Search

Unlike exhaustive rotational matching methods, the proposed approach evaluates only a limited set of orientations within a symmetric angular vicinity. The search therefore remains deterministic and bounded.

Because the candidate set has already been significantly reduced by structural normalization and pixel elimination, the additional computational cost of testing several nearby orientations remains small.

The angular vicinity search therefore provides a practical mechanism for handling small rotational uncertainties while preserving the deterministic nature of the recognition framework.

Structural Interpretation

The vicinity search does not represent a brute-force exploration of all possible rotations. Instead, it acts as a localized structural refinement step that compensates for discretization errors and minor rotational deviations.

Recognition therefore remains primarily driven by deterministic structural compatibility, while the limited angular search provides robustness against small alignment imperfections.

Computational Complexity and Deterministic Bounds

The proposed recognition framework operates through deterministic preprocessing followed by structural comparison and candidate elimination. Because the representation is finite and the operations are simple, the computational behavior of the algorithm can be analyzed directly.

Structural Representation Size

Let the normalized object be represented on an $N \times N$ grid. The total number of pixel elements is therefore $P = N^2$ (31)

For example, in the experimental configuration $N = 7$, which yields $P = 49$. Each object in the library is therefore represented as a vector of fixed length P .

Library Size

Let M denote the number of objects stored in the

library. The recognition task consists of identifying the correct object among these M candidates.

In a naive comparison approach, the worst-case computational cost of comparing a query object with every library object is proportional to

$$O(MP) \quad (32)$$

because each candidate object requires P pixel comparisons.

Progressive Elimination Effect

The deterministic elimination mechanism significantly reduces the effective computational cost.

During the pixel scan, incompatible objects are removed from the candidate set as soon as a structural conflict is detected. As additional pixels are examined, the candidate set shrinks progressively.

If the number of remaining candidates after evaluating pixel i is denoted by M_i , then

$$M_0 = M \quad (33)$$

$$M_1 \leq M \quad (34)$$

$$M_2 \leq M_1 \quad (35)$$

$$\dots$$

$$M_P \leq 1. \quad (36)$$

In practice, the candidate set typically collapses rapidly, often after only a small number of pixel evaluations. Consequently, the average computational effort is substantially smaller than the theoretical upper bound $O(MP)$.

Deterministic Upper Bound

Despite the elimination acceleration, the recognition process remains strictly bounded.

In the worst case the algorithm may examine all pixels of the grid for all candidate objects. The maximum number of pixel comparisons is therefore $C_{\max} = M \times P$. (37)

Because both M and P are finite, the recognition procedure always terminates after a finite number of operations. This deterministic bound ensures predictable computational behavior.

Angular Search Cost

When rotational uncertainty is present, the deterministic angle vicinity search described in the

previous section may evaluate several orientations.

Let $\Delta\theta$ denote the angular step size. The maximum number of tested orientations is

$$K = 180 / \Delta\theta. \quad (38)$$

For example,

$$K = 18. \rightarrow \Delta\theta = 10^\circ \quad (39)$$

The worst-case computational cost including angular search therefore becomes

$$O(KMP). \quad (40)$$

However, because the candidate set is strongly reduced by elimination and because the correct orientation is typically found early in the vicinity search, the practical computational cost remains significantly lower.

Deterministic Computational Structure

The algorithm involves only simple computational operations:

- binary pixel comparisons
- integer tally accumulation
- logical candidate elimination
- limited-angle rotation tests

No matrix inversion, iterative optimization, gradient descent, or training procedure is required. The computational behavior of the system is therefore fully deterministic and bounded.

Structural Scalability

As the grid resolution increases, the number of structural constraints also increases. Each additional pixel introduces an extra compatibility condition between the query object and the library objects.

Consequently, larger grids tend to strengthen the elimination process rather than significantly increase computational burden. Incompatible objects are typically eliminated earlier because more structural constraints are available.

Recognition therefore emerges from the accumulation of structural constraints rather than from computationally intensive optimization procedures. The recognition task is therefore governed by structural constraint accumulation rather than by computationally intensive optimization dynamics.

Experimental Validation (7 × 7 Library Experiment)

To demonstrate the deterministic recognition framework, a small-scale experimental library was constructed using objects represented on a fixed 7 × 7 grid. The purpose of this experiment is not to achieve large-scale classification performance but to illustrate the deterministic elimination mechanism under controlled structural conditions.

Library Construction

The experimental library consists of M objects represented on a 7 × 7 binary grid.

$$P = N^2 = 49 \rightarrow N = 7 \quad (41)$$

Each object is therefore represented as a vector obtained by flattening the grid row by row.

$$v = [v_1, v_2, \dots, v_{49}]^T \quad (42)$$

Each pixel takes a binary value

$$0 \rightarrow \text{empty pixel} \quad (43)$$

$$1 \rightarrow \text{active object pixel} \quad (44)$$

The library is stored as a matrix

$$L = [v(1) \ v(2) \ \dots \ v(M)] \quad (45)$$

where each column corresponds to one stored object.

Each object is assigned a unique identifier (UID) to enable deterministic identification during the elimination process.

Query Object

A blind object is selected from the same structural class and used as the query object.

The query object undergoes the deterministic normalization process described earlier:

- centroid alignment
- scale normalization
- orientation correction

After normalization, the object is embedded into the same 7 × 7 reference grid used by the library.

The normalized query vector is therefore

$$q = [q_1, q_2, \dots, q_{49}]^T. \quad (46)$$

Deterministic Recognition Procedure

Recognition is performed through the deterministic

pixel elimination search.

The algorithm evaluates pixels sequentially and retrieves the list of library objects compatible with each pixel value.

Illustrative elimination sequence:

pixel 1 → object 12, object 45, object 78

pixel 2 → object 12, object 45

pixel 3 → object 12

As additional pixels are examined, incompatible objects are progressively removed from the candidate set. The process continues until a single compatible object identifier remains. The final remaining identifier corresponds to the recognized object.

Rotation Robustness Test

To evaluate the robustness of the recognition procedure, rotated versions of the query object were also tested.

The object was rotated by a small angle relative to the library orientation. The deterministic angle vicinity search described earlier was then applied.

Search sequence example:

$$0^\circ, +10^\circ, -10^\circ, +20^\circ, -20^\circ \quad (47)$$

For each tested orientation the normalized object was rotated and the deterministic elimination procedure was executed. In the tested cases the correct object was identified after only a few orientation evaluations because the correct orientation was typically located near the initial reference orientation.

Observed Behavior

The experiment confirmed several structural properties of the proposed recognition method.

First, the candidate set shrinks rapidly during the elimination process. Even when the library contains many objects, only a small number of pixels are typically required to isolate the correct object.

Second, the deterministic angle vicinity search successfully compensates for small rotational deviations without requiring exhaustive rotation across the full angular range.

Third, increasing the grid resolution strengthens the recognition process because each additional pixel introduces an additional structural compatibility constraint.

These observations confirm that object recognition in the proposed framework emerges from deterministic structural filtering rather than from statistical training procedures.

Purpose of the Experiment

The 7×7 experiment serves primarily as a conceptual demonstration of the deterministic recognition mechanism.

The goal is to illustrate how the following components combine to produce a bounded and efficient recognition process:

- finite structural representation
- deterministic normalization
- pixel-wise elimination

The same principles can be extended to larger grids and larger object libraries, where the progressive elimination mechanism becomes even more effective as the number of structural constraints increases. The experiment demonstrates that recognition can emerge from deterministic structural compatibility rather than from data-driven learning or optimization-based training.

Conclusion

This study presented a deterministic framework for structured object recognition based on finite-grid representations and structural compatibility filtering. The proposed method demonstrates that object recognition can be formulated as a deterministic structural process rather than as a statistical learning problem.

Objects are represented on finite $N \times N$ grids and transformed into a common structural reference frame through centroid alignment, scale normalization, and orientation correction. Once the blind object and the library objects share the same geometric coordinate system, recognition can be performed through direct structural comparison.

The core recognition mechanism is based on deterministic pixel-wise tally evaluation combined with progressive candidate elimination. Each

evaluated pixel introduces an additional compatibility constraint that gradually removes inconsistent objects from the candidate set. As the number of evaluated pixels increases, the candidate set collapses until a single compatible object remains.

To address small residual orientation differences, a deterministic angular vicinity search is applied. Instead of performing exhaustive rotation across the full angular range, the search expands symmetrically around the reference orientation using a bounded sequence of angles. This approach maintains deterministic computational limits while providing robustness against small alignment deviations.

The computational structure of the proposed framework is simple and predictable. The algorithm requires only basic arithmetic operations, binary comparisons, and logical elimination steps. No gradient descent, optimization procedure, matrix inversion, or training phase is required. Because the representation is finite and the number of operations is bounded, the recognition process remains computationally stable and deterministic.

Experimental validation using a 7×7 object library demonstrated that recognition emerges naturally from accumulated structural constraints. As additional pixels are evaluated, incompatible objects are progressively eliminated, allowing the correct object identifier to be isolated efficiently. The results illustrate how deterministic structural filtering can provide a practical recognition mechanism for structured object libraries.

The purpose of this work is not to compete with large-scale statistical learning systems designed for highly variable natural image datasets. Instead, the study demonstrates that deterministic structural representations combined with elimination-based filtering can provide an efficient and conceptually transparent recognition strategy in structured environments.

Contribution Statement

This work introduces a deterministic framework for structured object recognition based on finite-grid representations and structural compatibility filtering. Unlike conventional recognition systems that rely on statistical training or optimization-based learning,

the proposed method formulates recognition as a deterministic structural elimination process.

The main contributions of this study are summarized as follows:

1. Deterministic structural representation. Objects are represented on finite $N \times N$ grids that preserve spatial structure while providing a fixed-dimensional representation suitable for direct comparison.
2. Normalization through deterministic geometric operations. The blind object is transformed into the same reference frame as the library objects using centroid alignment, scale normalization, and covariance-based orientation correction.
3. Pixel-wise elimination-based recognition. Recognition is performed through deterministic pixel comparison combined with progressive candidate elimination, allowing incompatible objects to be removed early in the comparison process.
4. Bounded angular vicinity search. Small rotational deviations are handled through a deterministic symmetric angle search rather than exhaustive rotational scanning.
5. Deterministic computational structure. The recognition algorithm requires only simple arithmetic operations, binary comparisons, and logical elimination steps. No training procedures, gradient descent, matrix inversion, or iterative optimization are required.

These contributions demonstrate that object recognition in structured environments can emerge from deterministic structural filtering rather than from statistical learning procedures.

Limitations of the Method

Although the proposed deterministic recognition framework provides an efficient structural filtering mechanism, several limitations should be noted.

First, the method assumes that objects can be represented within a consistent finite-grid structure. This assumption is appropriate for structured object libraries but may become restrictive when dealing with highly complex natural images containing large variations in illumination, deformation, or background clutter.

Second, the recognition mechanism relies on spatial structural compatibility between the query object and the library objects. If an object undergoes severe deformation or if significant portions of the object are occluded, the pixel-wise structural comparison may lose reliability.

Third, the current implementation uses relatively low-resolution grid representations such as 7×7 for conceptual demonstration. While this representation is sufficient to illustrate the deterministic elimination principle, higher-resolution grids may be required for recognizing more complex object structures.

Fourth, the deterministic angle vicinity search assumes that the residual rotational misalignment between the query object and the library representation is relatively small. Extremely large rotational variations may increase the number of tested orientations.

Finally, the present framework has been demonstrated on structured prototype libraries. Large-scale recognition problems involving millions of highly variable objects may require additional indexing mechanisms or hierarchical filtering strategies to maintain efficiency.

These limitations do not invalidate the deterministic framework but rather define the conditions under which the proposed approach is most applicable.

Future Research Directions

Several directions may be explored to extend the deterministic structural recognition framework proposed in this work.

One possible extension involves increasing the grid resolution of the structural representation. Higher-resolution grids introduce additional structural constraints that may strengthen the elimination mechanism and improve discrimination between similar objects.

Another promising direction is the incorporation of additional pixel attributes, such as color descriptors or intensity values. Extending the representation beyond binary pixels may allow the deterministic elimination process to exploit richer structural information.

The development of efficient indexing mechanisms

for large object libraries also represents an important research direction. Structured database indexing methods may further accelerate candidate elimination when the number of stored objects becomes large.

Future studies may also investigate hierarchical recognition strategies in which coarse structural filtering is applied first, followed by finer structural comparison stages. Such multi-level approaches could maintain deterministic behavior while improving scalability.

Finally, the deterministic structural filtering framework may be applied to other domains where structured spatial data appear, including industrial inspection, symbol recognition, and structured document analysis.

These directions suggest that deterministic structural recognition may serve as a complementary paradigm alongside statistical learning approaches in applications where structured representations are available.

Author Contributions

Huseyin Murat Cekirge is the sole author. The author read and approved the final manuscript.

Conflicts of Interest

The author declares no conflicts of interest.

References

1. Bishop CM (2006) Pattern Recognition and Machine Learning. Springer <https://link.springer.com/book/9780387310732>.
2. Gonzalez RC, Woods RE (2008) Digital Image Processing <https://www.cl72.org/090imagePLib/books/Gonzales,Woods-Digital.Image.Processing.4th.Edition.pdf>.
3. Hu MK (1962) Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 8: 179-187.
4. Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22: 4-37.
5. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521: 436-444.
6. Szeliski R (2010) Computer Vision: Algorithms and Applications. Springer <https://link.springer.com/book/10.1007/978-3-030-34372-9>.
7. Duda RO, Hart PE, Stork DG (2001) Pattern Classification. Wiley <https://tjzhifei.github.io/links/PC2.pdf>.
8. Forsyth DA, Ponce J (2012) Computer Vision: A Modern Approach. Pearson https://www.academia.edu/65495130/Computer_Vision_A_modern_approach.
9. Sonka M, Hlavac V, Boyle R (2008) Image Processing, Analysis, and Machine Vision. Cengage Learning <https://www.cl72.org/090imagePLib/books/sonka,hlavac,boyle-imageProc.pdf>.
10. Brunelli R (2009) Template Matching Techniques in Computer Vision: Theory and Practice. Wiley <https://onlinelibrary.wiley.com/doi/book/10.1002/9780470744055>.
11. Zhang D, Lu G (2004) Review of shape representation and description techniques. Pattern Recognition 37: 1-19.
12. Ballard DH, Brown CM (1982) Computer Vision. Prentice Hall https://en.wikipedia.org/wiki/Computer_vision.
13. Kephart JO, Chess DM (2003) The vision of autonomic computing. IBM Systems Journal.
14. IBM Autonomic Computing Initiative (2005) An architectural blueprint for autonomic computing.
15. Huebscher MC, McCann JA (2008) A survey of autonomic computing—degrees, models, and applications. ACM Computing Surveys.